

Unit BLDC Driver

| 案例程序

电机匀速旋转

Setup

bldc_driver_0 Init device I2C address 0x65 (0x01 ~ 0x7f)

bldc_driver_0 Set motor HIGH_SPEED model

bldc_driver_0 Set OPEN_LOOP mode

Loop

bldc_driver_0 Set open loop pwm dutycycle value 500 (0 ~ 2047)

bldc_driver_0 Set motor rpm 500 value in float

print rpm value: + bldc_driver_0 Get read back rpm value in str (return str)

print pwm value: + bldc_driver_0 Get open loop pwm dutycycle value (return int)

```
from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x222222)
bldc_driver_0 = unit.get(unit.BLDC_DRIVER, unit.PORTA)

bldc_driver_0.init_i2c_address(0x65)
bldc_driver_0.set_motor_model(1)
bldc_driver_0.set_mode(0)
while True:
    bldc_driver_0.set_open_loop_pwm(500)
    bldc_driver_0.set_rpm_float(500)
    print((str('rpm value:') + str((bldc_driver_0.get_read_back_rpm_str))))
    print((str('pwm value:') + str((bldc_driver_0.get_open_loop_pwm))))
    wait_ms(2)
```

| 功能说明

bldc_driver_0 Init device I2C address 0x65 (0x01 ~ 0x7f)

```
bldc_driver_0.init_i2c_address(0x65)
```

- 初始化 BLDC Driver 并设置 I2C 地址

bldc_driver_0 Get device **FIRMWARE_VERSION** spec detail (return int)

```
print((str('Version:') + str((bldc_driver_0.get_device_spec(0xFE)))))
```

- 获取设备的当前版本
 - FIRMWARE_VERSION
 - I2C_ADDRESS

bldc_driver_0 Get motor current direction (return 0 or 1)

```
print((str('Current direction:') + str((bldc_driver_0.get_motor_current_direction))))
```

- 获取电机当前转动方向

bldc_driver_0 Get current mode (return 0 or 1)

```
print((str('Current mode:') + str((bldc_driver_0.get_current_mode))))
```

- 获取当前模式

bldc_driver_0 Get motor current model (return 0 or 1)

```
print((str('Mofot Current mode:') + str((bldc_driver_0.get_motor_current_model))))
```

- 获取电机当前的工作模式

bldc_driver_0 Get motor number of pole pairs (return int)

```
print((str('pole pairs:') + str((bldc_driver_0.get_motor_pole_pairs))))
```

- 获取电机的极对数

bldc_driver_0 Get PID value (return tuple)

```
print((str('PID value:') + str((bldc_driver_0.get_pid_value))))
```

- 获取 PID 控制器的参数值

bldc_driver_0 Get open loop pwm dutycycle value (return int)

```
print((str('duty cycle value:') + str((bldc_driver_0.get_open_loop_pwm))))
```

- 获取开环模式下 PWM 的占空比值

bldc_driver_0 Get read back freq value in float (return float)

```
print((str('freq value') + str((bldc_driver_0.get_read_back_freq_float))))
```

- 获取反馈的频率值（浮点数格式）

bldc_driver_0 Get read back freq value in int (return int)

```
print((str('freq value:') + str((bldc_driver_0.get_read_back_freq_int))))
```

- 获取反馈的频率值（整数格式）

bldc_driver_0 Get read back freq value in str (return str)

```
print((str('freq value:') + str((bldc_driver_0.get_read_back_freq_str))))
```

- 获取反馈的频率值（字符串格式）

bldc_driver_0 Get read back rpm value in float (return float)

```
print((str('freq value:') + str((bldc_driver_0.get_read_back_freq_float))))
```

- 获取反馈的转速值（RPM，浮点数格式）

bldc_driver_0 Get read back rpm value in int (return int)

```
print((str('rpm value:') + str((bldc_driver_0.get_read_back_rpm_int))))
```

- 获取反馈的转速值（RPM，整数格式）

bldc_driver_0 Get read back rpm value in str (return str)

```
print((str('rpm value:') + str((bldc_driver_0.get_read_back_rpm_str))))
```

- 获取反馈的转速值（RPM，字符串格式）

bldc_driver_0 Get rpm value in float (return float)

```
print((str('rpm value:') + str((bldc_driver_0.get_read_back_rpm_float))))
```

- 获取设定的转速值 (RPM, 浮点数格式)

bldc_driver_0 Get rpm value in int (return int)

```
print((str('rpm value:') + str((bldc_driver_0.get_rpm_int))))
```

- 获取设定的转速值 (RPM, 整数格式)

bldc_driver_0 Get motor status (return int)

```
print((str('motor status:') + str((bldc_driver_0.get_motor_status))))
```

- 获取电机的当前状态

bldc_driver_0 Save current motor data in flash

```
bldc_driver_0.save_data_in_flash()
```

- 将当前电机数据保存到闪存中

bldc_driver_0 Set motor direction FORWARD

```
bldc_driver_0.set_direction(0)
```

- 设置电机转动方向为正向
 - FORWARD
 - BACKWARD

bldc_driver_0 Set device I2C address 0x65 (0x01 ~ 0x7f)

```
bldc_driver_0.set_i2c_address(0x65)
```

- 设置设备的 I2C 地址

bldc_driver_0 Set OPEN_LOOP mode

```
bldc_driver_0.set_mode(0)
```

- 设置电机模式
 - OPEN_LOOP:开环
 - COLSE_LOOP:闭环

bldc_driver_0 Set motor LOW_SPEED model

```
bldc_driver_0.set_motor_model(0)
```

- 设置电机速度模式
 - LOW_SPEED
 - HIGH_SPEED

bldc_driver_0 Set open loop pwm dutycycle value 500 (0 ~ 2047)

```
bldc_driver_0.set_open_loop_pwm(500)
```

- 设置开环模式下 PWM 的占空比

bldc_driver_0 set P 14200 I 5536 D 1580 value

```
bldc_driver_0.set_pid_value(14200, 5536, 1580)
```

- 设置 PID 控制器的参数

bldc_driver_0 Set 7 number of pole pairs

```
bldc_driver_0.set_pole_pairs(7)
```

- 设置电机的极对数

bldc_driver_0 Set motor rpm 500 value in float

```
bldc_driver_0.set_rpm_float(500)
```

- 设置电机转速（浮点数格式）

bldc_driver_0 Set mortor rpm 500 value in int

```
bldc_driver_0.set_rpm_int(500)
```

- 设置电机转速（整数格式）