2.1.6 Obtain drive board data

In addition to the control function, the driver board also carries out data transmission. This requires the agreement we provided.

The following is the data that can be Reported.

1) Turn on the report switch, which is used for remote programs
2) Reporting encoder data
3) Reporting voltage data
4) Pitch
5) Roll
6) Yaw

Library API.

BoardData_Get(index)

Range of index is 0~9.

0 -- Start Reporting
1~4 -- Four encoder
5 -- voltage data
6 -- Pitch
7 -- Roll
8 -- Yaw
9 -- Stop Reporting

Code path:

/home/pi/Yahboom_Project/2.Hardware_Control_course/6_Obtain_data.ipynb

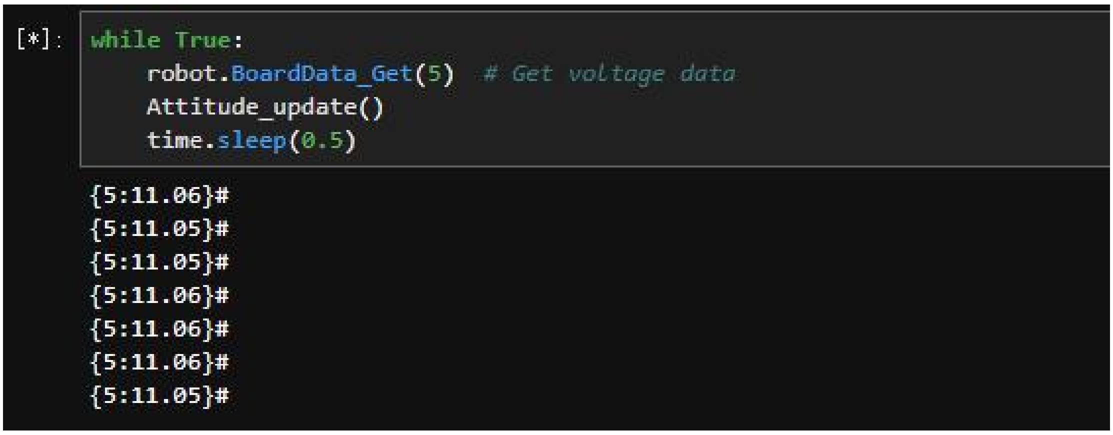
```
# Import Raspblock drive library
from Raspblock import Raspblock
robot = Raspblock()
```

```
import RPi.GPIO as GPIO
import time
import string
import serial

# Open serial
ser = serial.Serial("/dev/ttyAMA0", 115200)
ser.flushInput()
def Attitude_update():
    # Get receive buffer character
    count = ser.inWaiting()
    if count != 0:
        recv = list(ser.read(count))
        recv = str(bytes(recv), encoding='UTF-8')
        if( recv.find("{") != -1 and recv.find("}#") != -1 ):
            print(recv)
    #reg                                                                  =
re.compile('^{A(?P<Pitch>[^ ]*):(?P<Roll>[^ ]*):(?P<Yaw>[^ ]*):(?P<Voltage>[^ ]*)}#')

    # Clear receive buffer
    ser.flushInput()
```

```
while True:
    robot.BoardData_Get(5)    # Get voltage
    Attitude_update()
    time.sleep(0.5)
```

```
del robot    #The object needs to be released after use, otherwise, when the next program needs to use this object module, it will be occupied and will become unusable
```

As show below.



```
[*]: while True:
         robot.BoardData_Get(5)  # Get voltage data
         Attitude_update()
         time.sleep(0.5)

     {5:11.06}#
     {5:11.05}#
     {5:11.05}#
     {5:11.06}#
     {5:11.06}#
     {5:11.06}#
     {5:11.05}#
```