

Ubuntu範例

原始碼:可直接下載 .zip 檔

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 mapper      tty38      ttyS1      userio
cdrom        mcelog    mem        tty39      ttyS10     vcs
cdrw         mmi        memory_bandwidth tty40      ttyS11     vcs1
char         midi       midi        tty41      ttyS12     vcs2
console      mq         mq          tty42      ttyS13     vcs3
core         net        net         tty43      ttyS14     vcs4
cpu_dma_latency network_latency network_throughput tty44      ttyS15     vcs5
cuse         null       null        tty45      ttyS16     vcs6
disk         port       port        tty46      ttyS17     vcs7
dmmidi       ppp        ppp         tty47      ttyS18     vcsa
dri          psaux     psaux       tty48      ttyS19     vcsa1
dvd          ptmx      ptmx        tty49      ttyS20     vcsa2
ecryptfs     pts       pts         tty50      ttyS21     vcsa3
fb0          random    random      tty51      ttyS22     vcsa4
fd           rfkill    rfkill      tty52      ttyS23     vcsa5
full         rtc       rtc         tty53      ttyS24     vcsa6
fuse         rtc0      rtc0        tty54      ttyS25     vcsa7
hidraw0      sda       sda         tty55      ttyS26     vfio
hpet         sda1     sda1        tty56      ttyS27     vga_arbiter
hugepages    sda2     sda2        tty57      ttyS28     vhci
hwrng        sda5     sda5        tty58      ttyS29     vhost-net
initctl      sda6     sda6        tty59      ttyS30     vhost-vsock
input        sda7     sda7        tty60      ttyS31     vmci
kmsg         sg0       sg0         tty61      ttyS32     vsock
lightnvme    sg1       sg1         tty62      ttyS33     zero
log          shm       shm         tty63      ttyS34     zero
loop0        shm       shm         tty64      ttyS35     zero
loop1        shm       shm         tty65      ttyS36     zero
loop2        shm       shm         tty66      ttyS37     zero
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       ttyS9
bsg          loop6      sr0        tty35      tty9       ttyUSB0
btrfs-control loop7      stderr     tty36      ttyprintk  uinput
bus          loop-control mapper      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。

鮑率設置

在Ubuntu環境中，鮑率支援到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，改為更高的鮑率。

```
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hipnuc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：

```
device id: 0
frame rate: 50Hz
  Acc:  8      973      -222
  Gyo: -6       0        -2
  Mag: -67     -497     207
Eular(P R Y): -0.48  102.76  5.20
Please enter ctrl + 'c' to quit...
^C
```

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地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：

```
open_port: Unable to open SerialPort: Bad file descriptor
Please check the usb port name!!!
such as " sudo ./main ttyUSB0 "
```

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。