EC200&UC200 Linux&Android驱动移植说明

EC200&UC200 需要用到2个驱动，这2个驱动都是内核自带的。

1. usb-serial-option, USB转串口驱动，生成/dev/ttyUSB0~2. 总共3个串口，分别是DM、AT、PPP.

需要使能的内核选项如下

USB\_SERIAL=y

USB\_SERIAL\_WWAN=y

USB\_SERIAL\_OPTION=y

1. USB网卡驱动，模块可通过AT配置成rndis或者ecm，默认是ecm。

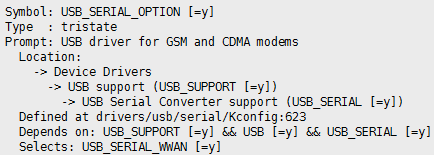
需要使能的内核选项如下

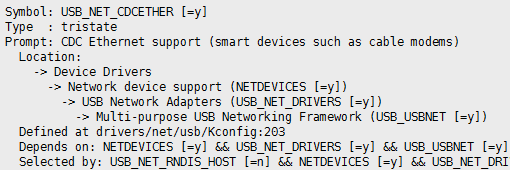
USB\_USBNET=y

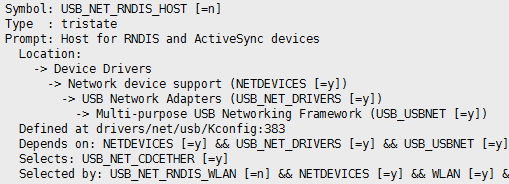
USB\_NET\_CDCETHER=y #用ECM 使能此项

USB\_NET\_RNDIS\_HOST=y #用RNDIS 使能此项

内核配置项截图如下







1. usb-serial-option usb串口驱动与公司其他高通平台模块的兼容

UC200&EC200和公司的其他基于高通平台的模块(后面叫EC21&EC25)，都是用usb-serial-option驱动，但是USB interface的顺序不同. 所以如果客户之前移植过EC21&EC25的代码，会导致UC200&EC200的PPP口识别不到。所以建议客户先还原drivers/usb/serial/option.c的修改，重新按照 [EC200\_UC200\_usb\_serial\_option.patch](#patch)移植。这个修改方法同时UC200&EC200和EC21&EC25模块。

1. dmesg log

[17738.104336] usb 1-1.4: new high-speed USB device number 5 using ehci-pci

[17738.215240] usb 1-1.4: New USB device found, idVendor=2c7c, idProduct=6026

[17738.215243] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3

[17738.215246] usb 1-1.4: Product: EC26

[17738.215248] usb 1-1.4: Manufacturer: Quectel

[17738.215250] usb 1-1.4: SerialNumber: 200806006809080000

[17738.217710] rndis\_host 1-1.4:1.0 eth0: register 'rndis\_host' at usb-0000:00:1a.0-1.4, RNDIS device, ac:af:fc:d4:e0:61

[17738.218236] option 1-1.4:1.2: GSM modem (1-port) converter detected

[17738.218485] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB0

[17738.218729] option 1-1.4:1.3: GSM modem (1-port) converter detected

[17738.218843] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB1

[17738.219089] option 1-1.4:1.4: GSM modem (1-port) converter detected

[17738.219186] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB2

/dev/ttyUSB0 is DM Port

/dev/ttyUSB1 is AT Port

/dev/ttyUSB2 is PPP port

1. EC200\_UC200\_usb\_serial\_option.patch

diff --git a/drivers/usb/serial/option.c b/drivers/usb/serial/option.c

index b6320e3..8ca8dbe 100644

--- a/drivers/usb/serial/option.c

+++ b/drivers/usb/serial/option.c

@@ -687,6 +687,22 @@ static const struct option\_blacklist\_info yuga\_clm920\_nc5\_blacklist = {

};

static const struct usb\_device\_id option\_ids[] = {

+#if 1 //Added by Quectel

+ { USB\_DEVICE(0x05C6, 0x9090) }, /\* Quectel UC15 \*/

+ { USB\_DEVICE(0x05C6, 0x9003) }, /\* Quectel UC20 \*/

+ { USB\_DEVICE(0x2C7C, 0x0125) }, /\* Quectel EC25 \*/

+ { USB\_DEVICE(0x2C7C, 0x0121) }, /\* Quectel EC21 \*/

+ { USB\_DEVICE(0x05C6, 0x9215) }, /\* Quectel EC20 \*/

+ { USB\_DEVICE(0x2C7C, 0x0191) }, /\* Quectel EG91 \*/

+ { USB\_DEVICE(0x2C7C, 0x0195) }, /\* Quectel EG95 \*/

+ { USB\_DEVICE(0x2C7C, 0x0306) }, /\* Quectel EG06/EP06/EM06 \*/

+ { USB\_DEVICE(0x2C7C, 0x0296) }, /\* Quectel BG96 \*/

+ { USB\_DEVICE(0x2C7C, 0x0435) }, /\* Quectel AG35 \*/

+ { USB\_DEVICE(0x2C7C, 0x0512) }, /\* Quectel EG12/EG18 \*/

+ { USB\_DEVICE(0x2C7C, 0x6026) }, /\* Quectel EC200 \*/

+ { USB\_DEVICE(0x2C7C, 0x6120) }, /\* Quectel UC200 \*/

+ { USB\_DEVICE(0x2C7C, 0x6000) }, /\* Quectel EC200/UC200 \*/

+#endif

{ USB\_DEVICE(OPTION\_VENDOR\_ID, OPTION\_PRODUCT\_COLT) },

{ USB\_DEVICE(OPTION\_VENDOR\_ID, OPTION\_PRODUCT\_RICOLA) },

{ USB\_DEVICE(OPTION\_VENDOR\_ID, OPTION\_PRODUCT\_RICOLA\_LIGHT) },

@@ -2088,6 +2104,9 @@ static struct usb\_serial\_driver option\_1port\_device = {

#ifdef CONFIG\_PM

.suspend = usb\_wwan\_suspend,

.resume = usb\_wwan\_resume,

+#if 1 //add by Quectel

+ .reset\_resume = usb\_wwan\_resume,

+#endif

#endif

};

@@ -2105,6 +2124,31 @@ static int option\_probe(struct usb\_serial \*serial,

struct usb\_device\_descriptor \*dev\_desc = &serial->dev->descriptor;

const struct option\_blacklist\_info \*blacklist;

+#if 1 //Added by Quectel

+ //Quectel UC20's interface 4 can be used as USB Network device

+ if (serial->dev->descriptor.idVendor == cpu\_to\_le16(0x05C6) && serial->dev->descriptor.idProduct == cpu\_to\_le16(0x9003)

+ && serial->interface->cur\_altsetting->desc.bInterfaceNumber >= 4)

+ return -ENODEV;

+

+ //Quectel EC20's interface 4 can be used as USB Network device

+ if (serial->dev->descriptor.idVendor == cpu\_to\_le16(0x05C6) && serial->dev->descriptor.idProduct == cpu\_to\_le16(0x9215)

+ && serial->interface->cur\_altsetting->desc.bInterfaceNumber >= 4)

+ return -ENODEV;

+

+ if (serial->dev->descriptor.idVendor == cpu\_to\_le16(0x2C7C)) {

+ \_\_u16 idProduct = le16\_to\_cpu(serial->dev->descriptor.idProduct);

+

+ //Quectel EC200&UC200's interface 0 can be used as USB Network device (ecm, rndis)

+ if (serial->interface->cur\_altsetting->desc.bInterfaceClass != 0xFF)

+ return -ENODEV;

+

+ //Quectel EC25&EC21&EG91&EG95&EG06&EP06&EM06&BG96&AG35&EG12&EG18's interface 4 can be used as USB network device (qmi,ecm,mbim)

+ if ((idProduct != 0x6026 && idProduct != 0x6126)

+ && serial->interface->cur\_altsetting->desc.bInterfaceNumber >= 4)

+ return -ENODEV;

+ }

+#endif

+

/\* Never bind to the CD-Rom emulation interface \*/

if (iface\_desc->bInterfaceClass == 0x08)

return -ENODEV;

diff --git a/drivers/usb/serial/qcserial.c b/drivers/usb/serial/qcserial.c

index 613f91a..4a06a2a 100644

--- a/drivers/usb/serial/qcserial.c

+++ b/drivers/usb/serial/qcserial.c

@@ -63,7 +63,7 @@ static const struct usb\_device\_id id\_table[] = {

{DEVICE\_G1K(0x05c6, 0x9202)}, /\* Generic Gobi Modem device \*/

{DEVICE\_G1K(0x05c6, 0x9203)}, /\* Generic Gobi Modem device \*/

{DEVICE\_G1K(0x05c6, 0x9222)}, /\* Generic Gobi Modem device \*/

- {DEVICE\_G1K(0x05c6, 0x9008)}, /\* Generic Gobi QDL device \*/

+ //{DEVICE\_G1K(0x05c6, 0x9008)}, /\* Generic Gobi QDL device \*/

{DEVICE\_G1K(0x05c6, 0x9009)}, /\* Generic Gobi Modem device \*/

{DEVICE\_G1K(0x05c6, 0x9201)}, /\* Generic Gobi QDL device \*/

{DEVICE\_G1K(0x05c6, 0x9221)}, /\* Generic Gobi QDL device \*/

@@ -88,7 +88,7 @@ static const struct usb\_device\_id id\_table[] = {

{USB\_DEVICE(0x03f0, 0x241d)}, /\* HP Gobi 2000 QDL device (VP412) \*/

{USB\_DEVICE(0x03f0, 0x251d)}, /\* HP Gobi 2000 Modem device (VP412) \*/

{USB\_DEVICE(0x05c6, 0x9214)}, /\* Acer Gobi 2000 QDL device (VP413) \*/

- {USB\_DEVICE(0x05c6, 0x9215)}, /\* Acer Gobi 2000 Modem device (VP413) \*/

+ //{USB\_DEVICE(0x05c6, 0x9215)}, /\* Acer Gobi 2000 Modem device (VP413) \*/

{USB\_DEVICE(0x05c6, 0x9264)}, /\* Asus Gobi 2000 QDL device (VR305) \*/

{USB\_DEVICE(0x05c6, 0x9265)}, /\* Asus Gobi 2000 Modem device (VR305) \*/

{USB\_DEVICE(0x05c6, 0x9234)}, /\* Top Global Gobi 2000 QDL device (VR306) \*/

diff --git a/drivers/usb/serial/usb\_wwan.c b/drivers/usb/serial/usb\_wwan.c

index 107e64c..ee0190c 100644

--- a/drivers/usb/serial/usb\_wwan.c

+++ b/drivers/usb/serial/usb\_wwan.c

@@ -499,6 +499,20 @@ static struct urb \*usb\_wwan\_setup\_urb(struct usb\_serial\_port \*port,

usb\_sndbulkpipe(serial->dev, endpoint) | dir,

buf, len, callback, ctx);

+#if 1 //Added by Quectel for Zero Packet

+ if (dir == USB\_DIR\_OUT) {

+ struct usb\_device\_descriptor \*desc = &serial->dev->descriptor;

+ if (desc->idVendor == cpu\_to\_le16(0x05C6) && desc->idProduct == cpu\_to\_le16(0x9090))

+ urb->transfer\_flags |= URB\_ZERO\_PACKET;

+ if (desc->idVendor == cpu\_to\_le16(0x05C6) && desc->idProduct == cpu\_to\_le16(0x9003))

+ urb->transfer\_flags |= URB\_ZERO\_PACKET;

+ if (desc->idVendor == cpu\_to\_le16(0x05C6) && desc->idProduct == cpu\_to\_le16(0x9215))

+ urb->transfer\_flags |= URB\_ZERO\_PACKET;

+ if (desc->idVendor == cpu\_to\_le16(0x2C7C))

+ urb->transfer\_flags |= URB\_ZERO\_PACKET;

+ }

+#endif

+

return urb;

}