

第五章 KUBOT 下位機參數調整與校正

本章將說明由 ROS 上位機調整 KUBOT 下位機，引用 rqt 動態參數調整，這些參數攸關於整車安全性，調整時請多留意。若覺得 KUBOT 的馬達行走有偏差，可以使用本章方法校正下位機。

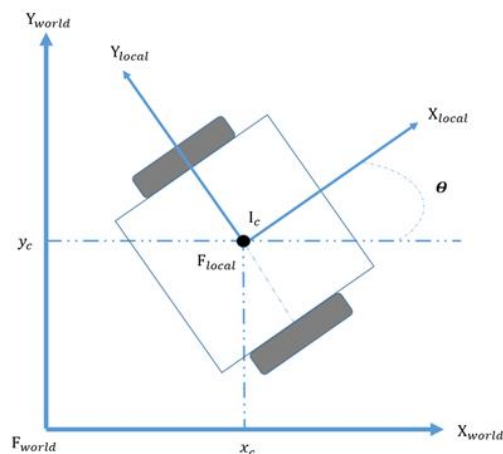
本章為高手境界，攸關整車行為精確度、穩定度、安全性以及產品壽命，一經調整本司不負任何保固責任。

事前準備：

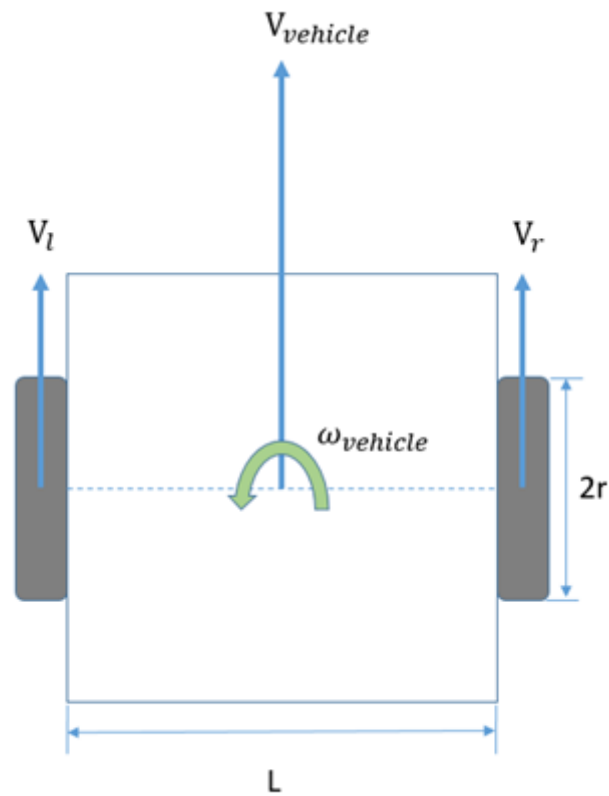
1. 充好電的 KUBOT 小車一台
2. Linux 環境電腦一台
3. 清醒的大腦
4. 將 KUBOT 架高
5. 捲尺

5-1 運動模型說明

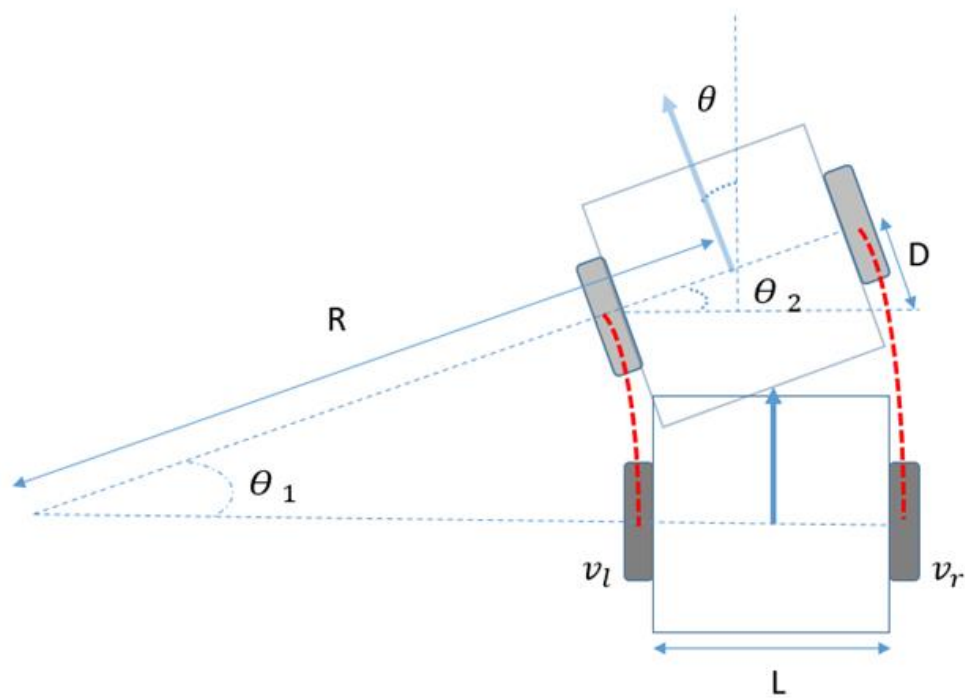
運動解算兩輪差分模型，自由度僅 2，X 軸往前或往後，繞著 Z 軸順時針或逆時間 Yaw 旋轉，因此 Y、Z 方向位移、Roll、Pitch 均為 0。



決定運動模型之變數為 輪徑、馬達(input)、兩輪中心距。



兩輪差速可以實現曲線移動，決定於差分角變化。



因此定義下位機運動模型與里程計代碼如下：

```
void get_odom(struct Odom* odom, float* motor_dis, unsigned long interval)
{
    float dxy_ave = (-wheel_diameter [0] + wheel_diameter [1]) / 2.0;
    float dth = (wheel_diameter [0] + wheel_diameter [1]) / (2* wheel_track);
    float vxy = 1000 * dxy_ave / interval;
    float vth = 1000 * dth / interval;
```

```
    dx = cos(dth) * dxy_ave;
    dy = -sin(dth) * dxy_ave;
    odom->x += (cos(odom->z) * dx - sin(odom->z) * dy);
    odom->y += (sin(odom->z) * dx + cos(odom->z) * dy);
    odom->z += dth;
```

需正規化之變數為 wheel_diameter 與 wheel_track

其中 wheel_diameter 由輪徑定義，由代碼可知，小車 X 方向位移影響之變數僅輪徑，
另 wheel_diameter [0]+ wheel_diameter [1]為 0 讓小車往前直走，因次第一階段先測試往前直走。
確定 wheel_diameter 變數後，再解 body_radius(兩輪中心距)即可。

下位機變數表

Wheel_diameter	輪徑
Wheel_track	輪距 (兩輪差分模型)
Encoder_resolution	編碼器解析度
Do_pid_interval	PID 計算的間隔時間 建議 10
Kp	P
Ki	I
Kd	D
ko	PID 係數 (kp/ko ki/ko kd/ko)
Cmd_last_time	超過該時間沒有新的命令，下位機會進入休眠
Max_v_linear_x	馬達下位機限速，X 方向直線速度，需考慮馬達極限，否則馬達會燒毀 單位 cm/s
Max_v_linear_y	馬達下位機限速，Y 方向直線速度，由上面的運動解可知，兩輪差分為 0
Max_v_angular_z	馬達下位機限速，Z 方向旋轉，需考慮馬達極限，否則馬達會燒毀
Imu_type	固定 69 (此下位機解用 GY85)

5-2 基礎參數調整

此調整方法僅調適下為機 odom，並沒有使用 ekf 演算法，因此建議在空曠平坦的場地測試。

Step.1 開啟 KUBOT 小車。

Step.2 遠端 KUBOT 小車請參考第三章，進入 SSH。亦可以使用外接螢幕、鍵盤滑鼠進入車端。

Step.3 第一個終端機啟動 KUBOT 下位機。(必須經過 SSH 或車端)

kubot_bri

```
[ INFO] [1571369479.932271050]: connected to main board
[ INFO] [1571369481.932705060]: end sleep
[ INFO] [1571369481.939605008]: robot version:v1.0.1 build time:20180329
[ INFO] [1571369481.943575581]: subscribe cmd topic on [cmd_vel]
[ INFO] [1571369481.984096518]: advertise odom topic on [odom]
[ INFO] [1571369482.057768133]: RobotParameters: 109 158 800 10 75 2500 0 10 250 40 1 150 69
```

其實要讓小車動作都會使用到此 launch 檔，最後一行即為下發下位機之參數，RobotParameter。若最後一行並非變數，而是其他，必定會跳警告並無法啟動小車。

詳細看附件問題集。

Step.4 第二個終端機啟動線性調整腳本。(必須經過 SSH 或車端)，此腳本會發送馬達控制命令。

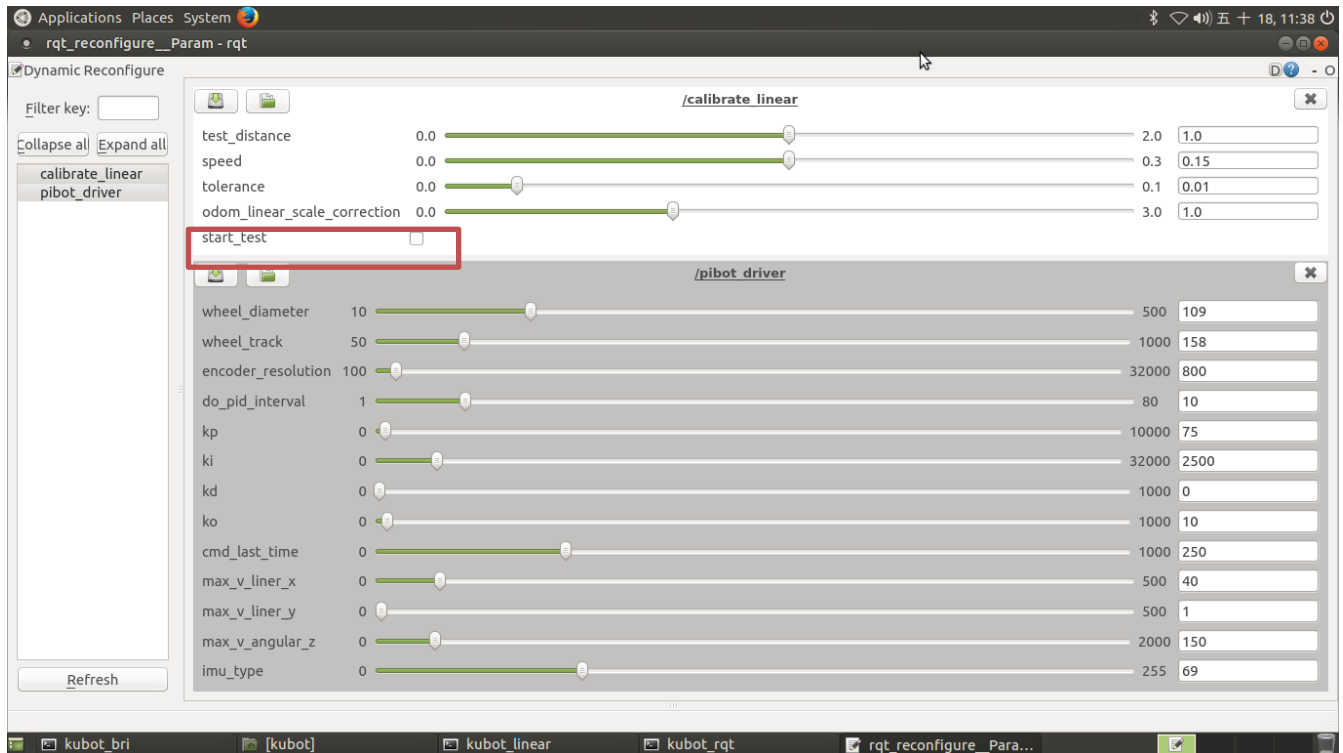
kubot_linear

```
File Edit View Search Terminal Help
zsh: corrupt history file /home/kubot/.zsh_history
kubot@kubot ~ 1144 11:36:51
zsh: corrupt history file /home/kubot/.zsh_history
kubot@kubot ~ kubot_lindar 1266 11:36:53
zsh: command not found: kubot_lindar
kubot@kubot ~ kubot_linear 127 1267 11:36:57
[INFO] [1571369831.575592]: Bring up rqt_reconfigure to control the test.
```

Step.5 第三個終端機啟動 rqt 動態參數調整。(本地從者或車端)

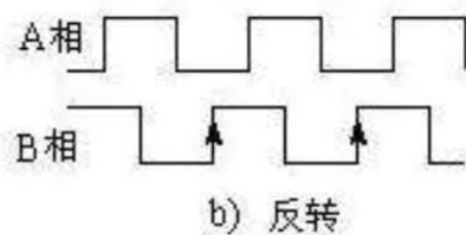
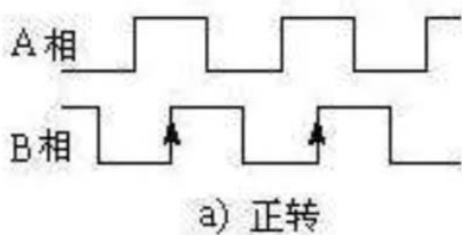
kubot_rqt

KUBOT 用戶手冊



即可看到上圖，左邊有兩個項目可以選擇，一個是 X 方向行走命令控制 calibrate，一個是 kubot_driver。原則上小車出廠均調校好基本參數。若無則參照此圖的初步參數調整。

Step.6 首先驗證馬達解析度，小車出廠 datasheet 有註明所使用的馬達齒輪箱及編碼器。



For example :

IG32 04type 139 比+encoder7N14P2C

即為減速比 139 的直流有刷馬達，並使用 7 個極性，旋轉一圈產生 14 個脈衝，2 通道通訊，馬達運轉 A B 相永遠相差 90°並且用 4 倍頻計算，因此解析度為：

$$\text{Encoder_resolution} = 4 \times 139 \times 7 = 3892$$

將此參數 key 入 driver。

KUBOT

用戶手冊

Step.7 此時可以將小車放在空曠且平坦的地面上，選擇車體一個地方作為參考點並鋪好捲尺，按下 `start_test` 小方塊小車就會往前行走 100cm(理論值)，若小車參數誤差過大導致無限往前請關閉程式或斷電重來。

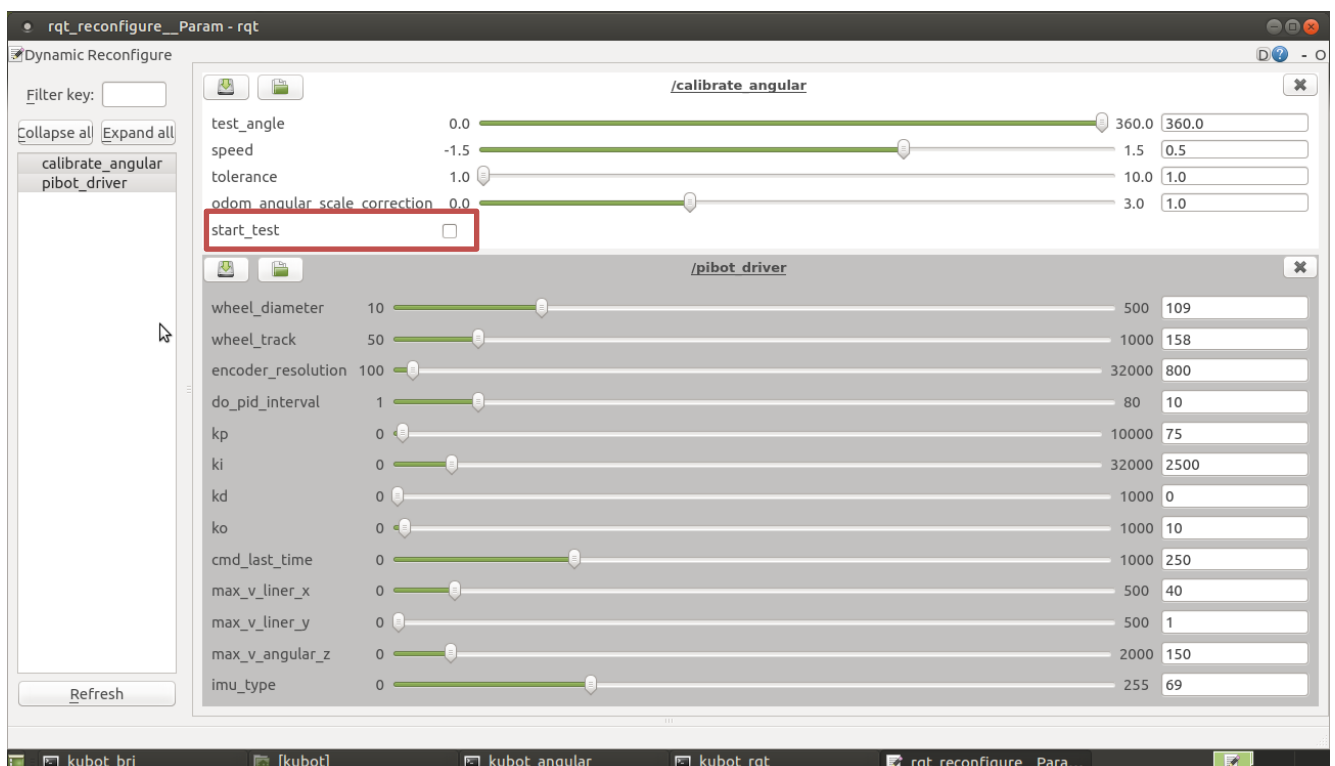


Step.8 理論與實際肯定不同，此時需做正規化，將下位機參數驅動調整。若下位機計算出來的里程計 $> 100\text{cm}$ ，則 `Wheel_diameter` 應調小，反之里程計 $< 100\text{cm}$ ，則 `Wheel_diameter` 應調大。

Step.9 不斷反覆 Step7~8，直到線性誤差在 $\pm 1\text{cm}$ 以內。

`Wheel_diameter` 調整好後完成第一階段測試，接著解 Z 軸旋轉，即定義 `Wheel_track`。

Step.10 關閉 `kubot_rqt` 與 `kubot_linear`，並開啟 `kubot_angular` 並再開一次 `kubot_rqt`。



Step.11 同樣找一個參照點並按下 `start_test` 小方塊，小車會開始旋轉 360 度(理論值)。

Step.12 同樣理論與實際肯定不同，若旋轉一圈後 $> 360^\circ$ 代表理論的車體半徑過大，wheel_track 應調小，反之 $< 360^\circ$ 則應調大。

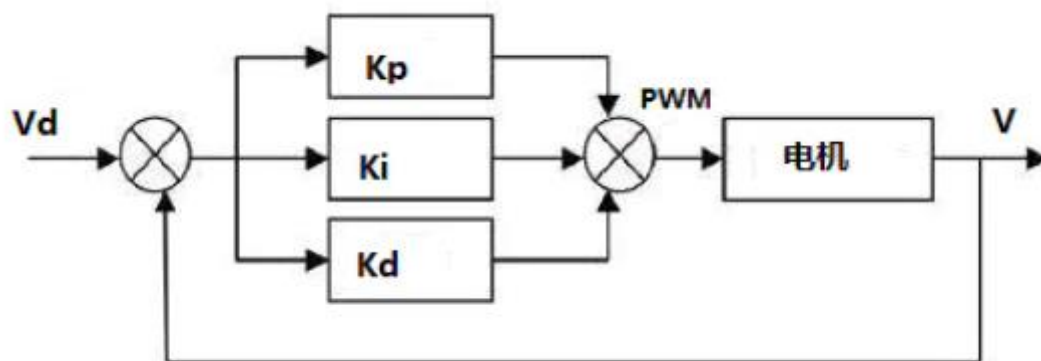
Step.13 反覆調適 Step11~12 直到偏差角 $\pm 1^\circ$ 。

5-3 PID 參數調整

PID 為世界三大美女..... 由左至右 P 比莉、I 維芬、D 姬芬



若不知道何為 PID 控制不建議調整此組參數。請直接跳過本章。



P 可以調整響應時間、I 可以改善穩態誤差、D 可以改善暫態響應

Step.1 開啟 KUBOT 小車。

Step.2 遠端 KUBOT 小車請參考第三章，進入 SSH。亦可以使用外接螢幕、鍵盤滑鼠進入車端。

Step.3 第一個終端機啟動 KUBOT 下位機。(必須經過 SSH 或車端)

```
kubot_bri
```

Step.4 第二個終端機啟動鍵盤遙控。(遠端或車端)

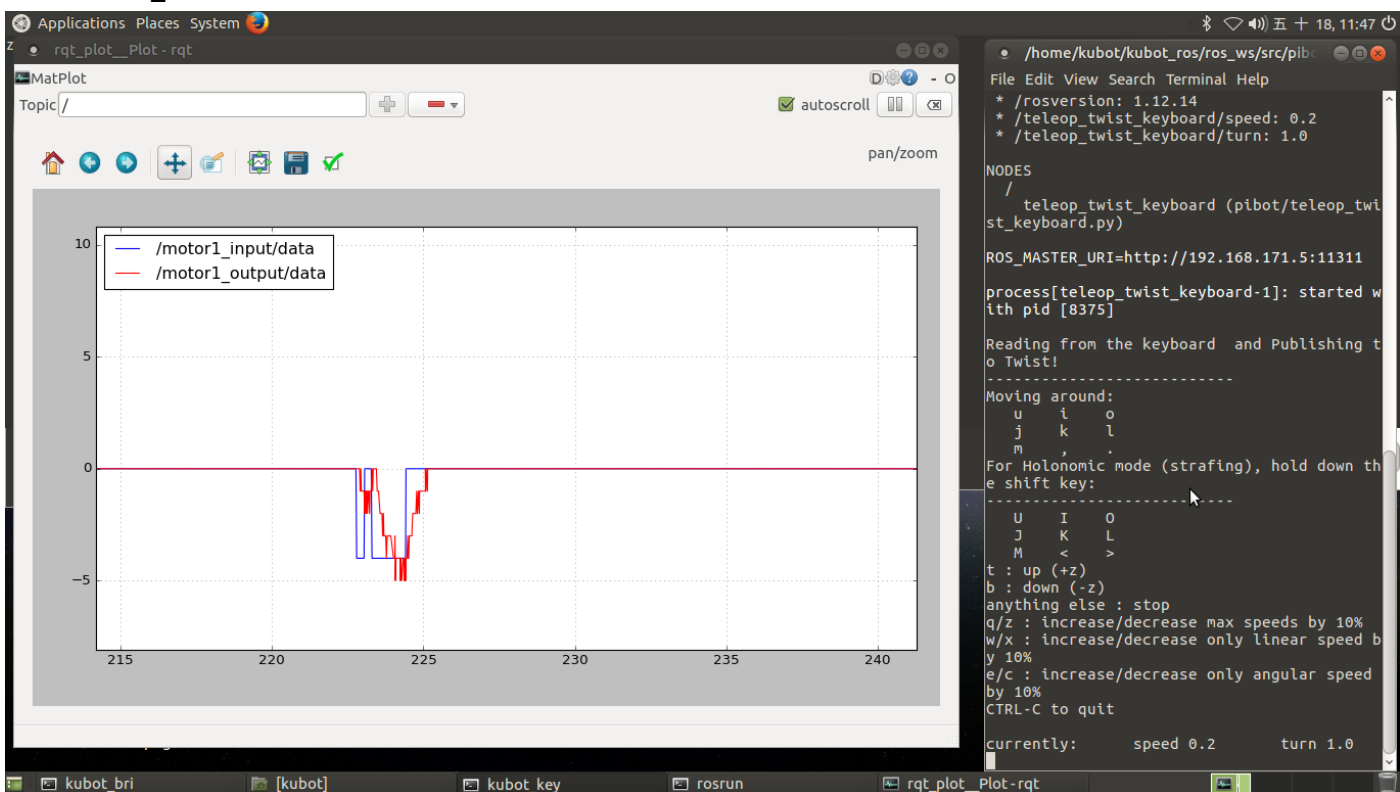
kubot_key

Step.5 將 KUBOT 小車架起來，讓輪子離開地面

Step.6 第三個終端機，呼叫圖形化節點。(遠端或車端)

roslaunch rqt_plot rqt_plot /motor1_input /motor1_output

試著下達馬達方向控制 即可看到馬達輸入、輸出響應圖：(更改為 motor2 即可看另一顆馬達)
或是下達 cmd_vel X Y Z 讓馬達持續轉動



可以再開一個終端機呼叫 rqt 動態參數 pibot_driver 調整，微調 kp、ki、kd 三個值直到調出理想的馬達運動動態曲線。

筆者建議：不要調整 kd，非常容易導致馬達、保險絲燒壞

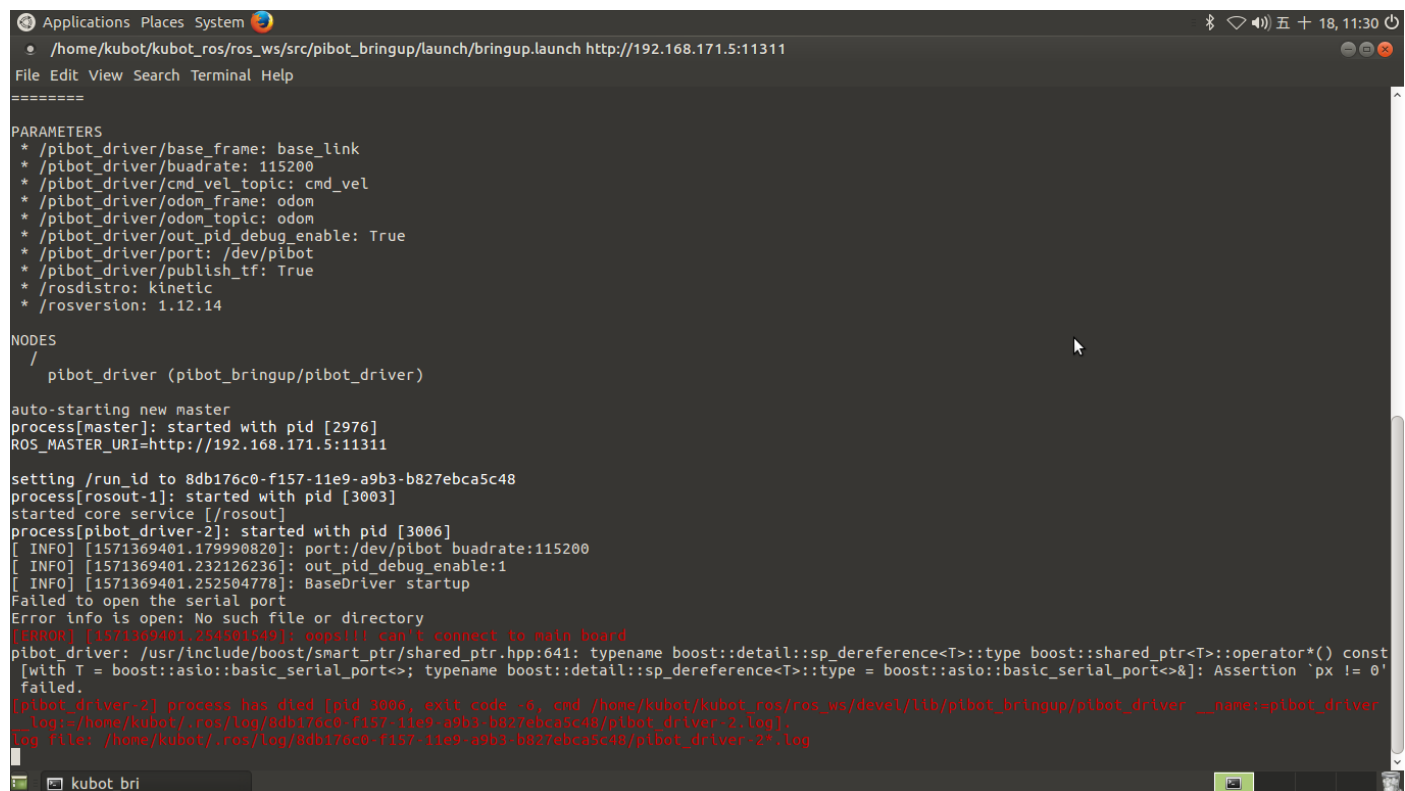
KUBOT

用戶手冊

ISSUE 集

執行 kubot_bri 時會做下位機檢查，

終端機畫面若如下：



```
Applications Places System
/home/kubot/kubot_ros/ros_ws/src/pibot_bringup/launch/bringup.launch http://192.168.171.5:11311
File Edit View Search Terminal Help
=====
PARAMETERS
* /pibot_driver/base_frame: base_link
* /pibot_driver/buadrate: 115200
* /pibot_driver/cmd_vel_topic: cmd_vel
* /pibot_driver/odom_frame: odom
* /pibot_driver/odom_topic: odom
* /pibot_driver/out_pid_debug_enable: True
* /pibot_driver/port: /dev/pibot
* /pibot_driver/publish_tf: True
* /roscdistro: kinetic
* /rosversion: 1.12.14


NODES
/
  pibot_driver (pibot_bringup/pibot_driver)

auto-starting new master
process[master]: started with pid [2976]
ROS_MASTER_URI=http://192.168.171.5:11311

setting /run_id to 8db176c0-f157-11e9-a9b3-b827ebca5c48
process[rosout-1]: started with pid [3003]
started core service [/rosout]
process[pibot_driver-2]: started with pid [3006]
[ INFO] [1571369401.179990820]: port:/dev/pibot buadrate:115200
[ INFO] [1571369401.232126236]: out_pid_debug_enable:1
[ INFO] [1571369401.252504778]: BaseDriver startup
Failed to open the serial port
Error info is open: No such file or directory
[ERROR] [1571369401.254581549]: oops!!! can't connect to main board
pibot_driver: /usr/include/boost/smart_ptr/shared_ptr.hpp:641: typename boost::detail::sp_dereference<T>::type boost::shared_ptr<T>::operator*() const
[with T = boost::asio::basic_serial_port<>; typename boost::detail::sp_dereference<T>::type = boost::asio::basic_serial_port<>]: Assertion 'px != 0'
failed.
[pibot_driver-2] process has died [pid 3006, exit code -6, cmd /home/kubot/kubot_ros/ros_ws/devel/lib/pibot_bringup/pibot_driver __name:=pibot_driver
__log:=/home/kubot/.ros/log/8db176c0-f157-11e9-a9b3-b827ebca5c48/pibot_driver-2.log].
Log file: /home/kubot/.ros/log/8db176c0-f157-11e9-a9b3-b827ebca5c48/pibot_driver-2*.log
```

可能原因：	解決方法
樹莓派沒有連接下位機	檢查下位機 USB 是否正常插入樹梅派
下位機的 USB 壞了	換一條 USB 傳輸線
下位機壞了	返廠維修

在終端機輸入 `ls /dev/pibot -l` 應回覆如下圖：



```
LSM: corrupt history file /home/kubot/.LSM_history
kubot@kubot ~$ ls /dev/pibot -l
lrwxrwxrwx 1 root root 7 18 11:36 /dev/pibot -> ttyUSB0
kubot@kubot ~$
```

若不是，請拔掉所有 USB 後重插。或定義 USB kernel、rule

用戶手冊

若如下圖：

[illegible]

可能原因：	解決方法
與下位機波特率錯誤	依照第二章重新設置基礎環境
IMU 故障	換一顆 GY85 返廠維修
馬達未接	檢查馬達接線是否有接入下位機

若如下圖：

[illegible]

可能原因：	解決方法
IMU 故障	換一顆 GY85

KUBOT

用戶手冊

	返廠維修
--	------

若測試 kubot_linear 時小車均無法直線行走(些微誤差正常)、或是偏差度太大。

可能原因：	解決方法
馬達壽命已到	返廠維修
馬達過度使用	返廠維修
輪子上有異物(髒污、口香糖等)	手動處理
地面打滑	換場地試試

若測試 kubot_angular 時小車均無法原地旋轉(些微誤差正常)、或是偏差角度太大。

可能原因：	解決方法
馬達壽命已到	返廠維修
馬達過度使用	返廠維修
輪子上有異物(髒污、口香糖等)	手動處理
地面打滑	換場地試試