# HWDrive CAN通信协议

## 电调上报报文

CAN标识符: 0x100+电调ID

帧格式：DATA

帧类型：标准帧

DLC：8字节

电调ID为1~8，每个电调拥有唯一ID，由电调拨码开关指定

报文内容：

| 数据域 | 内容 |
| --- | --- |
| DATA[0] | 转子机械角度[15:8] |
| DATA[1] | 转子机械角度[7:0] |
| DATA[2] | 转子转速[15:8] |
| DATA[3] | 转子转速[7:0] |
| DATA[4] | 电调功率[15:8] |
| DATA[5] | 电调功率[7:0] |
| DATA[6] | 电调温度[7:0] |
| DATA[7] | reserved |

转子机械角度为16位无符号整数，$$[0,65536)$$对应实际角度$[0,2\pi)rad$

转子转速为16位有符号整数，$[-32768,32768)$对应转速$[-400,400) rad/s$

电调功率为16位有符号整数，$[-32768,32768)$对应转速$[-1000,1000) W$

电调温度为8位无符号整数，直接读取即可，单位℃，

示例代码：

```
switch (hcan->pRxMsg->StdId){
        case 0x101:
            {
                HW_Motor.mechTheta = (((hcan->pRxMsg->Data[0]) << 8) | (hcan-
>pRxMsg->Data[1])) / 65536.0f * 2.0f *3.141592653f;
                HW_Motor.w = (int16_t)(((hcan->pRxMsg->Data[2]) << 8) | (hcan-
>pRxMsg->Data[3])) / 32768.0f * 400.0f; //rad/s
                HW_Motor.Pbus = (int16_t)(((hcan->pRxMsg->Data[4]) << 8) |
(hcan->pRxMsg->Data[5])) / 32768.0f * 1000.0f;//Watt
                HW_Motor.temprature = (hcan->pRxMsg->Data[6]);//℃
            }
}
```

# 电调接收报文

向电调发送转矩电流控制指令，控制电流为16位有符号整数，$[-32768,32768)$对应转矩电流$[-100,100)A$

电调连续20ms未接收到控制指令将进入Idle状态

0xFF报文控制电机1~4, 0x100报文控制电机5~8

CAN标识符: 0xFF

帧格式：DATA

帧类型：标准帧

DLC：8字节

| 数据域 | 内容 | 电机ID |
| --- | --- | --- |
| DATA[0] | 控制电流值[15:8] | 1 |
| DATA[1] | 控制电流值[0:7] | 1 |
| DATA[2] | 控制电流值[15:8] | 2 |
| DATA[3] | 控制电流值[0:7] | 2 |
| DATA[4] | 控制电流值[15:8] | 3 |
| DATA[5] | 控制电流值[0:7] | 3 |
| DATA[6] | 控制电流值[15:8] | 4 |
| DATA[7] | 控制电流值[0:7] | 4 |

CAN标识符: 0x100

帧格式：DATA

帧类型：标准帧

DLC：8字节

| 数据域 | 内容 | 电机ID |
| --- | --- | --- |
| DATA[0] | 控制电流值[15:8] | 5 |
| DATA[1] | 控制电流值[7:0] | 5 |
| DATA[2] | 控制电流值[15:8] | 6 |
| DATA[3] | 控制电流值[7:0] | 6 |
| DATA[4] | 控制电流值[15:8] | 7 |
| DATA[5] | 控制电流值[7:0] | 7 |
| DATA[6] | 控制电流值[15:8] | 8 |
| DATA[7] | 控制电流值[7:0] | 8 |