

payload on RaspberryPi

Tags

安裝payload sdk

Payload SDK

<https://developer.dji.com/doc/payload-sdk-tutorial/cn/quick-start/run-sample-code.html>

document

[Raspberry Pi] 啟用Raspberry Pi 3B+ 序列埠功能

Raspberry Pi 3B+內建硬體式的PL011 UART，與半硬體式的Mini UART，本文介紹兩種UART的差異，以及如何開啟Raspberry Pi 3B+序列通訊介面，進一步將原本用於藍芽的PL011 UART與Mini

 <https://dumbcatnote.blogspot.com/2020/04/raspberry-pi-enable-serial-port.html>

```
W-Pi-1 root disk 1, 6 Apr 22 11:15 ram6
W-Pi-1 root disk 1, 7 Apr 22 11:15 ram7
W-Pi-1 root disk 1, 8 Apr 22 11:15 ram8
W-Pi-1 root disk 1, 9 Apr 22 11:15 ram9
W-Pi-1 root disk 1, 8 Apr 22 11:15 random
W-Pi-1 root root 60 Jan 1 1970 raw
W-Pi-1 root root 58 Apr 22 11:15 rkill1
W-Pi-1 root root 7 Apr 22 11:15 serial1 -> ttyAMA0
W-Pi-1 root root 40 Nov 4 2016
W-Pi-1 root root 160 Apr 22 11:15 snd
W-Pi-1 root root 15 Nov 4 2016 stderr -> /proc/self/fd/2
W-Pi-1 root root 15 Nov 4 2016 stdin -> /proc/self/fd/0
W-Pi-1 root root 15 Nov 4 2016 stdout -> /proc/self/fd/1
W-Pi-1 root tty 5, 0 Apr 22 11:15 tty
W-Pi-1 root tty 4, 0 Apr 22 11:15 tty0
W-Pi-1 root tty 4, 1 Apr 22 11:15 tty1
W-Pi-1 root tty 4, 10 Apr 22 11:15 tty10
W-Pi-1 root tty 4, 11 Apr 22 11:15 tty11
W-Pi-1 root tty 4, 12 Apr 22 11:15 tty12
W-Pi-1 root tty 4, 13 Apr 22 11:15 tty13
W-Pi-1 root tty 4, 14 Apr 22 11:15 tty14
```

```
#define LINUX_UART_DEV1 "/dev/ttyS0"
```

Raspberry Pi Pinout

The comprehensive add-on boards & GPIO Pinout guide for the Raspberry Pi

 https://pinout.xyz/pinout/pin22_gpio25

A+ Zero

- 在 `samples/sample_c/platform/linux/manifold2/application/dji_sdk_app_info.h` 文件中替换应用的名称、ID、Key、License、开发者账号和指定波特率。

```
#define USER_APP_NAME "your_app_name"
#define USER_APP_ID "your_app_id"
#define USER_APP_KEY "your_app_key"
#define USER_APP_LICENSE "your_app_license"
#define USER_DEVELOPER_ACCOUNT "your_developer_account_email"
#define USER_BAUD_RATE "460800"
```

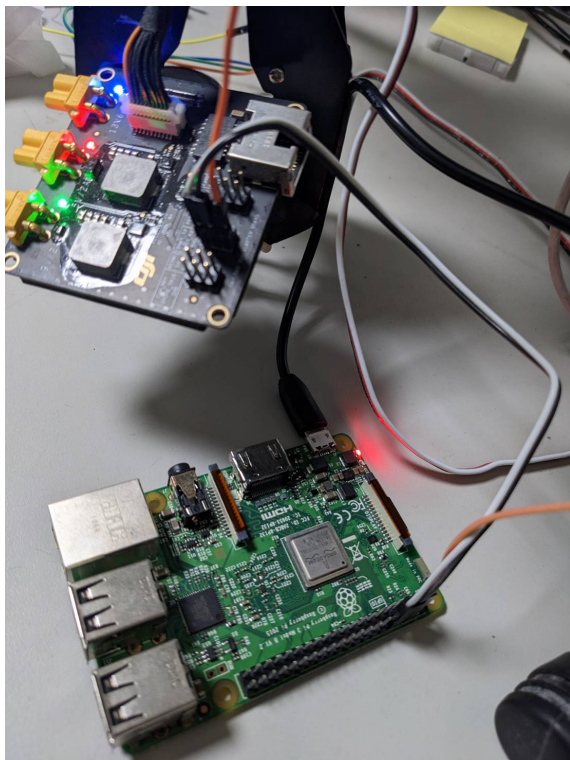
- 在 `samples/sample_c/platform/linux/manifold2/hal/hal_uart.h` 文件的 `LINUX_UART_DEV1` 和 `LINUX_UART_DEV2` 宏中填写对应的串口名称。

```
#define LINUX_UART_DEV1    "/dev/ttyS0"
#define LINUX_UART_DEV2    "/dev/your_com"
```

- 通过 `ifconfig` 命令，查看当前与无人机通讯的网口设备名称，并填写到 `samples/sample_c/platform/linux/manifold2/hal/hal_network.h` 文件的 `LINUX_NETWORK_DEV` 宏中。

```
#define LINUX_NETWORK_DEV    "wlan0"
```

```
git clone https://github.com/dji-sdk/Payload-SDK.git
cd Payload-SDK/samples/sample_c/platform/linux/manifold2
mkdir build
cd build
cmake ..
make
sudo build/bin/dji_sdk_demo_linux
```



skyport → RX ⇒ RBPi → TX(port8)
skyport → TX ⇒ RBPi → RX(port10)
skyport → GND ⇒ RBPi → GND(port6)

Gripper

motor → data ⇒ RBPi → GPIO26(port32)

[树莓派系列] 使用WiringPi库入门模拟舵机-SG90(C和Python)

难度：★★ 读者：适合有 C语言或 Python 编程基础，对舵机和 PWM等有一定了解的读者。往期相关文章：没有安装WiringPi的读者，可先阅读往期文章，搭建好环境。本文将介绍如何在树莓派4B上

[知 https://zhuanlan.zhihu.com/p/267042615?utm_id=0](https://zhuanlan.zhihu.com/p/267042615?utm_id=0)



1. `$ gpio mode 1 pwm # 设置GPIO1为PWM输出脚`
2. `$ gpio pwm-ms # 切换到占空比、传统模式`
3. `$ gpio pwmc 192 # 设置时钟分频`
4. `$ gpio pwmr 2000 # 每个刻度 0.01 ms, 2000 * 0.01ms = 20ms`
5. `$ gpio pwm 1 50 # 0.5 ms (-90°) => Maxrange * 占空比 = 2000 * 2.5% = 50`
6. `$ gpio pwm 1 100 # 1.0 ms (-45°) => 2000 * 5% = 100`
7. `$ gpio pwm 1 150 # 1.5 ms (0°) => 2000 * 7.5% = 150`
8. `$ gpio pwm 1200# 2.0 ms (45°) => 2000 * 10% = 200`



舵机的频率一般为频率为50HZ，也就是一個20ms左右的時基脈衝，而脈衝的高電平部分一般為0.5ms-2.5ms範圍。所以-90°⇒0.5ms, 90°⇒2.5ms，

Maxrange * 占空比

公式：

$$2000 \cdot \frac{\left(\frac{2ms}{180^\circ}\right) \cdot deg + 0.5}{20ms}$$