



Motor Driver HAT

用户手册

产品概述

Motor Driver HAT 是专为树莓派控制电机而设计的，使用 I2C 接口控制，适用于驱动小车电机。

特点

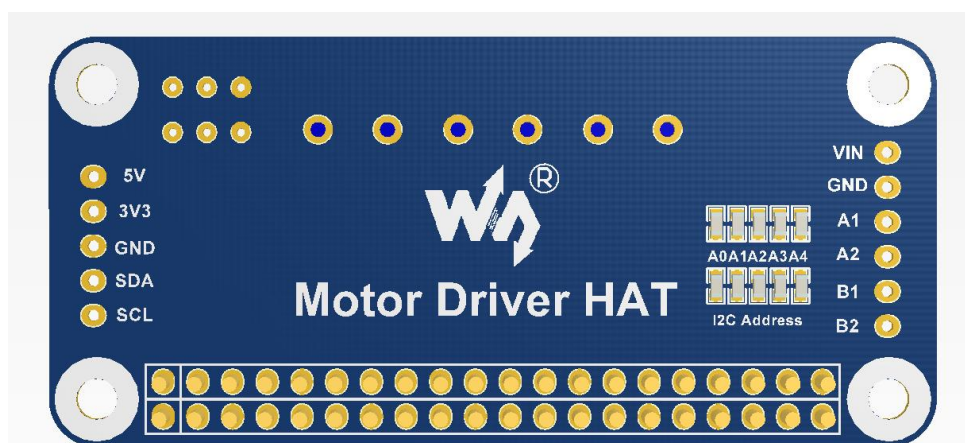
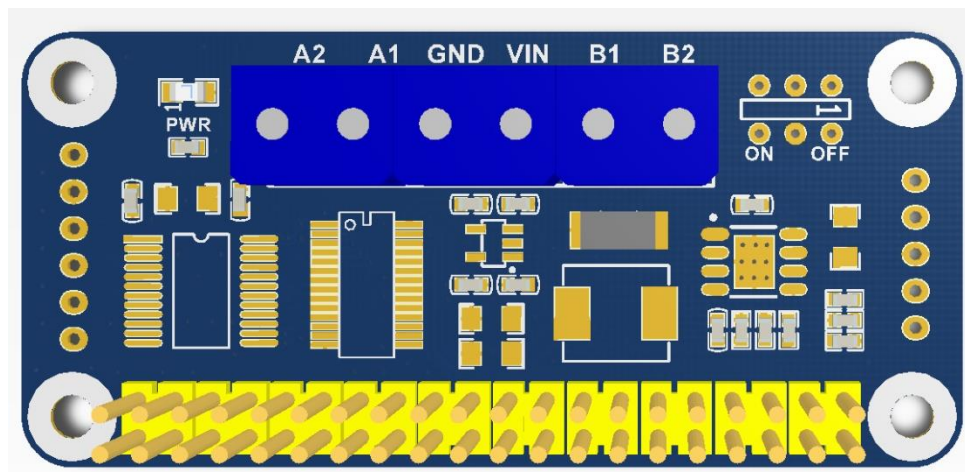
- 基于树莓派接口设计，适用于 Raspberry Pi Zero/Zero W/Zero WH/2B/3B/3B+
- I2C 接口控制，通过改变 5 个地址跳线可以同时接多达 32 个模块
- 板载 PCA9685 芯片，可输出 12 位硬件 PWM 控制电机转速
- 板载 TB6612FNG 双 H 桥电机驱动芯片，工作高效，不易发热
- 板载 5V 稳压芯片，输出可达 3A 电流，可通过 VIN 端子接入电池供电
- 预留 I2C 控制接口，方便接入其他主控板
- 提供完善的配套资料手册(提供 BCM2835、wiringPi 与 python 例程)

产品参数

| | |
|-----------|------------------|
| 供电电压: | 6V ~ 12V(VIN 端子) |
| 逻辑电压: | 3.3V |
| PWM 驱动芯片: | PCA9685 |
| PWM 芯片接口: | I2C |
| 电机驱动芯片: | TBA6612FNG |
| 产品尺寸: | 65mm x 30mm |
| 固定孔通径: | 3.0mm |

接口说明

模块实物图如下：

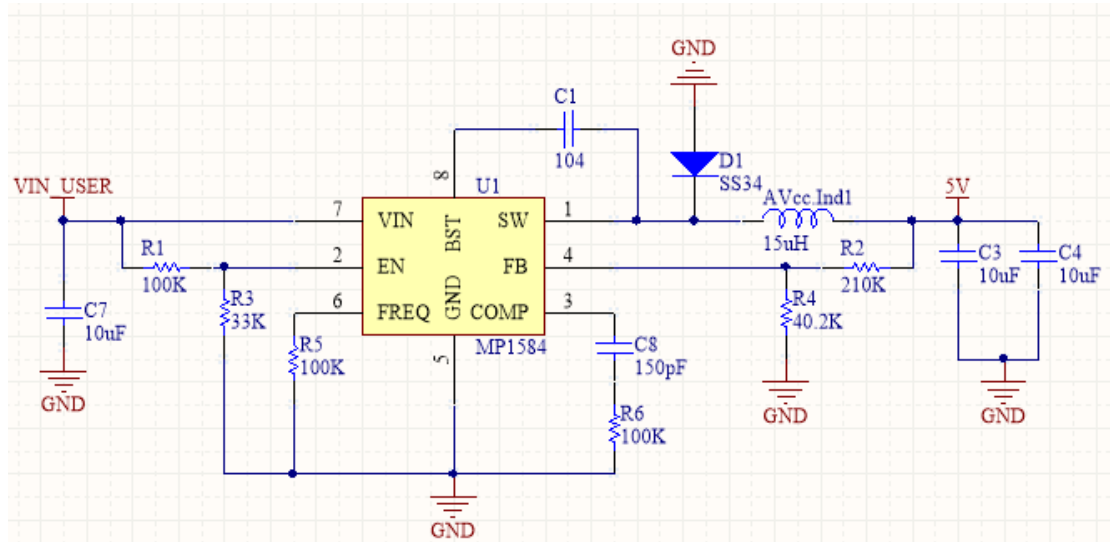


| 标识 | 描述 |
|-----|----------------|
| 5V | 5V 电源 |
| 3V3 | 3.3V 电源 |
| GND | 电源地 |
| SDA | I2C 数据线 |
| SCL | I2C 时钟线 |
| VIN | 电机驱动电压 (6-12V) |
| A1 | 电机 A 正极 |
| A2 | 电机 A 负极 |
| B1 | 电机 B 正极 |
| B2 | 电机 B 负极 |

硬件资源

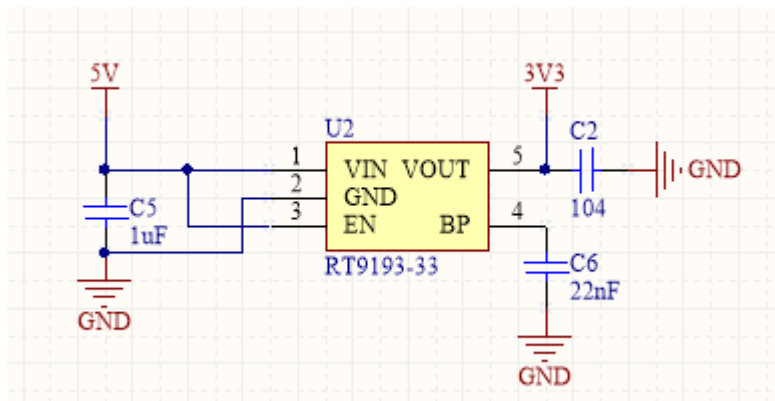
硬件上实际上由三部分构成：电源、PWM、电机驱动

电源



输入端采用 MP1584 稳压芯片，该芯片支持 4.5V 到 28V 的宽电压输入，输出电流达 3A。虽然芯片支持最高 28V 的输入，但是由于给电机的工作电压也由 VIN_USER 供给，实际上输入电压最好保持在 6-12V。

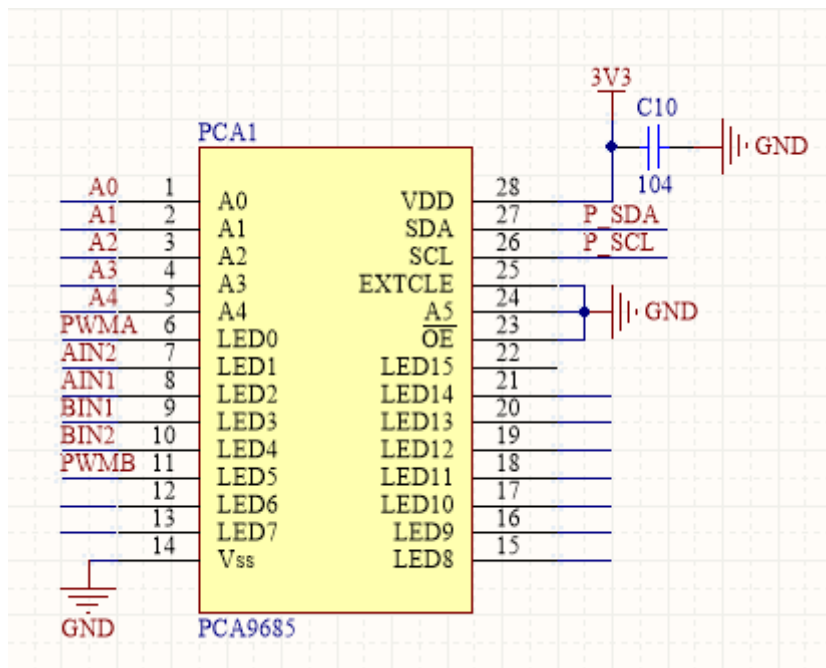
MP1584 稳压芯片输出 5V，这个 5V 用于给树莓派供电，然后 5V 再通过 RT9193-33 转成 3.3V 电平，该电平给 PWM、电机驱动这两部分提供逻辑电压



PWM

由于树莓派硬件 PWM 只有一个引脚 GPIO.1，而且，虽然 wiringPi 库与 python 都有软件 PWM，但是会占用部分 CPU 资源，因此本模块采用 PCA9685 芯片，该芯片使用 I2C 控制，可输出 16 路 12 位分辨率的 PWM，可控制输出频率 40HZ-1000HZ。

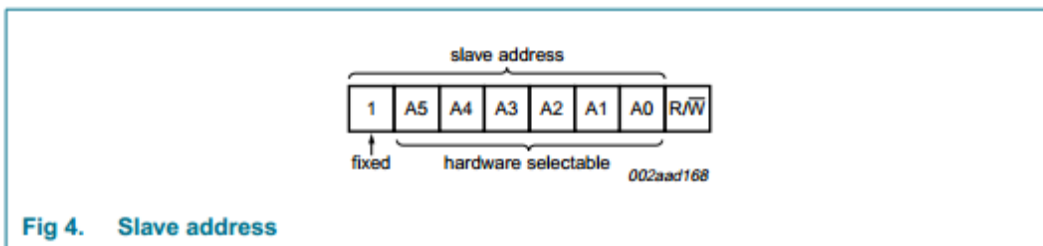
操作简单，只需要给芯片控制对应寄存器的值，即可一直输出 PWM 信号。



根据上面的原理图，我们可以看出 LED0-LED5 为控制电机驱动芯片管脚。

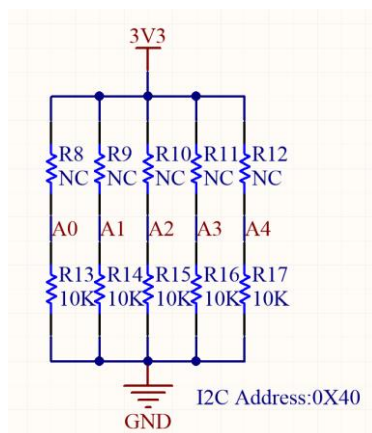
I2C 通信

查看数据手册 page6 – page8 可知：



I2C 从设备地址有 7 位，加上 1 位读写位，最高位固定为 1，A5-A0 可以通过硬件引脚配置。

在此模块中，默认 A5 接至地，可通过焊接或移除 A0-A4 电阻，来控制设备地址，焊接电阻为位 1，不焊接为位 0，I2C 设备地址设置范围为：0x40 到 0x5F



示例：

1. A5 为 0，A0-A4 不焊接为 0，那 I2C 设备地址为 0x40
2. A0-A4 全部焊接 0 欧姆电阻或者短接，在树莓派中可以使用 i2cdetect 工具，扫描 I2C 总线上的全部设备；

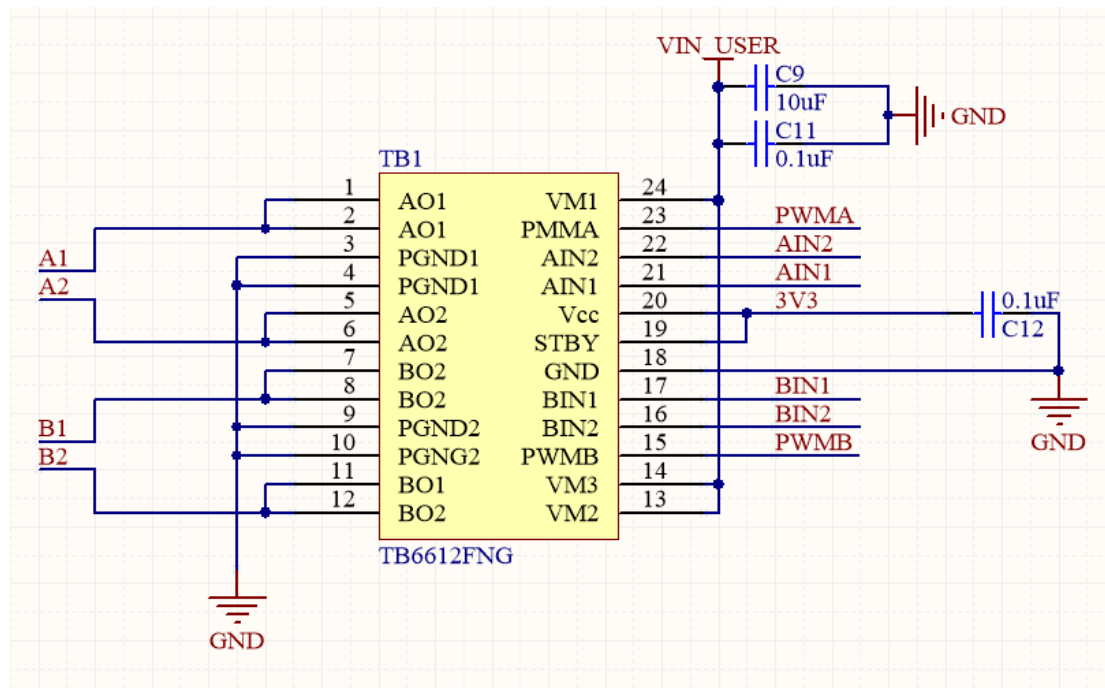
运行 `i2cdetect -y 1`，得到 I2C 从设备地址 0x5F：

```

pi@raspberrypi:~/c[redacted]/Motor_Driver_HAT_code/python $ i2cdetect -y 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- 5f
60:  -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- --

```

电机驱动



TB6612FNG 是一款双 H 桥电机驱动芯片，工作高效，不易发热。

VIN_USER 为输入电压，理论上该电压越大电机的转速也越快，建议输入控制在 6-12V。

其中 PWMA 与 PWMB 为控制两个电机的转速，AIN1 与 AIN2、BIN1 与 BIN2 分别控制两个电机的旋转方向。

A1 与 A2、B1 与 B2 分别接两个电机的正负极。

使用指南

打开 I2C 接口

执行如下命令进行树莓派配置：

```
sudo raspi-config
```

选择 Interfacing Options -> I2C ->yes 启动 i2C 内核驱动

```
Raspberry Pi Software Configuration Tool (raspi-config)

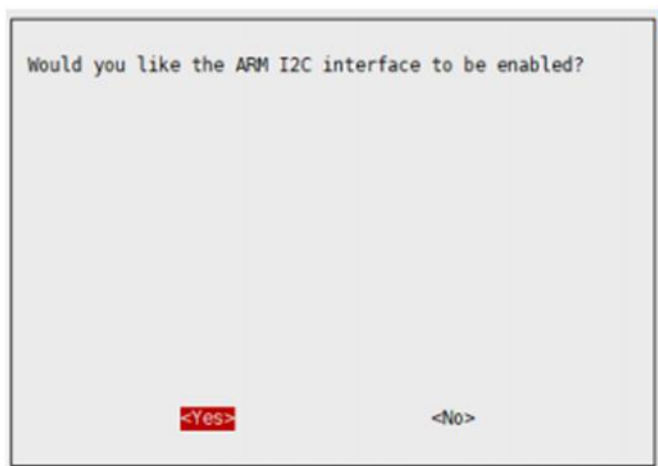
1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options    Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                                <Finish>
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

<Select>                                <Back>
```



注意：运行程序后有提示 I2C 错误，可以运行如下命令打开配置文件

```
sudo nano /etc/modules
```

如果没有这两行语句则添加上去，并保存退出。

```
i2c-dev
```

```
i2c-bcm2708
```

下载例程

在官网上找到对应产品，在产品资料打开下载路径，在 wiki 中下载示例程序：

文档

- [用户手册](#)
- [原理图](#)

程序

- [示例程序](#)

得到解压包，解压并将程序复制到树莓派中。

```
pi@raspberrypi:~/~/Motor_Driver_HAT_code $ ls  
bcm2835 python wiringPi
```

安装必要的函数库

需要安装必要的函数库 (wiringPi、bcm2835、python 库) , 否则以下的示例程序可能无法

正常工作。安装方法详见:

http://www.waveshare.net/wiki/Pioneer600_Datasheets

程序分析

本产品提供树莓派 BCM2835 库、wiringPi 库、python 代码。

BCM2835 库

目录及文件的作用

```
pi@raspberrypi:~/c...e/Motor_Driver_HAT_code/bcm2835 $ tree
.
├── bin
│   ├── DEV_Config.o
│   ├── main.o
│   ├── MotorDriver.o
│   └── PCA9685.o
├── Makefile
├── motor
├── obj
│   ├── Debug.h
│   ├── DEV_Config.c
│   ├── DEV_Config.h
│   ├── main.c
│   ├── MotorDriver.c
│   ├── MotorDriver.h
│   ├── PCA9685.c
│   └── PCA9685.h
└── README.txt
```

/bin:

makefile 生成的目标文件.o, 不需要理会

Makefile: 为工程的编译规则, 执行 make, 工程将会自动编译并生成可执行文件。

motor: 可执行文件, 执行: sudo ./motor 运行程序。

Obj/: 工作目录, 其下定义了多个函数, 其中:

Debug.h: 此函数为调试头文件, 把其中 USE_DEBUG 定义为 1,即可使用 DEBUG() 来打印调试信息, 与 printf()的作用是一样的;

DEV_Config.c(h): 定义了树莓派的管脚及通信方式, 根据 BCM2835 与 WiringPi 的不同而不同;

PCA9685.c(h): 为 PCA9685 芯片的驱动程序, 通过 I2C 控制来输出 16 路 PWM;

MotorDriver.c(.h): 电机驱动芯片 TB6612FNG 驱动, 控制两个电机;

main.c: 主函数,

程序

1. 初始化 BCM2835 库

```
if(DEV_ModuleInit())  
  
    exit(0);
```

2. 初始化电机, 并初始化 PCA9685

```
Motor_Init();
```

3. 控制电机

```
Motor_Run(MOTORA, FORWARD, 100);
```

第一个参数可以选择: MOTORA 或者 MOTORB

第二个参数可以选择: FORWARD 或者 BACKWARD

第三个参数可以选择: 0-100 范围内的整数

因此如果是控制小车的话, 只需要这个函数即可完成全部的动作。

本例程使用了异常处理机制, 因为终止了程序的运行, 但是电机是不会停止的, 当执行

CTRL+c 时, PCA9685 控制寄存器中的值没有被清除:

```
signal(SIGINT, Handler);
```

signal 是 linux 系统的信号处理函数，当执行 CTRL+c 终止程序时会产生信号 SIGINT，那么就会执行 Handler () 函数；Handler 函数实现了：

```
Motor_Stop(MOTORA);
```

```
Motor_Stop(MOTORB);
```

```
DEV_ModuleExit();
```

```
exit(0);
```

使用

由于例程并没有可执行文件，需要先执行 make 生成可执行文件 motor，然后再运行 motor 文件来运行程序

如果你重新编辑了函数，或新定义了函数，则执行 make 重新生成新的可执行文件

如果更改了头文件中的值，则先执行 make clear,再执行 make。

wiringPi 库

目录及文件的作用

```
pi@raspberrypi:~/.../Motor_Driver_HAT_code/wiringPi $ tree
.
├── bin
│   ├── DEV_Config.o
│   ├── main.o
│   ├── MotorDriver.o
│   └── PCA9685.o
├── Makefile
├── motor
├── obj
│   ├── Debug.h
│   ├── DEV_Config.c
│   ├── DEV_Config.h
│   ├── main.c
│   ├── MotorDriver.c
│   ├── MotorDriver.h
│   ├── PCA9685.c
│   └── PCA9685.h
└── README.txt
```

可以看出与 BCM2835 库是一样的目录，不同的地方在于两点：

DEV_Config.c(h):中调用函数是不一样的。

Makefile 中编译链接的库是不一样的。

使用

由于例程并没有可执行文件，需要先执行 `make` 生成可执行文件 `motor`，然后再运行 `motor` 文件来运行程序

如果你重新编辑了函数，或新定义了函数，则执行 `make` 重新生成新的可执行文件

如果更改了头文件中的值，则先执行 `make clear`，再执行 `make`。

python

目录及文件的作用

```
pi@raspberrypi:~/code/module/Motor_Driver_HAT_code/python $ ls
main.py PCA9685.py
```

PCA9685.py 为 PCA9685 芯片的驱动程序，通过 I2C 控制来输出 16 路 PWM

main.py: 为驱动电机程序

程序

1. 实例化 PCA9685 库

```
pwm = PCA9685(0x40, debug=True)
```

第一个参数为：PCA9685 设备地址，这个地址请根据实际的焊接电阻来设置，否则无法通信

第二个参数为：是否显示调试信息

```
pwm.setPWMPFreq(50)
```

设置 PWM 的频率，范围 40-1000

2. 初始化电机控制引脚

```
class MotorDriver():
```

```
    def __init__(self):
```

```
        self.PWMA = 0
```

```
        self.AIN1 = 1
```

```
        self.AIN2 = 2
```

```
self.PWMB = 5
```

```
self.BIN1 = 3
```

```
self.BIN2 = 4
```

```
Motor = MotorDriver()
```

使用 PCA9685 的输出通道 0 到通道 5

3. 控制 PWM:

```
pwm.setDutycycle(self.PWMA, speed)
```

第一个参数为控制 PCA9685 的输出通道

第二个参数为占空比, 其值为 0-100

4. 控制电平:

```
pwm.setLevel(self.AIN1, 1)
```

第一个参数为控制 PCA9685 的输出通道

第二个参数为电平, 0 为低电平, 1 为高电平

5. 控制电机

```
Motor.MotorRun(0, 'forward', 0)
```

第一个参数可以选择: 0 或者 1, 代表电机 1 或者电机 2

第二个参数可以选择: forward 或者 backward, 控制正转反转

第三个参数可以选择: 0-100 范围内的整数, 控制速度

更多使用

对于树莓派而言，使用 python 控制则还可以使用蓝牙或者 wifi 来进行控制

具体的搭建可以参考 Servo Driver HAT 控制：

[http://www.waveshare.net/w/upload/e/e3/Servo Driver HAT User Manual CN.pdf](http://www.waveshare.net/w/upload/e/e3/Servo_Driver_HAT_User_Manual_CN.pdf)