root\_task(void \*p\_arg)

bsp\_GpioInit();这个初始化gpio,给传感器用

**rs485和modbus**

初始化usart3,以dma方式传输

bsp\_Modbus\_Init

初始化modbus

ModbusMaster\_begin

接收数据：

USART3\_IRQHandler

HAL\_UART\_Receive\_DMA

Modbus\_Master\_Rece\_Handler

读

ModbusMaster\_readHoldingRegisters

Modbus\_Master\_Read

写

ModbusMaster\_writeSingleRegister

ModbusMaster\_ModbusMasterTransaction

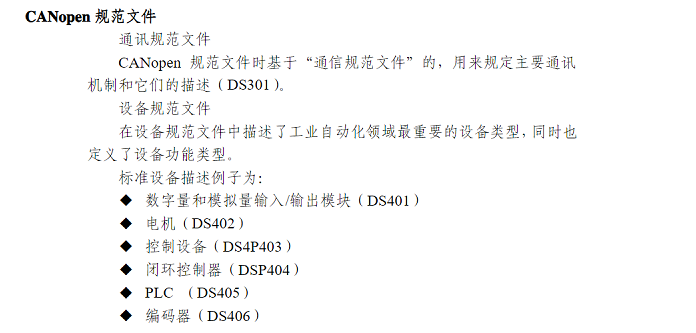
包装u8ModbusADU，然后调用底层接口发出去

Modbus\_Master\_Write(u8ModbusADU,\*u8ModbusADUSize);

Shuttle\_Run中执行拨叉的控制任务。然后通过FB\_Latch\_Move\_Control调用

ModbusMaster\_writeSingleRegister下发命令

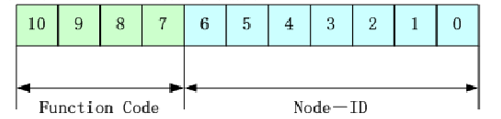
CAN和CANOPEN



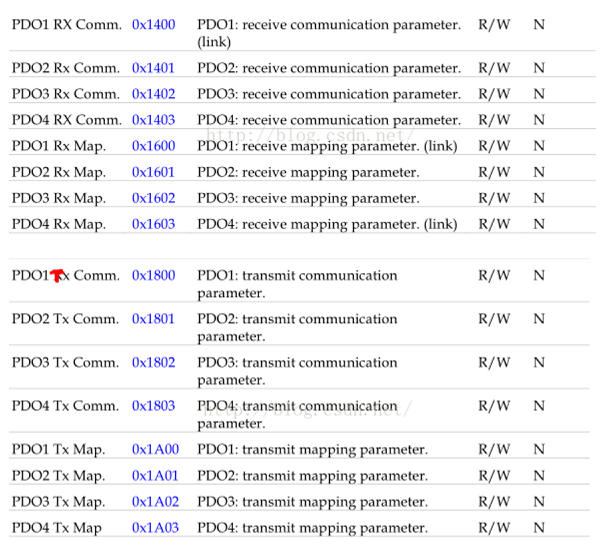
以下是一个标准的 CANopen [页框](https://baike.baidu.com/item/%E9%A1%B5%E6%A1%86)：

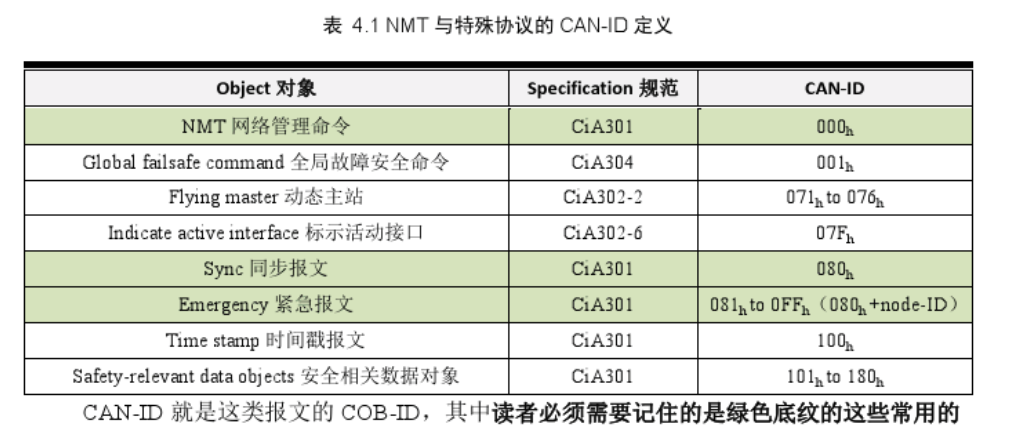
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 功能码 | 节点 ID | RTR | 资料长度 | 资料 |
| 长度 | 4 位元 | 7 位元 | 1 位元 | 4 位元 | 0-8 字节 |

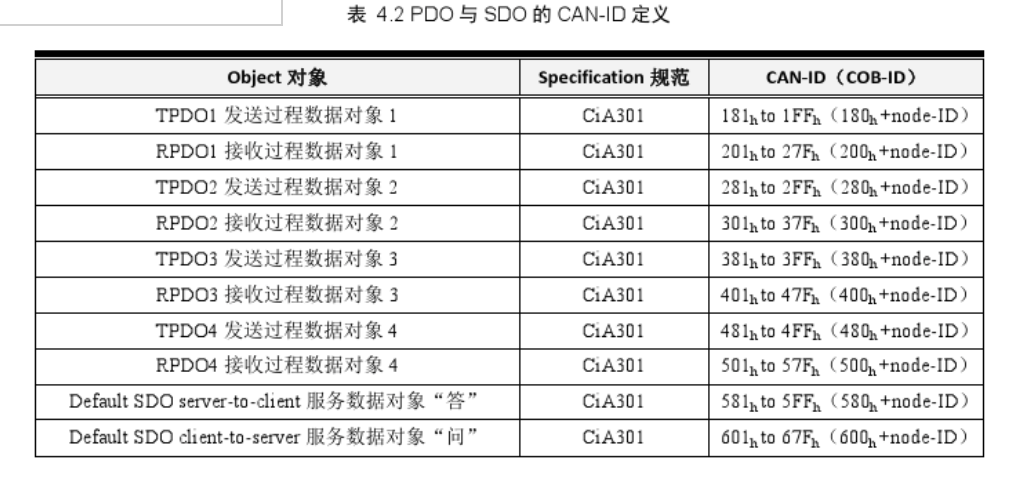
CANopen 将 CANbus 的 11 位元 ID 称为通讯对象 ID (COB-ID)

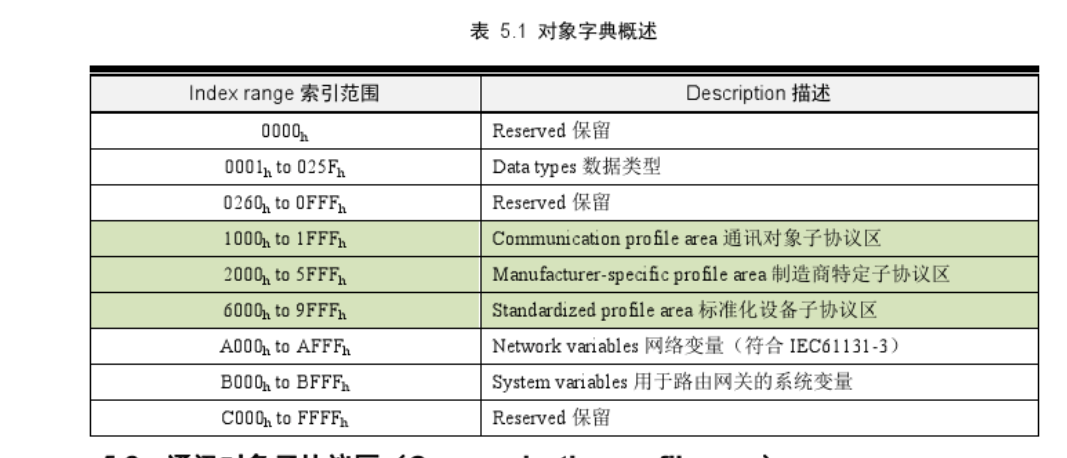


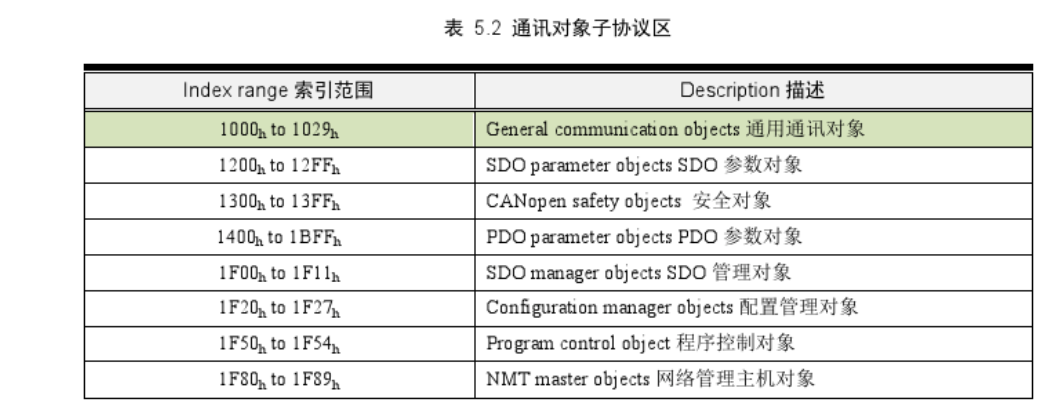












SDO

写4字节数据0x60120208到节点5，index 是1603，subindex是1

605 23 03 16 01 08 02 12 60

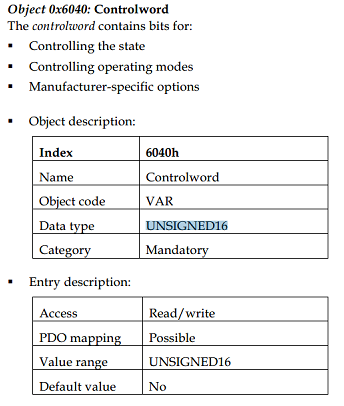
600 +5 23 03 16 01 08 02 12 60

COB-ID ? index subindex 数据

585 60 03 16 01 00 00 00 00

手册CAN402IG.pdf中描述的SDO

例如



PDO：

通过阅读下面这篇文章可以理解PDO

<https://wenku.baidu.com/view/bac4802f3169a4517723a3a5.html?rec_flag=default&sxts=1542096894470>

通讯参数：就是描述发送的cobid和一些熟悉。

接收通讯参数0x1400 0x1401 0x1402 0x1403

发送通讯参数 0x1800 0x1801 0x1802 0x1803

映射参数：实际上就是发送的数据对应的目的的哪个对象。

接收映射参数0x1600 0x1601 0x1602 0x1603

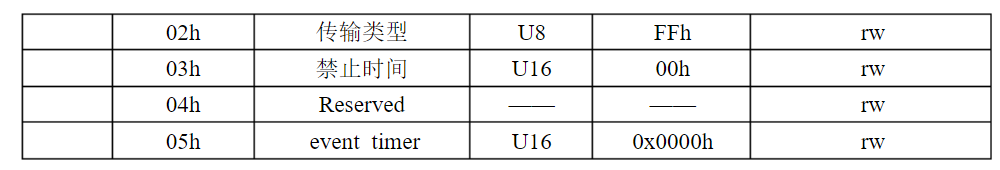
发送映射参数0x1A00 0x1A01 0x1A02 0x1A03

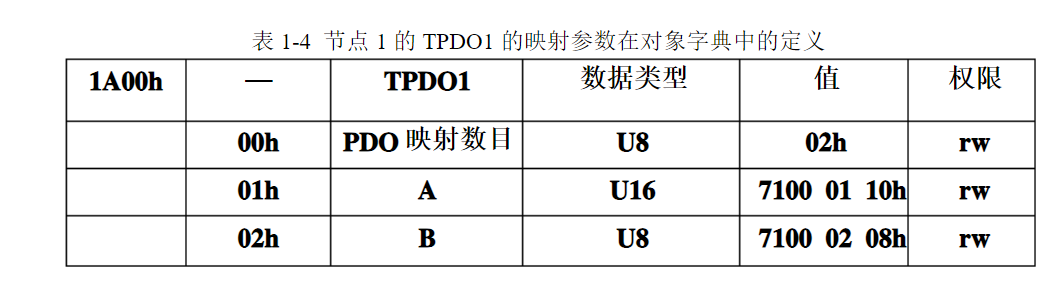
这些参数是相对电机说的，电机的发送就是cpu的接收

以发送TPDO1为例如下3个图，









代码中：

canopenSendSDOReq(serveNodeID, 0x1A00, 0, 0);

设置TPDO1的映射参数0x1A00映射条目为0？不解

canopenSendSDOReq(serveNodeID, 0x1A00, 1, 0x60690020);

canopenSendSDOReq(serveNodeID, 0x1A00, 2, 0x60630020);

0x1A00是TPDO1的映射参数，这两端代码的意思是设置TPDO1的映射参数为：

给0x6069对象，子索引0的位置发送32bit的数据

给0x6063对象，子索引为0的位置发送32bit的数据。

下面这句代码的意思就是把pdo2\_data的值发送给电机的0x6069，0x6063对象。

canopenSendRPDO2\_64((\*interface\_data[i]).server\_id, 8, pdo2\_data);

canopenSendSDOReq(serveNodeID, 0x1800, 2, 1)

设置TPDO1的通信参数subindex2的值为1。301手册上看什么意思。

canopenSendSDOReq(serveNodeID, 0x1a00, 0, 2);

设置TPDO1的映射参数0x1A00映射条目为2

canopenSendSDOReq(serveNodeID, 0x1603, 0, 0);

设置RPDO4的映射参数条目数为0

canopenSendSDOReq(serveNodeID, 0x1603, 1, 0x60ff0020);

设置RPDO4的映射对象。

canopenSendSDOReq(serveNodeID, 0x1403, 2, 0xFF);

设置RPDO4的通信参数subindex 2的值。

canopenSendSDOReq(serveNodeID, 0x1603, 0, 1);

设置RPDO4的映射参数条目数为1

步科伺服电机资料

[**https://wenku.baidu.com/view/db06e2215901020207409ccc.html**](https://wenku.baidu.com/view/db06e2215901020207409ccc.html)

**https://wenku.baidu.com/view/4613175d4b7302768e9951e79b89680202d86b65.html?pn=51**

[**https://blog.csdn.net/iamplane/article/details/49931319**](https://blog.csdn.net/iamplane/article/details/49931319)

[**https://wenku.baidu.com/view/ee8af2eb7fd5360cba1adbc3.html?from=search**](https://wenku.baidu.com/view/ee8af2eb7fd5360cba1adbc3.html?from=search)

[**https://wenku.baidu.com/view/bac4802f3169a4517723a3a5.html?rec\_flag=default&sxts=1542096894470**](https://wenku.baidu.com/view/bac4802f3169a4517723a3a5.html?rec_flag=default&sxts=1542096894470)

**SDO：**

**https://wenku.baidu.com/view/3148623cf11dc281e53a580216fc700abb6852af.html**

**AXIS\_Wheel 轮子电机**

**AXIS\_DB 倍程电机**

**AGV\_canserve\_run();**

**AGV\_canserve\_init(500); /\*初始化CAN\*/**

**canserve\_task(void \*param)**

**canserveMsgProc**

**这个是一个接收电机返回的报文处理。**

**SERVE\_NodeOnline(CanRxMsg \*RxMsg)**

**servePDOMapProcess()**

**mapPdoAction**

**这是一个状态机，一次执行mapPdoAction定义的操作（初始化电机？）**

**Shuttle\_Control()**

**这个好像是业务，控制电机运动，全局变量最后由motionctrl\_task**

**下发命令**

**启动任务shuttlectrl\_task**

**任务里循环执行Shuttle\_Run()**

**这里面对AXIS\_Wheel，AXIS\_DB全局变量赋值**

**并通过下面两个函数根据上次下发命令的情况修正？**

**FB\_Axis(&AXIS\_Wheel);**

**FB\_Axis(&AXIS\_DB);**

**这个FB\_Axis函数里面有最重要的一个就是对wheel\_data或pull\_data这两个全局变量进行修改。这两个全局变量就是motionctrl\_task任务通过CAN总线发给电机的。**

**ctrl\_data = &wheel\_data;**

**motionctrl\_task\_run();**

**启动任务motionctrl\_task（给电机发命令）**

**循环执行下面两个函数**

**can\_communite();**

**canopenSendSync();**

**can\_communite()通过CANOPEN协议发的就是wheel\_data和pull\_data两个全局变量了。**