

單元：Ubuntu 範例

原始碼(瀏覽器下載並解壓): [anrot_demo_ubuntu.zip](#)

介紹如何在 Ubuntu 環境中讀取 HI221/HI221Dongle/HI226/HI229/CH100/CH110/CH10x 的數據，提供了c代碼，用於讀取模組的數據。

- 測試環境：Ubuntu 16.04
- 測試設備：HI226 HI229

Step 1. 尋找 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** 打開終端機。
2. 終端機切換到 **dev** 目錄下，並列出所有檔案

```
1 | cd /dev
2 | ls
```

如下圖，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。

```

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  nqueue    tty12      tty43      ttyS15     vcs5
cuse         net        tty13      tty44      ttyS16     vcs6
disk         network_latency  tty14      tty45      ttyS17     vcs7
dmideid      network_throughput  tty15      tty46      ttyS18     vcsa
dri          null       tty16      tty47      ttyS19     vcsa1
dvd          port       tty17      tty48      ttyS20     vcsa2
ecryptfs     ppp        tty18      tty49      ttyS21     vcsa3
fb0          psaux      tty19      tty5       ttyS22     vcsa4
fd           ptmx       tty2       tty50      ttyS23     vcsa5
full         pts        tty20      tty51      ttyS24     vcsa6
fuse         random     tty21      tty52      ttyS25     vcsa7
hidraw0      rfkill     tty22      tty53      ttyS26     vfio
hpet         rtc        tty23      tty54      ttyS27     vga_arbiter
hugepages    rtc0       tty24      tty55      ttyS28     vhci
hwrng        sda        tty25      tty56      ttyS29     vhost-net
initctl      sda1       tty26      tty57      ttyS30     vhost-vsock
input        sda2       tty27      tty58      ttyS31     vmci
kmsg         sda5       tty28      tty59      ttyS32     vsock
lightnvm     sda6       tty29      tty6       ttyS4      zero
log          sda7       tty3       tty60      ttyS5
loop0        sg0        tty30      tty61      ttyS6
loop1        sg1        tty31      tty62      ttyS7
loop2        shm        tty32      tty63
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，連接模組，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  stdin      tty37      ttyS0

```

`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。

Step 2. 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。

Step 3. 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，按下打開終端機，到下載好的原始碼目錄下

```

1 | cd CH_demo_ubuntu
2 | make
3 | 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...
AC

```

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，需要清理之前生成的舊 `.o` 和 `main` 文件，重新生成 `main` 這個可執行文件。

```

1 | make clean
2 | make

```

- 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 "
linux@ubuntu:~/chaobe/gitrepo/products/examples/Ubuntu$

```

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