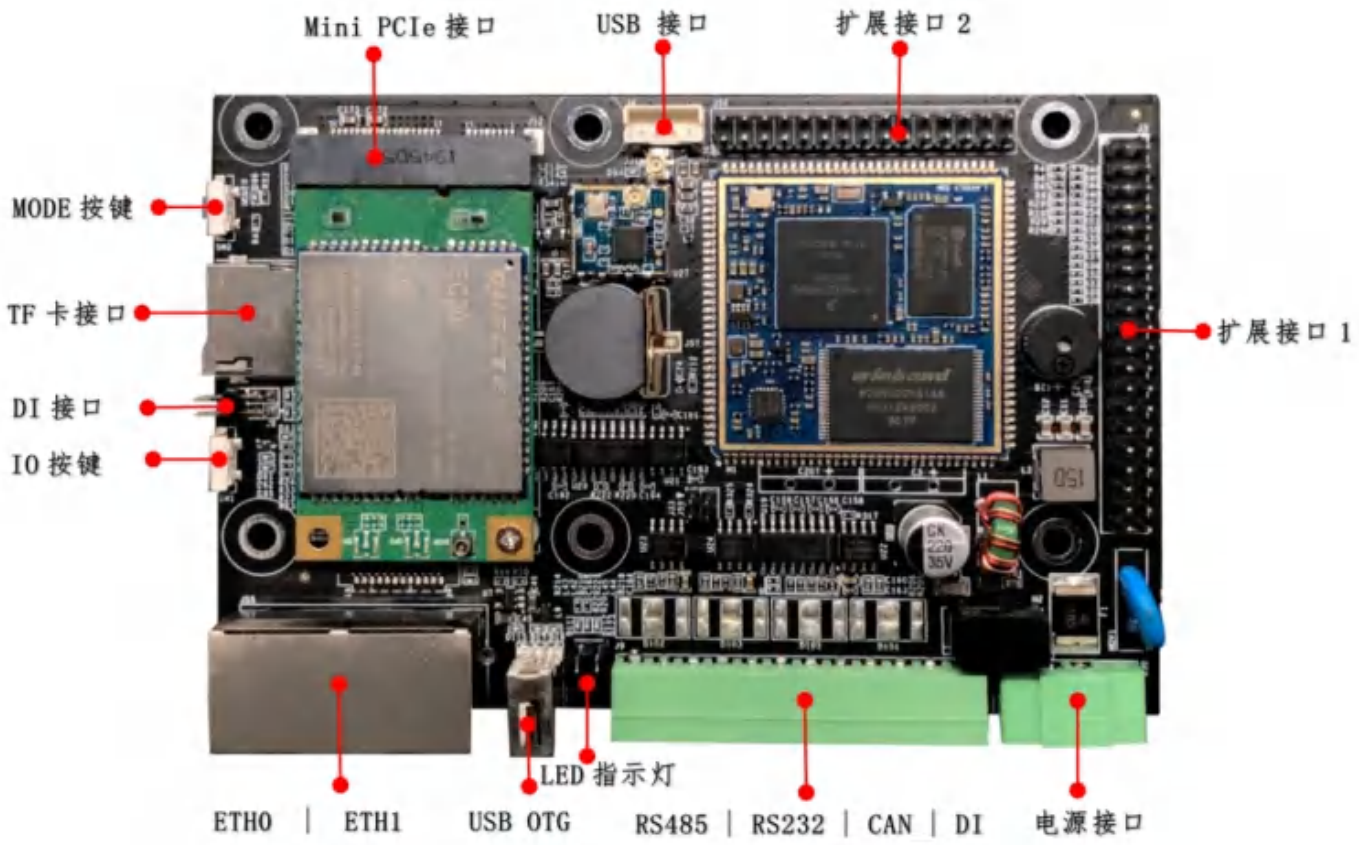
IDO-SBC6Y15-Linux使用说明

1、开发板软硬件资源概况

1.1硬件接口定义图



正面

背面

1.2系统及设备资源

1.2.1开发板资源列表

序号	名称	描述
1	Bootloader	U-Boot 2016.03
2	Kernel	Linux 4.1.15-2.1.0
3	FileSystem	Buildroot根文件系统
4	CPU	MX6Y,ARMV7,800MHZ
5	内存	256MB/512MB DDR3
6	MMC	支持eMMC、TF设备，1路外部接口
7	NAND	支持gpmi Nand
8	USB	2路USB host接口
9	WiFi	支持RTL8723BU、ATBM6032I
10	LTE	支持EC20、Air720
11	Ethernet	2路10/100Mbps以太网接口
12	RS232	3路RS232接口
13	RS485	2路RS485接口
14	CAN	2路CAN接口
15	SPI	2路SPI接口
16	I2C	4路I2C接口
17	PWM	1路PWM
18	RTC	HYM8563
19	LED	系统指示灯+LTE信号灯
20	DI	1路
21	KEY	1个外部按键，一个启动模式按键
22	Buzzer	1个有源蜂鸣器

1.2.2资源设备节点列表

序号	资源设备名称	设备节点
1	Nand Flash	/dev/mtd*
2	eMMC	/dev/mmcblk1*
3	SD Card	/dev/mmcblk0*
4	USB存储设备	/dev/sd*
5	LTE	/dev/ttyUSB0~3 或 /dev/ttyUSB1-4
6	WiFi	wlan0和p2p0
7	Ethernet	eth0和eth1
8	RS232	/dev/ttymxc1 /dev/ttymxc2 /dev/ttymxc6
9	RS485	/dev/ttymxc3 /dev/ttymxc4
10	CAN	can0和can1
11	SPI	/dev/spidev0.0 /dev/spidev1.0
12	I2C	/dev/i2c0 /dev/i2c1 /dev/i2c3 /dev/i2c4
13	pwm	/sys/class/pwm/pwmchip0
14	RTC	/dev/rtc0
15	DI	/dev/di1

16	KEY	/dev/input/event1
17	Buzzer	/dev/bz0

1.3存储资源分配

本开发板有根据不同需求而产生两种配置方案，分别为Nand Flash配置和EMMC配置，具体配置参数请查看下面的表格。

DDR大小	Flash大小	Flash分区信息			
256M	256M	bootloader	kernel	dtb	rootfs
		2M	8M	1M	–

表1 Nand Flash资源分配

DDR大小	Flash大小	Flash分区信息		
512M	4G	bootloader	kernel+dtb	rootfs
		5M	11M	–

表2 EMMC资源分配

2、开发板快速启动

2.1供电电源

开发板的供电接口位于J30，使用12V的直流电源给开发板供电。

2.2Boot Mode

开发板提供3种启动模式，分别为从Flash启动、从SD卡启动和从USB启动（烧录）。
下面的表格说明了如何从这3种方式启动：

启动方式	所需操作
Flash	直接上电

SD卡	按住SW2按键上电
USB	移除SD卡，按住SW2按键上电

2.3Debug

系统提供2种调试方式，分别为串口和SSH。

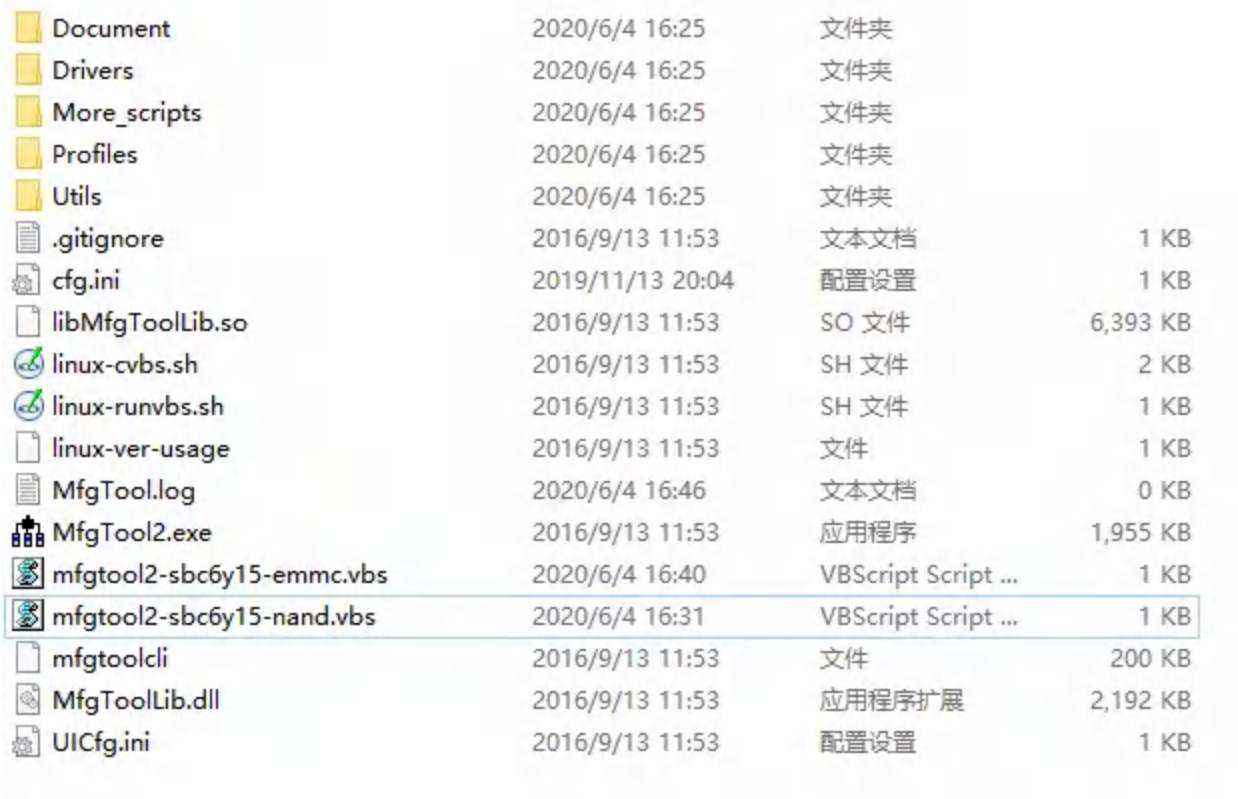
串口调试参数为115200,8N1。SSH端口默认为22。

两种调试方式使用相同的登陆账号的密码，账号：root，密码：wise-kit。

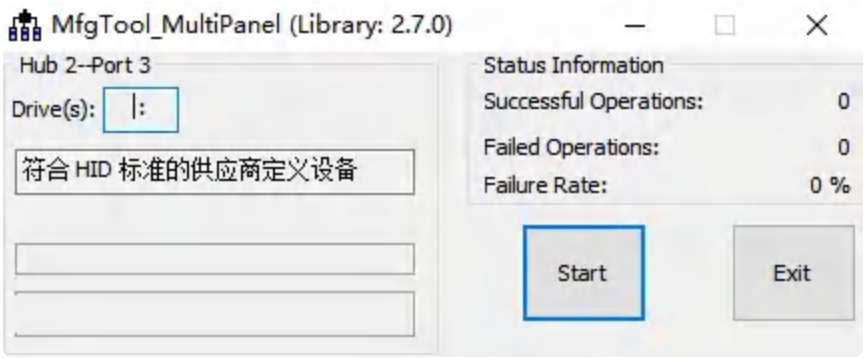
2.4系统烧录

系统的烧录使用MfgTool烧录系统

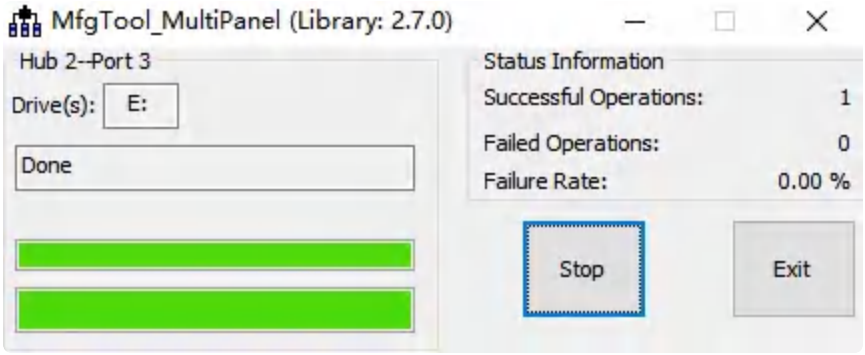
- 确认开发板的SD卡槽上没有插入SD卡；
- 按住SW2按键，然后给开发板上电；
- 使用双工头USB线连接开发板和电脑；
- 进入MfgTool目录



- 根据开发板的Flash类型，双击运行mfgtool2-sbc6y15-emmc.vbs或mfgtool2-sbc6y15-nand.vbs：



- 点击Start按钮，开始下载，大约2分钟后，下载完成：



- 依次点击Stop，Exit按钮，本次烧录完成。

3、开发板功能测试方法

3.1 LED

本开发板配置2个LED指示灯，分别为系统指示灯和4G信号指示灯。

3.1.1系统指示灯

系统启动后，系统指示灯默认为[heartbeat]功能，用于指示系统的运行状态，设备节点目录位于“/sys/class/leds/heartbeat/”。系统指示灯灯的操作方法如下：

1、使用cat命令查看LED的触发条件

```
# cat /sys/class/leds/heartbeat/trigger
none nand-disk mmc0 timer oneshot [heartbeat] backlight gpio cpu0 default-on
```

2、输出内容中，中括号括起来的即为当前的LED的触发条件。

设置系统灯触发条件为none，控制灯的亮灭，如果要将LED设置为用户可控状态，则需要将触发条件设置为none。

```
# echo none > /sys/class/leds/heartbeat/trigger          //修改触发模式
# echo 1 > /sys/class/leds/heartbeat/brightness          //LED亮
# echo 0 > /sys/class/leds/heartbeat/brightness          //LED灭
# echo heartbeat > /sys/class/leds/heartbeat/trigger     //恢复为心跳灯
```

3.1.2信号指示灯

4G状态灯正级连接3.3V电源，负极连接在4G模块的LED_WWAN引脚上，用于指示4G模块的运行状态，不可以由用户控制。状态灯的含义如下所示：

状态	含义
慢闪（200ms 高/1800ms 低）	找网状态
慢闪（1800ms 高/200ms 低）	待机状态
快闪（125ms 高/125ms 低）	正在数据传输
高电平	通话中

3.2按键

开发板共配置2个按键SW1和SW2，其中SW2用于控制启动模式，关于SW2详细介绍请参考2.2。SW1为GPIO按键，位于调试串口的上方，该按键在系统中对应的设备节点为/dev/input/event1，键值为KEY_1。以下该按键的一个测试应用程序，可以获取按键的按下和弹起事件：

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <linux/input.h>
int main(void)
{
    int keys_fd;
    struct input_event t;
    keys_fd = open("/dev/input/event1", O_RDONLY);
    if (keys_fd < 0)
    {
        perror("open /dev/input/event1");
        return –1;
    }
    while(1)
    {
        if (read(keys_fd,&t,sizeof(t)) == sizeof(t))
        {
```

```
            if (t.type==EV_KEY)
            {
                if (t.value==0 || t.value==1)
                {
                    switch(t.code)
                    {
                        case KEY_1:
                            /* 当按键按下t.value=1 , 当按键弹起t.value=0 */
                            printf("key1 %s\n",(t.value)?"Pressed":"Released");
                            break;
                        default:
                            break;
                    }
                }
            }
        }
        else
        {
            break;
        }
    }
    close(keys_fd);
    return -1;
}
```

3.3RS232

开发板配置有3路RS232接口，各路接口的硬件位置与软件设备节点信息详见下表：

接口名称	接口硬件位置	软件设备节点	支持波特率
UART2	J3的23和24引脚	/dev/ttymxc1	最高115200
UART3	J9的R3和T3	/dev/ttymxc2	最高115200
UART7	J56的17、18、27和28引脚	/dev/ttymxc6	最高115200

3.4RS485

开发板配置有2路RS485接口，各路接口的硬件位置与软件设备节点信息详见下表：

接口名称	接口硬件位置	软件设备节点	支持波特率
UART4	J9的A1和B1	/dev/ttymxc3	最高115200
UART5	J9的A2和B2	/dev/ttymxc4	最高115200

3.5CAN

开发板配置有2路CAN总线，分别为CAN0和CAN1，CAN0位于J9接口，CAN1位于J3的25和26引脚。可以使用USBCAN设备连接开发板CAN接口或者将两个开发板CAN接口互连来测试CAN的通信功能。

本节以两个本开发板A和B互连的方式，说明CAN总线的测试方法。将两个CAN接口以H-H，L-L的方式连接。

使用下面的方法，设置开发板B发送数据到开发板A。

设置开发板A为接收

```
# ifconfig can0 down
# ip link set can0 type can bitrate 125000 triple-sampling on
# ifconfig can0 up
# candump can0
```

设置开发板B为发送

```
# ifconfig can0 down
# ip link set can0 type can bitrate 125000 triple-sampling on
# ifconfig can0 up
# cansend can0 5A1#1122334455667788
```

3.6USB

开发板配置有1路USB接口，模式为HOST，位于J7。

3.6.1热插拔功能

系统支持U盘热拔插功能，以下使用U盘进行测试。

将FAT32格式16GB存储的U盘，插入开发板中，会提示如下信息：

```
# [ 9783.727327] usb 1-1: new high-speed USB device number 3 using ci_hdrc
[ 9783.907066] usb 1-1: New USB device found, idVendor=0951, idProduct=1666
[ 9783.927357] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 9783.934550] usb 1-1: Product: DataTraveler 3.0
[ 9783.947412] usb 1-1: Manufacturer: Kingston
[ 9783.951626] usb 1-1: SerialNumber: 4CEDFB74A337F32111987027B
[ 9783.971870] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 9784.007967] scsi host1: usb-storage 1-1:1.0
[ 9785.018593] scsi 1:0:0:0: Direct-Access     Kingston DataTraveler 3.0      PQ: 0 ANSI: 6
[ 9785.039547] sd 1:0:0:0: [sda] 30218842 512-byte logical blocks: (15.4 GB/14.4 GiB)
[ 9785.083433] sd 1:0:0:0: [sda] Write Protect is off
[ 9785.098056] sd 1:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 9785.124090]  sda: sda1
[ 9785.140294] sd 1:0:0:0: [sda] Attached SCSI removable disk
[ 9785.237408] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

并且U盘的第一个分区会自动挂载到/dev/udisk/目录下。

3.6.2查看U盘信息

使用df –h命令查看U盘的设备容量和挂载目录等信息，可以看到U盘的第一个分区已经挂载到/dev/udisk/目录下。

```
# df  -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                 6.5G    221.2M      5.9G   4% /
devtmpfs                 87.7M         0     87.7M   0% /dev
tmpfs                   247.9M         0    247.9M   0% /dev/shm
tmpfs                   247.9M       2.3M   245.6M   1% /tmp
tmpfs                   247.9M      44.0K   247.8M   0% /run
/dev/sda1                14.4G       2.1G    12.3G  15% /udisk
```

3.6.3测试U盘的读写速度

使用dd命令可以测试U盘的读写速度。

time dd if=/dev/zero of=/udisk/test bs=1k count=10240 conv=fsync

time dd if=/udisk/test of=/dev/null bs=1k count=10240

3.7WIFI

本系统默认支持 RTL8723BU和ATBM60321两种WIFI型号。并且文件系统已添加了相关的工具。

3.7.1基本操作

- 使用ifconfig命令可以查看WiFi网卡信息：

```
# ifconfig wlan0
wlan0  Link encap:Ethernet  HWaddr DC:29:19:3B:DC:10
       UP BROADCAST MULTICAST  MTU:1500  Metric:1
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- WiFi开启和关闭：

```
# ifconfig wlan0 up      //打开wlan0
# ifconfig wlan0 down  //关闭wlan0
```

- 使用iwlist命令进行扫描操作：

```
#iwlist wlan0 scan
wlan0     Scan completed :
        Cell 01 – Address: 3C:46:D8:9E:A5:1C
                  Channel:1
                  Frequency:2.412 GHz (Channel 1)
                  Quality=60/70  Signal level=–50 dBm
                  Encryption key:on
                  ESSID:"Luman_AP"
                  Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                            9 Mb/s; 12 Mb/s; 18 Mb/s
                  Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
                  Mode:Master
                  Extra:tsf=7fffffffffffffff
                  Extra: Last beacon: 1100ms ago
                  IE: Unknown: 00084C756D616E5F4150
                  IE: Unknown: 010882848B960C121824
                  IE: Unknown: 030101
                  IE: Unknown: 2A0102
                  IE: IEEE 802.11i/WPA2 Version 1
                      Group Cipher : TKIP
                      Pairwise Ciphers (2) : CCMP TKIP
                      Authentication Suites (1) : PSK
```

3.7.2STA模式

STA模式是将WIFI模块配置为子设备，然后主动去连接路由器。

根据路由器使用的加密方式差异（“WPA–PSK”和“WEP/WPA”），设置WiFi模块为STA模式连接路由器的方法有两种。

1. WPA–PSK加密方式：

在文件系统的“/etc/wifi/”目录下，已经添加WiFi相关的配置文件和操作脚本，其中“wpa.conf”配置文件用于保存要连接的路由器参数，“wifi_sta_start”脚本用于连接路由器，“wifi_sta_stop”脚本用于终止连接路由器。

- 修改/etc/wifi/wpa.conf，设置目标热点的ssid（名称）和psk（密码）：

```
# cat /etc/wifi/wpa.conf
ctrl_interface=/var/run/wpa_supplicant
network={
        ssid="HiWiFi_190B6A"
        psk="12345678"
        key_mgmt=WPA-PSK
        priority=2
}
```

- 执行“wifi_sta_start”脚本连接路由器

```
#/etc/wifi/wifi_sta_start
```

- 上一步执行完成后，查看连接结果，当看到wlan0分配到有效地址，说明连接成功。

```
# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr DC:29:19:3B:DC:10
          inet addr:192.168.1.28  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::859e:917b:94c0:98b9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86 errors:0 dropped:0 overruns:0 frame:0
          TX packets:154 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20327 (19.8 KiB)  TX bytes:19628 (19.1 KiB)
```

1. WPA/WEP加密方式：

- 打开wlan0

```
# ifconfig wlan0 up
```

- 设置连接热点名称和密码

```
# iwconfig wlan0 essid “industio”
# iwconfig wlan0 key “12345678”
```

- 设置网关

```
# route add default gw 192.168.1.1 dev wlan0
```

3.7.3AP模式

设置WiFi模块开启AP模式的脚本和配置文件位于“/etc/wifi/”目录，文件及功能说明如下表所示：

序号	文件名	说明
1	hostapd.conf	hostapd配置文件，配置热点名称、密码和加密方式等参数

2	udhcpd.conf	udhcpd配置文件，配置分配的IP范围、router和dns等参数
3	nat_start	配置WiFi模块支持IP包转发功能，实现WiFi模块上网
4	wifi_ap_start	开启WiFi AP模式
5	wifi_ap_stop	停止WiFi AP

- 修改“/etc/wifi/hostapd.conf”配置文件，设置模块热点名称和密码等信息：

- 修改“/etc/wifi/udhcpd.conf”，配置分配的IP范围、router和dns等参数，注意分配的IP段应该和wlan0的IP（在/etc/wifi/wifi_ap_start种设置）处于同一个段：

```
# The start and end of the IP lease block
start           192.168.2.20
end             192.168.2.254

# The remainder of options are DHCP options and can be specified with the
# keyword 'opt' or 'option'. If an option can take multiple items, such
# as the dns option, they can be listed on the same line, or multiple
# lines.
# Examples:
opt     dns     8.8.8.8 192.168.2.2
option  subnet  255.255.255.0
opt     router  192.168.2.2
opt     wins    8.8.8.8
option  dns     8.8.8.8 # appended to above DNS servers for a total of 3
option  domain  local
option  lease   864000          # default: 10 days
option  msstaticroutes  10.0.0.0/8 10.127.0.1           # single static route
option  staticroutes    10.0.0.0/8 10.127.0.1, 10.11.12.0/24 10.11.12.1
# Arbitrary option in hex form:
option  0x08    01020304        # option 8: "cookie server IP addr: 1.2.3.4"
```

分配IP范围 20~254

设置dns和router等信息

- 修改”/etc/wifi/nat_start，配置wifi数据包转发到eth0，可以按照实际情况修改为eth1或PPP0。

iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

- 执行“wifi_ap_start”脚本，开启AP模式。
- 通过子设备连接热点是否成功、连接后是否分配到有效IP、是否可以通过AP的数据上网等测试来判断AP是否正常工作。

3.8以太网

开发板配置有2路100M以太网接口。

- 通过ifconfig命令可以查看当前有效的以太网设备：

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 36:72:C3:0A:FE:B3
          inet addr:192.168.0.139  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::24cb:6cf8:a8a3:4ef8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10799 (10.5 KiB)  TX bytes:3867 (3.7 KiB)

eth1      Link encap:Ethernet  HWaddr CA:2B:AF:94:07:5B
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:11728 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11728 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:867872 (847.5 KiB)  TX bytes:867872 (847.5 KiB)
```

- 开启和关闭网口

ifconfig eth0 up

ifconfig eth0 down

- 设置IP地址

ifconfig eth0 192.168.0.23

- 设置MAC地址

ifconfig eth0 hw ether 36:72:C3:0A:FE:B3

- 设置子网掩码

ifconfig eth0 netmask 255.255.255.0

- 设置广播地址

ifconfig eth0 broadcast 192.168.0.255

- 添加和删除网关

route add default gw 192.168.0.1

route del default gw 192.168.0.1

- 设置DNS，修改配置文件/etc/resolv.conf（不存在则创建），如需要设置DNS为114.114.114.114，则在末尾添加：nameserver 114.114.114.114
- 设置动态IP

#udhcpc –i eth0

- 设置静态IP，首先需要将/etc/init.d/S41dhcpcd服务移除，然后修改配置文件/etc/init.d/S40network，以下为设置eth0静态IP为192.168.0.10，eth1静态IP为192.168.1.11的示例：

```
# Debian ifupdown needs the /run/network lock directory
mkdir -p /run/network

case "$1" in
  start)
        printf "Starting network: "
        /sbin/ifup -a
        [ $? = 0 ] && echo "OK" || echo "FAIL"
        ifconfig eth0 192.168.0.10 up
        ifconfig eth1 192.168.1.11 up
        ;;
  stop)
        printf "Stopping network: "
        /sbin/ifdown -a
        [ $? = 0 ] && echo "OK" || echo "FAIL"
        ;;
  restart|reload)
        "$0" stop
        "$0" start
        ;;
  *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

3.94G

开发板默认支持EC20和Air720两种型号的4G模块，硬件接口上，他们是兼容的。

3.9.1电源控制

系统添加4G模块的电源和复位控制，对应的设备节点为/dev/4g_pwr和/dev/4g_reset。

打开4G模块的电源：

#echo ON > /dev/4g_pwr

关闭4G模块的电源：

#echo OFF > /dev/4g_pwr

复位4G模块：

#echo ON > /dev/4g_reset

取消复位4G模块：

#echo OFF > /dev/4g_reset

3.9.2获取模块信息

当系统正常启动，并且4G模块正常工作，可以看到4G模块的设备节点/dev/ttyUSB0-3（Air720是/dev/ttyUSB1-4）。我们可以通过这些设备节点来获取4G模块相关信息。

- 检测SIM卡是否插好

#cat /dev/ttyUSB2 &

#echo –e "AT+CPIN?\r\n" > /dev/ttyUSB2

执行以上两条命令，如果SIM卡插入成功，返回结果应如下图所示

```
# echo -e "AT+CPIN?\r\n" > /dev/ttyUSB2
#

+CPIN: READY


OK
```

- 获取4G信号信号

#cat /dev/ttyUSB2 &

echo –e "AT+CSQ\r\n" > /dev/ttyUSB2

执行以上两条命令，返回结果如下图所示，该结果表示信号强度为17：

```
# echo -e "AT+CSQ\r\n" > /dev/ttyUSB2
#

+CSQ: 17,99


OK
```

- 获取4G模块IMEI号

#cat /dev/ttyUSB2 &

#echo –e "AT+GSN\r\n" > /dev/ttyUSB2

执行以上两条命令，将返回一串15位的数字，即为IMEI号。

- 获取SIM卡注册状态

#cat /dev/ttyUSB2 &

#echo –e "AT+CGREG?\r\n" > /dev/ttyUSB2

执行以上两条命令，如果返回”+CREG: 0,1”，则说明注册成功，否则需检查SIM卡是否无效。

```
# echo -e "AT+CGREG?\r\n" > /dev/ttyUSB2
#

+CGREG: 0,1


OK
```

3.9.3拨号上网

当4G模块正常工作，SIM卡已注册时，可以通过4G模块进行拨号上网。

如果4G模块为EC20，则通过以下命令进行拨号上网：

#/etc/ppp/quectel/quectel-pppd.sh &

如果4G模块为EC20，则通过以下命令进行拨号上网：

#pppd call air720-ppp &

拨号需要10秒左右时间完成，如果拨号成功，则会产生“ppp0”的网络节点：

```
# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.138.61.162  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:116 errors:0 dropped:0 overruns:0 frame:0
          TX packets:132 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:9084 (8.8 KiB)  TX bytes:9938 (9.7 KiB)
```

接着我们可以通过ping外网网址的方式，测试拨号上网是否正常：

#ping www.baidu.com

另外，系统添加了自动拨号脚本/etc/ppp/lte_start和服务/etc/init.d/S60lte ，该脚本兼容EC20和Air720拨号上网，同时实现掉线重连功能。

3.10 SPI

开发板配置有2路SPI接口，各路接口的硬件位置与软件设备节点信息详见下表：

接口名称	接口硬件位置	软件设备节点
SPI1	J3的1-4引脚	/dev/spidev0.0
SPI2	J3的5-8引脚	/dev/spidev1.0

3.11 I2C

开发板配置有4路I2C接口，各路接口的硬件位置与软件设备节点信息详见下表：

接口名称	接口硬件位置	软件设备节点
I2C1	J3的11和12引脚	/dev/i2c-0
I2C2	J3的13和14引脚	/dev/i2c-1
I2C3	J56的11和12引脚	/dev/i2c-2
I2C4	J56的13和14引脚	/dev/i2c-3

3.12 RTC

开发板配置1路RTC，对应的设备节点为/dev/rtc0。

3.13 DI

板子带有3个DI，各个DI的硬件位置与软件设备节点信息详见下表：

接口名称	接口硬件位置	软件设备节点	空闲电平
DI1	J9的DI	/dev/di1	低
DI2	J8的IO1	/dev/di2	高
DI3	J8的IO2	/dev/di3	高

通过对设备节点进行读操作，获取DI的输入电平。读到’0’，说明输入低电平；读到’1’，说明输入高电平。以下为一个测试用例：

```
#include <stdio.h>
#include <unistd.h>
#include <error.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#define ID_DEV "/dev/di1"
int main(void)
{
        int fd;
        char buf;
        int ret;
```

```
        fd = open(ID_DEV, O_RDONLY);
        if (fd < 0){
                perror("open dev fail");
                return -1;
        }
        while(1)
        {
                ret = read(fd, &buf, 1);
                if (ret != 1){
                        perror("read");
                        close(fd);
                        return -1;
                }
                printf("di1 input:%s\n", (buf=='0')?"LOW":"HIGH");
                sleep(1);
        }

        return 0;
}
```

3.14 PWM

开发板配置有1路PWM，位于J3的第9引脚，在系统中对应的设备为/sys/class/pwm/pwmchip0，以下为一个测试用例：

```
#echo 0 >  /sys/class/pwm/pwmchip0/export
#echo 1000000000 > /sys/class/pwm/pwmchip0/pwm0/period    //设置周期为1s
#echo 500000000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle //设置占空比为50%
#echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable //开启pwm
```

3.15 Buzzer

开发板配置了一个有源蜂鸣器，对应的设备节点为/dev/bz0，以下为一个测试用例：

蜂鸣器响

```
#echo ON > /dev/bz0
```

蜂鸣器不响

#echo OFF > /dev/bz0

3.16 FTP

系统添加了vsftp服务，可以通过此此服务进行文件传输。登陆账号：root，密码：wise-kit。其权限可通过修改/etc/vsftpd.conf进行配置。

3.17断电检测服务

板子带有两个2.7V 20F规格的电容，可以让系统在断电后继续运行10秒以上。同时，系统提供了断电响应服务，当系统检测到断电后，会立刻执行/usr/app/power_off_handle此程序，power_off_handle可以是脚本，也可以是一个应用程序。

3.18开机自动执行脚本

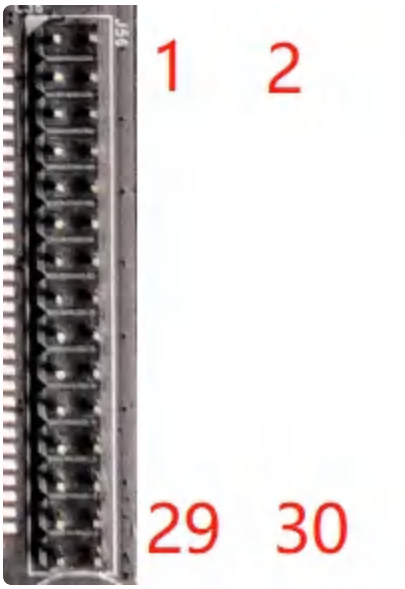
如果我们需要将自定义的脚本实现开机启动，只需要将脚本文件放在/etc/init.d目录下，并改名为大写英文字母'S'开头即可，可参考同目录下其他脚本的格式。

3.19 GPIO

开发板预留了15个GPIO在扩展排针J56上，各个GPIO对应的位置见下表：

GPIO	GPIO Num	排针编号
GPIO3-IO0	64	3
GPIO3-IO1	65	4
GPIO3-IO3	67	7
GPIO3-IO2	66	8
GPIO5-IO0	128	10
GPIO3-IO9	73	15
GPIO3-IO10	74	16
GPIO3-IO13	77	19
GPIO3-IO14	78	20
GPIO3-IO15	79	21
GPIO3-IO16	80	22

GPIO3–IO17	81	23
GPIO3–IO18	82	24
GPIO3–IO19	83	25
GPIO3–IO20	84	26



通过系统GPIO驱动，可以配置这些GPIO为输入或输出功能。以下为对GPIO3–IO10的一个测试用例：

- 作为输入

```
# echo 74 > /sys/class/gpio/export
# echo in > /sys/class/gpio/gpio74/direction
# cat /sys/class/gpio/gpio74/value
  0  //代表输入低电平
# cat /sys/class/gpio/gpio74/value
  1  //代表输入高电平
```

- 作为输出

```
# echo 74 > /sys/class/gpio/unexport
# echo 74 > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio74/direction
# echo 0 > /sys/class/gpio/gpio74/value   //输出低电平
```

```
# echo 1 > /sys/class/gpio/gpio74/value  //输出高电平
```