

iTOP-6818- Ubuntu-4G_EC20 的移植

EC20 模块封装成标准的 PCIe 接口，和开发板主要通过 usb 通讯。此次提供的文件有 pppd 源码、编译好的 chat 和 pppd 工具、wcdma，wcdma_back，wcdma-chat-connect、wcdma-chat-disconnect 以及要在文档最后使用的 pppd_conf.sh 脚本，如下图所示。

名称	大小
ppp-2.4.4.tar.gz	673 KB
pppd	613 KB
chat	44 KB
wcdma	1 KB
wcdma_back	1 KB
pppd_conf.sh	1 KB
wcdma-chat-connect	1 KB
wcdma-chat-disconnect	1 KB

在进行编译之前要先修改编译器为 arm-2009q3 编译器，如何设置编译器参见手册 5.2.2 “交叉编译工具的安装”。另外为了避免使用环境变量设置编译器而可能出现的问题，文档中大部分编译是使用编译器的绝对路径，用户也应先找到自己编译器的绝对路径待用。下图是本次编译使用的编译器以及编译器压缩包。

```
root@ubuntu:/usr/local/arm# ls
4.3.2
4.4.1
arm-2009q3
arm-2009q3.tar.bz2
arm-2014.05
arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
arm-linux-4.4.1.tar.gz
arm-linux-gcc-4.3.2.tar.gz
arm-linux-gcc-4.5.1-v6-vfp-20120301.tgz
gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12
gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12.tar.gz
root@ubuntu:/usr/local/arm#
```

由上图可知该编译器的绝对路径为 “/usr/local/arm/arm-2009q3/bin/ arm-none-linux-gnueabi-gcc” 。

1 驱动支持

首先需要对 Linux 内核驱动做一定的修改，使操作系统能够支持 EC20，首先增加 VID 和 PID，即 usb 的厂商 ID 和设备 ID。在内核源码中的 kernel /drivers/usb/serial/option.c 文件里添加如下两项内容。

```
#define EC20_VID 0x1E0E
#define EC20_PID 0x9001
```

如下图所示。

```
static void option_instat_callback(struct urb *urb);
/* Vendor and product IDs */
#define EC20_VID 0x1E0E
#define EC20_PID 0x9001

#define OPTION_VENDOR_ID 0x0AF0
#define OPTION_PRODUCT_COLT 0x5000
```

然后搜索 option_ids，在 option_ids 列表中增加以下内容。

```
{ USB_DEVICE(EC20_VID, EC20_PID) },
```

如下图所示。

```
};

static const struct usb_device_id option_ids[] = {
    //add
    { USB_DEVICE(EC20_VID, EC20_PID) },
    //end add
    { USB_DEVICE(OPTION_VENDOR_ID, OPTION_PRODUCT_COLT) },
    { USB_DEVICE(OPTION_VENDOR_ID, OPTION_PRODUCT_RICOLA) },
};
```

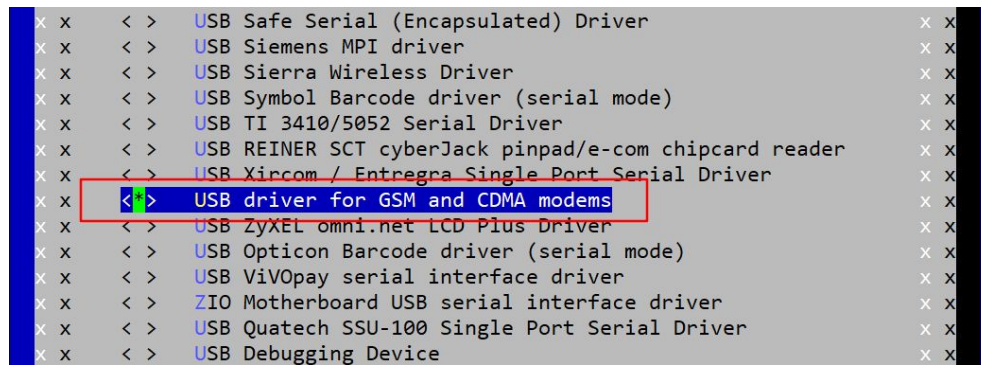
回到 kernel 目录，配置 make menuconfig 以下两项。

[*] Device Drivers →

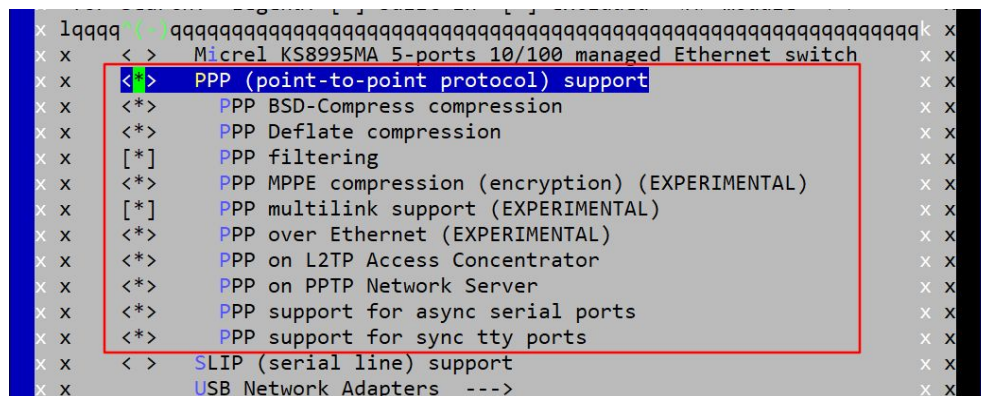
[*] USB Support →

[*] USB Serial Converter support →

[*] USB driver for GSM and CDMA modems



以及在 Device Drivers > Network device support 的<*> PPP***全部选中



然后编译内核，并烧写到开发板。

2 编译 ppp 拨号工具

将提供的源码解压到 Ubuntu，进入解压的文件夹，如下图所示。

```

root@ubuntu:/home/frao/EC20# 1
ppp-2.4.4.tar.gz
root@ubuntu:/home/frao/EC20# tar -xf ppp-2.4.4.tar.gz
root@ubuntu:/home/frao/EC20# cd ppp-2.4.4/
root@ubuntu:/home/frao/EC20/ppp-2.4.4# 1
Changes-2.3  etc.ppp/  PLUGINS  README.cbcp  README.MSCHAP81  SETUP
chat/        FAQ        pppd/     README.eap-srp  README.pppoe     solaris/
common/      include/  pppdump/  README.linux    README.pwfd
configure*   linux/    pppstats/ README.MPPE     README.sol2
contrib/     modules/  README    README.MSCHAP80 scripts/

```

使用 “./configure” 命令配置源码，如下图所示。

```
root@ubuntu:/home/frao/EC20/ppp-2.4.4# ./configure
Configuring for Linux
Creating Makefiles.
Makefile <= linux/Makefile.top
pppd/Makefile <= pppd/Makefile.linux
pppstats/Makefile <= pppstats/Makefile.linux
chat/Makefile <= chat/Makefile.linux
pppdump/Makefile <= pppdump/Makefile.linux
pppd/plugins/Makefile <= pppd/plugins/Makefile.linux
pppd/plugins/rp-pppoe/Makefile <= pppd/plugins/rp-pppoe/Makefile.linux
pppd/plugins/radius/Makefile <= pppd/plugins/radius/Makefile.linux
pppd/plugins/pppoeatm/Makefile <= pppd/plugins/pppoeatm/Makefile.linux
root@ubuntu:/home/frao/EC20/ppp-2.4.4#
```

使用命令 “make CC=/usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-gcc” 编译源码，注意这里使用编译器的绝对路径。

```
root@ubuntu:/home/frao/EC20/ppp-2.4.4# make CC=/usr/local/arm/arm-2009q3/bin/arm-
none-linux-gnueabi-gcc
cd chat; make all
make[1]: Entering directory `/home/frao/EC20/ppp-2.4.4/chat'
/usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-gcc -c -O2 -g -pipe -DTERMI
OS -DSIGTYPE=void -UNO_SLEEP
-DNDELAY=O_NDELAY -o chat.o chat.c
chat.c: In function 'put_string':
chat.c:1306: warning: trigraph ??) ignored, use -trigraphs to enable
/usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-gcc -o chat chat.o
make[1]: Leaving directory `/home/frao/EC20/ppp-2.4.4/chat'
cd pppd/plugins; make all
make[1]: Entering directory `/home/frao/EC20/ppp-2.4.4/pppd/plugins'
```

编译到此完成，可以看到在 pppd 文件夹生成了 pppd 工具，在 chat 文件夹生成了 chat 工具，如下图所示。

```
root@ubuntu:/home/frao/EC20/ppp-2.4.4/pppd# ll -tr
total 168
-rw-r--r-- 1 root root 10724 Jan 24 23:34 md4.o
-rw-r--r-- 1 root root 27196 Jan 24 23:34 chap_ms.o
-rw-r--r-- 1 root root 16340 Jan 24 23:34 sha1.o
-rw-r--r-- 1 root root 8032 Jan 24 23:34 pppcrypt.o
-rw-r--r-- 1 root root 24152 Jan 24 23:34 multilink.o
-rw-r--r-- 1 root root 58188 Jan 24 23:34 tdb.o
-rw-r--r-- 1 root root 7080 Jan 24 23:34 spinlock.o
drwxr-xr-x 3 topeet topeet 4096 Jan 24 23:34 /
-rwxr-xr-x 1 root root 626753 Jan 24 23:34 pppd*
root@ubuntu:/home/frao/EC20/ppp-2.4.4/pppd# ll -tr
total 168
-rw-r--r-- 1 topeet topeet 263 Sep 7 2002 Makefile.sol2
-rw-r--r-- 1 topeet topeet 34667 Feb 1 2004 chat.c
-rw-r--r-- 1 topeet topeet 18815 Nov 13 2004 chat.8
-rw-r--r-- 1 topeet topeet 693 Jun 3 2006 Makefile.linux
drwxr-xr-x 14 topeet topeet 4096 Jan 24 23:32 ../
-rw-r--r-- 1 root root 694 Jan 24 23:32 Makefile
-rw-r--r-- 1 root root 46716 Jan 24 23:34 chat.o
drwxr-xr-x 2 topeet topeet 4096 Jan 24 23:34 /
-rwxr-xr-x 1 root root 44841 Jan 24 23:34 chat*
root@ubuntu:/home/frao/EC20/ppp-2.4.4/chat#
```

3 拷贝工具

将上面步骤中生成的 pppd 和 chat 工具下载到开发板/usr/sbin/目录下。

将压缩包提供的脚本 wcdma，wcdma-chat-connect，wcdma-chat-disconnect 拷贝到开发板的/etc/ppp/peers/目录下。

最后在开发板执行以下指令，建立软连接。

```
In /lib/ld-linux-armhf.so.3 /lib/ld-linux.so.3
```


4 测试

首先将天线安装到开发板上，如下图所示。



插入手机卡后重启开发板，并将脚本 `pppd_conf.sh` 拷贝到开发板，然后使用命令 `“./pppd_conf.sh &”` 运行脚本，运行结果如下图所示。

```
02.99.166.4>]
Could not determine remote IP address: defaulting to 10.64.64.64
local IP address 10.24.62.217
remote IP address 10.64.64.64
primary DNS address 202.99.160.68
secondary DNS address 202.99.166.4
Script /etc/ppp/ip-up started (pid 1586)
Script /etc/ppp/ip-up finished (pid 1586), status = 0x0

10.64.64.64
202.99.160.68
```

使用命令 `“ping www.baidu.com -c 4 -I ppp0 ”` 测试，可以看到连网成功（由于 4G 连接需要搜索信号，所以在信号较差的区域需等待一段时间，才能上网）。

```
root@iTOP4412-ubuntu-desktop:/# ping www.baidu.com -c 4 -I ppp0
PING www.a.shifen.com (61.135.169.121) from 10.24.62.217 ppp0: 56(84) bytes of data:
64 bytes from 61.135.169.121: icmp_req=1 ttl=53 time=25.4 ms
64 bytes from 61.135.169.121: icmp_req=2 ttl=53 time=39.3 ms
64 bytes from 61.135.169.121: icmp_req=3 ttl=53 time=38.3 ms
64 bytes from 61.135.169.121: icmp_req=4 ttl=53 time=35.2 ms

--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 25.430/34.585/39.302/5.490 ms
root@iTOP4412-ubuntu-desktop:/#
```

移植到此完成。

联系方式

北京迅为电子有限公司致力于嵌入式软硬件设计，是高端开发平台以及移动设备方案提供商；基于多年的技术积累，在工控、仪表、教育、医疗、车载等领域通过 OEM/ODM 方式为客户创造价值。

iTOP-6818 开发板是迅为电子基于三星最新八核处理器 6818 研制的一款实验开发平台，可以通过该产品评估 6818 处理器相关性能，并以此为基础开发出用户需要的特定产品。

本手册主要介绍 iTOP-6818 开发板的使用方法，旨在帮助用户快速掌握该产品的应用特点，通过对开发板进行后续软硬件开发，衍生出符合特定需求的应用系统。

如需平板电脑案支持，请访问迅为平板方案网“<http://www.topeet.com>”，我司将有能力为您提供全方位的技术服务，保证您产品设计无忧！

本手册将持续更新，并通过多种方式发布给新老用户，希望迅为电子的努力能给您的学习和开发带来帮助。

迅为电子

2018 年 2 月