Ubuntu範例

本文檔介紹了如何在Ubuntu環境中讀取Hi229/226 的數據，本路徑提供了c語言範例代碼，生成的可執行文件用於讀取模組的數據。

測試環境： Ubuntu 16.04

測試設備：超核調試版　HI226　HI229

查找USB-UART設備

因為Ubuntu 系統自帶CP210x的驅動，所以不用專門去安裝相應序列埠驅動。將調試版連接到電腦上時，會自動識別設備。識別成功後，會在dev目錄下出現一個對應的設備文件。

檢查系統是否識別到USB-UART設備：

1、打開Ubuntu系統，按下 **ctrl + alt + t** 打開命令行窗口

2、在窗口上輸入 `cd /dev` 切換到dev目錄下，這個目錄下，是一些設備文件。

3、然後在dev目錄下執行`ls` 這個命令是查看當前目錄下都有哪些文件，然後按下 Enter 鍵，就會出現設備文件名稱，在這些文件名稱中，主要關心 **ttyUSB** 這個設備文件。後面數字代表USB設備號，由於Ubuntu USB設備號為從零開始依次累加，所以多個設備每次開機後設備號是不固定的，需要確定設備的設備號。下面用兩張圖片來描述:

```
linux@ubuntu:~$ cd /dev
linux@ubuntu:/dev$ ls
agpgart           loop3             snapshot  tty33  tty7       ttyS8
autofs            loop4             snd       tty34  tty8       ttyS9
block             loop5             sr0       tty35  tty9       uhid
bsg               loop6             stderr    tty36  ttyprintk  uinput
btrfs-control     loop7             stdin     tty37  ttyS0      urandom
bus               loop-control      stdout    tty38  ttyS1      userio
cdrom             mapper            tty       tty39  ttyS10     vcs
cdrw              mcelog            tty0      tty4   ttyS11     vcs1
char              mem               tty1      tty40  ttyS12     vcs2
console           memory_bandwidth  tty10     tty41  ttyS13     vcs3
core              midi              tty11     tty42  ttyS14     vcs4
cpu_dma_latency   mqueue            tty12     tty43  ttyS15     vcs5
cuse              net               tty13     tty44  ttyS16     vcs6
disk              network_latency   tty14     tty45  ttyS17     vcs7
dmmidi            network_throughput tty15    tty46  ttyS18     vcsa
dri               null              tty16     tty47  ttyS19     vcsa1
dvd               port              tty17     tty48  ttyS2      vcsa2
ecryptfs          ppp               tty18     tty49  ttyS20     vcsa3
fb0               psaux             tty19     tty5   ttyS21     vcsa4
fd                ptmx              tty2      tty50  ttyS22     vcsa5
full              pts               tty20     tty51  ttyS23     vcsa6
fuse              random            tty21     tty52  ttyS24     vcsa7
hidraw0           rfkill            tty22     tty53  ttyS25     vfio
hpet              rtc               tty23     tty54  ttyS26     vga_arbiter
hugepages         rtc0              tty24     tty55  ttyS27     vhci
hwrng             sda               tty25     tty56  ttyS28     vhost-net
initctl           sda1              tty26     tty57  ttyS29     vhost-vsock
input             sda2              tty27     tty58  ttyS3      vmci
kmsg              sda5              tty28     tty59  ttyS30     vsock
lightnvm          sda6              tty29     tty6   ttyS31     zero
log               sda7              tty3      tty60  ttyS4
loop0             sg0               tty30     tty61  ttyS5
loop1             sg1               tty31     tty62  ttyS6
loop2             shm               tty32     tty63  ttyS7
linux@ubuntu:/dev$
```

上圖為沒有插入USB設備的情況，這個時候，dev目錄下並沒有名為 **ttyUSB** 文件，插入USB線，連接調試板，然後再次執行`ls`：

dev目錄下多了幾個文件名稱, 如圖：

```
linux@ubuntu:/dev$ ls
agpgart           loop3             shm       tty32  tty63      ttyS7
autofs            loop4             snapshot  tty33  tty7       ttyS8
block             loop5             snd       tty34  tty8
bsg               loop6             sr0       tty35  tty9       ttyUSB0
btrfs-control     loop7             stderr    tty36  ttyprintk
bus               loop-control      stdin     tty37  ttyS0      uinput
```

ttyUSB0 文件就是調試版在ubuntu系統中生成的設備文件，對它進行讀寫，就可以完成序列埠通信。這個文件名稱我們把它記下來。後面的數字是不固定的，有可能為 **ttyUSB1** 或 **ttyUSB2** 等。

波特率設置

在Ubuntu環境中，波特率支援到115200,460800,921600，本範例使用的是115200。

如果需要輸出幀率超過100Hz，則需要需要修改main.c文件中的options.c_cflag參數，改為更高的波特率。

```
83      options.c_cflag = B115200 | CS8 | CLOCAL |CREAD;
84      options.c_iflag = IGNPAR;
85      options.c_oflag = 0;
86      options.c_lflag = 0;
87      options.c_cc[VTIME] = 0;
88      options.c_cc[VMIN] = 0;
89      tcflush(fd,TCIFLUSH);
90      tcsetattr(fd, TCSANOW, &options);
91      return (fd);
```

如圖，在第83行，將B115200修改為B460800或者是B921600。

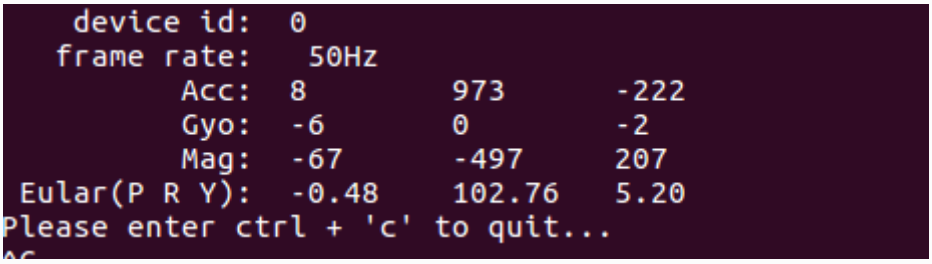
編譯並執行

我們開始在Ubuntu環境下生成一個可執行文件，專門用來解析模組的數據：

首先在Ubuntu系統中，按下 **ctrl + alt + t** 快捷鍵，在彈出的窗口上，執行`mkdir hipnuc` 建立hipunc目錄,然後將本文檔所在目錄下的所有文件複製到 **hipnuc** 目錄下。

執行`make`, 生成可自行文件`main`。 並執行`sudo ./main ttyUSB0`:

執行成功後，會出現這個畫面：

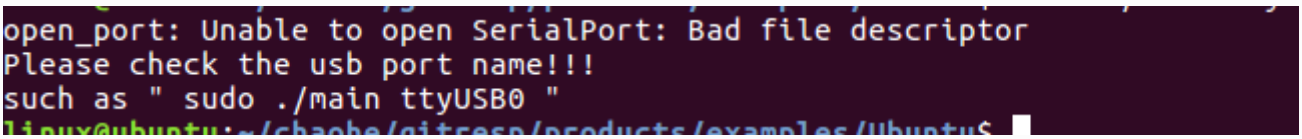


這個畫面上的數字會隨著模組位置的改變而發生變化。

如果後期修改了這些文件，需要執行`make clean`命令，進行清理之前生成的舊 **.o** 和 **main** 文件，然後再執行`make`，重新生成 **main** 這個可執行文件。

如果後期您需要在本路徑上添加其他文件，配合使用，請打開 **Makefile** 文件，在第一行的後面加上後添加文件的鏈接文件名，例如添加append_file.c文件，那麼在 **Makefile** 中第一行後面追加append_file.o文件名。如果後加的文件還需要鏈接第三方的庫，請在第二行的後面添加庫名字。格式為 **-l+lib_name** ("l" 是「L"的小寫的英文字母)。

如果出現：



表示未能找到序列埠，需要回到《**查找USB-UART設備**》**一節** 確認USB-UART設備已經被ubuntu識別。