ASR1802 模组 Linux 驱动配置指南

2017 年 9 月

ASR FAE team

文档概述

ASR1802 模组和 Linux 的主机进行通讯需要正确配置 Linux 内核，使能 RNDIS 驱动和 CDC ACM 驱动，ASR1802 模组的软件也需要做正确的配置。ASR1802 模组通过 USB 连接主机后正常情况下需要枚举出三个设备：一个 RNDIS 的网口设备和两个 ACM 设备。第一个 ACM 设备是 Log 口，linux 主机通过这个端口接收 1802 模组的 log 并保存到本地，第二个 ACM 设备是 AT 口，主机通过这个端口发送 AT 命令给模组并接收模组上报的信息。

本文档说明如何配置 Linux 主机的 RNDIS 驱动和 ACM 设备驱动，以及 ASR1802 模组的软件需要注意哪些地方。

1 Linux 主机配置 RNDIS 驱动

在 menuconfig 中配置内核的 rndis 驱动：

| Prompt: Host for RNDIS and ActiveSync devices (EXPERIMENTAL) |

| Defined at drivers/net/usb/Kconfig:234 |

| Depends on: NETDEVICES && USB && NET && USB_USBNET && EXPERIMENTAL |

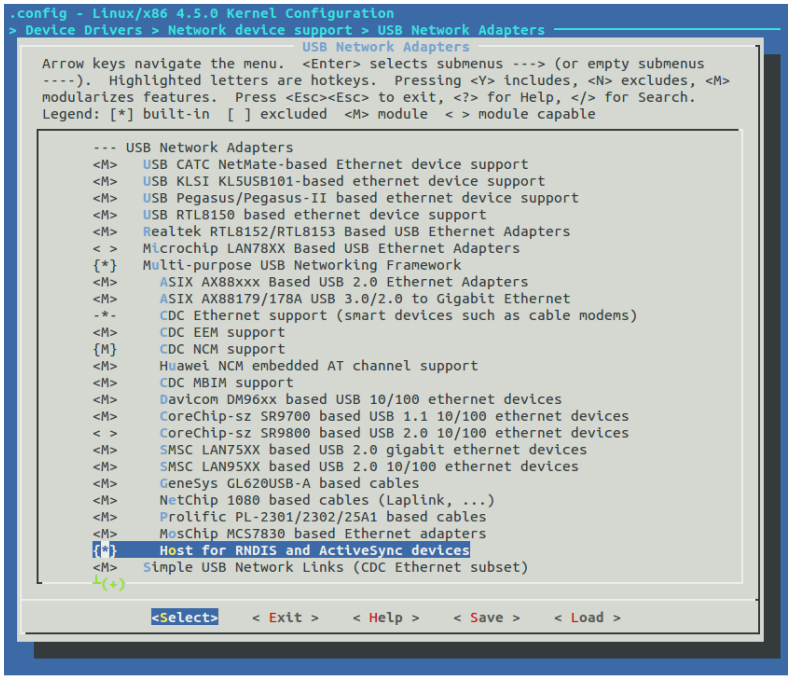
| Location: |

| -> Device Drivers |

| -> Network device support (NETDEVICES [=y]) |

| -> USB Network Adapters |

| -> Multi-purpose USB Networking Framework (USB_USBNET [=y]) |



配置完成后编译主机的内核，下载到目标板中，模组通过 USB 和目标板连接后可以看到以下的枚举过程（不同平台和版本 log 会有差异），主机会枚举出一个 ethx 的网络设备：

<4>[usb1]---------- Start Mount USB Device ----------

<4>[usb1] Device is connected

<4>[usb1]USB_EVENTS_CHECK_CONNECT

<4>[usb1] HS Reset

<4>[usb1] HS device connected

<6>usb 1-1: new high-speed USB device number 12 using MtkUsbHcdHub

<4>[usb1] HS Reset

<4>[usb1] HS device connected

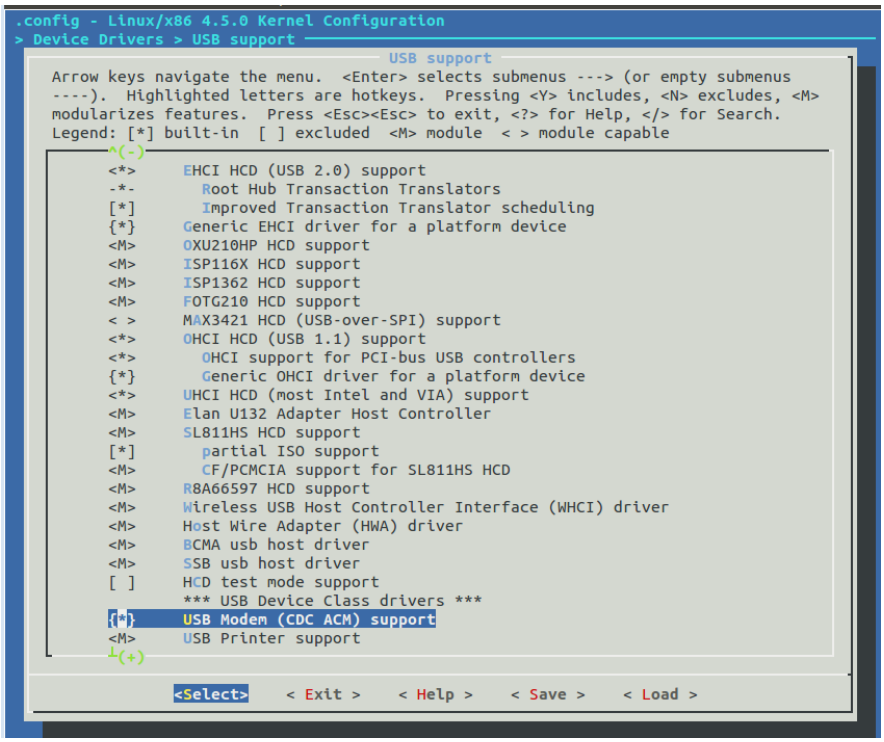
<4>[usb1]dequeue urb(0xe2defd00) L_EP:0 TX

<6>rndis_host 1-1:1.0: eth0: register 'rndis_host' at usb-MtkUsbHcdHub.0-1, RNDIS device, ac:15:b4:cf:56:8b

2 Linux 主机配置 CDC ACM 驱动

在 menuconfig 中打开 cdc acm 的驱动选项：

```
| Symbol: USB_ACM [=y] |
| Type  : tristate |
| Prompt: USB Modem (CDC ACM) support |
|   Location: |
|     -> Device Drivers |
|       -> USB support (USB_SUPPORT [=y]) |
|         -> Support for Host-side USB (USB [=y]) |
|   Defined at drivers/usb/class/Kconfig:6 |
|   Depends on: USB_SUPPORT [=y] && USB [=y] && TTY [=y] |
|   Selected by: USB_VL600 [=m] && NETDEVICES [=y] && USB_NET_DRIVERS [=y] && \ |
| USB_NET_CDCETHER [=y] && TTY [=y]       |
```



在 dirvers/usb/class/cdc-acm.c 文件中的 acm_ids 数组下面加入 ASR1802 的 VID 和 PID：

```
static const struct usb_device_id acm_ids[] = {
    /* quirky and broken devices */
    {USB_DEVICE(0x1286, 0x4e3d),}/*asr1802 acm devices*/
    { USB_DEVICE(0x076d, 0x0006), /* Denso Cradle CU-321 */
    .driver_info = NO_UNION_NORMAL, },/* has no union descript
    { USB_DEVICE(0x17ef, 0x7000), /* Lenovo USB modem */
    .driver_info = NO_UNION_NORMAL, },/* has no union descript
    { USB_DEVICE(0x0870, 0x0001), /* Metricom GS Modem */
    .driver_info = NO_UNION_NORMAL, /* has no union descriptor
    },
```

编译好的 kernel 下载到目标板后，ASR1802 通过 USB 连接到目标板，会在/dev 目录下枚举出 ttyACM0 和 ttyACM1（如果系统有其它 ACM 设备，设备序号会递增，不一定是这两个序号），ttyACM0 是 log 端口，ttyACM1 是 AT 端口。枚举过程的 log 如下（不同平台和版本 log 会有差异）：

<4>[usb1]---------- Start Mount USB Device ----------

<4>[usb1] Device is connected

<4>[usb1]USB_EVENTS_CHECK_CONNECT

<4>[usb1] HS Reset

<4>[usb1] HS device connected

<6>usb 1-1: new high-speed USB device number 12 using MtkUsbHcdHub

<4>[usb1] HS Reset

<4>[usb1] HS device connected

<3>cdc_acm 1-1:1.2: This device cannot do calls on its own. It is not a modem.

<6>cdc_acm 1-1:1.2: ttyACM0: USB ACM device

<3>cdc_acm 1-1:1.4: This device cannot do calls on its own. It is not a modem.

<6>cdc_acm 1-1:1.4: ttyACM1: USB ACM device

3 ASR1802 软件配置的注意事项

为了保证 linux 主机能正确枚举出 ASR1802 的设备，模组的软件需要作以下两点配置：

1. 在 hop/bsp/src/main.c 需要将 usbDrvCfg 的第一个选项设置为 USB_GENERIC_MIFI_DRIVER：

```
00407: /* Default USB driver configuration */
00408: Usb_DriverS          usbDrvCfg = {USB_GENERIC_MIFI_DRIVER, MASS_STORAGE_I
```

2. hal/usb_standart/src/usb_descriptor.c 的函数 USB2MgrUpdateDescriptor()里面，在 desc 变量的 switch 语句中找到 USB_GENERIC_MIFI_DESCRIPTOR 的配置，按照下面选中部分的代码进行设置：

```
01325:              /*INTERFACE ASSOCIATION DESCRIPTOR */
01326:              configDesc[configDesc_num++] = 0x08; // bLength
01327:              configDesc[configDesc_num++] = 0x0b; // INTERFACE ASSOCIATION DESCRIPTOR bI
01328:              configDesc[configDesc_num++] = 0x00; // bFirstInterface
01329:              configDesc[configDesc_num++] = 0x02; // bInterfaceCount
01330:              configDesc[configDesc_num++] = 0xe0;//0xef; // bFunctionClass
01331:              configDesc[configDesc_num++] = 0x01;//0x04; // bFunctionSubClass
01332:              configDesc[configDesc_num++] = 0x03;//0x01; // bFunctionProtocol
01333:              configDesc[configDesc_num++] = 0x05; // Index of string descriptor describi
01334:
01335:              /*************************************************************/
```

3. hal/usb_device/src/cidriver/mvUsbModem.c 文件的 mvUsbModem0RxHISR()函数里面，在接收到 AT 命令数据的代码中，加入如下条件过滤掉无效的 AT 命令，防止 ACM 设备的回显数据引起 ASR1802 的状态出错：

```
    while(length)
    {
        if (length > mvUsbModemRxSize)
        {
            txLength = mvUsbModemRxSize;
        }
        else
        {
            txLength = length;
        }

        buf_ptr = (char *)malloc(txLength);
        ASSERT(buf_ptr != NULL);

        memcpy(buf_ptr, RxPtr, txLength);

        sATPMode = Get_sATP_Mode(TEL_AT_CMD_ATP_0);

        if(sATPMode == MODEM_CONTROL_MODE)
        {
            ATParserMsg atMsg;

            if(txLength > 2 && (*buf_ptr == 'a' || *buf_ptr == 'A')
                && (*(buf_ptr+1) == 't' || *(buf_ptr+1) == 'T'))
            {
                atMsg.data = (char *)buf_ptr;
                atMsg.length = txLength;
                atMsg.sATPInd = TEL_AT_CMD_ATP_0;

                DBGMSG("sATP%d->AtChanThread", atMsg.sATPInd);

                osa_status = OSAMsgQSend(gATMsgQ, sizeof(atMsg), (UINT8*)&atMsg, OSA_NO_SUSPEND);
                ASSERT(osa_status == OS_SUCCESS);
            }
            else
            {
                CPUartLogPrintf("%s:give up the msg(%d):%s\r\n", __func__,txLength, buf_ptr);
            }
        } ? end if sATPMode==MODEM_CONTR... ?
        else
```