

目录

一、综合	13
1. 简称	13
2. 综合性能参数	14
二、对象字典	15
2.1 CIA301 控制字	15
0x1000, device type, VAR	15
0x1001, error register, VAR	15
0x1003, pre-defined error field, ARRAY	15
0x1005, COB-ID SYNC message, VAR	16
0x1006, communication cycle period, VAR	16
0x1007, synchronous window length, VAR	16
0x1008, manufacturer device name, ARRAY	16
0x1009, manufacturer hardware version, ARRAY	17
0x100A, manufacturer software version, ARRAY	17
0x1010, store parameters, ARRAY	17
0x1011, restore default parameters, ARRAY	18
0x1014, COB ID emergency, VAR	19
0x1015, inhibit time EMCY, VAR	19
0x1016, consumer heartbeat time, ARRAY	20
0x1017, producer heartbeat time, VAR	20
0x1018, identity object, RECORD	20
0x1019, synchronous counter overflow value, VAR	21
0x1029, error behavior object, ARRAY	21
0x1200, SDO server parameter, RECORD	21
0x1400, RPDO1 communication parameter, RECORD	21
0x1401, RPDO2 communication parameter, RECORD	22
0x1402, RPDO3 communication parameter, RECORD	22
0x1403, RPDO4 communication parameter, RECORD	23
0x1600, RPDO1 mapping parameter, RECORD	24
0x1601, RPDO2 mapping parameter, RECORD	24
0x1602, RPDO3 mapping parameter, RECORD	25
0x1603, RPDO4 mapping parameter, RECORD	26

0x1800, TPDO1 communication parameter, RECORD	27
0x1801, TPDO2 communication parameter, RECORD	28
0x1802, TPDO3 communication parameter, RECORD	28
0x1803, TPDO4 communication parameter, RECORD	29
0x1A00, TPDO1 mapping parameter, RECORD	30
0x1A01, TPDO2 mapping parameter, RECORD	31
0x1A02, TPDO3 mapping parameter, RECORD	32
0x1A03, TPDO4 mapping parameter, RECORD	32
0x1F80, NMT Startup, VAR	33
2.2. 厂商控制字	33
0x2001, CANOPEN config, RECORD	33
0x2002, motor para, RECORD	34
0x2003, CPR, RECORD	35
0x2005, additionalNMT, RECORD	36
0x2010, PID Current, RECORD	36
0x2012, PID Velocity, RECORD	37
0x2013, PID Position, RECORD	38
0x2014, brake control, RECORD	38
0x2015, servo strategy, RECORD	40
0x2016, filter, RECORD	42
0x2017, IIT, RECORD	43
0x2018, bus voltage, RECORD	43
0x2019, EMCY enable, RECORD	44
0x201A, feedforward, RECORD	45
0x201B, temperature, RECORD	46
0x201C, brake resistor, RECORD	47
0x201D, vbat, RECORD	47
0x201E, adaptive notch filter, RECORD	48
0x2030, torque window, VAR	48
0x2031, torque window timeout, VAR	48
0x2032, torqueModeAdditionalPara, RECORD	49
0x2033, sync in time, VAR	49
0x2034, internal target reach window, VAR	49
0x2035, sync delay time, VAR	49

0x203A, homing block configuration, RECORD	50
0x203B, limit switch, RECORD	50
0x203C, mode switch amplifier, VAR	50
0x2080, sync received, VAR	50
0x2240, digital inputs control, RECORD	51
0x2250, digital outputs control, RECORD	51
0x3000, absolute value of current target position, VAR	52
0x3001, current actual value, VAR	52
0x3100, negative limit switch position, VAR	52
0x3101, positive limit switch position, VAR	52
0x4008, can error count, RECORD	53
2.3. CIA402 控制字	53
0x603F, error code, VAR	54
0x6040, control word, VAR	54
0x6041, status word, VAR	54
0x605A, quickstop option code, VAR	54
0x605B, shutdown option code, VAR	54
0x605C, disable operation option code, VAR	55
0x605D, halt option code, VAR	55
0x605E, fault reaction option code, VAR	55
0x6060, mode of operation, VAR	56
0x6061, mode of operation display, VAR	56
0x6062, position demand value, VAR	56
0x6064, position actual value, VAR	56
0x6065, following error window, VAR	57
0x6066, following error timeout, VAR	57
0x6067, position window, VAR	57
0x6068, position window time, VAR	57
0x606B, velocity demand value, VAR	57
0x606C, velocity actual value, VAR	57
0x606D, velocity window, VAR	58
0x606E, velocity window time, VAR	58
0x606F, velocity threshold, VAR	58
0x6070, velocity threshold time, VAR	58

0x6071, target torque, VAR	58
0x6072, max torque, VAR.....	58
0x6074, torque demand, VAR.....	59
0x6075, motor rated current, VAR	59
0x6077, torque actual value, VAR.....	59
0x6079, dclink circuit voltage, VAR	59
0x607A, target position, VAR	59
0x607C, home offset, VAR	60
0x607D, software position limit, ARRAY	60
0x607E, polarity, VAR.....	60
0x607F, max profile velocity, VAR	60
0x6080, max motor speed, VAR.....	61
0x6081, profile velocity, VAR.....	61
0x6083, profile acceleration, VAR.....	61
0x6084, profile deceleration, VAR	61
0x6085, quickstop deceleration, VAR.....	61
0x6086, motion profile type, VAR.....	62
0x6087, torque slope, VAR	62
0x6093, position factor, RECORD	62
0x6095, velocity factor, RECORD	63
0x6097, acceleration factor, RECORD	64
0x6098, homing method, VAR	65
0x6099, homing speeds, ARRAY	65
0x609A, homing acceleration, VAR.....	65
0x60A3, profile jerk use, VAR.....	65
0x60A4, profile jerk, ARRAY	65
0x60B0, position offset, VAR	66
0x60B1, velocity offset, VAR	66
0x60B2, torque offset, VAR.....	66
0x60C0, interpolation submode select, VAR.....	66
0x60C1, interpolation data record, ARRAY.....	66
0x60C2, interpolation time period, RECORD	67
0x60C5, max acceleration, VAR.....	67
0x60C6, max deceleration, VAR	67

0x60F2, position option code, VAR.....	67
0x60F4, following error actual value, VAR	68
0x60FD, digital inputs, VAR	68
0x60FE, digital outputs, ARRAY	68
0x60FF, target velocity, VAR.....	68
0x6502, supported drive modes, VAR.....	69
2.4 保存对象	69
2.4.1 communication parameters.....	69
2.4.2 manufacturer and application parameters	70
三、功能、流程	72
3.1 CANopen.....	72
3.1.1. NMT	72
3.1.2. Node Guarding	73
3.1.3. Heartbeat.....	74
3.1.4. SYNC.....	74
3.1.5. EMCY	74
3.1.6. PDO.....	74
3.1.7. SDO.....	75
3.2 control and status.....	76
3.2.1. status word	76
3.2.2. control word.....	77
3.2.2. event and action	78
3.3 单位, units.....	78
3.3.1. 位置的用户单位.....	79
3.3.2. 速度的用户单位.....	80
3.3.3. 加速度的用户单位	80
3.4 边界, limit.....	81
3.5 IO 接线.....	82
3.6 扭矩模式的零速带载.....	82
3.7 内置的 CAN 总线的 120 欧姆电阻	83
3.8 PID 整定.....	83
3.8.1. 电流环的 PID 整定	83
3.8.2. 速度环的 PID 整定	83
3.8.3. 位置环的 PID 整定	83
3.9 多圈实际位置	83

3.10 S 曲线规划.....	84
四、控制模式.....	85
4.1 profile position mode.....	85
4.1.1. 说明.....	85
4.1.2. 激活.....	85
4.1.3. 控制字.....	85
4.1.4. 状态字.....	85
4.1.5. 相关对象.....	86
4.1.6. 规划框图.....	87
4.1.7. 控制框图.....	89
4.1.8. 配置实例.....	89
4.2 profile velocity mode.....	89
4.2.1. 说明.....	89
4.2.2. 激活.....	89
4.2.3. 控制字.....	90
4.2.4. 状态字.....	90
4.2.5. 相关对象.....	90
4.2.6. 规划框图.....	91
4.2.6. 控制框图.....	92
4.2.7. 配置实例.....	92
4.3 profile torque mode.....	92
4.3.1. 说明.....	92
4.3.2. 激活.....	93
4.3.3. 控制字.....	93
4.3.4. 状态字.....	93
4.3.5. 相关对象.....	93
4.3.6. 规划框图.....	94
4.3.7. 控制框图.....	94
4.3.8. 配置实例.....	94
4.4 homing mode.....	95
4.4.1. 说明.....	95
4.4.2. 激活.....	95
4.4.3. 控制字.....	95
4.4.4. 状态字.....	95
4.4.5. 回零方法.....	96

4.4.6. 相关对象	107
4.4.7. 规划框图	107
4.4.8. 控制框图	109
4.4.9. 配置实例	109
4.5 interpolated position mode	109
4.5.1. 说明.....	109
4.5.2. 激活.....	110
4.5.3. 控制字.....	110
4.5.4. 状态字.....	110
4.5.5. 相关对象	110
4.5.6. 规划框图	111
4.5.6. 控制框图	112
4.5.7. 配置实例	112
4.6 cyclic synchronous position mode.....	112
4.6.1. 说明.....	113
4.6.2. 激活.....	113
4.6.3. 控制字.....	113
4.6.4. 状态字.....	113
4.6.5. 相关对象	113
4.6.6. 规划框图	114
4.6.6. 控制框图	115
4.6.7. 配置实例	115
4.7 cyclic synchronous velocity mode	116
4.7.1. 说明.....	116
4.7.2. 激活.....	116
4.7.3. 控制字.....	116
4.7.4. 状态字.....	116
4.7.5. 相关对象	116
4.7.6. 规划框图	117
4.7.7. 控制框图	118
4.7.8. 配置实例	118
4.8 cyclic synchronous torque mode	118
4.8.1. 说明.....	118
4.8.2. 激活.....	118

4.8.3. 控制字	118
4.8.4. 状态字	119
4.8.5. 相关对象	119
4.8.6. 规划框图	119
4.8.7. 控制框图	120
4.8.8. 配置实例	120
4.9 step dir position mode	120
4.9.1. 说明	120
4.9.2. 激活	121
4.9.3. 控制字	121
4.9.4. 状态字	121
4.9.5. 相关对象	121
4.9.6. 规划框图	121
4.9.7. 控制框图	122
4.9.8. 配置实例	122
4.10 step dir velocity mode	123
4.10.1. 说明	123
4.10.2. 激活	123
4.10.3. 控制字	123
4.10.4. 状态字	123
4.10.5. 相关对象	123
4.10.6. 规划框图	123
4.10.7. 控制框图	123
4.10.8. 配置实例	123
五、错误告警	124
5.1 LED 告警	124
5.2 EMERGENCY 告警	125
0x2310, over current, short	125
0x2314, over current, IIT	126
0x2320, stall	126
0x3110, over voltage	126
0x3111, over max voltage	126
0x3120, under voltage	126
0x3121, under min voltage	127
0x3201, battery voltage low	127

0x4310, amplifier over temperature	127
0x4311, motor over temperature	127
0x4320, amplifier under temperature	128
0x4321, motor under temperature	128
0x4350, temperature sensor error.....	128
0x7114, unwanted brake	128
0x7115, unwanted unbrake	129
0x8111, canopen tx overrun.....	129
0x8112, canopen rx overrun.....	129
0x8130, heartbeat lost.....	129
0x8140, can bus off	130
0x8131, remote node bootup.....	130
0x8201, canopen sync timeout.....	130
0x8202, canopen tpdo out window.....	130
0x8210, canopen rpdo short.....	131
0x8220, canopen rpdo long	131
0x8240, canopen sync overlength.....	131
0x8250, canopen rpdo timeout.....	131
0x8401, over velocity	131
0x8611, following error	132
0xFF01, wrong limit	132
0xFF02, negative limit switch reached.....	132
0xFF03, positive limit switch reached	133
0xFF04, wrong initial condition when homing	133
5.3 SDO abort code	133
六、其它	135
6.1 LED	135
6.1.1. 绿灯	135
6.1.2. 红灯	135

零、更新记录

变更后版本	变更内容	变更时间
A. 00	a. 0x605A 的默认值从 1 改为 2, 也就是默认快停减速度从 0x6084 改为 0x6085.	2022. 02. 01
A. 01	a. 0x6060 添加描述, 回零模式回零成功后能切换为其它模式.	2022. 02. 12
A. 02	a. 修改 2014_brakeRelated, 增加 2014:08h, unBrake Trigger Current, 为有抱闸的伺服的松闸判定电流. 2014:00h 从 7 改为 8. b. 修改 2016_filter, 增加 2016:08h input position filter bandwidth. 同时将 2016:00h 从 7 改为 8. c. 增加 3100h negative limit switch position. 负极限传感器的位置. d. 增加 3101h positive limit switch position. 正极限传感器的位置. e. 增加 0xFF01/0xFF02/0xFF03/0xFF04 等告警的定义. f. 添加章节 3.4 边界, limit 的描述.	2022. 07. 06
A. 03	a. 添加 0x3000_absolute value of target position 这个 object. b. 修正默认值错误. 对象 0x2240_digital inputs control 的默认值修正为 7. c. 添加 object 0x203B_limit switch.	2022. 07. 12
A. 04	a. 电机参数 2002, 增加 Kv, 为 2002:04, 原先 subindex 大于 4 的依次后移. b. 对象母线电压 6079_dcLinkVoltage 的属性描述错误, 从 uint16 修正为 uint32. c. 增加 pp/pv/pt/ip/csp/csv/cst 等模式的状态字里面的 bit14(nls)和 bit15(pls) 的描述.	2022. 07. 25
A. 05	a. 增加对象 60FD/60FE 的描述, 该 2 个对象一直有, 不是新增, 之前是忘记在本文件中添加.	2022. 08. 08
A. 06	a. 增加 2012:05 的描述, 方便单参数调节速度环 PID, 增加了默认值. b. 修正错误, 2014:03 的单位从 mV 改到 mA. c. 自动抱闸部分的描述从 2014:05 挪到 2014:04. d. 2017 的 IIT 中的 iit value 的 subindex 错误的标记为 3, 修正为 2. e. 2019:00 的默认值修正为 16. f. 201A:01 和 201A:02 的 type 修正为 uint16. g. 修正 201C:01 和 201C:02 的 PDO 属性为 NO. h. 修正 2033 的 PDO 属性为 NO. i. 修正 203A:01 和 203A:02 的 PDO 属性为 NO. 修正它们的读写属性为 rw. j. 修正 2240:00 的默认值为 3. 修正 2240:03 的 PDO 属性为 TPDO. k. 修正 2250:03 的 PDO 属性为 TPDO. l. 修正 6075 的 PDO 属性为 NO. m. 修正 6081 的 PDO 属性为 TPDO+RPDO. n. 去除 6083/6084 中强制修改 0 为 1 的描述, 实际设置为 0 会返回 SDO abort, 不会配置成功. o. 简化各个模式的配置实例, 改用最少的 SDO 来实现功能. p. 对象 6093/5/7 的属性修正为 record. q. 对象 60FE 的属性修正为 ARRAY. r. pp/pv/pt 等模式的状态字的 bit10 需要结合控制字的 bit8 来一起描述, 原先没有清楚注明这 2 个 bit 分别状态字和控制字. s. 修正 3.1.5 中 EMCY 报文中 byte3 和 byte4~7 的描述.	2022. 08. 18
A. 07	a. 增加对象 4008 的描述, 4008 是个调试对象, 记录 can 底层错误出现次数	2022. 08. 25
A. 08	a. 增加 3.2.2 event and action 中. 状态字从 STATUS_0x217_QuickstopActive 到 STATUS_0x260_SwitchOnDisabled 的路径. b. 增加 3001h, 电流实际值, 毫安单位. c. 增加 2034 的描述.	2022. 12. 19

	d. 章节 4.1.4 状态字和章节 4.2.4 状态字中, target reach 和 halt 的配合部分描述失误, 修正. 章节 4.3.4 状态字中的描述正确. e. 修改 2014:06 的 PDO 属性为 TPD0. f. 去除 ip 模式下 halt 的描述, ip 模式不再支持 halt. g. 章节 4.4.5.1 增加了一些负数的回零模式. h. 章节 4.9 step dir mode 更名为 step dir position mode. i. 增加章节 4.10 step dir velocity mode. j. 章节 2.2 厂商控制字里面的 0x2034, 标注弃用. k. 对象 0x2015, 增加 2015:06, 用于脉冲速度模式. l. 增加章节 3.5/3.6/3.7/3.8.	
A. 09	a. 增加 2003:02, 多圈编码器的配置. b. 增加 201D, 外接电池的配置. c. 增加 201E, 增加自适应带阻滤波器的配置. d. 章节 3.2.1 的 status word 增加 bit7 的描述, 用于多圈产品. e. 增加 0x3201(外置电池低压)告警及其的 LED 配置. f. 增加章节 3.9 多圈实际位置.	2023. 04. 14
A. 10	a. 增加 2035, sync delay time 的描述配置. b. 修正章节 4.1.4 状态字的描述错误, 修改前 target reach 描述反了. c. 201D:01 low voltage detect threshold, 属性从 ro 改为 rw.	2023. 05. 17
A. 11	a. 6071 的描述中, 6071 的参考对象从 2002:09 修正为 2002:0A. b. 1010:00 的属性从 ro 修正为 const. c. 1011:00 的属性从 ro 修正为 const. d. 1400:00 的属性从 ro 修正为 const. e. 607D:00 的属性从 ro 修正我 const. f. 60C2:00 的属性从 rw 修正为 const. g. 修正 4.x.6 章节, 相关对象. h. 60F2 position option code 默认值从 0 改到 2, 更符合用户需求. i. 605A quickstop option code 默认值从 2 改到 6, 更符合用户需求. j. 增加 2240:04, estop 部分功能描述, 修改 2240:00 从 3 到 4.	2023. 07. 19
A. 12	a. 增加 2015:07 homing stop at zero position 的配置和描述. b. 增加 203C 的配置和描述. c. 修正章节 2.4.2 里的表格.	2023. 08. 11
A. 13	a. 修正章节 3.2.2 control word 里面的表格, 去除大部分的控制字组合, 以免困扰.	2023. 08. 12
A. 14	a. 修改 0x1F80 的默认值从 operational 改为 pre-operational. b. 修改 TPD01/TPD02/TPD03/TPD04/RPD01/RPD02/RPD03/RPD04 的默认配置. c. 修改 0x1017 的默认值为 1000, 也就是 1 秒发送一次心跳. d. 修改 0x6502 的默认值, 脉冲模式改为占用 bit14/15.	2023. 09. 07
A. 15	a. 厂商控制字 201B:00 从 6 改为 10, 添加 201B:07/08/09/10 的描述. b. 章节 3.2.1 status word 添加 bit8/bit14/bit15 的描述.	2023. 09. 16
A. 16	a. 60C2:01, interpolation time period value 的属性修改为 RPD0.	2023. 09. 26
A. 17	a. 2018 bus voltage 增加 2018:05 percent usage, 默认值 100. b. 605E fault reaction code 增加 5/6 选项. c. 在章节 3.2.2 event and action 中增加从 fault 到 operationalEnabled 路径, 并说明对应前提.	2023. 11. 10
A. 18	a. 0x605A/605B/605C/605D/605E 的描述中, 在轴状态的描述上, 加上停机后描述, 为了更清晰的表达.	2023. 11. 16
A. 19	a. 增加 2015:08, 标准位置模式下增加无过冲控制参数.	2023. 11. 19
A. 20	a. 章节 3.1.1 NMT, 增加厂家自定义 NMT 的描述. b. 增加 index 2005, 厂家自定义 NMT 的使能.	2023. 11. 24

A. 21	a. 增加 index 6086 motion profile type. b. 增加 index 60A3 profile jerk use. c. 增加 index 60A4 profile jerk. d. 增加章节 3.10 S 曲线规划. e. 修改章节 4.1 profile position mode, 增加 S 曲线内容. f. 修正章节 3.1.1 NMT 中的文字错误.	2023. 11. 28
A. 22	a. 修正章节 4.4.5.1 回零方法表格示意里面的内容. b. 关于 3 个 I0 的编号, 部分地方描述为 I00/1/2, 部分地方描述为 I01/2/3, 现在统一为 I01/2/3. c. 修正一些文字错误.	2024. 01. 16
A. 23	a. 添加对象 2080. b. 章节 5.2 中增加堵转告警(0x2320). c. 章节 4.5.1 插补模式增加位置前瞻部分描述.	2024. 05. 08

一、综合

1. 简称

简称	全称	备注
CCW	count clockwise	逆时针方向
COB	communication object	通信对象
COS	change of state	状态变更
CSPM	cyclic synchronous position mode	循环同步位置模式
CSTM	cyclic synchronous torque mode	循环同步扭矩模式
CSVM	cyclic synchronous velocity mode	循环同步速度模式
CW	clockwise	顺时针方向
fsm	finite state machine	有限状态机
HM	homing mode	回零模式
HS	home switch	原点开关
IPM	interpolation position mode	插补位置模式
ms	manufacture specified	生厂商自定义
ms	millisecond	毫秒
NBL	negative block limit	负向阻挡极限
NLS	negative limit switch	负向极限开关
oe	operation enabled	操作使能
oms	operation mode specified	由模式确定
PDO	process data object	过程数据对象
pds	power drive system	动力驱动系统
PBL	positive block limit	正向阻挡极限
PLS	positive limit switch	正向极限开关
PPM	profile position mode	标准位置模式，也叫规划位置模式
PTM	profile torque mode	标准扭矩模式，也叫规划扭矩模式
PVM	Profile velocity mode	标准速度模式，也叫规划速度模式
qs	quick stop	快停
RPM	revolution per minute	圈每分钟，一种速度单位
RPS	revolution per second	圈每秒，一种速度单位
rtso	ready to switch on	
ro	read only	只读
RT	rated torque	额定扭矩
rw	read write	读写
SDO	service data object	服务数据对象
so	switched on	
sod	switched on disabled	
tr	target reached	到达目标
ve	voltage enabled	上电

2. 综合性能参数

大类	条目	说明
CAN 接口	协议	CANOpen
	波特率	1M/500K/250K/125K/100K/50K/20K/10K
	最大节点数	127 个
	终端匹配电阻	120 Ω 内置, 配置 0x2001:03 来使能
	支持服务	NMT、SDO、PDO、SYNC、EMCY、HEART BEATING
	帧类型	数据帧
	PDO 类型	async, sync, cos, event-drive
	PDO 数量	4 TPDO, 4 RPDO
	SDO 传输方式	expedited、segment、block
	支持模式	标准位置模式, profile position mode 标准速度模式, profile velocity mode 标准扭矩模式, profile torque mode 回零模式, homing mode 插补位置模式, interpolated position mode 循环同步位置模式, cyclic synchronous position mode 循环同步速度模式, cyclic synchronous velocity mode 循环同步扭矩模式, cyclic synchronous torque mode
	HeartBeat	支持 4 个 consumer, 1 个 producer
	NodeGuarding	仅支持查询 NMT 状态

二、对象字典

2.1 CIA301 控制字

0x1000, device type, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1000	0x00	ro	uint32	/	/	NO

Description:

value	description
0x10000	PMSM 伺服
0x20000	BLDC 伺服
0x40000	两相步进伺服
0x80000	三相步进伺服

0x1001, error register, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1001	0x00	ro	uint8	/	0	TPDO

Description:

- 任何错误的发生, generic error 位总是置位的.

Bit value	description
0	generic error
1	current error
2	voltage error
3	temperature error
4	communication error
5	device profile specific error
6	reserved
7	manufacture error

0x1003, pre-defined error field, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x00	rw	uint8	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x01	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x02	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x03	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x04	ro	uint32	/	0	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x05	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x06	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x07	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x08	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x09	ro	uint32	/	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1003	0x0A	ro	uint32	/	0	NO

Description:

- subIndex=0x00, 为当前记录的 error 的个数.
- 对 0x00 写入 0 会清除所有错误.
- subIndex=0x01~0x0A 为当前记录的 error, 0x01 记录的 error 最新, 如果当前无错误记录, 那么尝试读取会返回 SDO abort 0x08000024.

0x1005, COB-ID SYNC message, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1005	0x00	const	uint32	/	0x00000080	NO

0x1006, communication cycle period, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1006	0x00	rw	uint32	us	0	NO

Description:

- 设置为 0 会 disable sync message.
- 单位是微秒(us), 实际内部分辨精度是 500us, 也就是 0.5ms, 建议设置为 0.5ms 的整倍数.
- sync 超时按照 0x1006 的 1.5 倍作为 sync 超时判断依据, sync time out 详见 [0x8201, canopen sync timeout](#)

0x1007, synchronous window length, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1007	0x00	rw	uint32	us	0	NO

Description:

- 设置为 0 会 disable sync window.
- 单位是微秒(us), 实际内部分辨精度是 500us, 也就是 0.5ms, 建议设置为 0.5ms 的整倍数.
- 收到 sync 后, 在 0x1007 规定的时间段内可以收发 PDO, 如有 TPDO 的收发未在规定时间内完成, 那么这些 TPDO 会被丢弃, 发出紧急报文为 [0x8202, canopen tpdo out window](#).

0x1008, manufacturer device name, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1008	0x00	ro	uint8	/	0x04	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1008	0x01	ro	uint8	/	"F"	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1008	0x02	ro	uint8	/	"L"	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1008	0x03	ro	uint8	/	"O"	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1008	0x04	ro	uint8	/	"W"	NO

0x1009, manufacturer hardware version, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1009	0x00	ro	uint8	0x04	0x04	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1009	0x01	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1009	0x02	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1009	0x03	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1009	0x04	ro	uint8	/	/	NO

Description:

- 为产品硬件版本号, 具体内容随产品而定.

0x100A, manufacturer software version, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x100A	0x00	ro	uint8	/	0x04	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x100A	0x01	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x100A	0x02	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x100A	0x03	ro	uint8	/	/	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x100A	0x04	ro	uint8	/	/	NO

Description:

- 为产品软件版本号, 具体内容随产品而定.

0x1010, store parameters, ARRAY

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1010	0x00	const	uint8	/	0x3	NO

store all parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1010	0x01	rw	uint32	/	0x1	NO

Description:

- all parameters = communication + manufacturer + application parameters.
- 被保存的参数在[章节 2.4](#)中有描述;
- 读时返回 0x01, 表示依据命令存储.
-
- in order to avoid storage of parameters by mistake, storage shall be only executed when a specific signature is written to appropriate sub-index. the signature is:

MSB			LSB
e	v	a	s
0x65	0x76	0x61	0x73

store communication parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1010	0x02	rw	uint32	/	0x1	NO

Description:

- communication parameter index from 0x1000~0x1FFF.
- 具体那些 index 会被保存请参见[章节 2.4.1](#).
- 读时返回 0x01, 表示依据命令存储.
- in order to avoid storage of parameters by mistake, storage shall be only executed when a specific signature is written to appropriate sub-index. the signature is:

MSB			LSB
e	v	a	s
0x65	0x76	0x61	0x73

store manufacturer and application parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1010	0x03	rw	uint32	/	0x1	NO

Description:

- manufacturer parameter index from 0x2000~0x6FFF.
- 具体那些 index 会被保存请参见[章节 2.4.2](#).
- 读时返回 0x01, 表示依据命令存储.
- in order to avoid storage of parameters by mistake, storage shall be only executed when a specific signature is written to appropriate sub-index. the signature is:

MSB			LSB
e	v	a	s
0x65	0x76	0x61	0x73

0x1011, restore default parameters, ARRAY

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1011	0x00	const	uint8	/	0x3	NO

restore all parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1011	0x01	rw	uint32	/	0x1	NO

Description:

- all parameters = communication + manufacturer + application parameters.
- 被恢复的参数在章节 2.4 中有描述.
- 读时返回 0x01, 表示依据命令恢复.
- 为避免误操作, 需要写入特殊字符到本地地址才会执行恢复操作, 特殊字符如下:

MSB			LSB
d	a	o	l
0x64	0x61	0x6F	0x6C

restore communication parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1011	0x02	rw	uint32	/	0x1	NO

Description:

- communication parameter index from 0x1000~0x1FFF.
- 具体那些 index 会被恢复请参见章节 2.4.1.
- 读时返回 0x01, 表示依据命令恢复.
- 为避免误操作, 需要写入特殊字符到本地地址才会执行恢复操作, 特殊字符如下:

MSB			LSB
d	a	o	l
0x64	0x61	0x6F	0x6C

restore manufacturer and application parameters

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1011	0x03	rw	uint32	/	0x1	NO

Description:

- manufacturer parameter index from 0x2000~0x6FFF.
- 具体那些 index 会被恢复请参见章节 2.4.2.
- 读时返回 0x01, 表示依据命令恢复.
- 为避免误操作, 需要写入特殊字符到本地地址才会执行恢复操作, 特殊字符如下:

MSB			LSB
d	a	o	l
0x64	0x61	0x6F	0x6C

0x1014, COB ID emergency, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1014	0x00	const	uint32	/	0x00000080	NO

0x1015, inhibit time EMCY, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1015	0x00	rw	uint16	0.1ms	0	NO

Description:

- EMCY 报文发送的抑制时间, 也就是 EMCY 报文的最短发送间隔.
- 如果设置为 0, 表示不限制发送间隔.

0x1016, consumer heartbeat time, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1016	0x00	ro	uint32	/	0x04	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1016	0x01	rw	uint32	ms	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1016	0x02	rw	uint32	ms	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1016	0x03	rw	uint32	ms	0	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1016	0x04	rw	uint32	ms	0	NO

Description:

- 默认支持接受 4 路节点的 heartbeat 消息.
- 每个监控节点的数据格式如下:

byte3	byte2	byte1~byte0
reserved(0x00)	node-id	heartbeat time

0x1017, producer heartbeat time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1017	0x00	rw	uint16	ms	1000	NO

Description:

- 配置非零值以激活主动心跳报文.
- 心跳报文属于 NMT, 所以任意 NMT 状态下都会发送.

0x1018, identity object, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1018	0x00	const	uint8	/	0x4	NO

vendor ID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1018	0x01	ro	uint32	/	0x0	NO

product code

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1018	0x02	ro	uint8	/	0x0	NO

revision number

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1018	0x03	ro	uint32	/	0x0	NO

serial number

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1018	0x04	ro	uint8	/	/	NO

0x1019, synchronous counter overflow value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1019	0x00	rw	uint8	/	0	NO

Description:

- 带数据(编号)的 SYNC 帧, 其数据最大值(最大编号).
- 如果 SYNC 帧带数据, 作为 SYNC 的接收方, 本值需要被配置, 不能是 0 或者 1, 因为 0 表示 sync 帧不带数据, 1 没有意义.

0x1029, error behavior object, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1029	0x00	ro	uint8	/	2	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1029	0x01	rw	uint8	/	1	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1029	0x02	rw	uint8	/	1	NO

Description:

- subindex=0x01, 对应 bus off, heartbeat lost 情况.
- subindex=0x02, 对应其他通信错误.

value	description
0	控制器切换到预操作状态（如果之前处于操作状态）.
1	控制器不改变状态.
2	控制器切换到停止状态.

0x1200, SDO server parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1200	0x00	const	uint8	/	2	NO

client to server COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1200	0x01	const	uint32	/	0x00000600	NO

server to client COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1200	0x02	const	uint32	/	0x00000580	NO

0x1400, RPD01 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1400	0x00	const	uint8	/	0x05	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1400	0x01	rw	uint32	/	0x00000200	NO

Description:

- bit31=0 为使能, bit31=1 为禁止.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1400	0x02	rw	uint8	/	0x01	NO

Description:

- transmission 支持以下种类:

value	description
0x00~0xF0	synchronous, 同步
0xF1~0xFD	保留
0xFE	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警
0xFF	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1400	0x05	rw	uint16	/	0x00	NO

Description:

- 支持 event timer, 如果该值非 0, 那么需要在规定时间内收到 RPDO, 否则报错误 [0x8250, canopen rpdo timeout](#)

0x1401, RPDO2 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1401	0x00	const	uint8	/	0x05	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1401	0x01	rw	uint32	/	0x00000300	NO

Description:

- bit31=0 为使能, bit31=1 为禁止.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1401	0x02	rw	uint8	/	0x01	NO

Description:

- transmission 支持以下种类:

value	description
0x00~0xF0	synchronous, 同步
0xF1~0xFD	保留
0xFE	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警
0xFF	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1401	0x05	rw	uint16	/	0x00	NO

Description:

- 支持 event timer, 如果该值非 0, 那么需要在规定时间内收到 RPDO, 否则报错误 [0x8250, canopen rpdo timeout](#)

0x1402, RPDO3 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1402	0x00	const	uint8	/	0x05	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1402	0x01	rw	uint32	/	0x80000400	NO

Description:

- bit31=0 为使能, bit31=1 为禁止.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1402	0x02	rw	uint8	/	0xFF	NO

Description:

- transmission 支持以下种类:

value	description
0x00~0xF0	synchronous, 同步
0xF1~0xFD	保留
0xFE	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警
0xFF	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1402	0x05	rw	uint16	/	0x00	NO

Description:

- 支持 event timer, 如果该值非 0, 那么需要在规定时间内收到 RPDO, 否则报错误 [0x8250, canopen_rpdo_timeout](#)

0x1403, RPDO4 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1403	0x00	const	uint8	/	0x05	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1403	0x01	rw	uint32	/	0x80000500	NO

Description:

- bit31=0 为使能, bit31=1 为禁止.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1403	0x02	rw	uint8	/	0xFF	NO

Description:

- transmission 支持以下种类:

value	description
0x00~0xF0	synchronous, 同步
0xF1~0xFD	保留
0xFE	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警
0xFF	事件驱动, 超过 event-timer 规定时间没有收到 RPDO 则告警

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1403	0x05	rw	uint16	/	0x00	NO

Description:

- 支持 event timer, 如果该值非 0, 那么需要在规定时间内收到 RPDO, 否则报错误 [0x8250, canopen rpdo timeout](#)

0x1600, RPDO1 mapping parameter, RECORD

number of mapped application objects in PDO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x00	rw	uint8	/	0x03	NO

mapped objects 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x01	rw	uint32	/	0x60400010	NO

mapped objects 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x02	rw	uint32	/	0x60FF0020	NO

mapped objects 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x03	rw	uint32	/	0x60600008	NO

mapped objects 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x04	rw	uint32	/	0x00	NO

mapped objects 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x05	rw	uint32	/	0x00	NO

mapped objects 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x06	rw	uint32	/	0x00	NO

mapped objects 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x07	rw	uint32	/	0x00	NO

mapped objects 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1600	0x08	rw	uint32	/	0x00	NO

Description:

- mapped object 的 value 的组成如下表, 其中 length 以 bit 计算, 比如 uint8 的 length 是 8, uint32 的 length 是 32.

bit31~16	bit15~8	bit7~0
index	sub-index	length

- 0x1600 的 mapping parameter 配合 0x1400 的 communication parameter.

0x1601, RPDO2 mapping parameter, RECORD

number of mapped application objects in PDO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x00	rw	uint8	/	0x02	NO

mapped objects 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x01	rw	uint32	/	0x607A0020	NO

mapped objects 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x02	rw	uint32	/	0x60810020	NO

mapped objects 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x03	rw	uint32	/	0x00	NO

mapped objects 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x04	rw	uint32	/	0x00	NO

mapped objects 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x05	rw	uint32	/	0x00	NO

mapped objects 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x06	rw	uint32	/	0x00	NO

mapped objects 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x07	rw	uint32	/	0x00	NO

mapped objects 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1601	0x08	rw	uint32	/	0x00	NO

Description:

- mapped object 的 value 的组成如下表:

bit31~16	bit15~8	bit7~0
index	sub-index	length

- 0x1601 的 mapping parameter 配合 0x1401 的 communication parameter.

0x1602, RPD03 mapping parameter, RECORD

number of mapped application objects in PDO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x00	rw	uint8	/	0x00	NO

mapped objects 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x01	rw	uint32	/	0x00	NO

mapped objects 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x02	rw	uint32	/	0x00	NO

mapped objects 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x03	rw	uint32	/	0x00	NO

mapped objects 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x04	rw	uint32	/	0x00	NO

mapped objects 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x05	rw	uint32	/	0x00	NO

mapped objects 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x06	rw	uint32	/	0x00	NO

mapped objects 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x07	rw	uint32	/	0x00	NO

mapped objects 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1602	0x08	rw	uint32	/	0x00	NO

Description:

- mapped object 的 value 的组成如下表:

bit31~16	bit15~8	bit7~0
index	sub-index	length

- 0x1602 的 mapping parameter 配合 0x1402 的 communication parameter.

0x1603, RPD04 mapping parameter, RECORD**number of mapped application objects in PDO**

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x00	rw	uint8	/	0x00	NO

mapped objects 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x01	rw	uint32	/	0x00	NO

mapped objects 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x02	rw	uint32	/	0x00	NO

mapped objects 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x03	rw	uint32	/	0x00	NO

mapped objects 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x04	rw	uint32	/	0x00	NO

mapped objects 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x05	rw	uint32	/	0x00	NO

mapped objects 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x06	rw	uint32	/	0x00	NO

mapped objects 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x07	rw	uint32	/	0x00	NO

mapped objects 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1603	0x08	rw	uint32	/	0x00	NO

Description:

- mapped object 的 value 的组成如下表:

bit31~16	bit15~8	bit7~0
index	sub-index	length

- 0x1603 的 mapping parameter 配合 0x1403 的 communication parameter.

0x1800, TPD01 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x00	const	uint8	/	0x06	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x01	rw	uint32	/	0x00000180	NO

Description:

- 不支持修改 TPD0 的 COBID, 可以修改 bit31 (valid) 来使能/禁止 PDO.
- bit31=0 为使能, bit31=1 为禁止.
- bit30=1, 表示不支持 RTR. 将 RTR 置位会返回 SDO 告警.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x02	rw	uint8	/	0x01	NO

Description:

- transmission 支持以下种类:

value	description
0x00	async. 事件触发后的下一个 SYNC 帧到来时发送 TPD0
0x01~0xF0	sync. 如果值为 n, 那么每 n 个 SYNC 帧发送一次 TPD0
0xF1~0xFB	保留
0xFB	不支持
0xFC	不支持
0xFE	事件驱动, 触发事件是 mapping 参数变动 (COS).
0xFF	事件驱动, 触发事件是 event-timer 计时到 0.

- 当为事件驱动时, 发送时间间隔受到 inhibit time 的限制.
- 当 transmission = 0xFE 的设置. 建议用于 status word (6041h), 一旦 status word 已发生变化就发出 TPD0, 不需要等待 SYNC 帧的到来. 不建议用于 position Actual Value (6064h), 因为 position actual value 的更新频率是 1Khz.

inhibit time

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x03	rw	uint16	0.1ms	0	NO

Description:

- inhibit time 的单位是 0.1ms, 而 event timer 的单位是 ms, 需要注意.

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x05	rw	uint16	ms	0	NO

SYNC start value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1800	0x06	rw	uint8	/	0	NO

Description:

- 为了错开各节点在同步模式下同时发送导致总线拥堵而设计的, 各节点的 sync start value 不同就可以有效的错峰通信.

- 要使得 sync start value 有效, sync 帧必须是带数据的(长度为 1), 且各节点的 0x1019 需要被配置.

0x1801, TPD02 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x00	const	uint8	/	0x06	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x01	const	uint32	/	0x00000280	NO

Description:

- 不支持修改 TPD0 的 COBID, 可以修改 bit31 (valid) 来使能/禁止 PDO.
- bit31=0 为使能, bit31=1 为禁止.
- bit30=1, 表示不支持 RTR. 将 RTR 置位会返回 SDO 告警.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x02	rw	uint8	/	0x01	NO

Description:

- transmission 支持以下种类:

value	description
0x00	async. 事件触发后的下一个 SYNC 帧到来时发送 TPD0
0x01~0xF0	sync. 如果值为 n, 那么每 n 个 SYNC 帧发送一次 TPD0
0xF1~0xFB	保留
0xFB	不支持
0xFC	不支持
0xFE	事件驱动, 触发事件是 mapping 参数变动 (COS).
0xFF	事件驱动, 触发事件是 event-timer 计时到 0.

- 当为事件驱动时, 发送时间间隔受到 inhibit time 的限制.
- 当 transmission = 0xFE 的设置. 建议用于 status word (6041h), 一旦 status word 已发生变化就发出 TPD0, 不需要等待 SYNC 帧的到来. 不建议用于 position Actual Value (6064h), 因为 position actual value 的更新频率是 1Khz.

inhibit time

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x03	rw	uint16	0.1ms	0	NO

Description:

- inhibit time 的单位是 0.1ms, 而 event timer 的单位是 ms, 需要注意.

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x05	rw	uint16	ms	0	NO

SYNC start value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1801	0x06	rw	uint8	/	0	NO

Description:

- 为了错开各节点在同步模式下同时发送导致总线拥堵而设计的, 各节点的 sync start value 不同就可以有效的错峰通信.
- 要使得 sync start value 有效, sync 帧必须是带数据的(长度为 1), 且各节点的 0x1019 需要被配置.

0x1802, TPD03 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x00	const	uint8	/	0x06	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x01	const	uint32	/	0x80000380	NO

Description:

- 不支持修改 TPD0 的 COBID, 可以修改 bit31 (valid) 来使能/禁止 PDO.
- bit31=0 为使能, bit31=1 为禁止.
- bit30=1, 表示不支持 RTR. 将 RTR 置位会返回 SDO 告警.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x02	rw	uint8	/	0xFF	NO

Description:

- transmission 支持以下种类:

value	description
0x00	async. 事件触发后的下一个 SYNC 帧到来时发送 TPD0
0x01~0xF0	sync. 如果值为 n, 那么每 n 个 SYNC 帧发送一次 TPD0
0xF1~0xFB	保留
0xFB	不支持
0xFC	不支持
0xFE	事件驱动, 触发事件是 mapping 参数变动 (COS).
0xFF	事件驱动, 触发事件是 event-timer 计时到 0.

- 当为事件驱动时, 发送时间间隔受到 inhibit time 的限制.
- 当 transmission = 0xFE 的设置. 建议用于 status word (6041h), 一旦 status word 已发生变化就发出 TPD0, 不需要等待 SYNC 帧的到来. 不建议用于 position Actual Value (6064h), 因为 position actual value 的更新频率是 1Khz.

inhibit time

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x03	rw	uint16	0.1ms	0	NO

Description:

- inhibit time 的单位是 0.1ms, 而 event timer 的单位是 ms, 需要注意.

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x05	rw	uint16	ms	0	NO

SYNC start value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1802	0x06	rw	uint8	/	0	NO

Description:

- 为了错开各节点在同步模式下同时发送导致总线拥堵而设计的, 各节点的 sync start value 不同就可以有效的错峰通信.
- 要使得 sync start value 有效, sync 帧必须是带数据的 (长度为 1), 且各节点的 0x1019 需要被配置.

0x1803, TPD04 communication parameter, RECORD

maxSubIndex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x00	const	uint8	/	0x06	NO

COBID

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x01	const	uint32	/	0x80000480	NO

Description:

- 不支持修改 TPD0 的 COBID, 可以修改 bit31 (valid) 来使能/禁止 PDO.
- bit31=0 为使能, bit31=1 为禁止.
- bit30=1, 表示不支持 RTR. 将 RTR 置位会返回 SDO 告警.

transmission type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x02	rw	uint8	/	0xFF	NO

Description:

- transmission 支持以下种类:

value	description
0x00	async. 事件触发后的下一个 SYNC 帧到来时发送 TPD0
0x01~0xF0	sync. 如果值为 n, 那么每 n 个 SYNC 帧发送一次 TPD0
0xF1~0xFB	保留
0xFB	不支持
0xFC	不支持
0xFE	事件驱动, 触发事件是 mapping 参数变动 (COS).
0xFF	事件驱动, 触发事件是 event-timer 计时到 0.

- 当为事件驱动时, 发送时间间隔受到 inhibit time 的限制.
- 当 transmission = 0xFE 的设置. 建议用于 status word (6041h), 一旦 status word 已发生变化就发出 TPD0, 不需要等待 SYNC 帧的到来. 不建议用于 position Actual Value (6064h), 因为 position actual value 的更新频率是 1KHz.

inhibit time

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x03	rw	uint16	0.1ms	0	NO

Description:

- inhibit time 的单位是 0.1ms, 而 event timer 的单位是 ms, 需要注意.

event timer

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x05	rw	uint16	ms	0	NO

SYNC start value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1803	0x06	rw	uint8	/	0	NO

Description:

- 为了错开各节点在同步模式下同时发送导致总线拥堵而设计的, 各节点的 sync start value 不同就可以有效的错峰通信.
- 要使得 sync start value 有效, sync 帧必须是带数据的 (长度为 1), 且各节点的 0x1019 需要被配置.

0x1A00, TPD01 mapping parameter, RECORD

number of mapped application objects

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x00	rw	uint8	/	0x03	NO

mapped object 1

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x1A00	0x01	rw	uint32	/	0x60410010	NO
mapped object 2						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x02	rw	uint32	/	0x60FD0020	NO
mapped object 3						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x03	rw	uint32	/	0x60610008	NO
mapped object 4						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x04	rw	uint32	/	0x00	NO
mapped object 5						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x05	rw	uint32	/	0x00	NO
mapped object 6						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x06	rw	uint32	/	0x00	NO
mapped object 7						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x07	rw	uint32	/	0x00	NO
mapped object 8						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A00	0x08	rw	uint32	/	0x00	NO

0x1A01, TPD02 mapping parameter, RECORD

number of mapped application objects						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x00	rw	uint8	/	0x02	NO
mapped object 1						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x01	rw	uint32	/	0x60640020	NO
mapped object 2						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x02	rw	uint32	/	0x606C0020	NO
mapped object 3						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x03	rw	uint32	/	0x00	NO
mapped object 4						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x04	rw	uint32	/	0x00	NO
mapped object 5						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x05	rw	uint32	/	0x00	NO
mapped object 6						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x06	rw	uint32	/	0x00	NO

mapped object 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x07	rw	uint32	/	0x00	NO

mapped object 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A01	0x08	rw	uint32	/	0x00	NO

0x1A02, TPD03 mapping parameter, RECORD**number of mapped application objects**

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x00	rw	uint8	/	0x00	NO

mapped object 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x01	rw	uint32	/	0x00	NO

mapped object 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x02	rw	uint32	/	0x00	NO

mapped object 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x03	rw	uint32	/	0x00	NO

mapped object 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x04	rw	uint32	/	0x00	NO

mapped object 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x05	rw	uint32	/	0x00	NO

mapped object 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x06	rw	uint32	/	0x00	NO

mapped object 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x07	rw	uint32	/	0x00	NO

mapped object 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A02	0x08	rw	uint32	/	0x00	NO

0x1A03, TPD04 mapping parameter, RECORD**number of mapped application objects**

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x00	rw	uint8	/	0x00	NO

mapped object 1

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x01	rw	uint32	/	0x00	NO

mapped object 2

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x02	rw	uint32	/	0x00	NO

mapped object 3

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x03	rw	uint32	/	0x00	NO

mapped object 4

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x04	rw	uint32	/	0x00	NO

mapped object 5

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x05	rw	uint32	/	0x00	NO

mapped object 6

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x06	rw	uint32	/	0x00	NO

mapped object 7

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x07	rw	uint32	/	0x00	NO

mapped object 8

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1A03	0x08	rw	uint32	/	0x00	NO

0x1F80, NMT Startup, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x1F80	0x00	rw	uint32	/	0x00	NO

Description:

value	description
0x00	伺服启动正常启动后, NMT 进入 pre-operational 状态.
0x08	伺服启动正常启动后, NMT 进入 operational 状态.

2.2. 厂商控制字

0x2001, CANOPEN config, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2001	0x00	const	uint8	/	4	NO

can nodeid

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2001	0x01	rw	uint8	/	1	NO

Description:

- nodeid 的允许范围是 1~127, 如果尝试配置为 0 或者超过 127, 都会导致 SDO 反馈告警.

can bit rate

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2001	0x02	rw	uint16	/	1000	NO

Description:

- 允许的波特率配置如下表格:

value	bit rate	备注
10	10Kbps	最长通信长度 5000 米
20	20Kbps	最长通信长度 2500 米
50	50Kbps	最长通信长度 1000 米
100	100Kbps	最长通信长度 1000 米
125	125Kbps	最长通信长度 500 米
250	250Kbps	最长通信长度 250 米
500	500Kbps	最长通信长度 100 米
1000	1Mbps	最长通信长度 20 米

- 配置错误的 bit rate 都会收到 SDO abortcode 0x06090030.

can enable resistor

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2001	0x03	rw	bool	/	/	NO

Description:

- 在一条 CAN 通信物理线路上, 推荐只在菊花链线路的头尾两个节点上使能 120 匹配电阻;

value	description
0	断开内置的 120 欧姆串联电阻
1	使能内置的 120 欧姆串联电阻

- 匹配电阻使能的伺服, 其上电后绿灯是以 1HZ 的频率闪烁. 匹配电阻没有使能的伺服, 其上电后绿灯是常亮.

can show nodeid

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2001	0x04	rw	bool	/	/	NO

Description:

- 置位会触发红灯/绿灯闪烁, 用来通过 nodeid 来寻找对应的伺服.
- 闪烁方式: 绿灯闪烁次数为 nodeid 高半字节的数值, 红灯闪烁次数为低半字节数值. 比如 nodeid=47=0x2F. 那么显示是“先绿灯 2 次, 后红灯 15 次”.
- 闪烁频率为 1Hz.

value	description
0	无影响
1	驱动器表面的 LED 灯会通过闪烁的方式告知本驱动器的 nodeid.

0x2002, motor para, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x00	const	uint8	/	13	NO

phase resistor

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x01	ro	uint32	$10^{-6} \Omega$	/	NO

phase inductance

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x02	ro	uint32	10^{-6} H	/	NO
KE						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x03	ro	uint32	10^{-6} VS/RAD	/	NO
KV						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x04	ro	uint32	10^{-6} RPM/V	/	NO
KT						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x05	ro	uint32	10^{-6} N.m/A	/	NO
KJ						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x06	ro	uint32	10^{-6} Kg.m ²	/	NO
KB						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x07	ro	uint32	10^{-6} Kg.m ²	/	NO
rated voltage						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x08	ro	uint32	10^{-6} V	/	NO
rated current						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x09	ro	uint32	10^{-6} A	/	NO
rated torque						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x0A	ro	uint32	10^{-6} N.m	/	NO
inertia						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x0B	ro	uint32	10^{-6} N.m	/	NO
biscus						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x0C	ro	uint32	10^{-6} N.m	/	NO
inertiaTrigger						
Index	subIndex	Access	Type	Units	Default	Map PDO
0x2002	0x0D	ro	bool	/	/	NO

0x2003, CPR, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2003	0x00	const	uint8	/	2	NO

count per resolution

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2003	0x01	rw	uint32	/	/	NO

Description:

- 当处于“本地受限状态”时, count per resolution 不会被修改, 本地受限状态的定义请查看[本地受限状态](#).
- 电机转动一圈位置增加的脉冲数, 单圈产品默认 10000, 多圈产品默认 16384, 超过范围的会 SDO 报错.
- 不建议修改, 若想修改单位, 可以通过 0x6093/0x6095/0x6097 实现.

multiturn count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2003	0x02	r	uint32	/	/	NO

Description:

- 单圈产品默认为 1, 多圈产品默认为 262144 或者为 16384.

0x2005, additionalNMT, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2005	0x00	const	uint8	/	8	NO

nmt reboot

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2005	0x01	rw	bool	/	0	NO

Description:

- 置位后, 当接收到厂家自定义 NMT 命令 NMT_REBOOT 后, 重启一体伺服.
- NMT 命令描述见章节 3.1.1 NMT.

nmt quickstop

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2005	0x02	rw	bool	/	0	NO

Description:

- 置位后, 当接收到厂家自定义 NMT 命令 NMT_QUICKSTOP 后, 等同执行控制字的 QUICKSTOP, quickstop 的描述在章节 3.2.2.
- MNMT 命令描述见章节 3.1.1 NMT.

nmt shutdown

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2005	0x03	rw	bool	/	0	NO

Description:

- 置位后, 当接收到厂家自定义 NMT 命令 NMT_SHUTDOWN 后, 等同执行控制字的 SHUTDOWN, shutdown 的描述在章节 3.2.2.
- NMT 命令描述见章节 3.1.1 NMT.

0x2010, PID Current, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x00	const	uint8	/	6	NO

current kp default

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x01	ro	uint32	10^{-3}	/	NO

Description:

- kp、ki 无量纲, 10^{-3} 表示实际使用值是设置值的千分之一. 下同.

current ki default

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x2010	0x02	ro	uint32	10 ⁻³	/	NO
--------	------	----	--------	------------------	---	----

current kp

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x03	rw	uint32	10 ⁻³	/	NO

current ki

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x04	rw	uint32	10 ⁻³	/	NO

current bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x05	rw	uint32	/	/	NO

Description:

- 增大 bandwidth 会同比例增加 kp 和 ki, 减小同理.
- 2010:03 = 2010:01 * 2010:05/1000.
- 2010:04 = 2010:02 * 2010:05/1000.

autotune trigger

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2010	0x06	ro	bool	/	/	NO

Description:

value	description
0	无影响
1	触发自动调整电流环 PID 的 kp 和 ki, 调整后新的 kp 和 ki 立马生效, 需要使能” store manufacturer and application parameters “(index=0x1010, subindex=0x03) 才会永久生效.

- 一体伺服产品的电流环不建议使用自动整定, 因为电流环参数只和电机参数有关, 和负载/工况无关. 使用出厂配置即可.

0x2012, PID Velocity, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x00	const	uint8	/	6	NO

velocity kp default

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x01	ro	uint32	10 ⁻³	/	NO

Description:

- kp、ki 无量纲, 10⁻³ 表示实际使用值是设置值的千分之一. 下同.

velocity ki default

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x02	ro	uint32	10 ⁻³	/	NO

velocity kp

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x03	rw	uint32	10 ⁻³	/	NO

velocity ki

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x04	rw	uint32	10 ⁻³	/	NO

stiffness

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x2012	0x05	rw	uint32	10 ⁻³	1000	NO
--------	------	----	--------	------------------	------	----

Description:

- 千分比单位, 1000 表示 1 倍, 2000 表示 2 倍, 500 表示 0.5 倍.
- 速度刚性, 也可以理解为速度环带宽. 修改会同步修改速度环的 Kp 和 Ki. 计算公式如下:

$$2012:03 = 2012:01 * 2012:05/1000.$$

$$2012:04 = 2012:02 * 2012:05/1000.$$
- 好处单参数调整 PID, 比分开调节 P 和 I 容易多了.

autotune trigger

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2012	0x06	rw	bool	/	/	NO

Description:

value	description
0	无影响
1	触发自动调整速度环 PID 的 kp 和 ki, 调整后新的 kp 和 ki 立马生效, 需要使能” store manufacturer and application parameters “(index=0x1010, subindex=0x03) 才会永久生效.

- 出厂配置的速度环的 kp 和 ki 是针对空载情况的, 如负载有较大改变, 建议使用 velocity autotune 后再使用伺服产品, 若对自动整定不满意, 请尝试用 2012:05 来单参数配置 PID.

0x2013, PID Position, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2013	0x00	const	uint8	/	6	NO

position kp default

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2013	0x01	ro	uint32	10 ⁻³	/	NO

Description:

- kp、ki 无量纲, 10⁻³表示实际使用值是设置值的千分之一. 下同.

position kp

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2013	0x02	rw	uint32	10 ⁻³	/	NO

autotune trigger

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2013	0x03	rw	bool	/	/	NO

Description:

value	description
0	无影响
1	触发自动调整位置环 PID 的 kp, 调整后新的 kp 立马生效, 需要使能” store manufacturer and application parameters “(index=0x1010, subindex=0x03) 才会永久生效.

0x2014, brake control, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x00	const	uint8	/	8	NO

is there a brake

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x01	ro	bool	/	/	NO

Description:

value	description
0	没有刹车(抱闸)
1	有刹车(抱闸)

is inside control the brake

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x02	ro	bool	/	1	NO

Description:

value	description
0	外部设备控制刹车
1	伺服控制刹车

brake current

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x03	ro	uint16	mA	/	NO

Description:

- 抱闸电流, 松闸时电流较大, 大概 280mA.

auto brake

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x04	rw	bool	/	/	NO

Description:

value	description
0	关闭自动刹车(抱闸)
1	使能自动刹车(抱闸). 有内置抱闸时自动抱闸

自动抱闸在以下情况下起作用:

- 标准位置模式时到达目标位置且没有等待执行的新目标.
- 标准速度模式时到达目标速度且目标速度为 0.
- 标准扭矩模式时到达目标扭矩且目标扭矩为 0.
- 标准位置/标准速度/标准扭矩/回零等模式执行 HALT 后停止运动时.

有抱闸的一体伺服, 不管是否置位自动抱闸. 其使能(enable operation)时都会自动松闸, 失能(disable operation)后都会自动抱闸, 松闸时间需要大约 100ms. 所以在 csp/csv/cst/ip 模式下建议用(2014:07)提前手动松闸.

brake now

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x05	rw	bool	/	/	NO

Description:

value	description
0	无影响
1	刹车(抱闸). 有内置抱闸时立即抱闸

brake status

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x06	ro	uint8	/	/	TPDO

Description:

brake status	meaning
0	braked
1	brake in progress
2	unbraked
3	unbrake in progress

unbrake now

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x07	rw	bool	/	/	NO

Description:

value	description
0	无影响
1	松开刹车(抱闸). 有内置抱闸时立即松开抱闸.

unbrake trigger current

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2014	0x08	r	uint16	mA	/	NO

Description:

- 只读, 这个参数会跟随者伺服温度的变化而变化. 结合 2014:03h 可以判定是否有抱闸故障. 松闸后 2014:03 会大于 2014:08.

0x2015, servo strategy, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x00	const	uint8	/	8	NO

CSP do trajectory planning

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x01	rw	bool	/	1	NO

Description:

value	description
0	cyclic synchronous position mode 下, 用 target position(0x607A)+position offset(0x60B0) 之和作为目标位置, 直接赋值为伺服执行层的 position demand value(0x6062), velocity offset(0x60B1) 作为速度前馈.
1	cyclic synchronous position mode 下, 用 target position(0x607A)+position offset(0x60B0) 之和作为伺服执行的目标位置, 触发伺服的位置规划. 定时获取规划的输出 csptraj.positionDemandValue 作为伺服执行层的 position demand value(0x6062), 输出 csptraj.velocityDemadnValue 作为作为速度前馈.

- CSP 模式是否有规划(trajectory planning)对伺服的运动效果有明显影响. 举个对比的例子, 在时刻 0 设置 target position 为 1000 个脉冲, 设置插补周期(60C2)为 10ms. 那么在没有规划的时候伺服在时刻 1ms 开始运动, 可能在 2ms 这个时刻, 也可能在 5ms 这个时刻到达 pulse=1000 的位置, 之后剩余时间是保持不动. 在有规划的时候, 伺服的从时刻 1ms 开始运动, 在时刻 11ms 的时候刚好到达 pulse=1000 的位置. 所以当 CSP 模式发送目标的插补周期比较大(大于 5ms)的是时候, 有规划的运行起来明显比没有规划的平稳.

CSV do trajectory planning

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x02	rw	bool	/	1	NO

Description:

value	description
0	cyclic synchronous velocity mode 下, 用 target velocity(0x60FF)+velocity offset(0x60B1) 之和作为目标速度, 直接赋值为伺服执行层的 velocity demand value(0x606B).
1	cyclic synchronous velocity mode 下, 用 target velocity (0x60FF)+velocity offset(0x60B1) 之和作为伺服执行的目标速度, 触发伺服的速度规划, 定时获取规划的输出 csvtraj.velocityDemandValue 作为伺服执行层的 velocity demand value(0x606B).

- CSV 模式是否有规划(trajectory planning)对伺服的运动效果有明显影响. 可以参考 0x2015:01 的说明对比.

CST do trajectory planning

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x2015	0x03	rw	bool	/	1	NO
--------	------	----	------	---	---	----

Description:

value	description
0	cyclic synchronous torque mode 下, 用 target torque (0x6071)+torque offset (0x60B2) 之和作为目标扭矩, 直接赋值为伺服执行层的 torque demand value (0x6074).
1	cyclic synchronous torque mode 下, 用 target torque (0x6071)+torque offset (0x60B2) 之和作为伺服执行的目标扭矩, 触发伺服的扭矩规划, 定时获取规划的输出 csttraj. torqueDemandValue 作为伺服执行层的 torque demand value (0x6074).

- CST 模式是否有规划(trjectory planning)对伺服的运动效果有明显影响. 可以参考 0x2015:01 的说明对比.

torque with position

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x04	rw	bool	/	0	NO

Description:

value	description
0	扭矩模式(profile torque mode/cyclic synchronous torque mode)不带位置控制, 扭矩模式下的最大速度受到 torque mode max velocity (2032:01h) 的限制, 最大扭矩受到 (torque mode max velocity - velocityActualValue)*torque mode KP (2032:02h) 的限制, 当将 torque mode max velocity 设置的很小时, 低速也无法发挥大扭矩.
1	扭矩模式(profile torque mode/cyclic synchronous torque mode)带位置控制. 扭矩模式下的最大速度受到 torque mode max velocity (2032:01h) 的限制, 零速也能保持大扭矩.

- 置位时支持零速有输出扭矩, 意味着 control word 给 halt 命令或者 quickstop 命令时停下来后有输出扭矩.
- 清零时不支持零速无输出扭矩, 意味着 control word 给 halt 命令或者 quickstop 命令时停下来后无输出扭矩.

stepdir position without canopen

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x05	rw	bool	/	0	NO

Description:

value	description
0	脉冲位置模式的使能还需要通过 6040h 来实现, 也就是 pulse 和 dir 信号起作用, enable 信号不起作用. canopen 总线正常工作.
1	脉冲位置模式的控制完全由 3 个 IO 来实现, DI1 为 pulse, DI2 为 dir, DI3 为 enable.

- 置 1 后保存后重上电才能彻底关闭 canopen, 如果需要重新配置为其它模式需要清零本 bit 后保存再重上电.
- 置位本 bit 会清零 2015:06.

stepdir velocity without canopen

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x06	rw	bool	/	0	NO

Description:

value	description
0	脉冲速度模式的使能还需要通过 6040h 来实现, 也就是 pulse 和 dir 信号起作用, enable 信号不起作用. canopen 总线正常工作.
1	脉冲速度模式的控制完全由 3 个 IO 来实现, DI1 为 pulse, DI2 为 dir, DI3 为 enable.

- 置 1 后保存后重上电才能彻底关闭 canopen, 如果需要重新配置为其它模式需要清零本 bit 后保存再重上电.
- 置位本 bit 会清零 2015:05.

homing stop at zero position

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x07	rw	bool	/	1	NO

Description:

- 如果零点和原点相同, 即 home offset (0x607C) 的值为零, 那么本设置不导致任何差别.
- 如果零点和原点不同, 即 0x607C 不等于 0, 那么:

- 置 1 时,回零执行过程中到达原点后会按照 home offset (0x607C) 的值运动到零点,然后停在零点.
- 清零时,回零执行过程中到达原点后停在原点,停下后当前实际位置 (0x6064) 等于负数的 home offset.

profile position mode without overshoot

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2015	0x08	rw	bool	/	0	NO

Description:

- 在标准位置模式下,比如 T 型规划的时候,在 T 型的减速阶段更新规划,规划中不更改 pv (6081),不更改 pa (6083),不更改 pd (6084),但是运动里程减小,这种情况下:
 - 本设置=0,那么由于 pd 的限制,规划位置 (6062) 的最大值是未更新前的目标位置,规划最终位置是更新后的目标位置,而实际位置 (6064) 总是跟随规划位置,那么意味着有过冲.
 - 本设置=1,那么实际减速阶段用的 pd 将在 6084 和 60C6 之间选择,最大不超过 60C6 的值,当实际 pd 不超过 60C6 时,规划位置的最大值是新的目标位置,若实际 pd 超过 60C6,那么规划位置的最大值在新/旧目标位置之间. 这样可以大大减小过冲.

0x2016, filter, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x00	const	uint8	/	8	NO

velocity filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x01	rw	uint32	Hz	1000	NO

Description:

- 一阶速度滤波器带宽,用于过滤速度数据.
- 默认值依据电机特性而定,一般在 100~1000 之间.

acceleration filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x02	rw	uint32	Hz	1	NO

Description:

- 一阶加速度滤波器带宽,用于过滤加速度数据.

iit filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x03	rw	uint32	Hz	10	NO

Description:

- 一阶 IIT 滤波器带宽,用于过滤 IIT 数据.

power filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x04	rw	uint32	Hz	1	NO

Description:

- 一阶 power 滤波器带宽,用于过滤功率数据.

temperature filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x05	rw	uint32	Hz	10	NO

Description:

- 一阶温度滤波器带宽,用于过滤温度数据.

bus voltage filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x06	rw	uint32	Hz	10	NO

Description:

- 一阶 VBUS 滤波器带宽, 用于过滤母线电压数据.

brake current filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x07	rw	uint32	Hz	100	NO

Description:

- 有内置刹车地伺服, 用于过滤刹车电流数据.

input Position filter bandwidth

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2016	0x08	rw	uint32	Hz	10	NO

Description:

- 输入位置的滤波器的带宽, 目前只用于脉冲输入模式.

0x2017, IIT, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2017	0x00	const	uint8	/		NO

IIT limit

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2017	0x01	rw	uint32	/	12	NO

Description:

- IIT 过流检测, 常用于堵转检测, 当 IIT 触发累加后的 IIT value 超过 IIT limit 触发 IIT 告警.
- 默认的 12 意味着 Iqrms(有效值)为额定电流 1.5 倍运行 10 秒触发 IIT 告警.

IIT value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2017	0x02	ro	uint32	/	/	TPDO

Description:

- IIT 累加的实时值. 计算方式是 $(I_{qrms} * I_{qrms} - I_{rated} * I_{rated}) * second$.

IIT on threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2017	0x03	rw	uint32	/	1100	NO

Description:

- 千分比单位. 默认的 1100 意味着当 Iq 达到额定电流 (rated current 0x2002:08) 的 1.1 倍时开始 IIT 累加.

0x2018, bus voltage, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x00	const	uint8	/	5	NO

over voltage threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x01	rw	uint32	mV	/	NO

Description:

- 用户设置的过压保护值, 当母线电压超过该值时触发用户过压保护 ([0x3110](#)).
- 初始值为 $0.95 * max_voltage$ (0x2018:03), 设置是不能超过 max voltage.

under voltage threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x02	rw	uint32	mV	/	NO

Description:

- 用户设置的欠压保护值, 当母线电压低于该值时触发用户欠压保护.
- 默认值为 min voltage(0x2018:04), 设置时不能低于 min voltage.

max voltage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x03	ro	uint32	mV	/	NO

Description:

- 硬件过压保护值, 当母线电压超过该值时触发硬件过压保护.
- 触发硬件过压保护时, 为了保护伺服以及外部设备, 会强制短路伺服上的 MOSFET 来降压.
- 硬件过压保护和用户过压保护的不同之处是, 用户过压保护会执行 fault reaction(按照 0x605E 的配置执行), 硬件过压保护不会执行 fault reaction.
- 各伺服的默认值不同.

min voltage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x04	ro	uint32	mV	/	NO

Description:

- 欠压保护值, 当母线电压低于该值时触发硬件欠压保护.
- 触发硬件欠压保护时, 会执行 fault reaction, 需要注意的是, 实际未必有足够的供电电压/电流来执行完成 fault reaction.
- 各型号的默认值不同.

percent usage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2018	0x05	rw	uint8	/	/	NO

Description:

- 限制母线电压的使用比例, 默认 100 为 100%, 特殊场合有用.

0x2019, EMCY enable, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x00	const	uint8	/	16	NO

enable velocity lost

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x01	rw	bool	/	1	NO

Description:

- 置 1 时使能 [超速保护 \(0x8401\)](#).

enable IIT

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x02	rw	bool	/	1	NO

Description:

- 置 1 时使能 [IIT 过流保护 \(0x2314\)](#).

enable heartbeat lost

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x03	rw	bool	/	1	NO

Description:

- 置 1 时使能 [心跳丢失保护 \(0x8130\)](#).

enable sync timeout

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x2019	0x04	rw	bool	/	1	NO
--------	------	----	------	---	---	----

Description:

- 置 1 时使能[同步丢失保护 \(0x8201\)](#).

enable rx overrun

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x05	rw	bool	/	1	NO

Description:

- 置 1 时使能[CAN 接收 OVERRUN 保护 \(0x8112\)](#).

enable tx overrun

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x06	rw	bool	/	1	NO

Description:

- 置 1 时使能 CAN 发送[OVERRUN 保护 \(0x8111\)](#).

enable remote reset

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x07	rw	bool	/	1	NO

Description:

- 置 1 时使能[远程节点复位保护 \(0x8131\)](#).

enable TPD0 out window

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x08	rw	bool	/	1	NO

Description:

- 置 1 时使能[TPD0 out of window 保护 \(0x8202\)](#).

enable RPD0 wrong length

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x09	rw	bool	/	1	NO

Description:

- 置 1 时使能[RPD0 wrong length 保护 \(0x8210/0x8220\)](#).

enable RPD0 timeout

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2019	0x0A	rw	bool	/	1	NO

Description:

- 置 1 时使能[0x8250, canopen rpdo timeout 保护 \(0x8250\)](#).

0x201A, feedforward, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201A	0x00	const	uint8	/	2	NO

velocity feedforward

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201A	0x01	rw	uint16	10 ⁻³	1000	NO

Description:

- 设置为 1000 时表示使用 100%的速度前馈.

acceleration feedforward

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201A	0x02	rw	uint16	10 ⁻³	0	NO

Description:

- 设置为 1000 时表示使用 100%的加速度前馈.

0x201B, temperature, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x00	const	uint8	/	10	NO

amplifier temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x01	ro	int32	10 ⁻³ 度	/	TPDO

Description:

- 伺服驱动的实时温度.

motor temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x02	ro	int32	10 ⁻³ 度	/	TPDO

Description:

- 电机的实时温度.
- 注意, 实际未必支持, 一体伺服的测温经常在驱动部分, 电机部分的温度可以估算为 201B:01 再加上 15 度.

max amplifier temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x03	rw	int32	10 ⁻³ 度	95000	NO

Description:

- 伺服驱动允许工作的最高温度, 默认 95 度.

min amplifier temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x04	rw	int32	10 ⁻³ 度	-40000	NO

Description:

- 伺服驱动允许工作的最低温度, 默认-40 度.

max motor temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x05	rw	int32	10 ⁻³ 度	95000	NO

Description:

- 电机允许工作的最高温度, 默认 100 度.

min motor temperature

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x06	rw	int32	10 ⁻³ 度	-40000	NO

Description:

- 电机允许工作的最低温度, 默认-40 度.

board temperature warning on threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x07	rw	int32	10 ⁻³ 度	0x00	NO

Description:

- 电机温度告警触发阈值, 触发时会将 6041 的 bit8 置位.
- 最小设置值是 0, 表示任何温度都不触发.
- 最大设置值是 0x201B:03.

board temperature warning off threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x08	rw	int32	10 ⁻³ 度	0x00	NO

Description:

- 电机温度告警关闭阈值, 触发时会将 6041 的 bit8 清零.

- 最小设置值是 0, 表示任何温度都不触发.
- 最大设置值是 0x201B:03.

motor temperature warning on threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x09	rw	int32	10 ⁻³ 度	0x00	NO

Description:

- 预留, 实际无电机温度传感器, 只有驱动温度传感器.
- 电机温度告警触发阈值, 触发时会将 6041 的 bit8 置位.
- 最小设置值是 0, 表示任何温度都不触发.
- 最大设置值是 0x201B:03.

motor temperature warning off threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201B	0x0A	rw	int32	10 ⁻³ 度	0x00	NO

Description:

- 预留, 实际无电机温度传感器, 只有驱动温度传感器.
- 电机温度告警关闭阈值, 触发时会将 6041 的 bit8 清零.
- 最小设置值是 0, 表示任何温度都不触发.
- 最大设置值是 0x201B:03.

0x201C, brake resistor, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201C	0x00	const	uint8	/	4	NO

brake resistor value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201C	0x01	ro	uint32	mOhm	/	NO

Description:

- 外挂刹车电阻的阻值, 毫欧单位.
- 24V 供电的推荐 2.2 欧姆, 48V 供电的建议 4.7 欧姆.
- 注意, 是带刹车电阻的机型才有用到本设置.

brake start voltage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201C	0x02	ro	uint32	mV	/	NO

Description:

- 外挂刹车电阻开始起作用的起始电压, 毫伏单位.
- 注意其实电压要稍微高于供电电压.

brake end voltage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201C	0x03	rw	uint32	mV	/	NO

Description:

- 外挂刹车电阻完全打开的电压.

reserved

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201C	0x04	rw	uint32	/	/	NO

0x201D, vbat, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201D	0x00	const	uint8	/	2	NO

low voltage detect threshold

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201D	0x01	rw	uint32	mV	2500	NO

Description:

- 外接电池电压告警门限, 默认 2.5V, 不接电池不会告警.

battery voltage

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201D	0x02	ro	uint32	mV	/	NO

Description:

- 多圈产品使用, 多圈产品有接口外接电池.

0x201E, adaptive notch filter, RECORD**maxSubindex**

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201E	0x00	const	uint8	/	3	NO

enable

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201E	0x01	rw	bool	/	/	NO

Description:

- 置 1 使能自适应带阻滤波器, 用于速度采样滤波. 部分产品默认没有使能, 部分产品默认使能

q

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201E	0x02	rw	uint32	/	30	NO

Description:

- 自适应带阻滤波器的 Q 值.

low frequency limit

Index	subIndex	Access	Type	Units	Default	Map PDO
0x201E	0x03	rw	uint32	Hz	200	NO

Description:

- 自适应带阻滤波器的低频限制, 低频下自动关闭.

0x2030, torque window, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2030	0x00	rw	uint16	10 ⁻³ RT	10	NO

Description:

- profile torque mode 时, 和 0x2031 组合使用判断是否 target reached.

0x2031, torque window timeout, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2031	0x00	rw	uint16	10 ⁻³ S	1	NO

Description:

- profile torque mode 时, 和 0x2030 组合使用判断 PT 模式下是否 target reached.

0x2032, torqueModeAdditionalPara, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2032	0x00	const	uint8	/	2	NO

torque mode max velocity

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2032	0x01	rw	uint32	/	1000000	NO

Description:

- 扭矩模式的最大速度, 扭矩模式包括 profile torque mode 以及 cyclic sync torque mode. 其它模式的最大速度限制是 607F.
- 用户单位, 默认用户单位为 pulse/s.
- 可以设置为 0, 配合 2015:04=1 时实现零速有输出保持扭矩. 当 2015:04=0 时, 本对象设置为 0 无输出扭矩.

torque mode kp

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2032	0x02	rw	uint32	/	/	NO

Description:

- 当 2015:04=0 时, 本对象用来调节扭矩模式的扭矩输出, 当实际速度达到 2032:01 时无扭矩输出, 当实际速度远小于 2032:01 时能输出目标扭矩. 本对象值越大, 实际速度接近 2032:01 时能输出的扭矩越大. 本对象值为零时无扭矩输出.
- 当 2015:04=1 时, 本对象不影响输出扭矩.

0x2033, sync in time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2033	0x00	rw	uint16	10^{-6} S	300	TPDO

Description:

- 单位是 us, 微秒.
- CSP/CSV/CST 等工作状态下, 触发帧的 jitter 允许误差, 超过该值会被调整, 没有超过的认为是允许的 jitter.

0x2034, internal target reach window, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2034	0x00	rw	uint16	10^{-5}	10	NO

Description:

- 弃用, 本 object 还能读写, 但实际已经没有任何效果.

0x2035, sync delay time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2035	0x00	rw	uint16	ms	1	NO

Description:

- IP/CSP/CSV/CST 等模式的命令执行延时. 默认 1 毫秒.

- 用户并不需要改动, 除非单条 PDO/SDO 的报文时间超过 1 毫秒. 比如使用小于 100kbps 的 can 波特率来执行 IP/CSP/CSV/CST 等运动控制, 比如 10kbps 下, 单条 PDO/SDO 的报文时间达到 11.25 毫秒, 那么这时候 SYNC 帧的帧间隔误差可能被其他 SDO/PDO 所影响而远超 1ms. 那么需要将本值修改为大于 11.25ms, 比如 12ms.

0x203A, homing block configuration, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203A	0x00	const	uint8	/	2	NO

minimum current for block detection

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203A	0x01	rw	int32	10 ⁻³	-250	NO

Description:

- 堵转回零时的堵转检测阈值. 当配置为正值时, 该值单位为 mA, 比如配置为 100 意味着阈值为 100mA. 当配置为负值时, 该值为千分比单位, 比如配置为-100 时, 阈值为额定扭矩的千分之一百, 也就是 0.1 倍.
- 该值和 0x203A:02 配合用于 homing mode 中的堵转检测, 替代 limit switch 实体开关.

block duration

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203A	0x02	rw	uint16	ms	10	NO

Description:

- 堵转的持续时间, 超过这个时间判定 NBL(negative block limit)或者 PBL(positive block limit)发生有效变化, 单位毫秒.

0x203B, limit switch, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203B	0x00	const	uint8	/	1	NO

limit switch type

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203B	0x01	rw	uint32	/	0	NO

Description:

- bit0 对应 NLS, bit1 对应 PLS, 其余 bit 暂时没有用处.
- bit0=1 意味着 NLS 是电平型的开关, bit0=0 意味着 NLS 是边沿性开关.
- bit1=1 意味着 PLS 是电平型的开关, bit1=0 意味着 PLS 是边沿性开关.

0x203C, mode switch amplifier, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x203C	0x00	rw	uint16	/	1	TPDO

Description:

- 本值设置使能状态下切换工作模式的速度限制, 非使能状态下无限制.
- 当实际速度小于本值和 606F 的乘积是允许模式切换, 模式切换不考虑 6070 的值.
- 当有高速模式切换需求时, 建议修改本值, 不要修改 606F.

0x2080, sync received, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
-------	----------	--------	------	-------	---------	---------

0x2080	0x00	r	uint8	/	0	TPDO
--------	------	---	-------	---	---	------

Description:

- 本值在收到 SYNC 帧后被设置为 SYNC 帧的数据, 若 SYNC 帧不带数据, 则设置为 0xFF.
- 本值在 TPDO 发送后自动清零, 若没设置 TPDO, 那么也会自动清零, 所以 SDO 读到的都是 0x00.
- 本值的 TPDO 触发建议使用 SYNC 方式 (1~240) 或者 COS 方式 (254), 定时方式 (255) 会无效.

0x2240, digital inputs control, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2240	0x00	const	uint8	/	4	NO

special function enable

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2240	0x01	rw	uint32	/	0	NO

Description:

- 此位可关闭 (值 “0”) 或打开 (值 “1”) 输入的特殊功能。例如, 如果输入 1 未用作反向限位开关, 则必须关闭特殊功能, 从而防止对信号发生器做出错误响应。此对象对位 16 至 31 没有影响。
- 位 0: 将 DI1 作为负向限位开关 (NLS).
- 位 1: 将 DI2 作为正向限位开关 (PLS).
- 位 2: 将 DI3 作为零位开关 (HS).
- 置位 special function 会去使能对应 INPUT 的 estop 功能.

input inverted

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2240	0x02	rw	uint32	/	0	NO

Description:

- 输入信号的取反, 当输入设备类型为 NC 时常需要置位, 输入设备类型为 NO 时不需要置位.
- 位 0: DI1 的取反.
- 位 1: DI2 的取反.
- 位 2: DI3 的取反.

input raw value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2240	0x03	ro	uint32	/	0	TPDO

Description:

- 输入的原始值, 不受 2240:02 的影响.

estop enable

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2240	0x04	rw	uint32	/	0	NO

Description:

- estop 功能, 置位会触发快停, 按照 605A 的配置实现快停.
- bit0 对应 DI1, bit1 对应 DI2, bit2 对应 DI3, 三个 IO 可以同时都配置为 estop 功能, 只要有一个 IO 是 estop 且有效, 就处于 estop 状态.
- 处于 estop 状态时, 不能重新使能运动.
- 使能 estop 会去使能该 INPUT 对应的 special function.

0x2250, digital outputs control, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2250	0x00	const	uint8	/	3	NO

function enable

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2250	0x01	rw	uint32	/	0	NO

Description:

- bit0 置位时表示输出 1 的特殊功能使能, 特殊功能是 brake.

output inverted

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2250	0x02	rw	uint32	/	0	NO

Description:

- bit0 置位表示输出反相.

output raw value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x2250	0x03	ro	uint32	/	0	TPDO

Description:

- bit0 为输出 1 的电平.

0x3000, absolute value of current target position, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x3000	0x00	ro	int32	pulse	/	TPDO

Description:

- 只有 PP 模式使用. 为当前正在执行的目标位置.
- pp 模式有绝对目标位置和相对目标位置. 6071_targetPosition 读取回来的是写入的值, 也就是相对位置控制的时候, 如果了解当前运动的绝对目标位置, 可以从本 object 读取. 如果是绝对位置控制且立即执行, 那么 0x3000 和 0x607A 的值一样.

0x3001, current actual value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x3001	0x00	ro	int32	mA	/	TPDO

Description:

- 电流实际值, mA 单位.

0x3100, negative limit switch position, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x3100	0x00	ro	int32	pulse	/	TPDO

Description:

- 当处于 PP/PV/PT/IP/CSP/CSV/CST 等模式时, NLS 使能情况下被触发时的位置被记录在这里.
- 脉冲模式, IO1 作为脉冲输入用, 无法实现 NLS 功能.
- 回零模式下不记录

0x3101, positive limit switch position, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x3101	0x00	ro	int32	pulse	/	TPDO

Description:

- 当处于 PP/PV/PT/IP/CSP/CSV/CST 等模式时, PLS 使能情况下被触发时的位置被记录在这里.
- 脉冲模式, IO2 作为方向使用, 无法实现 PLS 功能.
- 回零模式下不记录.

0x4008, can error count, RECORD

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4003	0x00	const	uint8	/	7	NO

stuff error count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x01	r	uint32	/	0	NO

Description:

- 上电以来 stuff error 的出现次数.

formerror count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x02	r	uint32	/	0	NO

Description:

- 上电以来 form error 的出现次数.

ack error count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x03	r	uint32	/	0	NO

Description:

- 上电以来 ack error 的出现次数.

recessive error count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x04	r	uint32	/	0	NO

Description:

- 上电以来 recessive error 的出现次数.

dominant error count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x05	r	uint32	/	0	NO

Description:

- 上电以来 dominant error 的出现次数.

crc error count

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x06	r	uint32	/	0	NO

Description:

- 上电以来 crc error 的出线次数.

info code

Index	subIndex	Access	Type	Units	Default	Map PDO
0x4008	0x07	r	uint32	/	0	NO

Description:

- 当发生 0x8611 这个 emcy 时, 其 info code 会被更新记录在这里.
- 4008 是个调试对象, 所以 info code 临时保存在这里.

2.3. CIA402 控制字

0x603F, error code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x603F	0x00	ro	uint16	/	/	TPDO

Description:

- 保存最近一次 emergency 的 error code. error code 的定义详见章节“[错误告警](#)”

0x6040, control word, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6040	0x00	rw	uint16	/	/	RPDO

Description:

- 0x6040 和 0x6041 详见章节 [3.2. control and status](#).

0x6041, status word, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6041	0x00	ro	uint16	/	/	TPDO

Description:

- 0x6040 和 0x6041 详见章节 [3.2. control and status](#).

0x605A, quickstop option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x605A	0x00	rw	int16	/	2	NO

Description:

value	description
0	立即停机, 停机后轴处于自由转动状态, 有抱闸且自动抱闸使能时会抱闸, 停止后驱动器进入“STATUS_0x260_SwitchOnDisabled”状态.
1	按照减速斜坡减速停机, 停机后轴处于自由转动状态, 有抱闸且自动抱闸使能时会抱闸. 停止后驱动器进入“STATUS_0x260_SwitchOnDisabled”状态.
2	按照快停减斜坡减速停机, 停机后轴处于自由转动状态, 有抱闸且自动抱闸使能时会抱闸. 停止后驱动器进入“STATUS_0x260_SwitchOnDisabled”状态.
5	按照减速斜坡减速停机, 停机后轴处于锁定状态, 有抱闸且自动抱闸使能时会抱闸. 停止后驱动器进入“STATUS_0x217_QuickstopActive”状态.
6	按照快停减斜坡减速停机, 停机后轴处于锁定状态, 有抱闸且自动抱闸使能时会抱闸. 停止后驱动器进入“STATUS_0x217_QuickstopActive”状态.

- 在没有刹车存在的情况下, 在有重力做工的情况下, 谨慎使用 quickstop 功能.
- PP/PV/CSP/CSV/IP 模式的减速斜率由 0x6084 决定, 快减斜率由 0x6085 决定. CST/PT 模式的减速斜率和快减斜率都是 0x6087 决定. H 模式的减速斜率由 0x609A 决定, 快停斜率由 0x6085 决定.

0x605B, shutdown option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x605B	0x00	rw	int16	/	1	NO

Description:

value	description
-------	-------------

0	立即停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸，停止后驱动器进入“STATUS_0x231_ReadyToSwitchOn”状态。
1	按照减速斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x231_ReadyToSwitchOn”状态。

- 在有重力做工的情况下，谨慎使用 shutdown 功能。
- PP/PV/CSP/CSV/IP 模式的减速斜率由 0x6084 决定。CST/PT 模式的减速斜率是 0x6087 决定。H 模式的减速斜率由 0x609A 决定。

0x605C, disable operation option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x605B	0x00	rw	int16	/	1	NO

Description:

value	description
0	立即停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸，停止后驱动器进入“STATUS_0x233_SwitchedOn”状态。
1	按照减速斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x233_SwitchedOn”状态。

- 在有重力做工的情况下，谨慎使用 disable operation 功能。
- PP/PV/CSP/CSV/IP 模式的减速斜率由 0x6084 决定。CST/PT 模式的减速斜率是 0x6087 决定。H 模式的减速斜率由 0x609A 决定。

0x605D, halt option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x605D	0x00	rw	int16	/	1	NO

Description:

value	description
1	按照减速斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x237_OperationEnabled”状态。
2	按照快停减斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x237_OperationEnabled”状态。

- PP/PV/IP 模式的减速斜率由 0x6084 决定，快减斜率由 0x6085 决定。PT 模式的减速斜率和快减斜率都是 0x6087 决定。H 模式的减速斜率由 0x609A 决定，快停斜率由 0x6085 决定。
- CSP/CSV/CST 等模式不支持 halt。

0x605E, fault reaction option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x605E	0x00	rw	int16	/	1	NO

Description:

value	description
0	立即停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸，停止后驱动器进入“STATUS_0x238_Fault”状态。
1	按照减速斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x238_Fault”状态。
2	按照快停减斜坡减速停机，停机后轴处于自由转动状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x238_Fault”状态。

5	按照减速斜坡减速停机，停机后轴处于锁定状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x237_OperationEnabled”状态，最大保持扭矩为额定的 90%。
6	按照快停减斜坡减速停机，停机后轴处于锁定状态，有抱闸且自动抱闸使能时会抱闸。停止后驱动器进入“STATUS_0x237_OperationEnabled”状态，最大保持扭矩为额定的 90%。

- PP/PV/CSP/CSV/IP 模式的减速斜率由 0x6084 决定，快减斜率由 0x6085 决定。CST/PT 模式的减速斜率和快减斜率都是 0x6087 决定。H 模式的减速斜率由 0x609A 决定，快停斜率由 0x6085 决定。

0x6060, mode of operation, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6060	0x00	rw	int8	/	0x00	RPDO

Description:

- 支持以下这些工作模式：

value	mode	description
-2	step-dir velocity mode	脉冲速度模式
-1	step-dir position mode	脉冲位置模式
0	/	模式未配置
1	profile position mode	标准位置模式
3	profile velocity mode	标准速度模式
4	profile torque mode	规划力矩模式
6	homing mode	回零模式
7	interpolated position mode	插补位置模式
8	cyclic synchronous position mode	循环同步位置模式
9	cyclic synchronous velocity mode	循环同步速度模式
10	cyclic synchronous torque mode	循环同步扭矩模式

- 当电机已经使能后想切换模式，需要电机的实际速度 (606Ch) 小于初始速度 (606Fh)*(203Ch)。

0x6061, mode of operation display, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6061	0x00	ro	int8	/	0x00	TPDO

0x6062, position demand value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6062	0x00	ro	int32	pulse	/	TPDO

Description:

- 单位为用户单位，默认情况下的单位为 pulse，可通过配置 0x6093 position factor 来修改为需要的单位。

0x6064, position actual value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6064	0x00	ro	int32	pulse	0	TPDO

Description:

- 单位为用户单位，默认情况下的单位为 pulse，可通过配置 0x6093 position factor 来修改为需要的单位。

0x6065, following error window, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6065	0x00	rw	uint32	pulse	10000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse, 可通过配置 0x6093 position factor 来修改为需要的单位.
- 需要注意修改 position factor 不会自动更改 following error window 的数值, 需要手动更改配置.
- 配合 0x6066 实现 following error 错误判断, 判定方法详见” [0x8611, following error](#)”.

0x6066, following error timeout, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6066	0x00	rw	uint32	ms	1	RPDO

Description:

- 单位为 ms, 也就是毫秒.
- 配合 0x6065 实现 following error 错误判断, 判定方法详见” [0x8611, following error](#)”.

0x6067, position window, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6067	0x00	rw	uint32	pulse	10	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse, 可通过配置 0x6093 position factor 来修改为需要的单位.
- 需要注意修改 position factor 不会自动更改 position window 的数值, 需要手动更改配置.
- 和 0x6068 组合使用, 用于 profile position mode 下判定是否到达目标位置.

0x6068, position window time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6068	0x00	rw	uint16	ms	1	RPDO

Description:

- 和 0x6067 组合使用, 用于 PPM 和 IPM 模式下判定是否到达目标位置.

0x606B, velocity demand value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x606B	0x00	ro	int32	pulse/s	0	TPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.

0x606C, velocity actual value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x606C	0x00	ro	int32	pulse/s	0	TPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.

0x606D, velocity window, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x606D	0x00	rw	uint16	pulse/s	1666	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.
- 需要注意修改 velocity factor 不会自动更改 velocity window 的数值, 需要手动更改配置.
- 和 0x606E 组合使用, 用于 PVM 模式下判定是否到达目标速度.

0x606E, velocity window time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x606E	0x00	rw	uint16	ms	1	RPDO

Description:

- 和 0x606D 组合使用, 用于 PVM 模式下判定是否到达目标速度.

0x606F, velocity threshold, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x606F	0x00	rw	uint16	pulse/s	166	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.
- 需要注意修改 velocity factor 不会自动更改 velocity threshold 的数值, 需要手动更改配置.
- 和 0x6070 组合使用, 用于 PPM 和 PVM 模式下判定是否零速.

0x6070, velocity threshold time, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6070	0x00	rw	uint16	ms	1	RPDO

Description:

- 和 0x606F 组合使用, 用于 PPM 和 PVM 模式下判定是否零速.

0x6071, target torque, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6071	0x00	rw	int16	$10^{-3}RT$	/	TPDO+RPDO

Description:

- RT=rated torque, rated torque 定义于 2002h:0A.
- 目标扭矩, 用于 profile torque mode 和 cyclic synchronous torque mode.

0x6072, max torque, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6072	0x00	rw	int16	10 ⁻³ RT	/	RPDO

Description:

- RT=rated torque.
- 各型号伺服的默认值不同.

0x6074, torque demand, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6074	0x00	ro	int16	10 ⁻³ RT	/	TPDO

Description:

- RT=rated torque.

0x6075, motor rated current, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6075	0x00	ro	uint32	mA	/	NO

0x6077, torque actual value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6077	0x00	ro	int16	10 ⁻³ RT	0	TPDO

Description:

- RT=rated torque.
- 当值为 1000 的时候标示输出 rated torque (标称扭矩).

0x6079, dclink circuit voltage, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6079	0x00	ro	uint32	mV	/	TPDO

Description:

- 母线电压实时值, 单位是 mV, 毫伏
- 和 object 0x2018 vbus 配合判断有否发出如下错误告警:
 - [0x3110 over voltage](#)
 - [0x3111 over max voltage](#)
 - [0x3120 under voltage](#)
 - [0x3121 under min voltage](#)

0x607A, target position, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x607A	0x00	rw	int32	pulse	0	TPDO+RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse, 可通过配置 0x6093 position factor 来修改为需要的单位.

0x607C, home offset, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x607C	0x00	rw	int32	/	0x00	NO

Description:

- 定义 home position 和 zero position 的差值. **zero position= home position + home offset.**
- homing mode 的目标是寻找 home position, 停在 zero position, 当然如果 home offset 为零时, home position 和 zero position 重合.

0x607D, software position limit, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x607D	0x00	const	uint8	/	0x02	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x607D	0x01	rw	int32	/	0x80000000	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x607D	0x02	rw	int32	/	0x7FFFFFFF	NO

Description:

- subindex=01 的是 min position limit, subindex=02 的是 max position limit, 都是用户单位.
- 当 min position limit=0x80000000 时, min position limit 不起作用, 也就是不做限制.
- 当 max position limit=0x7FFFFFFF 时, max position limit 不起作用, 也就是不做限制.
- 当 position limit 不是默认值时会触发使用, 使用前会被校准, 校准方法为:
corrected min position limit = min position limit - home offset
corrected max position limit = max position limit - home offset
- 直接的理解方法是: min position limit 是从 home offset 往负方向移动的最大距离, 经常为 home switch 和 negative limit switch 之间的距离. max position limit 是从 home offset 往正方向移动的最大距离, 经常为 home switch 和 positive limit switch 之间的距离. 注意基准点(-home offset), 也就是 home position, 不是 zero position.

0x607E, polarity, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x607E	0x00	rw	uint8	/	0x00	RPDO

Description:

BIT	description
bit5	置 1 时, 扭矩模式下的目标扭矩取反.
bit6	置 1 时, 速度模式的目标速度取反.
bit7	置 1 时, 位置模式下的目标位置取反.

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看[本地受限状态](#)
- 当电机定义的 CW 和 CCW 和使用预期不一致时, 使用本 object.
- 实际这 3 个 bit 的作用是合一的, 任何 1 个 bit 的置位都会导致 3 个 bit 全部被置位. 比如置位 bit5, 目标扭矩取反, 那么导致实际扭矩/实际速度/实际位置都会取反. 又比如置位 bit7, 目标位置取反, 那么导致实际扭矩/实际速度/实际位置都取反.

0x607F, max profile velocity, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x607F	0x00	rw	uint32	pulse/s	1000000	RPDO

Description:

- 过速(error code =0x8401)的判定依据是当前速度的绝对值超过 1.2 倍的 max profile velocity. 且持续时间超过 10ms.
- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.
- 需要注意修改 velocity factor 不会自动更改 max profile velocity 的数值, 需要手动更改配置.
- 可以在电机运动时修改 maxProfileVelocity, 用户层的 maxProfileVelocity 修改后, 控制层的 maxProfileVelocity 会依据 profileAcceleration 或者 profileDeceleration 逐渐改变.
- max profile velocity 适用于 profile position mode/profile velocity mode/interpolated position mode.
- profile torque mode 的最大速度限制为 2032:01H.

0x6080, max motor speed, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6080	0x00	ro	uint32	RPM	6000	NO

Description:

- 该值仅供参考, 不可修改, 单位固定为 RPM, 不受用户单位调整影响.

0x6081, profile velocity, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6081	0x00	rw	uint32	pulse/s	500000	TPDO+RPDO

Description:

- 用于 profile position mode 模式时的巡航速度.
- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的单位.

0x6083, profile acceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6083	0x00	rw	uint32	pulse/s ²	500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的单位.

0x6084, profile deceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6084	0x00	rw	uint32	pulse/s ²	500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的单位.

0x6085, quickstop deceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6085	0x00	rw	uint32	pulse/s ²	1500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的单位.

0x6086, motion profile type, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6086	0x00	rw	int16	/	0	RPDO

Description:

value	description
0	T 型速度规划
3	S 型速度规划

- 目前只有标准位置模式支持 S 型速度规划.
- S 型规划描述见章节 3.10 S 曲线规划.

0x6087, torque slope, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6087	0x00	rw	uint32	10 ⁻³ RT/s	10000	RPDO

Description:

- 目前只支持 linear ramp, 就没有 0x6088.
- torque slope 理解为 torque 的加速度, 如果要 0.1 秒内将 torque 加到额定值, 那么 torque slope 就是 1000/0.1=10000.

0x6093, position factor, RECORD

max sub index

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6093	0x00	const	uint8	/	2	NO

numerator

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6093	0x01	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).

divisor

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6093	0x02	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).
- 使用 position factor 可以修改位置的单位. 计算方式如下公式,

$$\text{实际用户单位行程} = \text{默认用户单位行程} * \frac{\text{numerator}}{\text{divisor}}$$

- 位置的默认用户单位是脉冲, pulse.
- 简易的配置方法如下:
 - 将 divisor 设置为一圈脉冲数, 也就是 0x2003:01 的值.

- 将 numerator 设置为电机转动一圈对应的用户距离单位数值. 如果出现小数, 那么将 divisor 和 numerator 同比例放大即可.
- 具体配置例子可查看 [3.3.1, 位置的用户单位](#).
- position factor 对以下对象起作用:

index	name	备注
0x3000	absolute value of target position	
0x3010	negative limit switch position	
0x3011	positive limit switch position	
0x6062	position demand value	
0x6064	position actual value	
0x6065	following error window	
0x6067	position window	
0x607A	target position	
0x607C	home offset	
0x607D	software position limit	
0x60B0	position offset	
0x60C1:01	Interpolation data record	
0x60F4	following error actual value	

- 修改 position factor 后, 需要对应修改 0x6065 和 0x6067. 按照实际用户单位调整数值.

0x6095, velocity factor, RECORD

max sub index

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6095	0x00	const	uint8	/	2	NO

numerator

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6095	0x01	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).

divisor

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6095	0x02	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).
- 使用 velocity factor 可以修改速度的单位. 计算方式如下公式

$$\text{实际用户单位速度} = \text{默认用户单位速度} * \frac{\text{numerator}}{\text{divisor}}$$

- 速度的默认用户单位是 pulse/s, 脉冲数每秒.
- 简易的计算方法:
 - 取中间单位为 RPS, 圈每秒.
 - 将 divisor 设置为 1RPS 对应的脉冲数, 刚好 0x2003:01 的值.
 - 将 numerator 设置为 1RPS 对应的用户速度单位数值. 比如实际用户单位是 RPM, 那么 numerator=60 即可.
 - 如果出现小数, 那么将 divisor 和 numerator 同比例放大即可.
- 详细举例可查看 [3.3.2, 速度的用户单位](#):
- velocity factor 对以下对象起作用:

index	name	备注
0x606B	velocity demand value	
0x606C	velocity actual value	
0x606D	velocity window	
0x606F	velocity threshold	
0x607F	max profile velocity	
0x6081	profile velocity	
0x6099	homing speeds	
0x60B1	velocity offset	
0x60FF	target velocity	

- 修改 velocity factor 后, 需要对应修改 0x606D 和 0x606F. 按照实际用户单位调整数值.

0x6097, acceleration factor, RECORD

max sub index

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6097	0x00	const	uint8	/	2	NO

numerator

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6097	0x01	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).

divisor

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6097	0x02	rw	uint32	/	1	NO

Description:

- 当处于“本地受限状态”时, 修改无效, 本地受限状态的定义请查看 [3.2.1.1 本地受限状态](#).
- 使用 acceleration factor 可以修改速度的单位. 计算方式如下公式, 我们举几个案例.

$$\text{实际用户单位加速度} = \text{默认用户单位加速度} * \frac{\text{numerator}}{\text{divisor}}$$
- 加速度的默认用户单位是 pulse/s².
- 简易的计算方法:
 - 取中间单位 RPS/S, 也就是 RPS².
 - 将 divisor 设置为 1 RPS² 对应数值, 也就是 0x2003:01 的值.
 - 将 numerator 设置为 1 RPS² 对应的用户加速度单位数值. 比如期望加速度单位为 RPM/S, 那么设置 numerator=60.
 - 如果出现小数, 那么将 divisor 和 numerator 同比例放大即可.
 - 如果觉得 RPS² 不好理解的, 可以用 6095 速度的值用在 6097 加速度上.
- velocity factor 对以下对象起作用:

index	name	备注
0x6083	profile acceleration	
0x6084	profile deceleration	
0x6085	quick stop deceleration	
0x609A	homing acceleration	
0x60C5	max acceleration	
0x60C6	max deceleration	

- 修改 acceleration factor 后, 需要对应修改 0x60C5 和 0x60C6. 按实际用户单位调整数值.

0x6098, homing method, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6098	0x00	rw	int8	/	35	NO

Description:

- homing method 涉及到的触发量有如下几个:
NLS: negative limit switch, 负向极限开关, 负向限位开关.
PLS: positive limit switch, 正向极限开关, 正向限位开关.
HS: home switch, 原点开关.
NBL: negative block limit, 负向阻挡极限, 负向堵转限位.
PBL: positive block limit, 正向阻挡极限, 正向堵转限位.
- 方法 3~6, 方法 19~22 合适于在整个行程中将 HS 分为两段的情况.
- 方法 7~14, 方法 23~30 合适于在整个行程中将 HS 分为三段的情况.

0x6099, homing speeds, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6099	0x00	ro	uint8	/	2	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x6099	0x01	rw	uint32	/	16666	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x6099	0x02	rw	uint32	/	1666	NO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的实际用户单位.
- subindex=0x01 的名称是 “speed during search for switch”, 本文中简称为 “速度 1”.
- subindex=0x02 的名称是 “speed during search for zero”, 本文中简称为 “速度 2”.

0x609A, homing acceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x609A	0x00	rw	int32	pulse/ s ²	500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的实际用户单位.

0x60A3, profile jerk use, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60A3	0x00	rw	uint8	/	1	NO

Description:

- 只支持 JERK1.

0x60A4, profile jerk, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60A4	0x00	ro	uint8	/	1	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x60A4	0x01	rw	uint32	pulse/s ³	1000000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的实际用户单位.

0x60B0, position offset, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60B0	0x00	rw	int32	pulse	0	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse, 可通过配置 0x6093 position factor 来修改为需要的实际用户单位.
- CSP 模式下使用, 可以理解为位置前馈. 在 CSP 模式下 target Position + position offset 为内部伺服的位置目标.

0x60B1, velocity offset, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60B1	0x00	rw	int32	pulse/s	0	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的实际用户单位.
- CSP 模式下使用, 可以理解为速度前馈.
- 实际 CSP 模式不需要 velocity offset, 因为速度前馈可以从 target position 中提取. 所以实际 0x60B1 velocity offset 无作用.

0x60B2, torque offset, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60B2	0x00	rw	int16	10 ⁻³ RT	0	RPDO

Description:

- 力矩前馈, 目前不使用, 修改无任何影响.

0x60C0, interpolation submode select, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C0	0x00	rw	int16	/	0	NO

Description:

- 只支持线性插值, 尝试修改为其它值会返回 SDO abort:

0x60C1, interpolation data record, ARRAY

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C1	0x00	const	uint8	/	0x01	NO
Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C1	0x01	rw	int32	pulse	0x00	TPDO+RPDO

Description:

- 用于线性插补位置模式的位置.
- 只支持 1 个数据.

0x60C2, interpolation time period, RECORD

max sub index

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C2	0x00	const	uint8	/	2	NO

interpolation time period value

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C2	0x01	rw	uint8	/	1	RPDO

interpolation time index

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C2	0x02	ro	int8	ms	-3	NO

Description:

- 0x60C2-01 和 0x60C2-02 组合成一个周期时间, 比如默认分别为 1 和 -3, 那么就是 1 毫秒.
- 0x60C2:02 默认为 -3, 也就是单位为毫秒, 且不可修改.

0x60C5, max acceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C5	0x00	rw	uint32	pulse/s ²	1500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的实际用户单位.

0x60C6, max deceleration, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60C6	0x00	rw	uint32	pulse/s ²	1500000	RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s², 可通过配置 0x6097 acceleration factor 来修改为需要的单位.

0x60F2, position option code, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60F2	0x00	rw	uint16	/	2	NO

Description:

- PP 模式下, 使用相对位置时, 用到本 object.

value	description
0	位置增量相对于上一次的 target position(0x607A).
1	位置增量相对于 position demand value(0x6062).
2	位置增量相对于 position actual value(0x6064).

0x60F4, following error actual value, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60F4	0x00	ro	uint32	pulse	0	TPDO

Description:

- 计算方式为 position actual value(0x6064) 减去 position demand value(0x6062).
- 单位为用户单位, 默认用户单位为 pulse, 可通过配置 0x6093 position factor 来修改为需要的实际用户单位.

0x60FD, digital inputs, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60FD	0x00	ro	uint32	/	0	TPDO

Description:

各 bit 的定义如下, NLS/PLS/HS 需要 object 2240:01h 里面配置使能后才会生效.

- bit0, NLS 状态, NLS=negative limit switch.
- bit1, PLS 状态, PLS=positive limit switch.
- bit2, HS 状态, HS=home switch.
- bit3~bit15, 未使用.
- bit16, DI1(第 1 个输入) 状态.
- bit17, DI2(第 2 个输入) 状态.
- bit18, DI3(第 3 个输入) 状态.
- bit19~bit31, 未使用.

0x60FE, digital outputs, ARRAY

maxSubindex

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60FE	0x00	const	uint8	/	1	NO

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60FE	0x01	rw	uint32	/	0	TPDO

Description:

各 bit 的定义如下, 01 的特殊功能需要 object 2250:01h 里面配置使能后才会生效.

- bit0, DO1 的特殊功能状态, 特殊功能是 alarm, 发生 fault 时置位.
- bit1~bit15, 未使用.
- bit16, DO1(第 1 个输出) 状态.
- bit17~bit31, 未使用.

0x60FF, target velocity, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x60FF	0x00	rw	int32	pulse/s	0	TPDO+RPDO

Description:

- 单位为用户单位, 默认情况下的单位为 pulse/s, 可通过配置 0x6095 velocity factor 来修改为需要的实际用户单位.

0x6502, supported drive modes, VAR

Index	subIndex	Access	Type	Units	Default	Map PDO
0x6502	0x00	rw	uint32	/	0xC3ED	NO

Description:

- 目前支持的模式如下表格:

	mode	是否支持
bit 0	profile position mode, 标准位置模式	是
bit 1	velocity mode, 速度模式	否
bit 2	profile velocity mode, 标准速度模式	是
bit 3	profile torque mode, 标准力矩模式	是
bit 4	reserved	/
bit 5	homing mode, 回零模式	是
bit 6	interpolated position mode, 插补位置模式	是
bit 7	cyclic synchronous position mode, 循环同步位置模式	是
bit 8	cyclic synchronous velocity mode, 循环同步速度模式	是
bit 9	cyclic synchronous torque mode, 循环同步扭矩模式	是
bit 14	step-dir position mode, 脉冲位置模式	带 3 个输入的硬件支持
bit 15	step-dir velocity mode, 脉冲速度模式	带 3 个输入的硬件支持

2.4 保存对象

2.4.1 communication parameters

Description:

index	对象名
0x1005	COB-ID SYNC message
0x1006	communication cycle period
0x1007	synchronous window length
0x1014	COB-ID EMCY
0x1015	inhibit time EMCY
0x1016	consumer heartbeat time
0x1017	producer heartbeat time
0x1019	synchronous counter overflow value
0x1029	error behavior
0x1200	SDO server parameter
0x1400	RPD01 communication parameter
0x1401	RPD02 communication parameter
0x1402	RPD03 communication parameter
0x1403	RPD04 communication parameter
0x1600	RPD01 mapping parameter

0x1601	RPD02 mapping parameter
0x1602	RPD03 mapping parameter
0x1603	RPD04 mapping parameter
0x1800	TPD01 communication parameter
0x1801	TPD02 communication parameter
0x1802	TPD03 communication parameter
0x1803	TPD04 communication parameter
0x1A00	TPD01 mapping parameter
0x1A01	TPD02 mapping parameter
0x1A02	TPD03 mapping parameter
0x1A03	TPD04 mapping parameter
0x1F80	NMT startup

2.4.2 manufacturer and application parameters

将“save”写入 0x1010:01 或者 0x1010:03 就会使得以下参数被保存；将“load”写入 0x1011:01 或者 0x1011:03 就会使得以下参数恢复位默认状态。

Index	对象名
0x2001	CANOPEN config
0x2003	CPR
0x2005	additionalNMT
0x2010	PID current
0x2012	PID Velocity
0x2013	PID Position
0x2014	brake related
0x2015	servo strategy
0x2016	filter bandwidth
0x2017	IIT
0x2018	bus voltage
0x2019	emergency enable
0x201A	feed forward
0x201B	temperature
0x201C	resistor
0x201D	vbat
0x201E	adaptive notch filter
0x2030	torque window
0x2031	Torque window timeout
0x2032	torque mode additional parameter
0x2033	syncInTime
0x2034	internal target reach window
0x2035	sync delay time
0x203A	block
0x203B	limit switch
0x203C	mode switch amplifier

0x2240	digital inputs control
0x2250	digital outputs control
0x6065	following error window
0x6066	following error timeout
0x6067	position window
0x6068	position window time
0x606D	velocity window
0x606E	velocity window time
0x606F	velocity threshold
0x6070	velocity threshold time
0x6072	max torque
0x607C	home offset
0x607D	software position limit
0x607E	polarity
0x607F	max profile velocity
0x6083	profile acceleration
0x6084	profile deceleration
0x6085	quick stop deceleration
0x6087	torque slope
0x6093	position factor
0x6095	velocity factor
0x6097	acceleration factor
0x6098	homing method
0x6099	homing speeds
0x609A	homing acceleration
0x60A3	profile jerk use
0x60A4	profile jerk
0x60C5	max acceleration
0X60C6	max deceleration

三、功能、流程

3.1 CANopen

CANopen 是一个用于自动化领域的嵌入式系统的通信协议和设备配置文件规范。就 OSI 模型而言，CANopen 实现了包括网络层在内的各层。CANopen 标准由一个寻址方案、几个小型通信协议和一个由设备配置文件定义的应用层组成。通信协议支持网络管理、设备监控和节点之间的通信，包括一个简单的传输层，用于消息分段/解段。

CANopen 的帧结构如下表格，由 11 位的 id，远程传输请求（RTR）位，数据长度(DLC)和 0~8 个字节的数据组成的短包。

	CAN-ID	RTR	DLC	DATA
length	11bits	1bit	4bits	0~8bytes

CANopen 将 11 位的标准 CAN 帧中的 id 分为 4bits 的功能代码和 7bits 的 CANopen 节点 ID(nodeId)。这使得 CANopen 网络中的设备数量限制在 127 个(0 被保留用于广播)。在 CANopen 中，一个 CAN 帧的 11 位 ID 被称为通信对象标识符，或 COB-ID。在传输碰撞的情况下，CAN 总线中使用的总线仲裁允许具有最小 id 的帧首先被传输，并且没有延迟。对时间关键性的功能使用低功能码号，可以确保尽可能低的延迟。CANopen 的功能码如下表格：

communication object	COB-id	relative object	communication model
NMT	0x000	/	master/slave
SYNC	0x080	0x1005, 0x1006, 0x1007	producer/consumer
EMCY	0x080+NodeId	0x1014	producer/consumer
TPDO1	0x180+NodeId	0x1800, 0x1A00	producer/consumer
TPDO2	0x280+NodeId	0x1801, 0x1A01	producer/consumer
TPDO3	0x380+NodeId	0x1802, 0x1A02	producer/consumer
TPDO4	0x480+NodeId	0x1803, 0x1A03	producer/consumer
RPDO1	0x200+NodeId	0x1400, 0x1600	producer/consumer
RPDO2	0x300+NodeId	0x1401, 0x1601	producer/consumer
RPDO3	0x400+NodeId	0x1402, 0x1602	producer/consumer
RPDO4	0x500+NodeId	0x1403, 0x1603	producer/consumer
TSDO	0x600+NodeId	0x1200	client/server
RSDO	0x580+NodeId	0x1200	client/server
NMT (heartbeat)	0x700+NodeId	0x1016, 0x1017	producer/consumer
NMT (node guarding)	0x700+NodeId	/	client/server

3.1.1. NMT

NMT 是网络管理（Network ManagemenT）的缩写。NMT 是一种用于启动网络和监控设备的服务。NMT 报文由 NMT 主机发送，对从机进行启动、监控和重启，在 CANopen 网络中只允许有一个活动的 NMT 主机。

NMT 还可以控制设备的状态，从而决定了该设备能使用网络的功能。CANopen 设备有以下几种状态：初始化、预操作、操作和停止。状态之间的转换是通过发送 NMT 通信对象来实现的。

NMT 的 COBID 是 0x00，意味着 NMT 报文的优先级是最高的。

NMT 报文的数据是 2 个 byte，其中 byte0 是 command code，byte1 是 addressed node，当 byte1=0x00 为群发报文。

NMT 的启动停止指的是网络的启动/停止，不是从站的运动的启动/停止。

NMT 帧数据部分结构如下：

COB-ID	Data byte 0	Data byte1
0x00	command code	addressed node

NMT 的 command code 如下表格：

NMT command code	Description
0x01 (0d001)	节点状态变更为 “operational”
0x02 (0d002)	节点状态变更为 “stopped”
0x80 (0d128)	节点状态变更为 “pre-operational ”
0x81 (0d129)	复位节点 “reset node”
0x82 (0d130)	复位通信 “reset communication”

NMT 的通信机制是 master/slave, 通信是单向的, master 发送了 NMT command 之后, slave 并没有回自身的 NMT state 消息. 如果 master 想知道某节点(node)的 NMT state, 有 2 种方法, 一种是 NMT master 发送 node guarding 消息后等待回应, 另一种是配置心跳机制后, heart beat 返回的数据里面带有 NMT state.

NMT 的 state code 如下表格：

NMT state code	Description
0x00	boot up/initialization
0x04	stopped
0x05	operational
0x7F	pre-operational

SYNC, PDO, SDO 等消息能否正常通信, 和 NMT 状态是相关的, 各 NMT 状态的允许通信对象如下表格：

NMT state code	allowed communication object
boot up/initialization	/
stopped	NMT
pre-operational	NMT, SDO, SYNC, EMCY
operational	NMT, SDO, SYNC, EMCY, PDO

Heartbeat 通信属于 NMT 类.

本产品支持以下这些额外的 NMT 命令

NMT command code	Description
0xF0 (0d240)	nm_reboot, 节点重启, 需要通过 2005:01 使能.
0xF1 (0d241)	nm_quickstop 节点执行 quickstop 命令, 具体同章节 3.2.2 的描述, 需要通过 2005:02 使能.
0xF2 (0d242)	nm_shutdown 节点执行 shutdown 命令, 具体同章节 3.2.2 的描述, 需要通过 2005:03 使能.

3.1.2. Node Guarding

Node guarding 功能并不推荐使用, 节点保护推荐使用 heart beat 机制实现. 不过为了上位机在开机后能够获得节点的 NMT state, 我们实现了用 RTR 来查询节点状态的功能, node guarding 关联的 object 比如 0x100C/0x100D 并不存在.

Node guarding 的帧数据结构如下：

COB-ID	RTR
0x700+nodeid	1

节点在收到 Node guarding 后返回节点状态, 帧结构如下, 需要注意到 data0 里面没有 toggle 这个 bit.

COB-ID	RTR	Data byte0
0x700+nodeid	0	NMT state code

3.1.3. Heartbeat

Heartbeat 是 CANopen 推荐的节点保护方式. heartbeat 是定时主动发送的, 通信模型是 producer/consumer 模型. 同时 heartbeat 一般是双向保护, 比如伺服作为一个节点, 通过配置 0x1016 consumer heartbeat time 来监视其他节点/master 发送的心跳, 超时就报心跳丢失. 也通过配置 0x1017 producer heartbeat time 来定时主动发出心跳包. 伺服作为 consumer, 最多同时监视 4 个节点/master 的心跳.

Heartbeat 的帧数据结构如下:

COB-ID	RTR	Data byte0
0x700+nodeid	0	NMT state code

3.1.4. SYNC

不带数据的 SYNC 报文的结构如下:

COB-ID
0x80

SYNC 报文常用于多节点同步, 比如 TPDO 和 RPDO 的传输类型设定为 SYNC 时, TPDO 或者 RPDO 报文只负责传输数据, 等 SYNC 帧发出时才是报文内容生效时.

SYNC 报文可以携带数据, 数据长度为 1 个 byte, 当 SYNC 报文携带数据时, 其数据是个计数器, 从 2 一次次累加到 0x1019 Synchronous Counter Overflow Value 规定的值后回到 2 开始. 这时候 SYNC 报文的结构如下:

COB-ID	Data byte0
0x80	count

对于各个 TPDO, 可在相应通信参数的子索引 06h 中定义同步计数器的起始值(如在 1800h:06h 中), 从站将从该值开始首次响应同步并发送 PDO. 只有在 0x1019 中设置大于 1 小于 241 的值后才能全局激活此功能.

3.1.5. EMCY

只要控制器中发生不是由 SDO 访问引起的错误, 就会发送 Emergency 类型的消息. 该服务通过 CAN-ID 0x80+节点的 ID 发送.

紧急事件消息的结构如下:

COB-ID	Data byte0~1	Data byte2	Data byte3	Data byte4~7
0x80+nodeid	error code	error register	manufacturer specific error field	manufacturer specific info field

- CANopen 协议规定是 byte3~7 都是 manufacturer specific error field, 实际只使用 byte4~7, byte3 没用到.
- byte2 的 error register 永远和 0x1001 error register 一致.
- emergency error code 的具体定义在第 5 章节有详细描述, 且和 0x603F 内容一致.

3.1.6. PDO

一个只包含过程数据的消息被称为“过程数据对象”(process data object, PDO), PDO 是为需要循环交换的数据准备的. PDO 报文背后的想法是将所有的附加信息(索引、子索引和数据长度)从 CAN 报文中删除, 只用数据来填充 CAN 报文. PDO 的源信息和目标信息被单独存储在所谓的 PDO 映射中.

PDO 只有在 NMT 状态机处于 operational 状态时才能使用. 本伺服产品总共支持 8 个独立的 PDO 映射 (4*TPDO, 4*RPDO), 每个相应的 PDO 消息最多可以有 8 个字节(=64bit)的用户数据. 每个 PDO 都有自己的配置, RPDO 的配置

由 RPDO communication parameter 和 RPDO mapping parameter 决定,TPDO 的配置由 TPDO communication parameter 和 TPDO mapping parameter 决定. 若 TPDO 通信发生错误,就会发送 Emergency 类型的消息. 该服务通过 CAN-ID 0x80+节点的 ID 发送.

PDO 的配置严格遵循以下流程:

- Destroy PDO by setting bit valid to 1b of sub-index 01h of the according PDO communication parameter.
- Disable mapping by setting sub-index 00h to 00h.
- Modify mapping by changing the values of the corresponding sub-indices.
- Enable mapping by setting sub-index 00h to the number of mapped objects.
- Create PDO by setting bit valid to 0b of sub-index 01h of the according PDO communication parameter.

PDO 消息的结构如下:

COB-ID	Data byte
0x180+nodeid	不定长度的数据
0x200+nodeid	不定长度的数据
0x280+nodeid	不定长度的数据
0x300+nodeid	不定长度的数据
0x380+nodeid	不定长度的数据
0x400+nodeid	不定长度的数据
0x480+nodeid	不定长度的数据
0x500+nodeid	不定长度的数据

- TPDO 不支持 RTR 请求.

3.1.7. SDO

SDO(server data object)允许对对象字典进行读或写访问. SDO 通信是 client/server 模型. 拥有对象字典是 server, 所以本伺服驱动是 server, 需要读写数据的 CAN 节点被称为 client. upload 是从 server 读取摸一个对象的值, download 是对 server 字典中个某个对象写入数据.

因为 SDO 类型的 CAN 报文包含大量的元数据, 所以 SDO 报文应该只用于配置控制器. 如果有必要在运行过程中循环交换数据, 建议使用 PDO 报文.

SDO 传输分为三种类型的访问:

- expedited transfer 用于传输最多 4 个字节的 object.
- segment transfer 用于传输任何数量的字节.
- block transfer 也是针对任何数量的字节.

配置伺服控制器只需要使用到 expedited transfer, 我们就描述这一种传输方式. expedited transfer 也有 upload 和 download 之分.

download 也就是读 server 中字典的数据, 往返报文的数据结构如下:

	COB-ID	Databyte0	Databyte1	Databyte2	Databyte3	Databyte4	Databyte5	Databyte6	Databyte7
read	0x600+nodeid	0x40	index		subindex	/	/	/	/
respond	0x580+nodeid	0x43	index		subindex	data			
		0x47	index		subindex	data			/
		0x4B	index		subindex	data		/	/
		0x4F	index		subindex	data	/	/	/
		0x80	index		subindex	abortion code			

upload 也就是写数据到 server 的字典中, 往返报文的数据结构如下:

	COB-ID	Databyte0	Databyte1	Databyte2	Databyte3	Databyte4	Databyte5	Databyte6	Databyte7
write	0x600+nodeid	0x23	index		subindex	data			
		0x27	index		subindex	data			/
		0x2B	index		subindex	data		/	/
		0x2F	index		subindex	data	/	/	/
respond	0x580+nodeid	0x60	index		subindex	/	/	/	/
		0x80	index		subindex	aborting code			

3.2 control and status

从 CANopen 接口通过配置 control word(0x6040)可以去修改伺服状态,从读取 status word(0x6041)可以读取伺服状态.

3.2.1. status word

status word 有 16bit, 如下表格:

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ms		oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rsto

其中 bit8, bit14~15 是 manufacturer specific. bit12~13 是 operation mode specific, 在本文档第四部分控制模式里面描述. bit10 是 target reached, bit11 是 internal limit active, 这 2 个 bit 是伺服运动过程中的一些状态, 和 control word 基本没关系, 这里也不描述, bit7 是 warning. 剩下的只有 bit0~6 以及 bit9. 本章节就关心 status word 的这 8 个 bit.

- bit0, rsto: ready to switch on
- bit1, so: switched on.
- bit2, oe: operation enabled.
- bit3, f: fault.
- bit4, ve: voltage enabled.
- bit5, qs: quick stop.
- bit6, sod: switch on disabled.
- bit7, w: warning, 多圈产品下, 定义为是否发生过内部保持位置的电池断电, 通过回零模式清零. 单圈产品不会置位.
- bit8, ms: 厂家自定义, 实际用于电机温度警告(非报错), 配合 0x201B:07/08 使用.
- bit9, rm: remote.
- bit14, ms: 厂家自定义, 用于负限位标记, 置位表示硬件负限位被触发. 堵转限位不会触发.
- bit15, ms: 厂家自定义, 用于正限位标记, 置位表示硬件正限位被触发. 堵转限位不会触发.

这些 bits 之间可能的组合有如下表格:

status	bit9 rm	bit6 sod	bit5 qs	bit4 ve	bit3 f	bit2 oe	bit1 so	bit0 rsto	说明
STATUS_0x000_NotReadyToSwitchOn	0	0	0	0	0	0	0	0	
STATUS_0x260_SwitchOnDisabled	1	1	1	0	0	0	0	0	
STATUS_0x231_ReadyToSwitichOn	1	0	1	1	0	0	0	1	
STATUS_0x233_SwitchedOn	1	0	1	1	0	0	1	1	
STATUS_0x237_OperationEnabled	1	0	1	1	0	1	1	1	本地受限状态
STATUS_0x217_QuickstopActive	1	0	0	1	0	1	1	1	本地受限状态
STATUS_0x23F_FaultReactionActive	1	0	1	1	1	1	1	1	本地受限状态
STATUS_0x238_Fault	1	0	1	1	1	0	0	0	本地受限状态

关于 status word 的这些状态,有如下的说明:

- rm 是 remote, 只要 CANopen 能开始通信,rm 就是 1.
- 初始状态下 status=0x000,这时候这些 bit 的定义无意义,因为初始化未完成.
- qs 这个 bit 的定义和其它 bit 不一样,qs=0 表示 active,其它 bit 是置 1 表示 active.
- 能发生 fault 的,都是在 operation enabled 基础上的,所以 oe/so/rsto 都是置 1 的.
- 后 4 个状态(operation enabled/quickstop active/fault reaction active/fault)定义为本地受限状态,本地受限状态下部分字典内容不能修改,若通过 SDO 尝试修改会返回 0x08000021(Data cannot be transferred or stored to application because of local control)

1.2.1.1. 本地受限状态

当 status word 为以下状态时,定义为本地受限状态:

- STATUS_0x237_OperationEnabled.
- STATUS_0x217_QuickstopActive.
- STATUS_0x23F_FaultReactionActive.
- STATUS_0x238 Fault.

3.2.2. control word

control word 同样有 16bit,如下表格,同样本章节关心通用的这几个 bit,so/ev/qs/eo/fr.

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ms					r	oms	h	fr	oms			eo	qs	ev	so

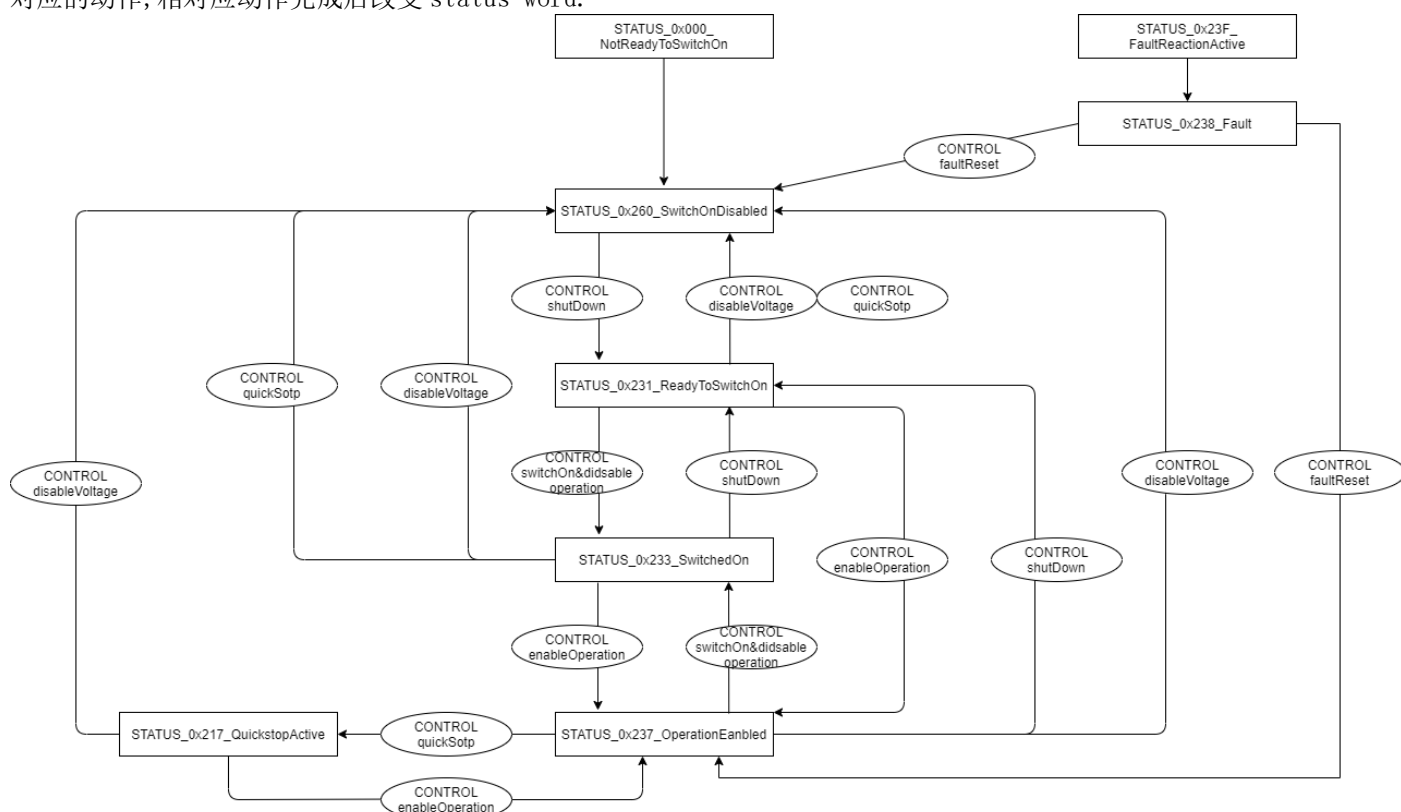
- so: switch on.
- ev: enable voltage.
- qs: quick stop.
- eo: enable operation.
- fr: fault reset.
- h: halt.

这些 bit 组合成如下命令,其中大写的 X 表示 don' t care.

control	bit7 fr	bit3 eo	bit2 qs	bit1 ev	bit0 so	可用的组合
shutDown	0	X	1	1	0	0x06
switchOn	0	0	1	1	1	0x07
enableOperation	0	1	1	1	1	0x0F
disableVoltage	0	X	X	0	X	0x00
quickStop	0	X	0	1	X	0x02
faultReset	1	X	X	X	X	0x80

3.2.2. event and action

伺服作为 control word 的接收方,解析的时候,解析顺序为 fr->ev->qs->so->eo. 伺服解析 control word 后执行相对应的动作,相对应动作完成后改变 status word.



关于 control 和 status 的交互,有如下的说明:

- STATUS_0x237_OperationEnabled 状态到 STATUS_0x231_ReadyToSwitchOn 由字典对象 0x605B shutdown operation code 决定.
- STATUS_0x237_OperationEnabled 状态到 STATUS_0x233_SwitchedOn 由字典对象 0x605C disable operation code 决定.
- 状态跳转到 STATUS_0x23F_FaultReactionActive 由字典对象 0x605E fault reaction option code 决定.
- 在状态 STATUS_0x238_Fault 时,执行命令 fault reset 时,根据 0x605E 的值来确定 reset 后的 status,如果 605E=5or6,那么 fault reset 后跳转到 STATUS_0x237_OperationEnabled,其它的跳转为 STATUS_0x260_SwitchOnDisabled.
- 如果在 STATUS_0x217_QuickstopActive 下发生错误,那么状态变到 STATUS_0x23F_FaultReactionActive,也就是 fault reaction 优先级比 quickstop 高.

3.3 单位, units

在用户单位体系(user-defined units)里,默认的用户单位是这样的,位置是 pulse,速度是 pulse/s,加速度是 pulse/s². 同时呢,也提供了修改选项,0x6093/0x6095/0x6097,通过配置这3个 object,可以将用户单位修改为任何单位.为了区分,我们将修改后的单位体系称为实际用户单位,默认的用户单位称为默认用户单位.

位置的用户单位修改,请详见 [0x6093 position factor](#).

速度的用户单位修改,请详见 [0x6095 velocity factor](#).

加速度的用户单位修改,请详见 [0x6097 acceleration factor](#).

总结下就是默认的用户单位为默认用户单位,修改后的用户单位为实际用户单位,如果不做修改,那么实际用户单位就是默认用户单位. 同时呢,默认用户单位和实际用户单位都是用户单位,不是内部单位.

3.3.1. 位置的用户单位

位置的默认用户单位是脉冲, pulse.

简易的配置方法如下:

- 将 divisor 设置为一圈脉冲数,也就是 0x2003:01 的值.
- 将 numerator 设置为电机转动一圈对应的用户距离单位数值.如果出现小数,那么将 divisor 和 numerator 同比例放大即可.

举例如下:

● 案例 1:

1. 需求:电机后置减速机,减速比为 5,减速机后配 1:1 的同步带轮,同步带轮后配齿轮,齿轮为一圈 32 齿.那么最终期望“齿”为用户单位,如何配置?
2. 计算:假定电机转动一圈,默认用户单位行程是 10000 个脉冲(取决于 0x2003:01),对应减速机转动 0.2 圈,对应同步带轮转动 0.2 圈也就是 6.4 齿,也就是实际用户单位行程是 32/5 齿,也就是:

$$\frac{32}{5} = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{32}{50000};$$

3. 结论:所以配置 numerator=32, divisor=50000,配置后,设置 target position(0x607A)为 1,执行后走动的就是 1 个齿的距离,对应默认用户单位走动 50000/32 个脉冲.

● 案例 2:

1. 需求:电机后置减速机,减速比为 5,减速机后置滚珠丝杆,滚珠丝杆为 10mm 每圈.最终期望按照“mm”为实际用户单位,如何配置?
2. 计算:假定电机转动一圈,默认用户单位行程是 10000 个脉冲(取决于 0x2003:01),对应减速机转动 0.2 圈,对应滚珠丝杆前进 2mm,也就是实际用户单位行程是 2mm:

$$2 = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{2}{10000};$$

3. 结论:所以配置 numerator=2, divisor=10000,配置后,设置 target resolution(0x607A)为 1,执行后走动的就是 1mm 的距离,对应默认用户单位走动 10000/2 个脉冲.

● 案例 3:

1. 需求:位置的实际用户单位为电机的“圈”,该如何配置?
2. 计算:假定默认用户单位下转动一圈的行程是 10000pulse(取决于 0x2003:01),对应实际用户单位行程是 1 圈,也就是:

$$1 = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{1}{10000};$$

3. 结论:所以配置 numerator=1 divisor=10000,配置后,设置 target position(0x607A)为 1,执行后走动的就是转动 1 圈,对应默认用户单位位置走动 10000 个脉冲.

3.3.2. 速度的用户单位

速度的默认用户单位是 pulse/s, 脉冲数每秒.

简易的计算方法:

- 取中间单位为 RPS, 圈每秒.
- 将 divisor 设置为 1RPS 对应的脉冲数, 刚好 0x2003:01 的值.
- 将 numerator 设置为 1RPS 对应的用户速度单位数值. 比如实际用户单位是 RPM, 那么 numerator=60 即可.
- 如果出现小数, 那么将 divisor 和 numerator 同比例放大即可.

举例如下:

● 案例 1:

1. 需求: 电机后置减速机, 减速比为 5, 减速机后配 1:1 的同步带轮, 同步带轮后配齿轮, 齿轮为一圈 32 齿. 那么最终期望“齿每秒”为实际用户单位, 如何配置?
2. 计算: 假定默认用户单位的速度是 10000 pulse/s, 也就是 60rpm(如果 0x2003:01 的值为 10000), 对应减速机是 12rpm, 对应同步带轮 12rpm, 对应齿转动 12*32 齿每分钟, 也就是用户单位速度是 12*32/60 齿每秒, 6.4 齿每秒, 也就是:

$$\frac{32}{5} = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{32}{50000};$$

3. 结论: 所以配置 numerator=32, divisor=50000, 配置后, 设置 profile velocity(0x6081) 为 1, 执行后走动速度是 1 齿每秒, 对应的默认用户单位的速度是 50000/32=1562 pulse/s.

● 案例 2:

2. 需求: 电机后置减速机, 减速比为 5, 减速机后置滚珠丝杆, 滚珠丝杆为 10mm 每圈. 最终期望按照“mm/s”为实际用户单位, 如何配置?
3. 计算: 假定默认用户单位速度是 10000 pulse/s, 也就是 60rpm(如果 0x2003:01 的值为 10000, 对应减速机速度是 12rpm, 对应滚珠丝杆速度是 (10*12/60)mm/s, 也就是实际用户单位速度是 2mm/s:

$$2 = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{2}{10000};$$

4. 结论: 所以配置 numerator=2, divisor=10000, 配置后, 设置 profile velocity(0x6081) 为 1, 执行后走动的就是 1mm/s 的速度, 对应的默认用户单位的速度是 5000pulse/s.

● 案例 3:

4. 需求: 实际用户单位速度为电机的“圈每分钟(rpm)”, 该如何配置?
5. 计算: 假定默认用户单位速度是 10000pulse/s, , 也就是 60rpm(如果 0x2003:01 的值为 10000, 对应实际用户单位速度是 60rpm, 也就是:

$$60 = 10000 * \frac{numerator}{divisor};$$

$$\frac{numerator}{divisor} = \frac{60}{10000};$$

6. 结论: 所以配置 numerator=60 divisor=10000, 配置后, 设置 profile velocity(0x6081) 为 1, 执行后走动的就是 1rpm 的速度, 对应默认用户单位速度是 166pulse/s.

3.3.3. 加速度的用户单位

加速度的默认用户单位是 pulse/s².

简易的计算方法:

- 取中间单位 RPS/S, 也就是 RPS².
- 将 divisor 设置为 1 RPS²对应数值, 也就是 0x2003:01 的值.
- 将 numerator 设置为 1 RPS²对应的用户加速度单位数值. 比如期望加速度单位为 RPM/S, 那么设置 numerator=60.
- 如果出现小数, 那么将 divisor 和 numerator 同比例放大即可.
- 如果觉得 RPS²不好理解的, 可以用 6095 速度的值用在 6097 加速度上.

举例如下:

● 案例 1:

1. 需求: 电机后置减速机, 减速比为 5, 减速机后配 1:1 的同步带轮, 同步带轮后配齿轮, 齿轮为一圈 32 齿. 那么最终期望“齿每平方秒”为实际用户单位, 如何配置?
2. 计算: 假定默认用户单位加速度是 10000 pulse/s², 也就是 60rpm/s (如果 0x2003:01 的值为 10000), 对应减速机加速度是 12rpm/s, 对应同步带轮加速度是 12rpm/s, 对应齿加速度是 6.4 齿每平方秒, 也就是用户单位速度是 32/5 齿每平方秒, 也就是:

$$\frac{32}{5} = 10000 * \frac{\text{numerator}}{\text{divisor}};$$

$$\frac{\text{numerator}}{\text{divisor}} = \frac{32}{50000};$$

3. 结论: 所以配置 numerator=32, divisor=50000, 配置后, 设置 profile acceleration(0x6083) 为 1, 执行后加速度是 1 齿每平方秒, 对应实际用户单位加速度是 50000/32=1562pulse/s².

● 案例 2:

1. 需求: 电机后置减速机, 减速比为 5, 减速机后置滚珠丝杆, 滚珠丝杆为 10mm 每圈. 最终期望按照“毫米每平方秒”为实际用户单位, 如何配置?
2. 计算: 假定默认电机单位加速度是 10000 pulse/s², 也就是 60rpm/s (如果 0x2003:01 的值为 10000), 对应减速机加速度是 12rpm/s, 对应滚珠丝杆速度是 2mm 每平方秒, 也就是:

$$2 = 10000 * \frac{\text{numerator}}{\text{divisor}};$$

$$\frac{\text{numerator}}{\text{divisor}} = \frac{2}{10000};$$

3. 结论: 所以配置 numerator=2, divisor=10000, 配置后, 设置 profile acceleration(0x6083) 为 1, 执行后加速度就是 1mm/s, 对应电机单位速度是 5000 pulse/s².

● 案例 3:

1. 需求: 实际用户单位加速度为“rpm/s”, 该如何配置?
2. 计算: 默认用户单位加速度是 10000 pulse/s²时, 对应加速度是 60rpm/s, 也就是:

$$60 = 10000 * \frac{\text{numerator}}{\text{divisor}};$$

$$\frac{\text{numerator}}{\text{divisor}} = \frac{60}{10000};$$

3. 结论: 所以配置 numerator=60 divisor=10000, 配置后, 设置 profile acceleration(0x6081) 为 1, 执行后加速度就是 1rpm/s 的速度, 对应实际用户单位速度是 166 pulse/s².

3.4 边界, limit

边界, 有软件编辑和硬件边界. 软件边界仅对位置模式有效, 包括标准位置模式(profile position mode), 插补位置模式(interpolated position mode)和周期同步位置模式(cyclic-synchronous position mode). 硬件边界对除脉冲模式(step-dir mode), 回零模式(homing mode)外的其他模式生效. 脉冲模式没有硬件边界是因为数字输入被占用, 回零模式有自己专门的边界处理逻辑.

软件边界由 607D 配置, 607D:01 配置负向最小值, 默认-2147483648 表示负向软件边界不生效. 非-2147483648 值表示负向软件边界生效. 607D:02 配置正向最大值, 默认 2147483647 表示正向软件边界不生效. 非 2147483647 值表示正向软件边界生效. 软件边界的数值的单位是用户单位. 生效后对输入的目标位置进行限制, 使其不能超过软件边界. 若实际

位置超过软件边界, 那么状态字的 bit11(internal limit active) 会置位, 实际位置回到软件边界内部才能清零控制字的 bit11.

硬件边界有传感器的触发实现, I01 的 NLS 功能使能后被触发意味着触发了硬件负向边界, I02 的 PLS 功能使能后被触发意味着触发了硬件正向边界. 硬件边界触发会发送紧急报文 0xFF02 或者 0xFF03. 触发后伺服会急停并保持在停止位置上, 后续只有回到边界内的目标(包括目标位置/目标速度/目标扭矩)会被执行. 同时触发后状态字的 bit14(nls)或 bit15(pls)被置位, 只有实际位置回到硬件边界内才会清零. 负向边界位置会被记录在 0x3100 nls position, 正向边界位置会被记录在 0x3101 pls position.

需要注意边界传感器有电平型和边沿性, 可以通过 0x203B limit switch 配置. 203B:01=0x03 适应电平型传感器输入, 203B:01=0x00 适应边沿型传感器输入.

3.5 IO 接线

大部分型号有 3 个输入, DI1/DI2/DI3. 在脉冲模式下(包括脉冲位置模式和脉冲速度模式), DI1 作为 pulse 信号, DI2 作为 dir 信号, DI3 作为 en 信号使用. 其他模式下, DI1/DI2/DI3 是普通 IO, 当然可以通过置位 2240:01H 开启特殊功能, DI1 的特殊功能是 NLS(negative limit switch), DI2 是 PLS(positive limit switch), DI3 是 HS(home switch).

3 个输入都支持 5V 到 48V 的信号输入, 因为要实现脉冲模式, DI1/DI2 支持高频输入(比如输入 1Mhz 的信号), DI3 只支持低频信号.

DI1/DI2/DI3 不支持反接, 只能保证反接不损坏硬件.
DI1/DI2/DI3 若为常闭, 当开启特殊功能(2240:01H)时, 需要将信号反转, 通过设置 0x2240:02H 实现, 详见该对象描述.

DI1/DI2/DI3 和接近开关的接线方法如下表格:

接近开关类型	接近开关信号	外接电源	伺服信号	是否需要开启反转信号
3 线 NPN, 常开	VCC	VCC	DI1/2/3P	否
	SIG	/	DI1/2/3N	
	GND	GND	/	
3 线 NPN, 常闭	VCC	VCC	DI1/2/3P	是
	SIG	/	DI1/2/3N	
	GND	GND	/	
3 线 PNP, 常开	VCC	VCC	/	否
	SIG	/	DI1/2/3P	
	GND	GND	DI1/2/3N	
3 线 PNP, 常闭	VCC	VCC	/	是
	SIG	/	DI1/2/3P	
	GND	GND	DI1/2/3N	

总结一下就是, 只要在伺服 DI1/2/3 的 P 和 N 之间形成电压差就是触发,

3.6 扭矩模式的零速带载

扭矩模式包括 profile torque mode 和 cyclic-sync torque mode. 出厂设置下, 扭矩模式的最大速度限制是 2032:01H(torque mode max velocity), 配合 2032:02H(torque mode Kp) 来调节最大速度和输出扭矩的平衡实现最大速度的限制. 当实际速度到达最大速度时无实际扭矩输出.

部分早期软件版本, 扭矩模式的最大速度限制是 607FH(max profile velocity), 实际用户使用从扭矩模式切换到其他模式后为及时更改 607FH 而报过速, 所以就将扭矩模式的最大速度限制修改为 2032:01H.

默认配置下, 扭矩模式时设置 2032:01h=0 时无扭矩输出, 当需要扭矩模式下零速有保持扭矩的时候, 需要置位 2015:04H(torque with p), 位置后扭矩模式能够实现零速有输出扭矩, 实际输出扭矩和负载相同(前提是 6071H, target torque 的设置大于负载扭矩). 当 2015:04H 置位后, 2032:02H 不起作用, 通过 2032:01H 来修改实际速度, 加减速是 6083/6084 控制. 且这时候 HOLD(暂停, 6040 的 bit8 置位)和 quickstop(605A=5/6) 都有保持力矩.

3.7 内置的 CAN 总线的 120 欧姆电阻

大部分型号有内置的 120 欧姆电阻可以用, 通过置位 2001:03H 使能内置电阻.

3.8 PID 整定

PID 的整定建议用 flowcan 来时间, flowcan 工具支持 pdo 数据的图形显示, 一目了然.

PID 的整定需要有内而外, 先内环后外环.

3.8.1. 电流环的 PID 整定

一体伺服, 对伺服控制器来说, 其电机参数是已知的, 其出厂的电流环 PID 已经是较优的了, 且电流环和负载无关, 没有更改负载带来的 PID 重新整定需求, 所以建议是不修改电流环的 PID.

3.8.2. 速度环的 PID 整定

当负载较大时, 建议修改速度环的 PID. 心流伺服的速度环的 PID 可以通过单参数实现调整. 单参数为 2012:05H, 速度环的带宽, 也可以理解为速度环的刚性. 通过对该参数增大或者减小来实现速度环的 PID.

可以配置伺服在 pv(profile velocity mode) 模式下, 定时修改目标速度 (60FF, target velocity), 观察实际速度 (606C, velocity actual value) 和需求速度 (606B, velocity demand value) 的重合度, 重合度越高的越好.

3.8.3. 位置环的 PID 整定

位置环是单 P(放大) 控制, 没有 I(积分). 所以对 P 进行放大或者缩小就可以调节位置环的 PID 了.

PID 的整定过程是由内往外, 在整定位置环前先整定速度环.

位置环的 P 的整定建议运行在 pp(profile position mode), 定时修改 607A(target position), 观察 60f4(following error actual value), 60F4 越小越好.

3.9 多圈实际位置

我司有多圈绝对值一体伺服, 也有单圈绝对值一体伺服. 可以通过读取 0x2003:02 来区分, 单圈绝对值的产品, 该值为 1. 多圈绝对值的产品, 该值为多圈的圈数. 比如产品 FCMT24S2P5754, 其 2003:02=262144=2 的 18 次方, 其 0x2003:01=16384=2 的 14 次方. 组合起来的多圈位置范围就是 2 的 32 次方. 和 CANopen 的位置类型数据要求吻合.

单圈绝对值的产品, 在主供电断开后, 只要电机转动的位置不超过正负半圈, 那么重新上电后也能恢复正确的实际位置, 若超过正负半圈, 那么重上电会其实际位置不正确, 其读出来的实际位置和真实的实际位置的差值是单圈脉冲数的整数倍.

多圈绝对值的产品, 在主供电断开后, 一定时间内随意转动电机轴, 重新上电后都能恢复正确的实际位置. 这里说的一定时间和备用电源的供电方法有关.

比如有外接电池作为备用电源的情况, 看电池容量, 主供电断开后多圈位置的维持电流为 8uA. 比如 3600mAH 的电池, 其理论待机时间 51 年.

在无接外部电池的情况下, 仅使用内部的储能电容做备用电源供电, 在储能电容充满电的情况下有 48 小时的供电能力(主电源上电 5 分钟, 储能电容有 8 小时的供电能力), 在该时间内, 主供电断开后, 随意转动电机, 其实际位置都不会丢失.

如果备用电源彻底没电了, 那么伺服重新上电后, 其 6041 的 bit7, warning 位会置位, 对该位清零需要使用任意一种回零方法回零成功即可.

3.10 S 曲线规划

S 曲线指的是速度的 S 曲线, 相对速度的 T 型曲线规划, 对减小震动有好处。

目前 S 曲线规划只能用于标准位置模式(profile position mode), 且有以下限制:

- 只支持初始速度和初始加速度为零的规划, 相当于标准位置模式下的非立即执行, 如果实际给的规划要求是立即执行, 也会被改为非立即执行;
- 只支持对称的规划, 加速度和减速度一致, 所以实际规划用的是 0x6083(profile acceleration). 0x6086(profile deceleration)用于 quickstop、shutdown、halt 等功能.
- profile jerk 只有 1 个, 也就是 0x60A4:01.

四、控制模式

4.1 profile position mode

4.1.1. 说明

标准位置模式(轮廓位置模式)用于点对点运动, 伺服根据输入的 target position, profile velocity, profile acceleration, profile deceleration 来规划运动路径.

- 支持零速规划, 未到达 target position 时使 profile velocity=0 即可实现停车.

4.1.2. 激活

要配置为 profile position mode, 需要将字典对象 0x6060(modes of operation)配置为 0x1.
更新 target position 并不会使得伺服立即执行该 target position, 伺服需要根据 control word 来执行更新 target position. 最常见的 target position 的执行更新, 需要 control word 从 0x0F 变到 0x3F.

4.1.3. 控制字

字典对象 0x6040(control word)中的以下位具有特殊功能:

位	描述	备注
bit 4	0 → 1: 表示预使能一段新的位移指令, 是否使能成功, 由伺服状态决定. 1 → 0: 表示预清零控制字 6041h 的 bit12, 是否成功清零, 由伺服状态决定.	边沿触发, 非电平触发.
bit 5	0: 0x607A 位置目标非立即更新. 1: 0x607A 位置目标立即更新.	
bit 6	0: 0x607A 位置目标是绝对位置指令; 1: 0x607A 位置目标是相对位置指令;	
bit 8	0: 位置指令正常执行. 1: 根据 0x605D 的值来执行暂停;	

4.1.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

bit8 of 6040	bit 10 of 6041	描述	备注
0	0	target position not reached.	
0	1	target position reached.	
1	0	decelerating	
1	1	velocity is 0	

位	描述	备注
bit 11	0: internal software limit not active. 1: internal software limit active.	
bit 12	0: 当前 setpoint 已经执行完毕, 等待新的 setpoint. 1: 当前 setpoint 正在被执行, 同时允许在执行中更改 setpoint.	
bit 13	1: 当跟随误差大于 following error window(0x6065) 设定值且时间超过 following error timeout(0x6066) 设定值时触发.	
bit 14	0: 硬件负向位置极限未触发. 1: 硬件负向位置极限触发.	需要 2240:01 对应的 nls 功能使能才会触发.
bit 15	0: 硬件正向位置极限未触发. 1: 硬件正向位置极限触发.	需要 2240:01 对应的 pls 功能使能.

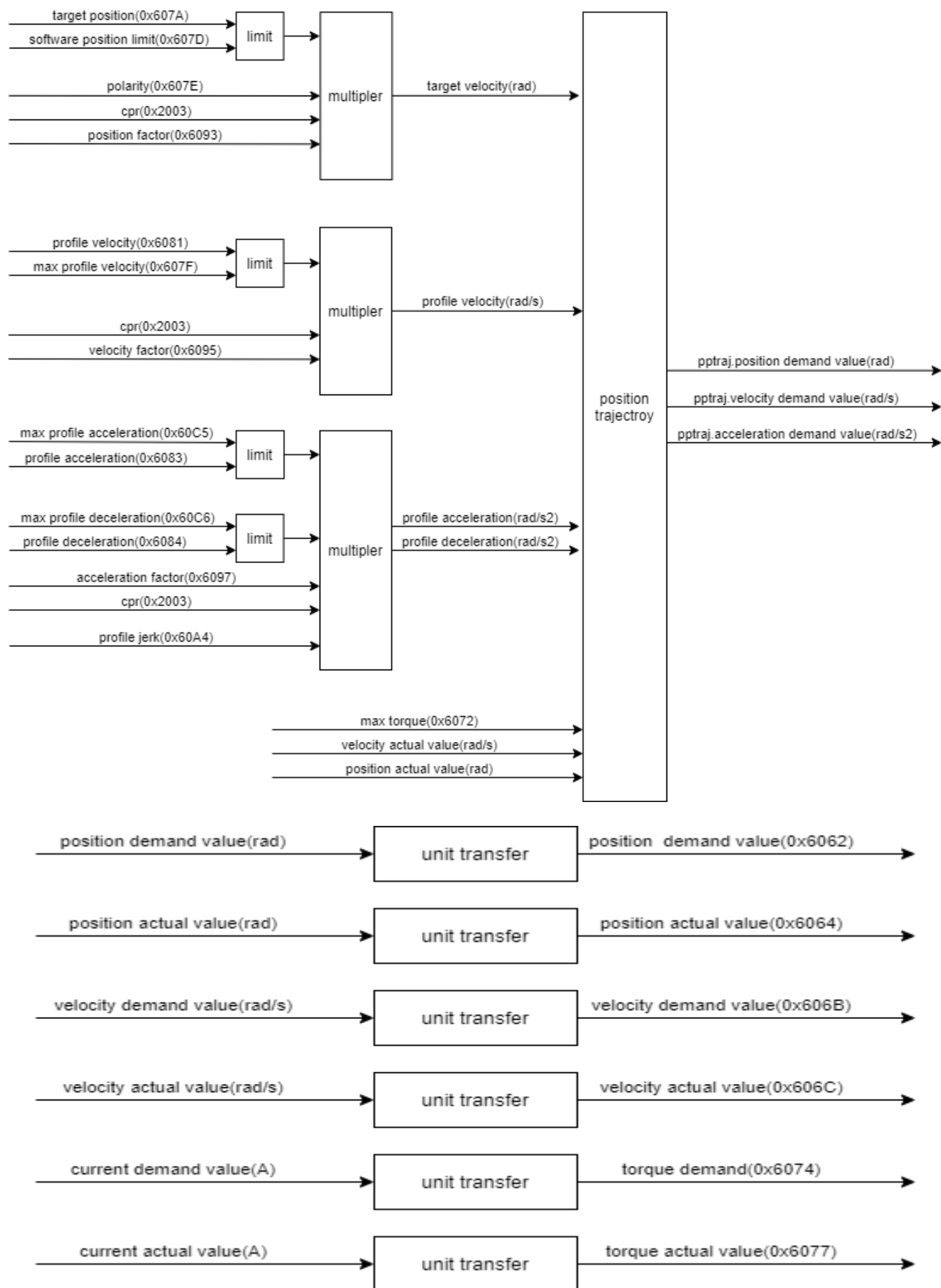
到达目标位置的判定依据是 position actual value(0x6064) 与 target position(0x607A) 的差值的绝对值小于 position window(0x6067) 的持续时间大于 position window time(0x68).

4.1.5. 相关对象

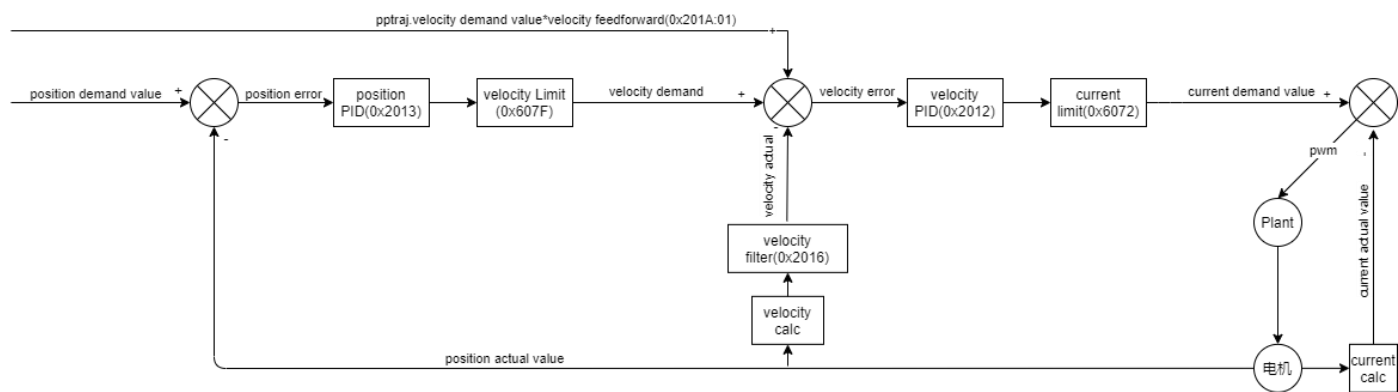
整个 profile position mode 涉及到的对象如下表格

index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2013	position loop PID, 位置环的 PID.
0x2015	servo strategy, 伺服控制策略.
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window time, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6077	torque actual value, 实际输出扭矩, 单位为千分之一的额定扭矩.
0x607A	target position, 目标位置, 用户单位.
0x607D	software position limit, 软件位置限制, 用户单位.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大速度, 用户单位.
0x6081	profile velocity, 规划速度, 用户单位.
0x6083	Profile acceleration, 规划加速度, 用户单位.
0x6084	profile deceleration, 规划减速度, 用户单位.
0x6085	quickstop deceleration, 快停减速度, 用户单位.
0x6086	motion profile type, 规划模式选择, 用户单位.
0x6093	position factor, 位置比例.
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x60A3	profile jerk use, 加加速度用况.
0x60A4	profile jerk, 规划加加速度, 用户单位.
0x60C5	max profile acceleration, 最大加速度, 用于限制 0x6083, 用户单位.
0x60C6	max profile deceleration, 最大减速度, 用于限制 0x6084, 用户单位.
0x60F4	following error actual value, 跟随误差实际值, 用户单位.

4.1.6. 规划框图



4.1.7. 控制框图



4.1.8. 配置实例

下表格对 nodeid=1 的实例进行 PPM 模式配置, 实现绝对位置控制, 立即执行, 运动到位置 10000:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 01 00 00 00	mode=0x01=profile position mode
601	23 7A 60 00 10 27 00 00	target position =0x2710=10000pulse
601	23 81 60 00 20 A1 07 00	profile velocity =0x2710=500000pulse/s=3000RPM
601	2B 40 60 00 06 00 00 00	control word = 0x06
601	2B 40 60 00 0F 00 00 00	control word = 0x0F
601	2B 40 60 00 3F 00 00 00	control word = 0x3F

下表格对 nodeid=1 的实例进行 PPM 模式配置, 实现相对位置控制运动, 立即执行, 前进 10000 个脉冲:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 01 00 00 00	mode=0x01=profile position mode
601	23 7A 60 00 10 27 00 00	target position =0x2710=10000pulse
601	23 81 60 00 20 A1 07 00	profile velocity =0x2710=500000pulse/s=3000RPM
601	2B 40 60 00 06 00 00 00	control word = 0x06
601	2B 40 60 00 0F 00 00 00	control word = 0x0F
601	2B 40 60 00 7F 00 00 00	control word = 0x7F

4.2 profile velocity mode

4.2.1. 说明

标准速度模式(规划速度模式, 轮廓速度模式), 伺服根据用户给定 profile velocity, profile acceleration, profile deceleration 后, 伺服规划电机的速度曲线.

4.2.2. 激活

要配置为 profile velocity mode, 需要将字典对象 0x6060(modes of operation)配置为 0x3. target velocity 的更新会触发伺服立即更新目标, 不需要 control word 的 bit4 的边沿变化. 也就是激活方式上和 PPM 有很大的不同.

4.2.3. 控制字

字典对象 0x6040(control word)中的以下位具有特殊功能:

位	描述	备注
Bit 8	0: 指令正常执行. 1: 根据 0x605D 的值来执行暂停;	

4.2.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

bit8 of 6040	bit 10 of 6041	描述	备注
0	0	target velocity not reached.	
0	1	target velocity reached.	
1	0	decelerating	
1	1	velocity is 0	

位	描述	备注
bit 12	0: velocity not equal 0. 1: velocity equal 0.	
bit 14	0: 硬件负向位置极限未触发. 1: 硬件负向位置极限触发.	需要 2240:01 对应的 nls 功能使能才会触发.
bit 15	0: 硬件正向位置极限未触发. 1: 硬件正向位置极限触发.	需要 2240:01 对应的 pls 功能使能.

速度等于零的判定方法是实际速度小于 velocity threshold(0x606F)的持续时间大于 velocity threshold time(0x6070).

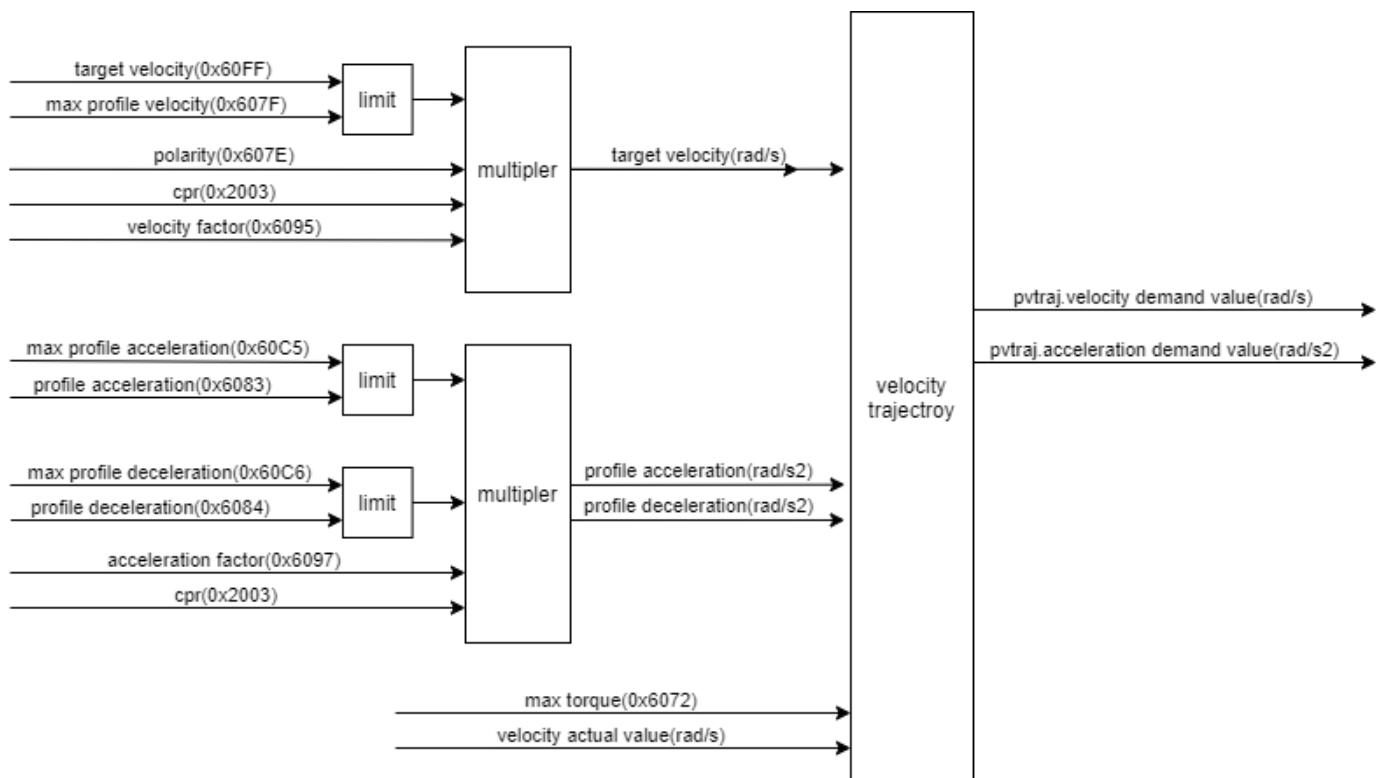
到达目标速度的判定方法是 velocity actual value(0x606C)与 target velocity(0x60FF)的差值的绝对值小于 velocity window(0x606D)的持续时间大于 velocity window time(0x606E).

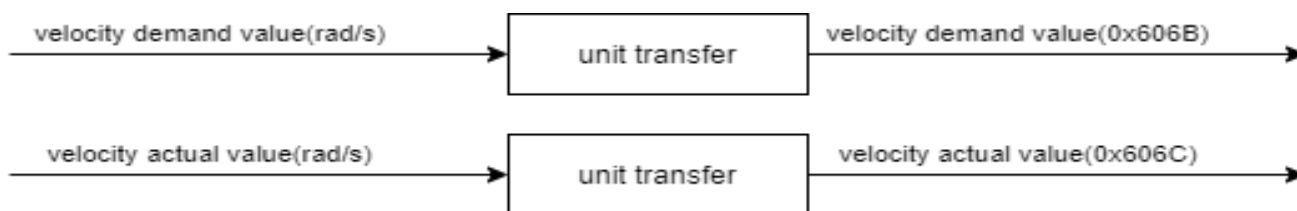
4.2.5. 相关对象

整个 profile velocity mode 涉及到的对象如下表格:

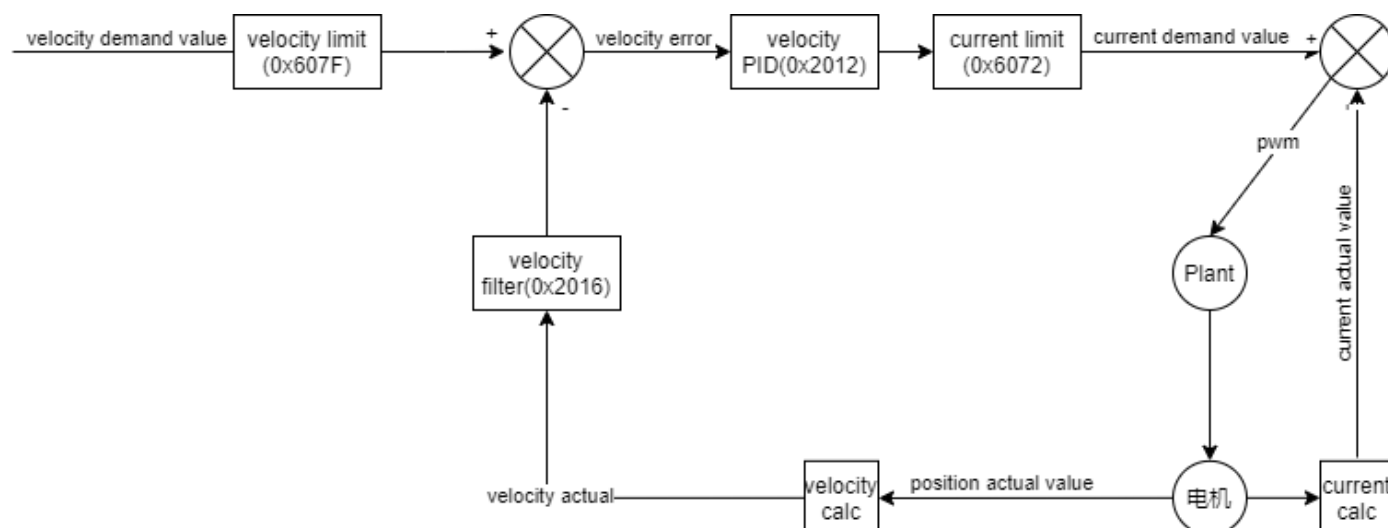
index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的PID.
0x2012	velocity loop PID, 速度环的PID.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x606D	velocity window, 速度窗口, 用户单位.
0x606E	velocity window time, 速度窗口时间.
0x606F	velocity threshold, 速度阈值, 用户单位.
0x6070	velocity threshold time, 速度阈值时间.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大规划速度, 用户单位.
0x6083	profile acceleration, 规划加速度, 用户单位.
0x6084	profile deceleration, 规划减速度, 用户单位.
0x6085	quickstop deceleration, 快停减速度, 用户单位
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x60C5	max profile acceleration, 最大加速度, 用于限制 0x6083, 用户单位.
0x60C6	max profile deceleration, 最大减速度, 用于限制 0x6084, 用户单位
0x60FF	target velocity, 目标速度, 用户单位.

4. 2. 6. 规划框图





4.2.6. 控制框图



4.2.7. 配置实例

下表格对 nodeid=1 的实例进行 PV 模式配置:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 03 00 00 00	mode=0x03=profile velocity mode
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	23 FF 60 00 20 A1 07 00	target velocity =0x7A120=500000pulse/s=3000RPM

4.3 profile torque mode

4.3.1. 说明

标准力矩模式用于力矩控制, 根据用户输入的 target torque 和 torque slope 的值规划需求力矩, 直到实际力矩直到稳定在 target torque.

profile torque mode 下, 电机的转速也受到 torque mode max velocity(0x2032:01H)的限制, 最大转速由 torque mode max velocity 决定.

若需要实现零速有大扭矩输出, 需要置位 2015:05H, torque with p. 详见 2015:04H 对象的描述. 当 2015:04H 置位后, HALT 时有力矩输出的。

4.3.2. 激活

要配置为 profile torque mode, 需要将字典对象 0x6060(modes of operation)配置为 0x4. 控制方式和 PVM 类似, 在伺服已经 enable 的状态下, 只要更新 target torque, 该 target torque 立马被执行。

4.3.3. 控制字

字典对象 0x6040(control word)中的以下位具有特殊功能:

位	描述	备注
Bit 8	0: 位置指令正常执行.	
	1: 根据 0x605D 的值来执行暂停;	

4.3.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

bit8 of 6040	bit 10 of 6041	描述	备注
0	0	target torque not reached.	
0	1	target torque reached.	
1	0	decelerating	
1	1	velocity is 0	

位	描述	备注
bit 14	0: 硬件负向位置极限未触发.	需要 2240:01 对应的 nls 功能使能才会触发.
	1: 硬件负向位置极限触发.	
bit 15	0: 硬件正向位置极限未触发.	需要 2240:01 对应的 pls 功能使能.
	1: 硬件正向位置极限触发.	

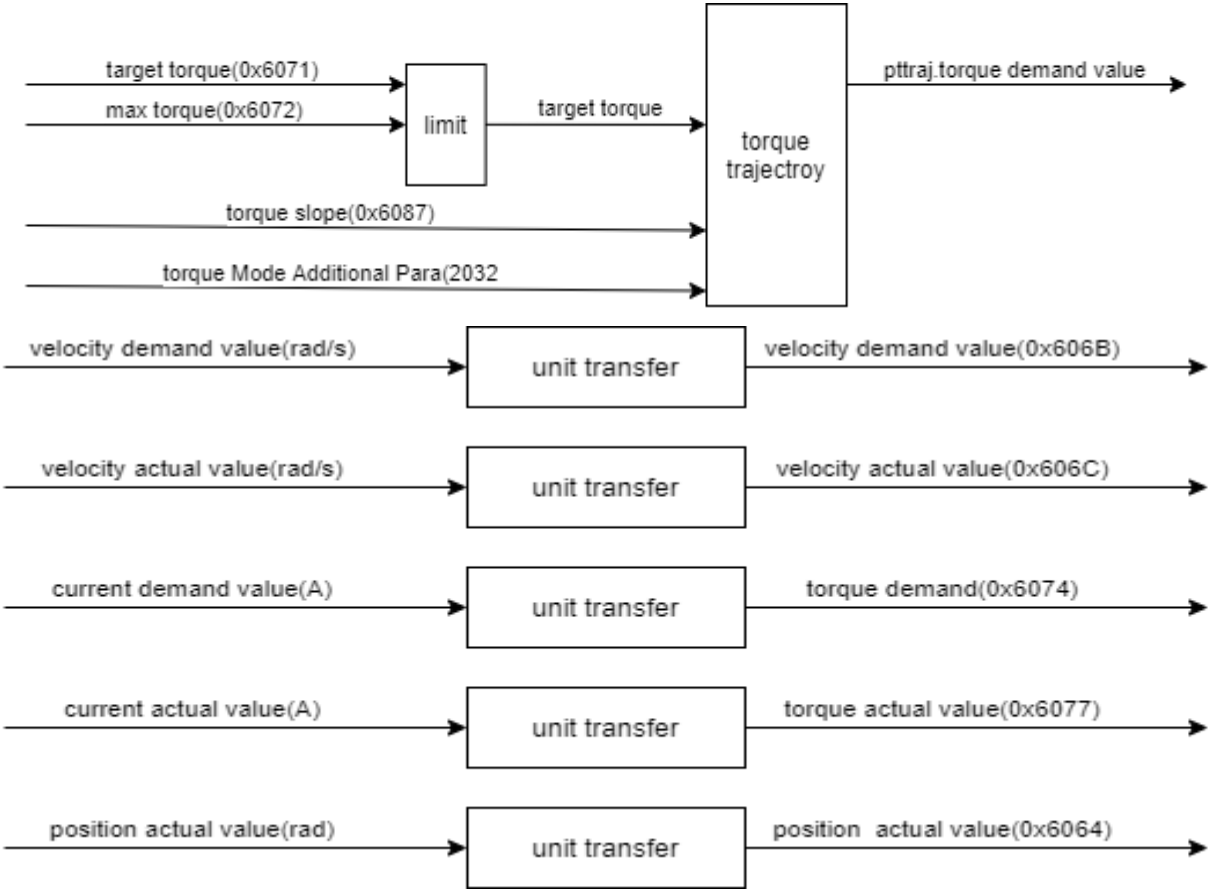
到达目标扭矩的判定方法是 torque actual value(0x6077) 和 target torque(0x6071) 的差值的绝对值小于 torque window(0x2030) 的持续时间大于 torque window time(0x2031).

4.3.5. 相关对象

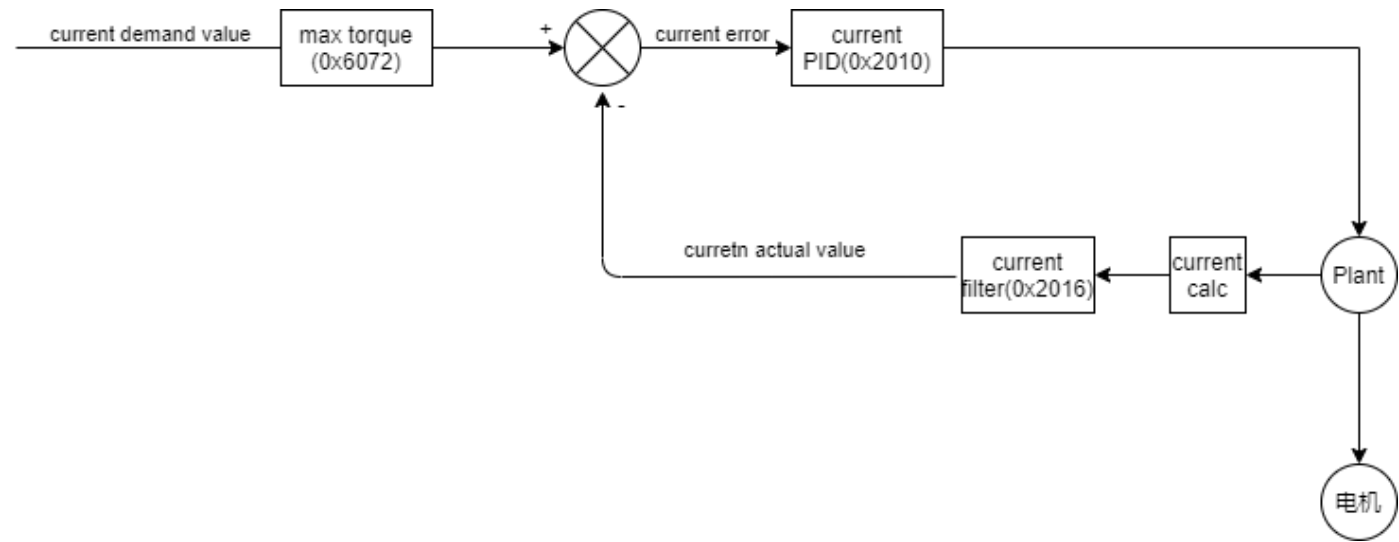
整个 profile torque mode 涉及到的对象如下表格

index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2013	position loop PID, 位置环的 PID.
0x2015	servo strategy, 伺服策略选择.
0x2031	torque window time, 单位是 ms.
0x2032	torque mode additional para, 扭矩模式的额外参数.
0x6071	target torque, 目标扭矩, 单位是千分之一的额定扭矩.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6074	demand torque, 需求扭矩, 单位是千分之一的额定扭矩.
0x6077	torque actual value, 实际扭矩, 单位是千分之一的额定扭矩.
0x6087	torque slope, 扭矩斜坡, 单位是千分之一的额定扭矩每秒.

4.3.6. 规划框图



4.3.7. 控制框图



4.3.8. 配置实例

下表格对 nodeid=1 的实例进行 PT 模式配置：

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 04 00 00 00	mode = 0x04 = profile torque mode
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	2B 71 60 00 64 00 00 00	target torque = 0x64=100, 10%*rated torque.

4.4 homing mode

4.4.1. 说明

回零模式用到 2 个位置, homing position 和 zero position.

回零模式需要外部 switch 或者 block 以及内部的 index 来定位, switch 包括原位开关(home switch), 负向限位开关(negative limit switch)和正向限位开关(positive limit switch). block 包括负向阻挡极限(negative block limit)和正向阻挡极限(positive block limit). index 为位置传感器的 index pulse. 根据这 6 个变量的使用选择, 以及它们的边沿选择衍生出总共 59 种回零方法(homing method).

这些 switch/block/index 触发的位置是原点(homing position), 伺服停下来的位置是零点(zero position). 两者之间的位置差距是 homing offset (0x607C), 如果 homing offset 为零, 那么 homing position 和 zero position 重合. 两者之间的关系为:

$$\text{zero position} = \text{homing position} + \text{homing offset}$$

也就是说如果用户期望 homing 过程中找到 home position 后继续往正方向移动, 那么 home offset 就是正的. 如果希望往负方向移动, home offset 是负的.

4.4.2. 激活

要配置为 homing mode, 需要将字典对象 0x6060(modes of operation)配置为 0x6.

4.4.3. 控制字

字典对象 0x6040(control word)中的以下位具有特殊功能:

位	描述	备注
bit 4	0:回零未使能. 0 → 1:表示使能回零. 1:回零进行中. 1 → 0:表示中断回零.	边沿触发, 非电平触发.
bit 8	0: 指令正常执行. 1: 根据 0x605D 的值来执行暂停;	

4.4.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

bit8 of 6040	bit 10 of 6041	描述	备注
0	0	当前速度不为 0.	
0	1	当前速度为 0.	
1	0	减速过程中	
1	1	当前速度为 0	

位	描述	备注
bit 12	0: 未取得原点. 1: 原点已取得.	
bit 13	0: 没有错误发生. 1: 有错误发生.	

回零成功结束的状态时 bit13=0,bit12=1,bit10=1.

4.4.5. 回零方法

- 回零方法由 0x6098(home method)决定：
- 下面描述的“速度 1”指的是 0x6099:01 的“speed during search for switch”.
 - 下面描述的“速度 2”指的是 0x6099:02 的“speed during search for zero”.
 - 方法 3~6, 方法 19~22 合适于在整个行程中将 HS 分为两段的情况.
 - 方法 7~14, 方法 23~30 合适于在整个行程中将 HS 分为三段的情况.
- 不带任何物理传感器的支持 7 种回零方法.
 只带 HS 的支持 31 种回零方法.
 带 NLS/PLS/HS 的支持 59 种回零方法.

4.4.5.1 回零方法表格示意

home method 的值有如下表格这么写选择, 且对应运动过程也如下表格所述, 另外部分方法的图形示意在表格后面.

value	description
-30	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 30.
-29	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 29.
-28	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 28.
-27	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 27.
-26	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 26.
-25	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 25.
-24	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 24.
-23	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 23.
-22	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 22.
-21	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 21.
-20	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 20.
-19	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 19.
-18	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 18.
-17	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 17.
-14	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错其余同方法 14.
-13	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错其余同方法 13.
-12	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错其余同方法 12.
-11	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错其余同方法 11.
-10	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 10.
-9	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 9.
-8	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 8.
-7	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 7.
-6	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 6.
-5	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 5.
-4	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 4.
-3	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL/NBL 状态变化不报错, 其余同方法 3.
-2	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 NBL 状态变化不报错, 其余同方法 2.
-1	将 NBL 替代 NLS, PBL 替代 PLS, 碰到 PBL 状态变化不报错, 其余同方法 1.
1	起步: NLS=OFF 则以速度 1 朝负向运动. NLS=ON 则以速度 2 朝正向运动. 过程: 负向运动时遇到 NLS 的 OFF→ON 时转向并以速度 2 运动. 原点: 正向运动时遇到 NLS 的 ON→OFF 后继续正向运动到第一个 Z 脉冲位置作为原点. 报错: 过程中碰到 PLS 的状态变化时停止并报错.
2	起步: PLS=OFF 则以速度 1 朝正向运动. PLS=ON 则以速度 2 朝负向运动. 过程: 正向运动时遇到 PLS 的 OFF→ON 时转向并以速度 2 运动. 原点: 负向运动是遇到 PLS 的 ON→OFF 后继续负向运动到第一个 Z 脉冲位置作为原点. 报错: 过程中碰到 NLS 的状态变化时停止并报错.

3	起步： HS=OFF 则速度 1 朝正向运动。 HS=ON 则速度 2 朝负向运动。 过程： 正向运动时遇到 HS 的 OFF→ON 时转向并切换为速度 2 运动。 原点： 负向运动时遇到 HS 的 ON→OFF 后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
4	起步： HS=OFF 则以速度 2 朝正向运动。 HS=ON 则以速度 1 朝负向运动。 过程： 负向运动时遇到 HS 的 ON→OFF 时转向并以速度 2 运动 原点： 朝正向运动时遇到 HS 的 OFF→ON 后继续正向运动找第一个 Z 脉冲位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
5	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时碰到 HS 的 OFF→ON 时转向并以速度 2 运动。 原点： 朝正向运动时遇到 HS 的 ON→OFF 后继续正向运动找第一个 Z 脉冲位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
6	起步： HS=OFF 则以速度 2 朝负向运动。 HS=ON 则以速度 1 朝正向运动 过程： 正向运动时遇到 HS 的 ON→OFF 时转向并以速度 2 运动。 原点： 朝负向运动时遇到 HS 的 OFF→ON 后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
7	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝负向运动。 原点： 负向运动时遇到 HS 的 ON→OFF 时以速度 2 运动，然后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。

8	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 负向运动时遇到 HS 的 ON→OFF 时转向。 原点： 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动，然后继续正向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
9	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 正向运动时遇到 HS 的 ON→OFF 时转向 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 原点： 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动，然后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
10	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝正向运动。 原点： 正向运动时遇到 HS 的 ON→OFF 时以速度 2 运动，然后继续正向运动找第一个 Z 脉冲位作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
11	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝正向运动。 原点： 正向运动时遇到 HS 的 ON→OFF 时以速度 2 运动，然后继续正向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。

12	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 正向运动时遇到 HS 的 ON→OFF 时转向。 原点： 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动，然后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
13	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 负向运动时遇到 HS 的 ON→OFF 时转向。 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 原点： 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动，然后继续正向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
14	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝负向运动。 原点： 负向运动时遇到 HS 的 ON→OFF 时以速度 2 运动，然后继续负向运动找第一个 Z 脉冲位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
17	起步： NLS 无效则以速度 1 朝负向运动。 NLS 有效则以速度 2 朝正向运动。 原点： 负向运动时遇到 NLS 的 OFF→ON 时换速度 2 反向运动。 正向运动时遇到 NLS 的 ON→OFF 时刻的位置为原点。 报错： 过程中碰到 PLS 的状态变化时停止并报错。
18	起步： PLS 无效则以速度 1 朝正向运动。 PLS 有效则以速度 2 朝负向运动。 原点： 正向运动时遇到 PLS 的 OFF→ON 时换速度 2 反向运动。 负向运动时遇到 PLS 的 ON→OFF 时刻的位置为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。

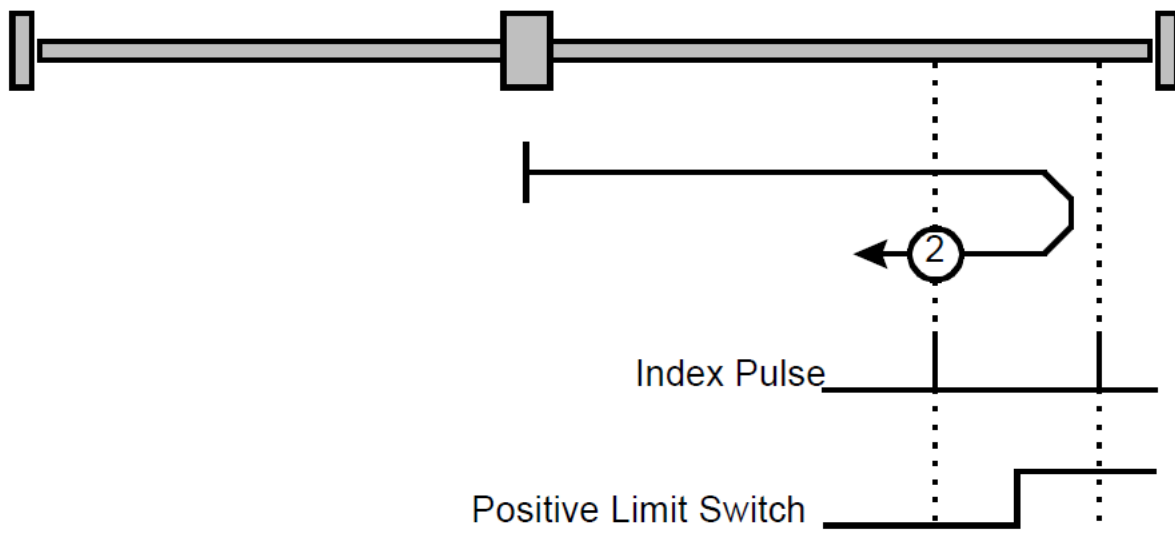
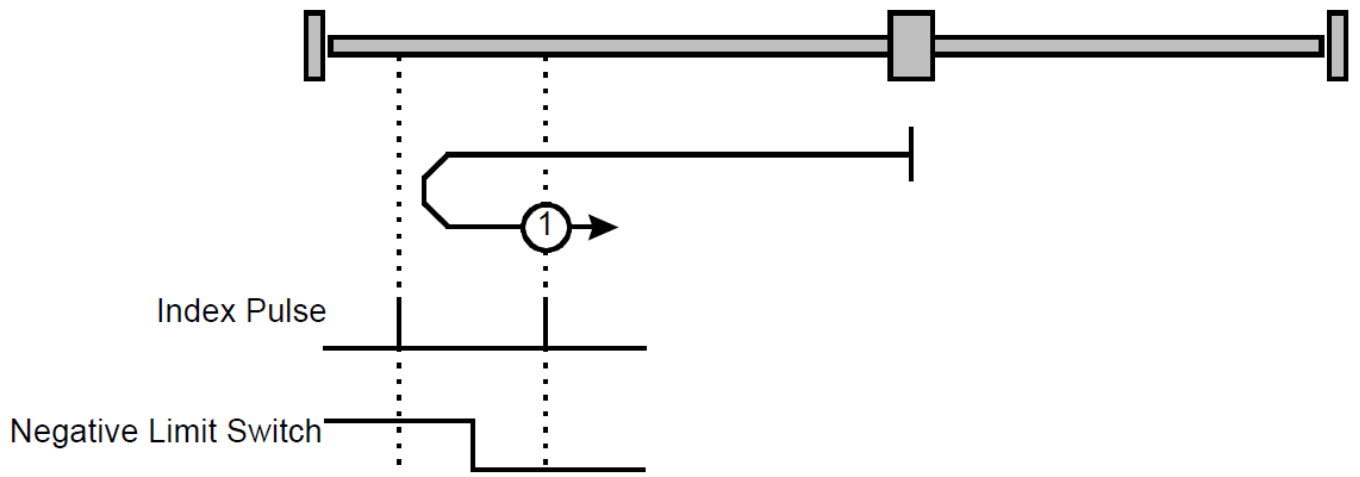
19	起步： HS=OFF 则速度 1 朝正向运动。 HS=ON 则速度 2 朝负向运动。 过程： 正向运动时遇到 HS 的 OFF→ON 时转向并切换为速度 2 运动。 原点： 负向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
20	起步： HS=OFF 则以速度 2 朝正向运动。 HS=ON 则以速度 1 朝负向运动。 过程： 负向运动时遇到 HS 的 ON→OFF 时转向并以速度 2 运动 原点： 朝正向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
21	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时碰到 HS 的 OFF→ON 时转向并以速度 2 运动。 原点： 朝正向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
22	起步： HS=OFF 则以速度 2 朝负向运动。 HS=ON 则以速度 1 朝正向运动 过程： 正向运动时遇到 HS 的 ON→OFF 时转向并以速度 2 运动。 原点： 朝负向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 过程中碰到 NLS 的状态变化时停止并报错。 过程中碰到 PLS 的状态变化时停止并报错。
23	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝负向运动。 原点： 负向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。

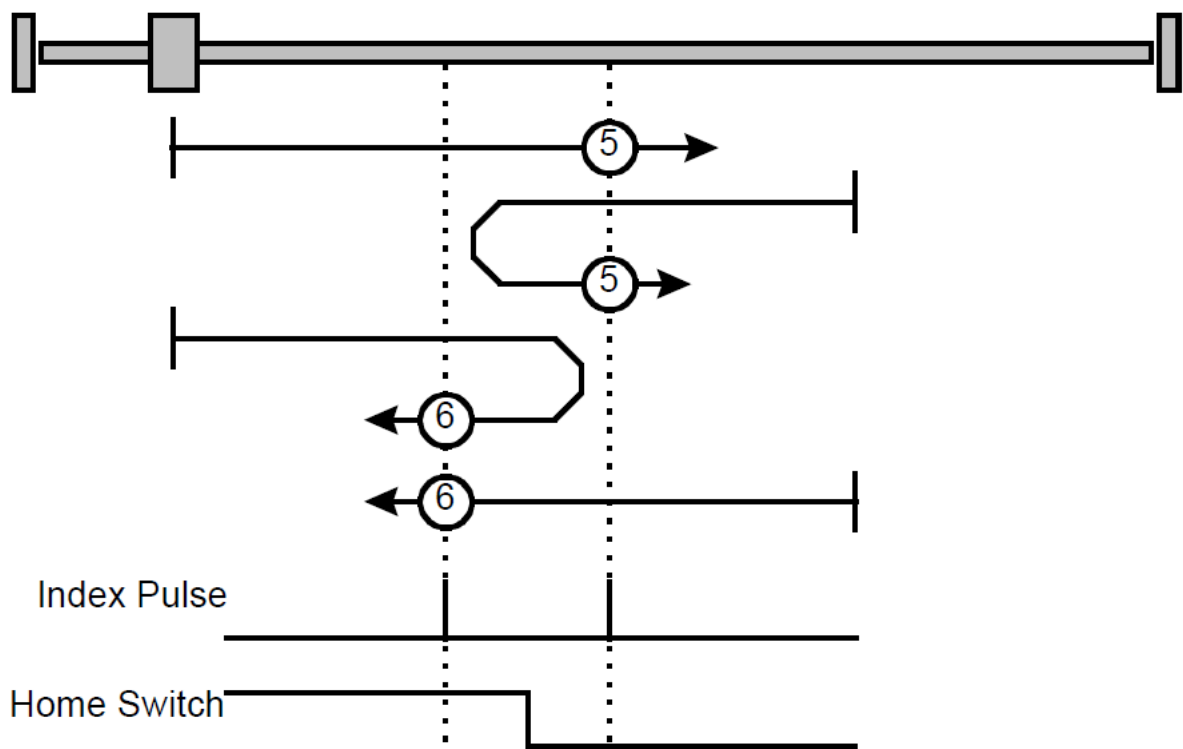
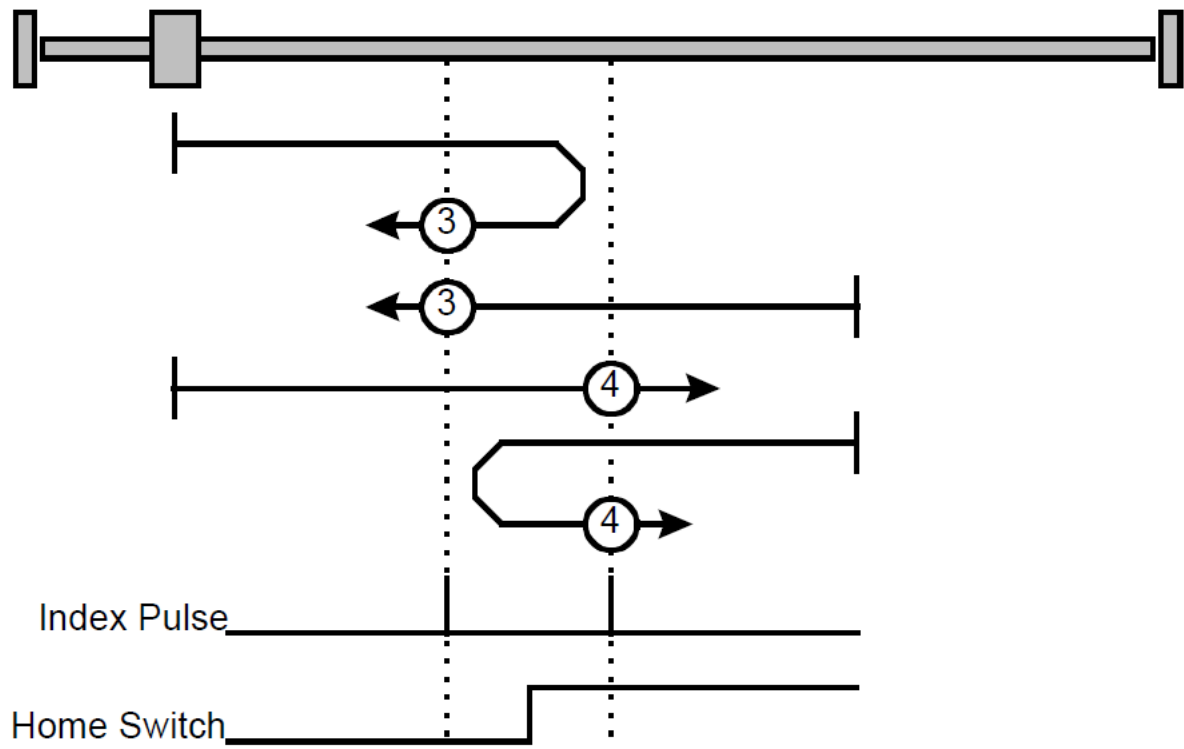
24	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 负向运动时遇到 HS 的 ON→OFF 时转向。 原点： 正向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
25	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 正向运动时遇到 HS 的 ON→OFF 时转向。 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 原点： 负向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
26	起步： HS=OFF 则以速度 1 朝正向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 正向运动时遇到 PLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝正向运动。 原点： 正向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 启动时 PLS 为 ON 则停止并报警。 过程中碰到 NLS 的状态变化就报警。
27	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝正向运动。 原点： 正向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。

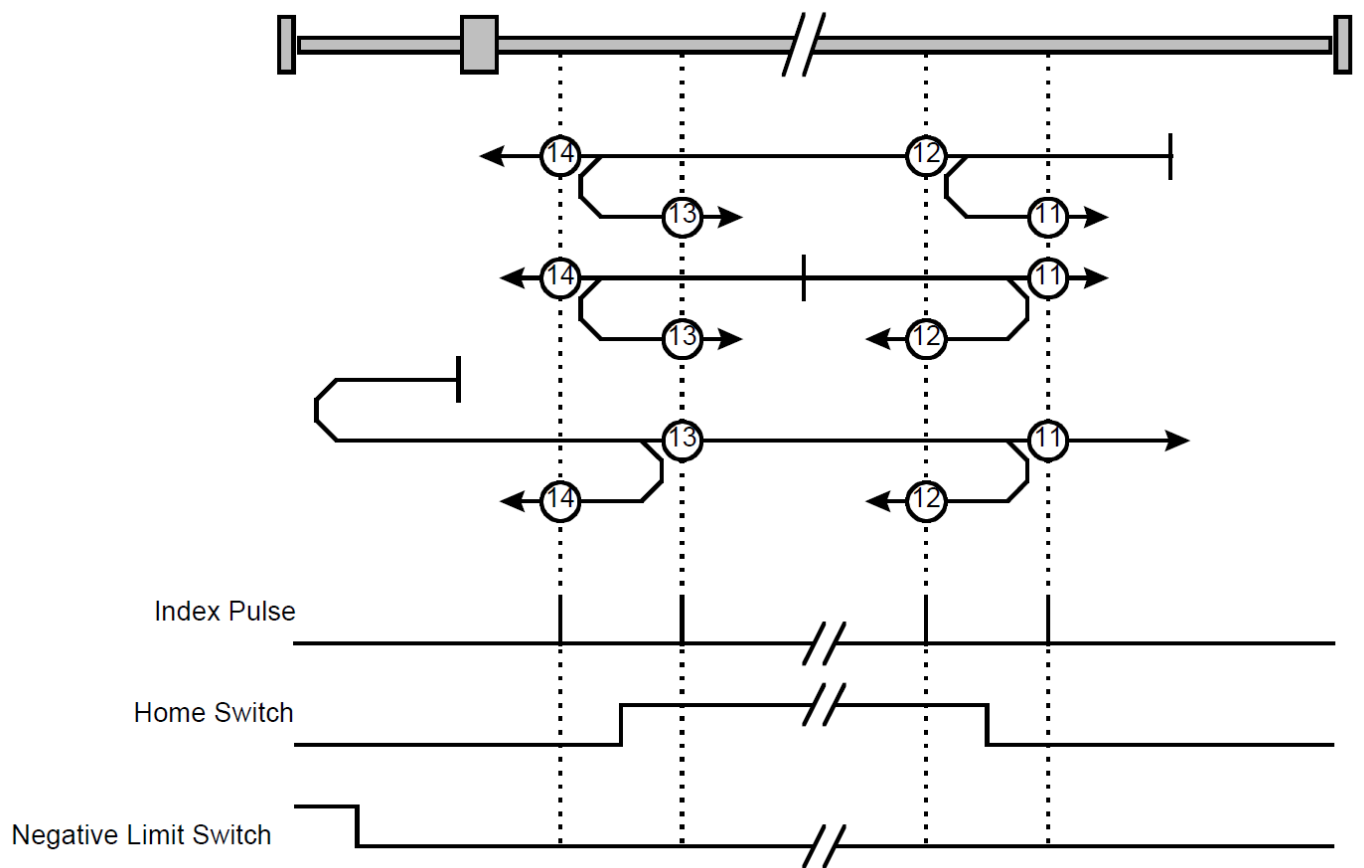
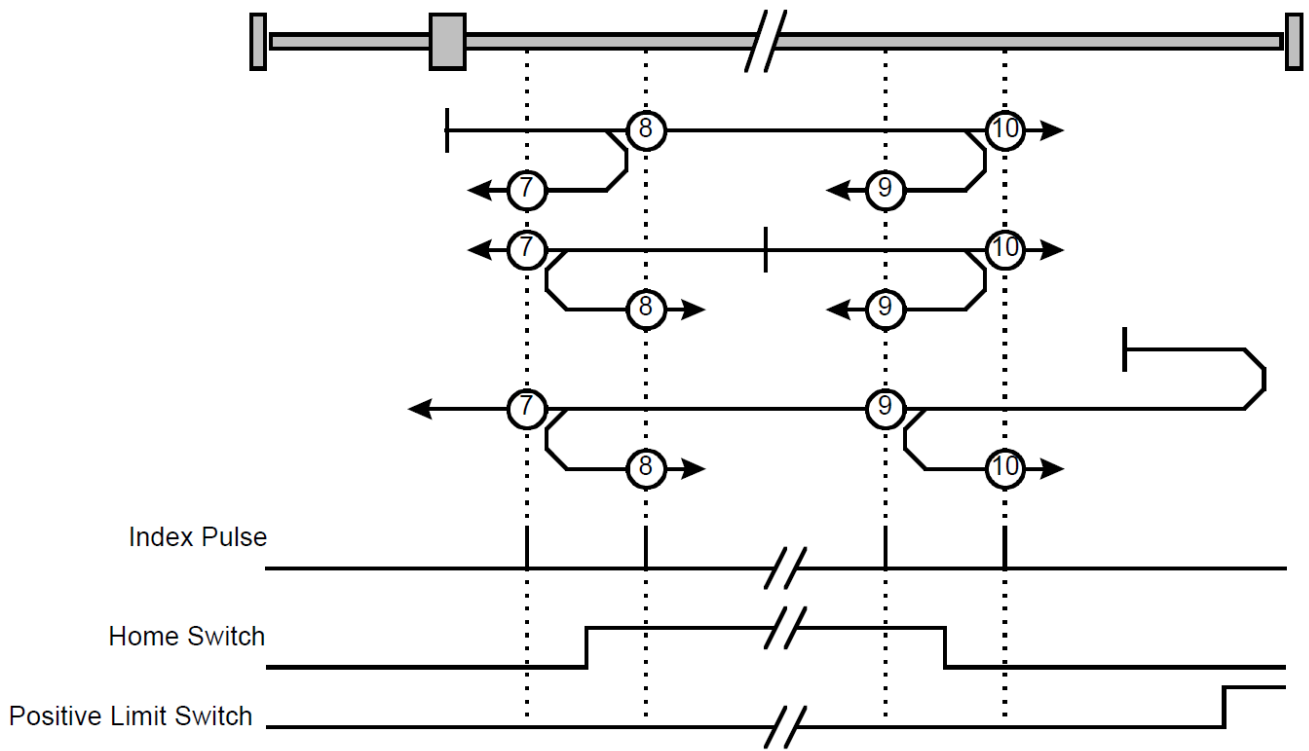
28	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝正向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 正向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 正向运动时遇到 HS 的 ON→OFF 时转向。 原点： 负向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
29	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 负向运动时遇到 HS 的 ON→OFF 时转向。 负向运动时遇到 HS 的 OFF→ON 时以速度 2 运动。 原点： 正向运动时遇到 HS 的 OFF→ON 时刻位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
30	起步： HS=OFF 则以速度 1 朝负向运动。 HS=ON 则以速度 2 朝负向运动。 过程： 负向运动时遇到 NLS 的 OFF→ON 时转向。 遇到 HS 的 OFF→ON 时以速度 2 朝负向运动。 原点： 负向运动时遇到 HS 的 ON→OFF 时刻位置作为原点。 报错： 启动时 NLS 为 ON 则停止并报警。 过程中碰到 PLS 的状态变化就报警。
33	起步时朝负向运动, 碰到第一个 Z 信号为原点。
34	起步时朝正向运动, 碰到第一个 Z 信号为原点。
35	以当前位置为原点。

4.4.5.2 回零方法图形示意

下面图形示意中, 往左定义为负方向, 往右定义为正方向,







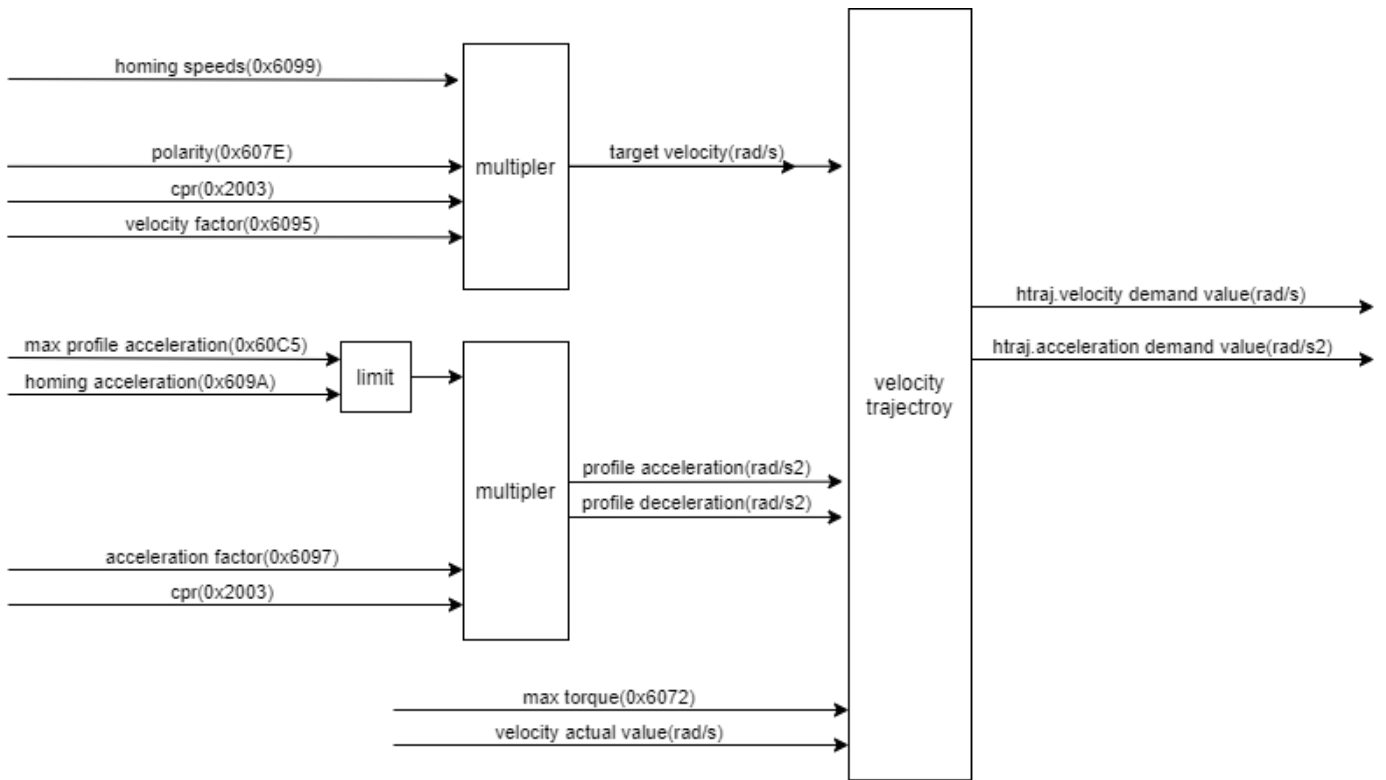
4.4.6. 相关对象

整个 homing mode 涉及到的对象如下表格

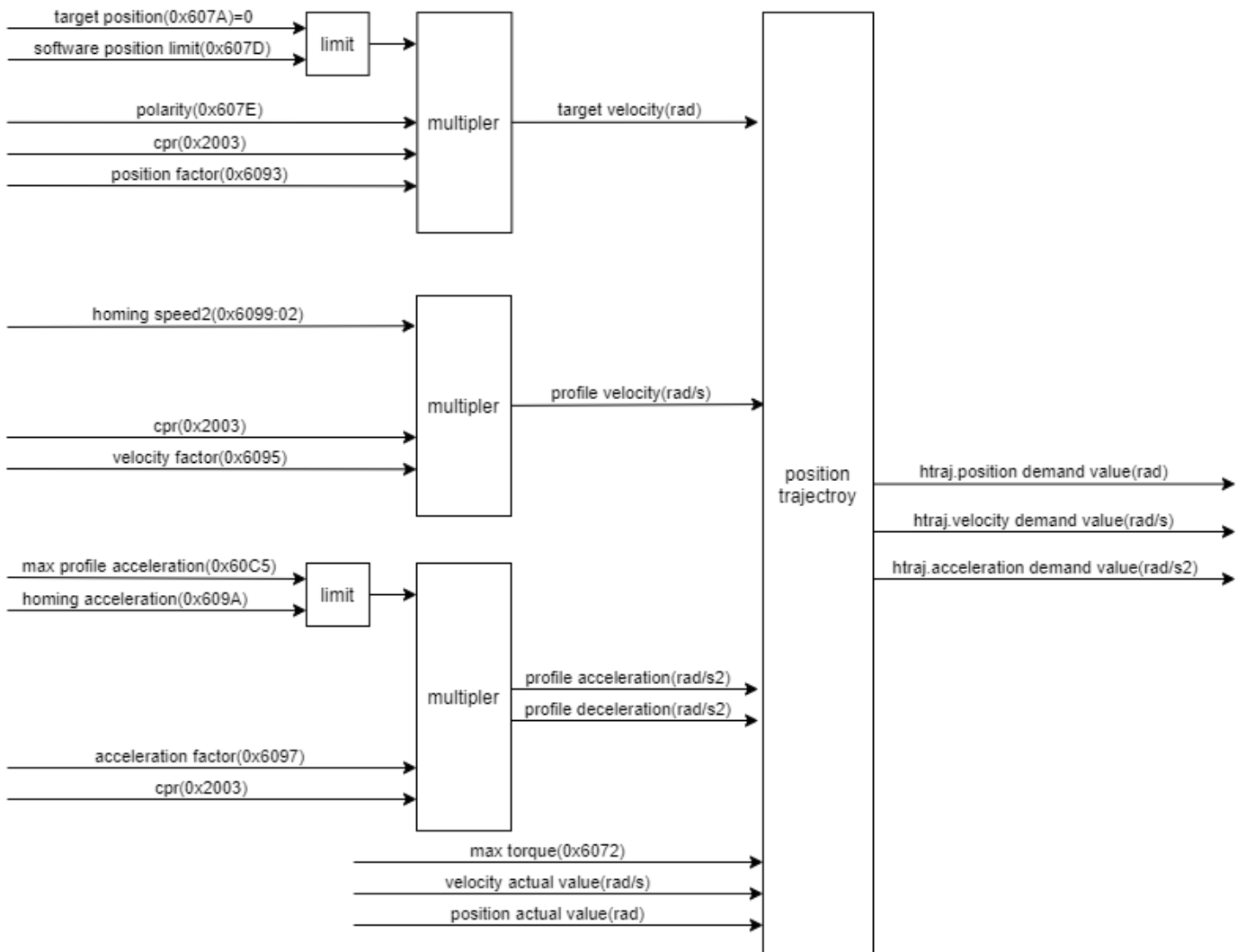
index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2013	position loop PID, 位置环的 PID.
0x203A	homing block, 回零堵转设置
0x2240	digital inputs control, 数字输入控制
0x2250	digital outputs control, 数字输出控制
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window time, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607C	homing offset, 原点的位置偏移, 用户单位.
0x607D	software position limit, 软件位置限制, 用户单位.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大速度, 用户单位.
0x6081	profile velocity, 规划速度, 用户单位.
0x6093	position factor, 位置比例.
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x6098	homing method, 回零方法.
0x6099	homing speeds, 回零速度.
0x609A	homing acceleration, 回零加速度.
0x60C5	max profile acceleration, 最大加速度, 用于限制 0x6083, 用户单位.
0x60C6	max profile deceleration, 最大减速度, 用于限制 0x6084, 用户单位.
0x60F4	following error actual value, 跟随误差实际值, 用户单位.
0x60FD	digital inputs, 数字输入
0x60FE	digital outputs, 数字输出

4.4.7. 规划框图

回零模式的规划有两种,在 home position 找到之前是速度规划,和 profile velocity 类似,如下图

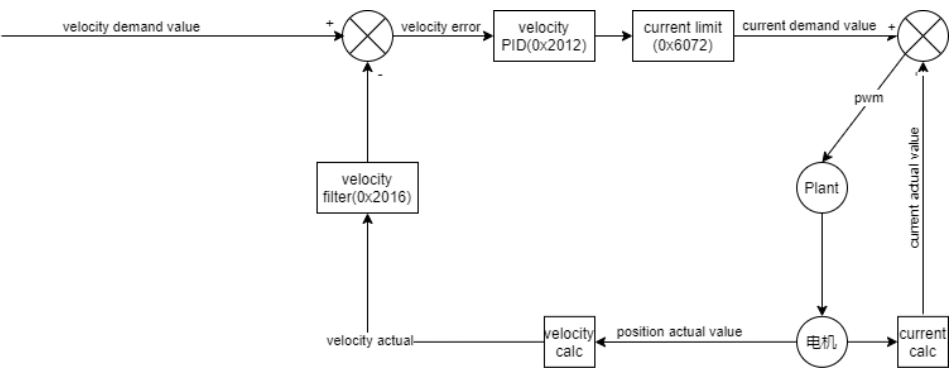


在 home position 找到后,前往 zero position 的过程中是位置规划,类似 profile position mode 的规划.

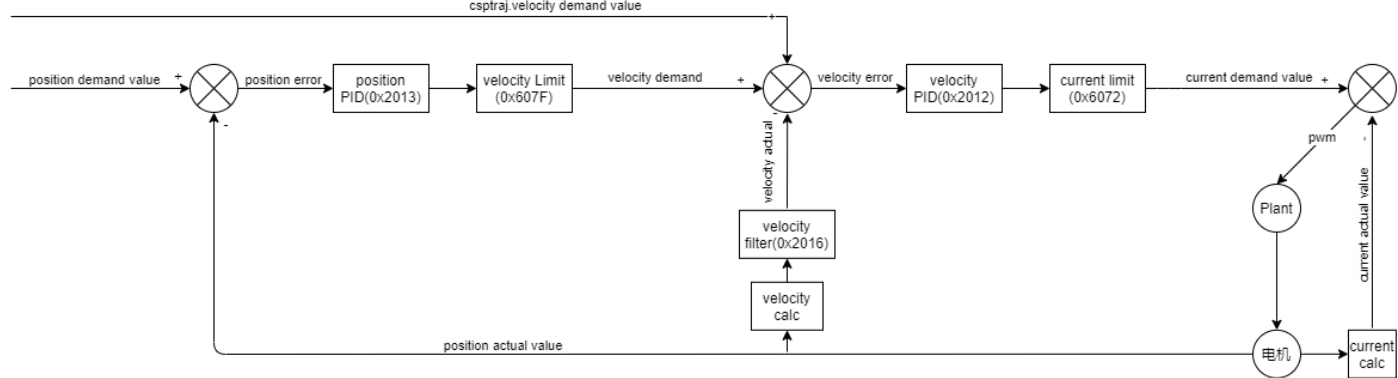


4.4.8. 控制框图

控制框图也一样,寻找 home position 时是速度控制,控制框图如下:



找到 home position 后,往 zero position 的过程中是位置控制,控制框图如下:



4.4.9. 配置实例

下表格对 nodeid=1 的实例进行 homing 模式配置.

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 98 60 00 08 00 00 00	homing method =8
601	2F 60 60 00 06 00 00 00	mode=0x06=homing mode
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	2B 40 60 00 1F 00 00 00	set control word = start homing

4.5 interpolated position mode

4.5.1. 说明

插补位置模式也是位置控制,支持同步更新指令,也支持异步更新指令。其位置指令经过上位控制器处理,周期性地发送给伺服驱动器,不是像轮廓位置控制模式一次给定最终的目标位置。此模式能够实现单轴或者多轴的

同步运动，适用于对同步要求较高的场合。不同于标准位置模式的地方就是，标准位置模式会依据收到的 target position 做出三段规划(加速，匀速，减速)，到达目标位置的速度是零. 而插补位置模式的执行的是单段规划。

插补位置模式只支持绝对位置，绝对位置的更新是通过更新 0x60C1(interpolated data record)来实现，不再是 0x607A(target position)。

如果想取得多轴的高同步精度，建议使用同步更新，也就是 SYNC 帧来同步位置，这样的控制方式可以获得较小的误差。

插补模式有位置前瞻能力，正常时主站是依照 60C2:01 的时间规定发送新的目标以及同步帧，实际情况有可能主站会有下列情形：

- 漏发新的目标位置。
- 漏发同步帧。
- 漏发目标位置帧和同步帧。
- 错发目标位置。

插补模式带位置前瞻，只要上面集中情形不是连续出现，那么从站能够自动修正，不影响控制效果。

插补模式的执行是延迟一次目标的。假定目标位置依次是 P0/P1/P2/P3/P4...，那么：

- 收到 P0 位置时不动作。
- 收到 P1 位置时，执行 P0 位置。
- 收到 P2 位置时，P0 位置执行完成，开始执行 P1 位置。
- 收到 P3 位置时，P1 位置执行完成，开始执行 P2 位置。
- 依次类推

4.5.2. 激活

要配置为 interpolated position mode，需要将字典对象 0x6060(modes of operation)配置为 0x7。

4.5.3. 控制字

字典对象 0x6040(control word)中的以下位具有特殊功能：

位	描述	备注
bit 4	0:表示插值运动控制未激活。 1:表示激活插值运动控制。	边沿触发，非电平触发。

4.5.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能：

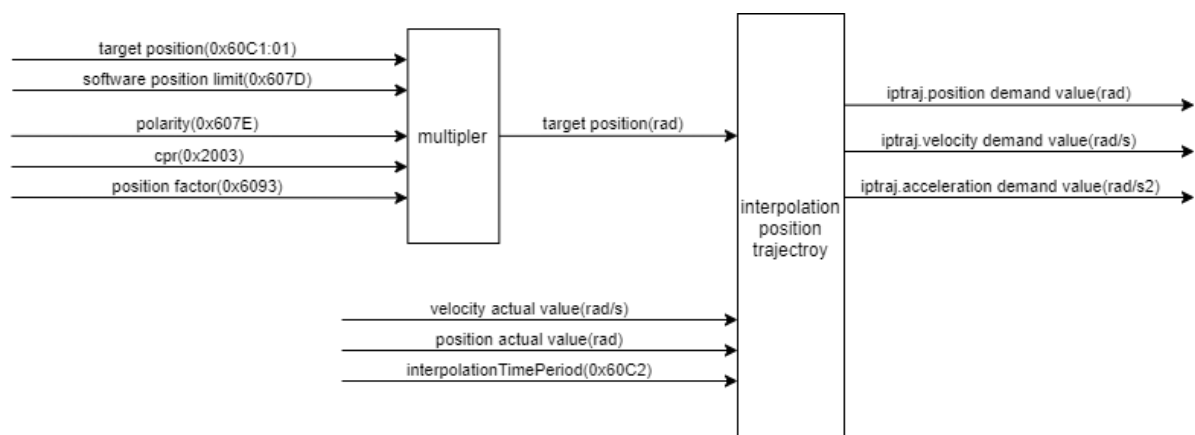
位	描述	备注
bit 10	0: 未到达目标。 1: 达到目标。	
bit 11	0: 未触发内部软件位置限制。 1: 触发内部软件位置限制。	
bit 12	0: 插值未使能。 1: 插值已经使能。	
bit 14	0: 硬件负向位置极限未触发。 1: 硬件负向位置极限触发。	需要 2240:01 对应的 nls 功能使能才会触发。
bit 15	0: 硬件正向位置极限未触发。 1: 硬件正向位置极限触发。	需要 2240:01 对应的 pls 功能使能。

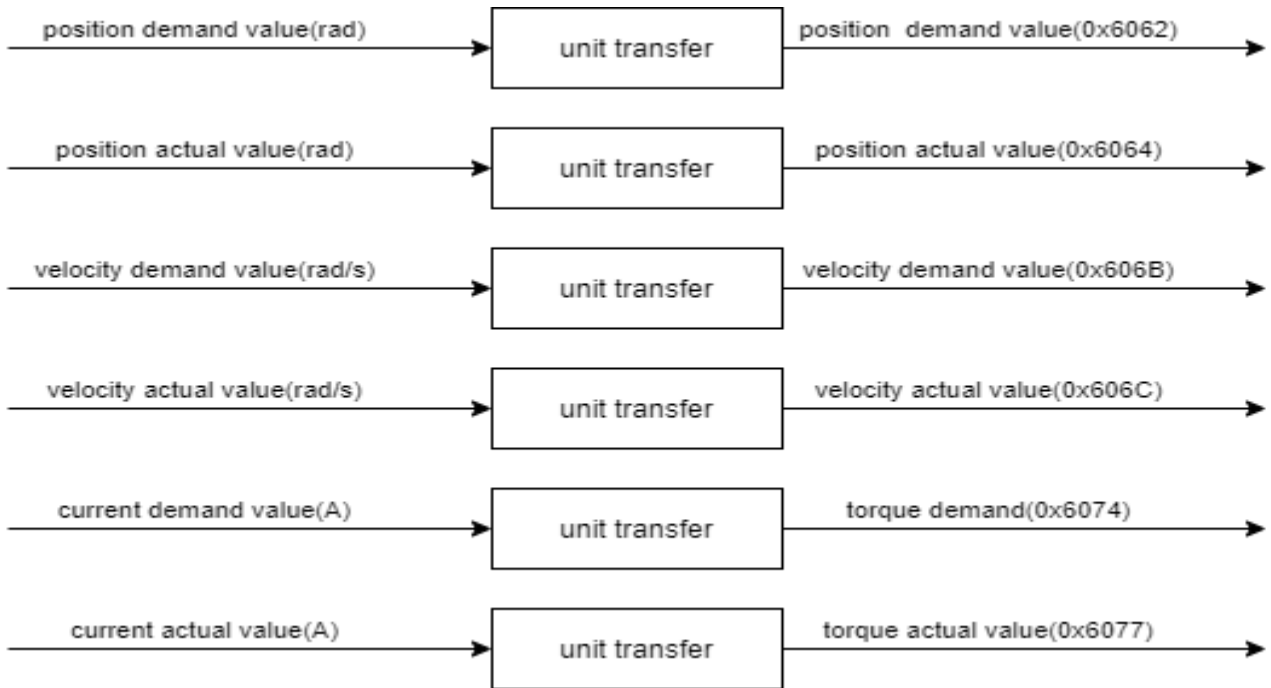
4.5.5. 相关对象

整个 interpolated position mode 涉及到的对象如下表格：

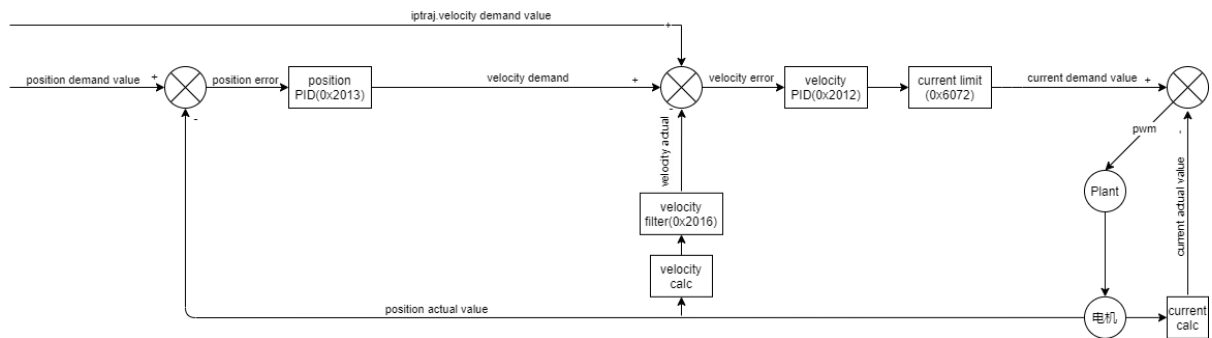
index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的PID.
0x2012	velocity loop PID, 速度环的PID.
0x2013	position loop PID, 位置环的PID.
0x2015	servo strategy, 伺服控制策略.
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window time, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607D	software position limit, 软件位置限制, 用户单位.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大速度, 用户单位.
0x6084	profile deceleration, 规划减速度, 用户单位.
0x6085	quickstop deceleration, 快停减速度, 用户单位.
0x6093	position factor, 位置比例.
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x60C0	interpolation submode select, 插补模式选择, 只支持线性插补
0x60C1	interpolation data record, 插补数据, 只支持 1 条绝对位置.
0x60C2	interpolation time period, 插补时间
0x60F4	following error actual value, 跟随误差实际值, 用户单位.

4.5.6. 规划框图





4.5.6. 控制框图



4.5.7. 配置实例

下表格对 nodeid=1 的实例进行 ip 模式配置:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 07 00 00 00	mode=0x07= interpolated position mode
601	2F C2 60 01 01 00 00 00	interpolation time period = 1ms
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	23 C1 60 01 42 40 0F 00	target position = target position =0x0F4240=1000000pulse
601	2B 40 60 00 1F 00 00 00	enable interpolation

4.6 cyclic synchronous position mode

4.6.1. 说明

循环同步位置模式也是位置控制, 不同于标准位置模式的地方就是, 标准位置模式会依据收到的 target position 做出三段规划(加速, 匀速, 减速), 到达目标位置的速度是零. 而循环同步位置模式的三段规划由命令发送方(上位机)来实现的, 所以伺服驱动器不需要执行三段规划了.

伺服驱动器收到 target position 后可以选择做位置插补(0x2015:01=1), 也可以不做位置插补(0x2015:01=0). 如果选择做位置插补, 那么伺服会根据目标位置和间隔时间(0x60C2)做规划, 输出出 cspTraj. positonDemandvalue 作为 position demand value(0x6062), 输出 cspTraj. velocityDemandValue 作为速度前馈. 如果不做规划, 是直接将目标位置作为 position demand value(0x6062), 将 velocity offset(0x60B1)作为速度前馈.

如果想取得多轴的高同步精度, 建议使用带规划的 CSP 模式, 建议使用 SYNC 帧来同步位置, 这样的控制方式可以获得小于 5us 误差的同步精度.

CSP 模式的执行延时固定为 1ms, 也就是从收到包含 target position 的那一帧(可能是 SDO, 可能是 PDO, 可能是 SYNC)的时刻的 1ms 后开始执行运动.

- 循环同步位置模式只支持绝对位置, 不支持相对位置.
- CSP 模式的位置规划, 不同于 PP 模式的三段规划, CSP 模式只做其中一段的规划.

4.6.2. 激活

要配置为 cyclic synchronous position mode, 需要将字典对象 0x6060(modes of operation)配置为 0x8.

4.6.3. 控制字

字典对象 0x6040(control word)没有模式相关位.

4.6.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

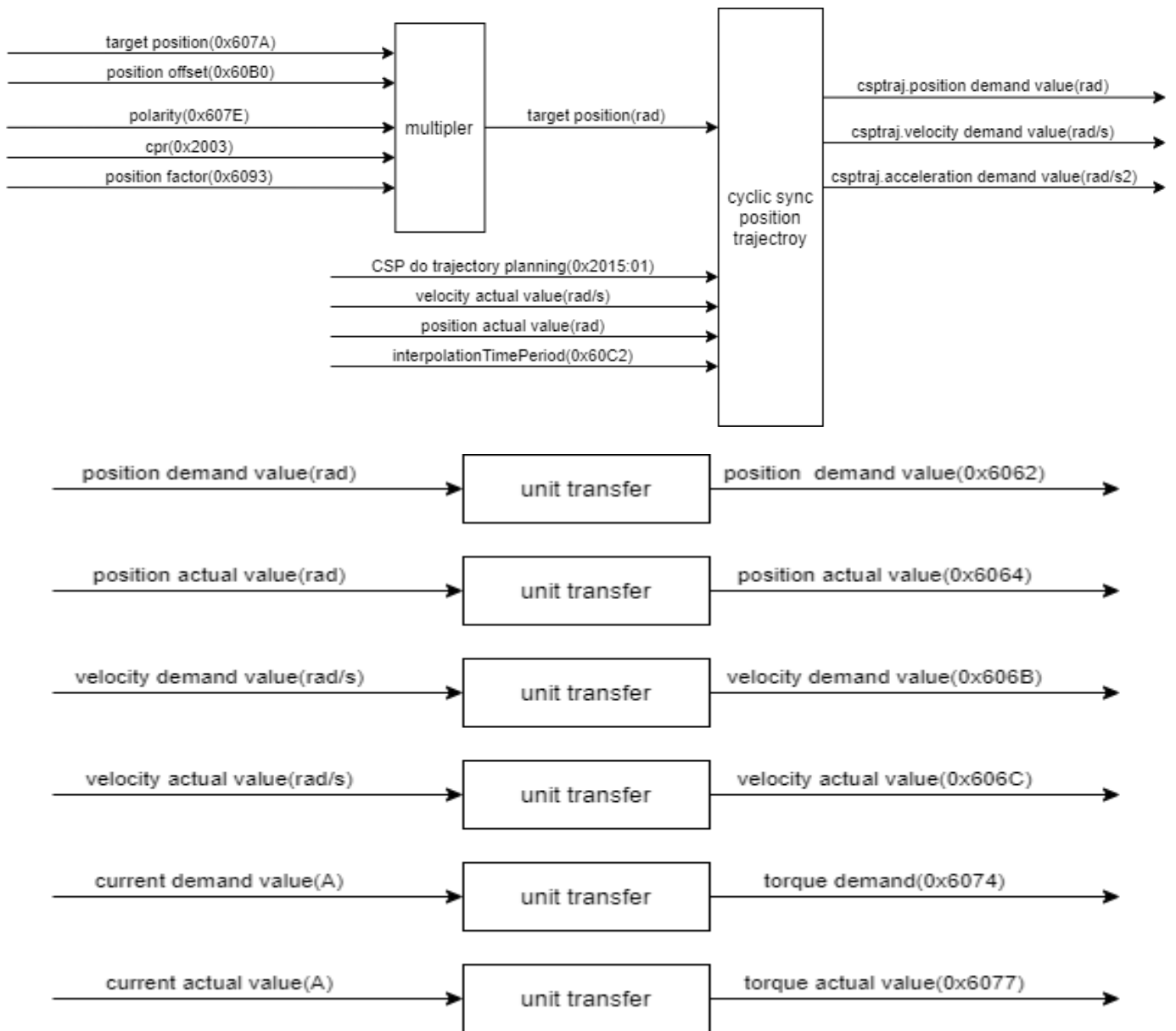
位	描述	备注
bit 12	0: target position ignored. 1: target position shall be used as input to position control loop.	
bit 13	0: no following error. 1: following error.	
bit 14	0: 硬件负向位置极限未触发. 1: 硬件负向位置极限触发.	需要 2240:01 对应的 nls 功能使能才会触发.
bit 15	0: 硬件正向位置极限未触发. 1: 硬件正向位置极限触发.	需要 2240:01 对应的 pls 功能使能.

4.6.5. 相关对象

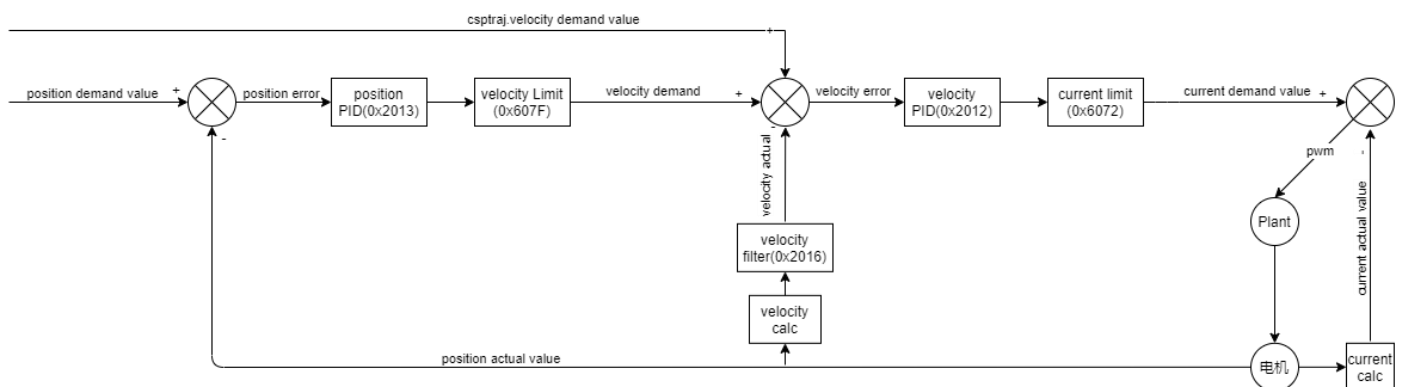
整个 cyclic synchronous position mode 涉及到的对象如下表格

index	描述
0x1006	communication Cycle Period, 同步周期.
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2013	position loop PID, 位置环的 PID.
0x2015	servo strategy, 伺服控制策略.
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window time, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607A	target position, 目标位置, 用户单位.
0X607D	software position limit, 软件位置限制, 用户单位.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大速度, 用户单位.
0x6081	profile velocity, 规划速度, 用户单位.
0x6084	profile deceleration, 规划减速度, 用户单位.
0x6085	quickstop deceleration, 快停减速度, 用户单位
0x6093	position factor, 位置比例.
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x60C2	interpolation time period, 插补周期.
0x60C5	max profile acceleration, 最大加速度, 用于限制 0x6083, 用户单位.
0x60C6	max profile deceleration, 最大减速度, 用于限制 0x6084, 用户单位.
0x60F4	following error actual value, 跟随误差实际值, 用户单位.

4.6.6. 规划框图



4.6.6. 控制框图



4.6.7. 配置实例

下表格对 nodeid=1 的实例进行 csp 模式配置.

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 08 00 00 00	mode=0x08=cyclic synchronous position mode
601	2F C2 60 01 01 00 00 00	interpolation time period = 1ms
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	23 7A 60 00 20 00 00 00	target position= 0x20

4.7 cyclic synchronous velocity mode

4.7.1. 说明

循环同步速度模式是速度控制,速度的规划由上位机实现,上位机定期发送 target velocity 以及 velocity offset 到伺服,伺服负责执行。

伺服层面,伺服根据 CSV do trajectory planning(0x2015:02)来决定是否做速度规划,如果该值为 true 就做规划,根据 target velocity(0x60FF)和插补周期(0x60C2)执行单段规划,输出 csvtraj.velocityDemandValue, 执行层定时获取规划输出做 velocity demand value(0x606B)。如果不执行规划, target velocity(0x60FF)和 velocity offset(0x60B1)只和直接作为 velocity demand value(0x606B)。

- CSV 模式可以速度规划,不同于 PV 模式的三段段速度规划,CSV 模式只做一段。

4.7.2. 激活

要配置为 cyclic synchronous velocity mode,需要将字典对象 0x6060(modes of operation)配置为 0x09。

4.7.3. 控制字

字典对象 0x6040(control word)中的没有特殊功能。

4.7.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

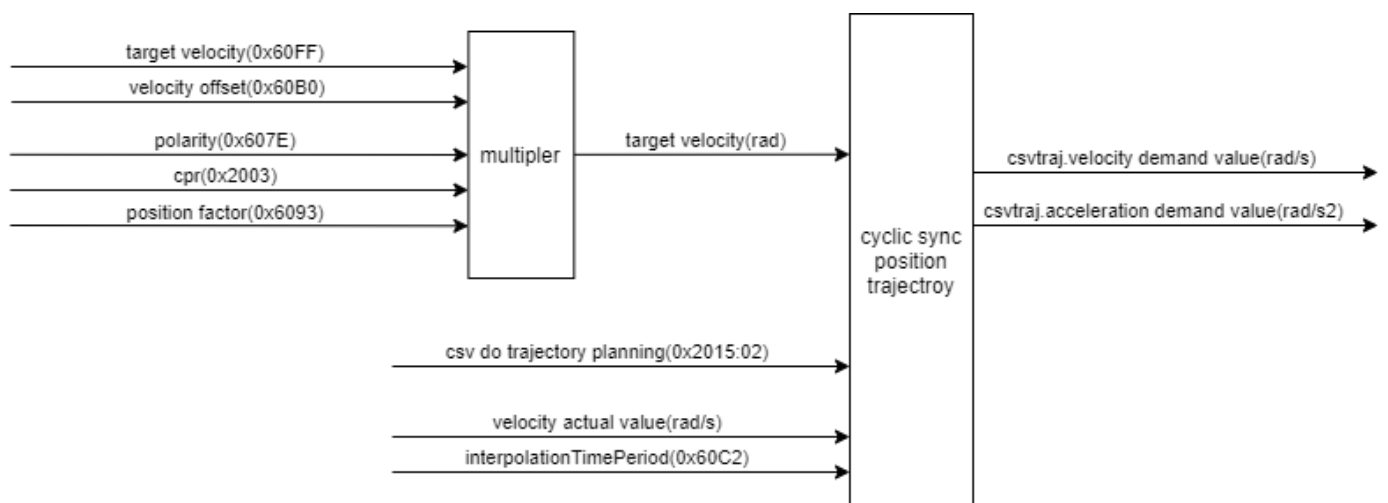
位	描述	备注
bit 12	0: target velocity ignored. 1: target velocity shall be used as input to velocity control loop.	
bit 14	0: 硬件负向位置极限未触发. 1: 硬件负向位置极限触发.	需要 2240:01 对应的 nls 功能使能才会触发.
bit 15	0: 硬件正向位置极限未触发. 1: 硬件正向位置极限触发.	需要 2240:01 对应的 pls 功能使能.

4.7.5. 相关对象

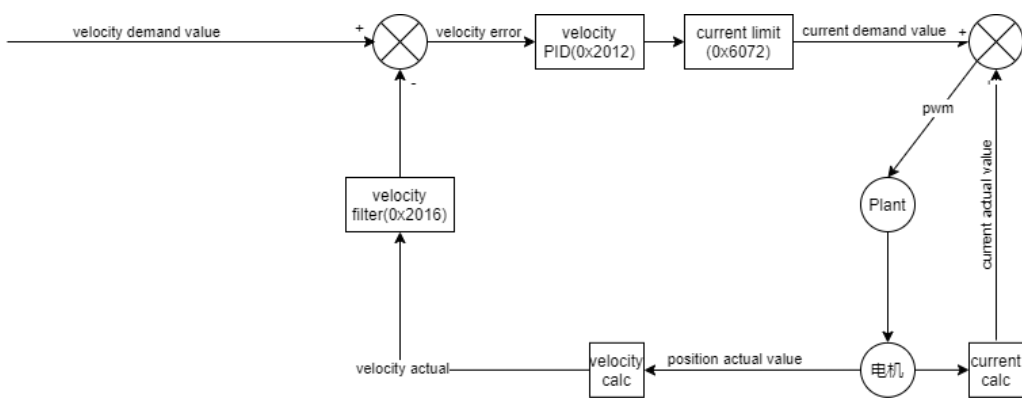
整个 cyclic synchronous velocity mode 涉及到的对象如下表格

index	描述
0x1006	communication Cycle Period, 同步周期.
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2015	servo strategy, 伺服控制策略.
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window tie, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607A	target position, 目标位置用户单位.
0x607E	polarity, 极性.
0x607F	max profile velocity, 最大规划速度, 用户单位.
0x6081	profile velocity, 规划速度, 用户单位.
0x6084	profile deceleration, 规划减速度, 用户单位.
0x6085	quickstop deceleration, 快停减速度, 用户单位
0x6093	position factor, 位置比例.
0x6095	velocity factor, 速度比例.
0x6097	acceleration factor, 加速度比例.
0x60C2	interpolation time period, 插补周期.
0x60C5	max profile acceleration, 最大加速度, 用于限制 0x6083, 用户单位.
0x60C6	max profile deceleration, 最大减速度, 用于限制 0x6084, 用户单位.
0x60F4	following error actual value, 跟随误差实际值, 用户单位.

4.7.6. 规划框图



4.7.7. 控制框图



4.7.8. 配置实例

下表格对 nodeid=1 的实例进行 CSV 模式配置:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 09 00 00 00	mode=0x09=cyclic synchronous velocity mode
601	2F C2 60 01 01 00 00 00	interpolation time period = 1ms
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	23 FF 60 00 0A 00 00 00	target velocity = 0x0A

4.8 cyclic synchronous torque mode

4.8.1. 说明

循环同步扭矩模式是扭矩控制，扭矩的规划由上位机实现，上位机定期发送 target torque 到伺服，伺服负责执行。伺服层面，伺服根据 CST do trajectory planning(0x2015:03) 来决定是否做扭矩规划，如果该值为 true 就做规划，根据 target torque(0x6071) 和插补周期(0x60C2) 执行单段规划，输出 csttraj.torqueDemandValue，执行层定时获取规划输出做 torque demand(0x6074)。如果不执行规划，target torque(0x6071) 和 torque offset(0x60B2) 只和直接作为 torque demand(0x6074)。

cyclic synchronous torque mode 下，电机的转速也受到 max profile velocity(0x607F) 的限制，最大转速由 max profile velocity 决定。当电机速度趋向于 max profile velocity 时，torqueDemand 从 target torque 梯形趋向于 0，梯形斜率由 velocity kp(0x2012:03) 决定，velocity kp 越大，梯形越陡，velocity kp 过大且 maxProfileVelocity 过小 (比如接近于 0) 会导致电机来回震荡。当实际转速到达 max profile velocity 时，torqueDemand 会是 0。

- CST 模式也做扭矩规划，不同于 PV 模式的二段扭矩规划，CST 模式只做一段。

4.8.2. 激活

要配置为 cyclic synchronous torque mode, 需要将字典对象 0x6060(modes of operation) 配置为 0xA。

4.8.3. 控制字

字典对象 0x6040(control word)中没有特殊功能:

4.8.4. 状态字

对象 0x6041(status word)中的以下位具有特殊功能:

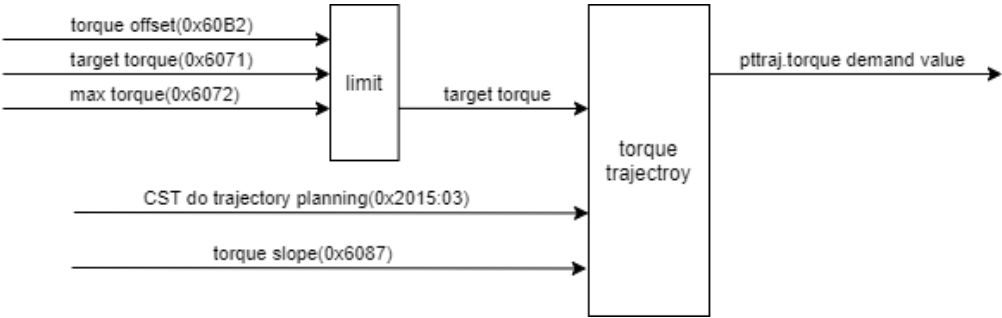
位	描述	备注
bit 12	0: target torque ignored. 1: target torque shall be used as input to torque control loop.	
bit 14	0: 硬件负向位置极限未触发. 1: 硬件负向位置极限触发.	需要 2240:01 对应的 nls 功能使能才会触发.
bit 15	0: 硬件正向位置极限未触发. 1: 硬件正向位置极限触发.	需要 2240:01 对应的 pls 功能使能.

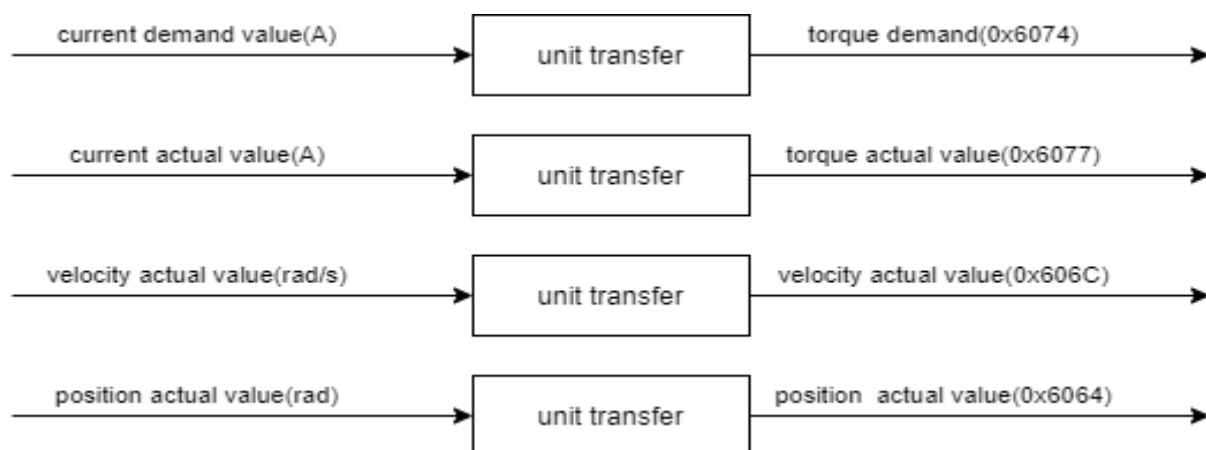
4.8.5. 相关对象

整个 cyclic synchronous torque mode 涉及到的对象如下表格

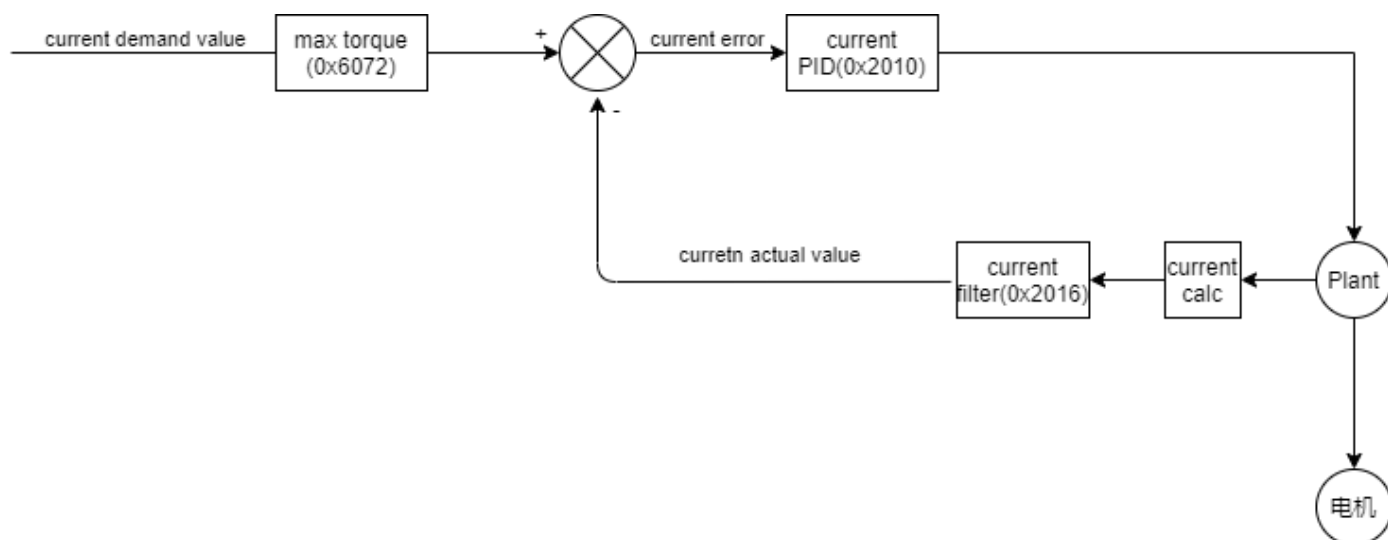
index	描述
0x1006	communication Cycle Period, 同步周期.
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2015	servo strategy, 伺服控制策略.
0x2032	torque mode additional para, 扭矩模式额外参数
0x60C2	interpolation time period, 插补周期.
0x6071	target torque, 目标扭矩, 以 rated torque 为基数的千分比.
0x6072	max torque, 最大扭矩, 以 rated torque 为基数的千分比.
0x6074	demand torque, 需求扭矩, 以 rated torque 为基数的千分比.
0x6077	torque actual value, 实际扭矩, 以 rated torque 为基数的千分比.
0x6087	torque slope, 扭矩斜坡, 千分比.

4.8.6. 规划框图





4.8.7. 控制框图



4.8.8. 配置实例

下表格对 nodeid=1 的实例进行 CST 模式配置:

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 0A 00 00 00	mode=0x0A=cyclic synchronous torque mode
601	2F C2 60 01 01 00 00 00	interpolation time period = 1ms
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation
601	2B 71 60 00 0A 00 00 00	target torque = 0x0A

4.9 step dir position mode

4.9.1. 说明

脉冲位置模式可以分为带 CANOPEN 的脉冲位置模式,也可以是不带 CANOPEN 的脉冲位置模式. 当 (2015:05h)置位并重启后, 脉冲模式不再需要 CANOPEN, CAN 线都可以不接. 当 (2015:05h)清零并重启后, 脉冲模式需要 CANOPEN, 模式的使能还是通过 6040 实现, 而方向和脉冲分别由 DI2 和 DI1 实现.

4.9.2. 激活

要配置为 step dir position mode, 需要将字典对象 0x6060(modes of operation)配置为 0xFF. 如果是不带 CANOPEN 的脉冲位置模式, 需要将 (2015:05h)置位并重启.

4.9.3. 控制字

字典对象 0x6040(control word)没有模式相关位.

4.9.4. 状态字

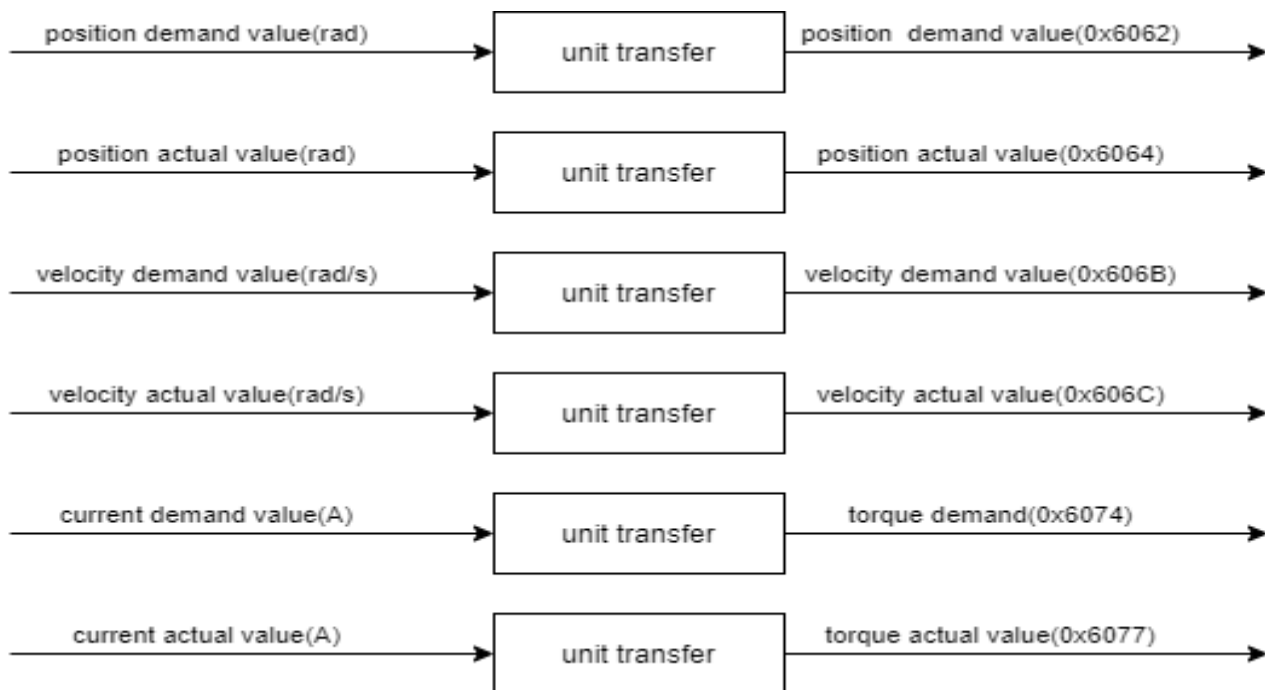
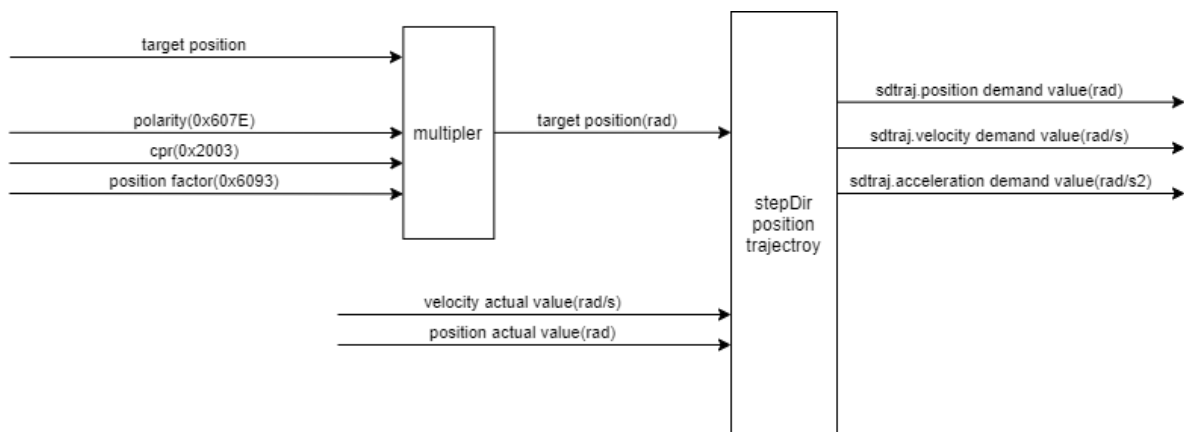
对象 0x6041(status word)中的没有特殊功能.

4.9.5. 相关对象

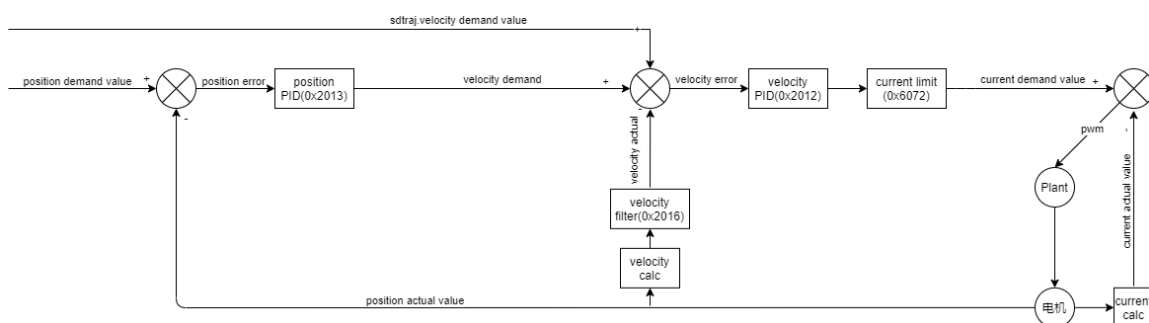
整个 step dir position mode 涉及到的对象如下表格:

index	描述
0x2003	count per resolution, 单圈脉冲数.
0x2010	current loop PID, 电流环/扭矩环的 PID.
0x2012	velocity loop PID, 速度环的 PID.
0x2013	position loop PID, 位置环的 PID.
0x2015	servo strategy, 伺服控制策略.
0x6062	position demand value, 位置规划值, 用户单位.
0x6064	position actual value, 位置当前值, 用户单位.
0x6065	following error window, 跟随错误窗口值, 用户单位.
0x6066	following error timeout, 跟随错误时间窗口.
0x6067	position window, 判定位置到达的位置窗口, 用户单位.
0x6068	position window time, 判定位置到达的时间窗口.
0x606B	velocity demand value, 速度规划值, 用户单位.
0x606C	velocity actual value, 速度实际值, 用户单位.
0x6072	max torque, 最大扭矩, 单位是千分之一的额定扭矩.
0x6074	torque demand value, 需求扭矩, 单位为千分比.
0x6077	torque actual value, 实际输出扭矩, 单位为千分比.
0x607E	polarity, 极性.
0x6093	position factor, 位置比例.
0x60F4	following error actual value, 跟随误差实际值, 用户单位.

4.9.6. 规划框图



4.9.7. 控制框图



4.9.8. 配置实例

下表格对 nodeid=1 的实例进行脉冲模式配置, 注意是带 CANOPEN 的脉冲模式.

- TPD01 配置了 position actual value 以及 velocity actual value.

COBID	Data	Description
000	01 01	NMT enter operational
601	2F 60 60 00 FF 00 00 00	mode=0xFF= step-dir mode
601	2B 40 60 00 06 00 00 00	set control word = enable voltage
601	2B 40 60 00 0F 00 00 00	set control word = enable operation

4.10 step dir velocity mode

4.10.1. 说明

脉冲速度模式和脉冲位置模式的区别不大, 脉冲速度模式根据输入脉冲频率控速度, 不再实现位置控制.

4.10.2. 激活

要配置为 step dir velocity mode, 需要将字典对象 0x6060 (modes of operation) 配置为 0xFE. 如果是不带 CANOPEN 的脉冲位置模式, 需要将 (2015:06h) 置位并重启. 注意, (2015:05h) 和 (2015:06h) 是互斥的, 一个的置位会导致另一个的清零.

4.10.3. 控制字

字典对象 0x6040 (control word) 没有模式相关位.

4.10.4. 状态字

对象 0x6041 (status word) 中的没有特殊功能.

4.10.5. 相关对象

同 step dir position mode 涉及到的对象.

4.10.6. 规划框图

同 step dir position mode 的规划框图.

4.10.7. 控制框图

同 step dir position mode 的控制框图.

4.10.8. 配置实例

同 step dir position mode 的配置实例.

五、错误告警

伺服驱动实现的告警方式有 2 种, 2 种同时实现, 一种是通过 CANopen 的 EMCY 报文发送, 另一种是在驱动器的表面通过 LED 的闪烁实现.

大部分的错误告警可以通过 object 0x2019 关闭部分, 少部分不能关闭, 比如短路过流 (0x2310), 硬件过压 (0x2314).

5.1 LED 告警

当没有发生告警的时候, LED 的状态是绿灯常亮或者闪烁 (120 欧姆电阻使能时闪烁), 红灯长灭. 当错误发生的时候, 除了 CANOPEN 发送对应的 EMERGENCY 报文外, LED 的状态也会变化, 根据错误的不同, LED 的闪烁方式不同. 当 LED 闪烁表示错误时:

- 绿灯长灭, 红灯按照 1Hz 的频率闪烁, 红灯闪烁的次数不同代表着不同的错误.
- 每一轮完整的闪烁警示之间有 2 秒的红/绿灯长灭表示区分间隔.
- 除非清除错误, 红灯会不停的闪烁提示错误.
- 当有多个错误发生时, LED 只提示最先发生的那一个错误.
- 需要注意 CAN 总线的一些错误, 当该错误消失的时候, LED 这里的告警也会关闭. CAN 总线的心跳丢失和 SYNC 超时也是可恢复的.

红灯闪烁次数	错误名称	对应 Errorcode
1	over voltage, user	0x3110
2	under voltage, user	0x3120
3	over current, short	0x2310
4	over current, IIT	0x2314
5	over temperature	0x4310/0x4311
6	under temperature	0x4320/0x4321
7	CAN bus passive	/
8	following error	0x8611
9	heartbeat lost	0x8130
10	over velocity	0x8401
12	unwanted brake	0x7114
13	unwanted unbrake	0x7115
14	temperature sensor error	0x4350
15	over voltage, hardware	0x3111
16	under voltage, hardware	0x3121
17	wrong limit	0xFF01
18	nls reached	0xFF02
19	pls reached	0xFF03
20	TPDO out window	0x8202
21	CAN bus off	0x8140
22	CAN TX overrun	0x8111
23	CAN RX overrun	0x8112
24	SYNC timeout	0x8201
25	SYNC length error	0x8240
26	RPDO wrong length	0x8210, 0x8220
27	RPDO time out	0x8250
28	wrong initial when homing	0xFF04
30	battery voltage low	0x3201

5.2 EMERGENCY 告警

canopen 接口发送的 emergency 有如下这些:

error code	错误名称	Description
0x2310	over current, short	过流, 基本是短路.
0x2314	over current, IIT	IIT 过流.
0x3110	over voltage, user	供电电压超过用户设定最高值(0x2018:01).
0x3111	over voltage, hardware	供电电压超过硬件允许最高值(0x2018:03)
0x3120	under voltage, user	供电电压低于用于设定最低值(0x2018:02)
0x3121	under voltage, hardware	供电电压低于硬件允许最高值(0x2018:04)
0x4310	amplifier over temperature	驱动器温度过高.
0x4311	motor over temperature	电机温度过高.
0x4320	amplifier under temperature	驱动器温度过低.
0x4321	motor under temperature	电机温度过低.
0x4350	temperature sensor error	温度传感器工作异常, 会导致温度不准.
0x7114	unwanted brake	检测到异常刹车.
0x7115	unwanted unbrake	检测到异常松刹车.
0x7305	encoder over speed	磁编检测到过速.
0x7306	encoder under magnet	磁编对应的磁铁磁性不够.
0x7307	encoder under voltage	磁编的供电电压不足.
0x8111	canbus tx overrun	发送溢出, 会导致消息丢失.
0x8112	canbus rx overrun	接收溢出, 会导致消息丢失.
0x8130	heartbeat lost	心跳丢失.
0x8131	remote node bootup	被监控对象重启了
0x8140	canbus bus off	can 总线硬件判定.
0x8201	canopen sync timeout	sync 超时
0x8202	canopen delete tpdo	因 sync 超时导致部分待发送的 tpdo 被删除
0x8210	canopen rpdo short	RPDO 收到的数据过短
0x8220	canopen rpdo long	RPDO 收到的数据过长
0x8240	canopen sync over length	sync 的帧长度过长.
0x8401	over velocity	过速
0x8611	following error	跟随误差.

0x2310, over current, short

- **描述:**
短路过流, 基本上是发生异常短路, 硬件过流阈值不可配置.
- **触发条件:**
母线电流超过预设值的短路阈值.
- **后续动作:**
驱动器会立即断电, 不会执行 fault reaction(0x605E)过程.
- **是否能够关闭:**
不能.

0x2314, over current, IIT

- **描述:**
IIT 短路过流, 一种用户可以配置的过流检测, 可用于检测堵转.
- **触发条件:**
当实际驱动电流 (I_q) 大于触发阈值 (0x2017:03, IIT on threshold) 时, IIT value (0x2017:02) 开始累加, 当 IIT value 大于 IIT limit (0x2017:01) 时激活 IIT over current 告警.
在累加过程中一旦出现驱动电流 (I_q) 小于 IIT off threshold (0x2017:04), IIT value (0x2017:02) 就停止累加并清零.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止.
- **是否能够关闭:**
能, 但不建议. 对 0x2019:02 清零即可关闭该错误的后续 fault reaction.

0x2320, stall

- **描述:**
堵转告警, 用于堵转保护. 堵
- **触发条件:**
转 0.5 秒即触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止.
- **是否能够关闭:**
不能.

0x3110, over voltage

- **描述:**
母线电压超过用户设定的过压值, 该阈值 (0x2018:01) 可配置.
- **触发条件:**
母线电压 (0x6079) 大于触发阈值 (0x2018:01) 时触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止.
- **是否能够关闭:**
不能.

0x3111, over max voltage

- **描述:**
母线电压超过驱动器内置的硬件过压值, 该阈值不可配置.
- **触发条件:**
母线电压 (0x6079) 大于触发阈值 (0x2018:03) 时触发告警.
- **后续动作:**
驱动器立即停止驱动, 并短路电机各相来紧急刹车, 以免电压过高损坏驱动器或者外部设备. 也就是说驱动器不会执行 fault reaction.
- **是否能够关闭:**
不能.

0x3120, under voltage

- **描述:**
母线电压低于用户配置的欠压值, 该阈值 (0x2018:02) 可配置.
- **触发条件:**
母线电压 (0x6079) 低于触发阈值 (0x2018:02) 时触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止.
- **是否能够关闭:**
不能.

0x3121, under min voltage

- **描述:**
母线电压低于驱动器内置的欠压值, 该阈值不可配置.
- **触发条件:**
母线电压 (0x6079) 低于触发阈值 (0x2018:04) 时触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x3201, battery voltage low

- **描述:**
多圈位置保持的外置电池电压过低.
- **触发条件:**
多圈位置保持的外置电池电压低于触发阈值 (0x201D:01) 时触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不接电池不会触发.

0x4310, amplifier over temperature

- **描述:**
驱动器的温度过高.
- **触发条件:**
驱动器温度 (0x201B:01) 高于触发阈值 (0x201B:03) 时触发告警.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x4311, motor over temperature

- **描述:**
电机的温度过高.
- **触发条件:**

电机温度(0x201B:02)高于触发阈值(0x201B:05)时触发告警.

- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x4320, amplifier under temperature

- **描述:**
驱动器的温度过低.
- **触发条件:**
驱动器温度(0x201B:01)低于触发阈值(0x201B:04)时触发告警.
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x4321, motor under temperature

- **描述:**
电机的温度过高.
- **触发条件:**
电机温度(0x201B:02)低于触发阈值(0x201B:06)时触发告警.
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x4350, temperature sensor error

- **描述:**
温度传感器检测到错误.
- **触发条件:**
伺服控制单元检测到温度传感器异常.
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x7114, unwanted brake

- **描述:**
发生异常抱闸情况.
- **触发条件:**
伺服控制单元检测到实际发生抱闸, 而没有期待抱闸时触发.
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.

- **是否能够关闭:**
不能.

0x7115, unwanted unbrake

- **描述:**
发生异常松抱闸情况.
- **触发条件:**
伺服控制单元检测到实际发生松抱闸, 而没有期待松抱闸时触发.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x8111, canopen tx overrun

- **描述:**
canopen 数据包发送过载, 发送包会丢失.
- **触发条件:**
硬件判定.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:06 清零即可关闭该错误的后续 fault reaction.
- **可能的原因:**
 - a. can 总线波特率是 1Mbps 时, 1 毫秒内最多 8 个满载的 PDO 帧, 配置过多的 PDO 会出现本告警.

0x8112, canopen rx overrun

- **描述:**
canopen 数据包接收过载, 接收包会丢失.
- **触发条件:**
硬件判定.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:05 清零即可关闭该错误的后续 fault reaction.

0x8130, heartbeat lost

- **描述:**
心跳丢失.
- **触发条件:**
开启心跳监控后, 对方在规定时间内 (0x1016) 内没有发来 heartbeat.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**

能. 对 0x2019:03 清零即可关闭该错误的后续 fault reaction.

0x8140, can bus off

- **描述:**
CAN 总线断开, MCU 上 CAN 控制器的底层驱动反馈的一个报警.
- **触发条件:**
CAN 总线通信断开了.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x8131, remote node bootup

- **描述:**
远程节点重启.
- **触发条件:**
检测到其它节点发来的 bootup 包.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:07 清零即可关闭该错误的后续 fault reaction.

0x8201, canopen sync timeout

- **描述:**
sync 帧丢失.
- **触发条件:**
在规定时间内 (0x1019) 内没有收到 sync 帧.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:04 清零即可关闭该错误的后续 fault reaction.

0x8202, canopen tpdo out window

- **描述:**
sync 帧丢失.
- **触发条件:**
同步的 TPDO 帧没有在规定时间内 (0x1007) 内发出去.
- **后续动作:**
来不及发出去的同步 TPDO 帧被清除.
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:08 清零即可关闭该错误的后续 fault reaction.

0x8210, canopen rpdo short

- **描述:**
收到的 RPDO 帧的长度小于预期.
- **触发条件:**
实际收到的 RPDO 帧的长度比预设定的小.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:09 清零即可关闭该错误的后续 fault reaction.

0x8220, canopen rpdo long

- **描述:**
收到的 RPDO 帧的长度大于预期.
- **触发条件:**
实际收到的 RPDO 帧的长度比预设定的大.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能. 对 0x2019:09 清零即可关闭该错误的后续 fault reaction.

0x8240, canopen sync overlength

- **描述:**
SYNC 帧的帧长度不对.
- **触发条件:**
SYNC 帧的数据长度, 和预期不符.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
不能.

0x8250, canopen rpdo timeout

- **描述:**
RPDO 超时.
- **触发条件:**
RPDO 的时间间隔超过 RPDO communication parameters 中的 event timer 规定的值.
- **后续动作:**
根据 fault reaction option code (0x605E) 的配置执行减速过程后停止.
- **是否能够关闭:**
能. 对 0x2019:0A 清零即可关闭该错误的后续 fault reaction.

0x8401, over velocity

- **描述:**

运动过速.

- **触发条件:**
实际速度(606C)超过 max profile velocity(0x607F)的 1.2 倍的持续时间超过 10 毫秒.
实际速度小于 100RPM 时, 不会触发该错误.
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能, 设置 max profile velocity 为 0xFFFFFFFF 即可关闭, 但不建议.

0x8611, following error

- **描述:**
位置模式时发生跟随错误.
- **触发条件:**
following error actual value(0x60F4)的值大于 following error window(0x6065)的持续时间大于 following error timeout(0x6066).
- **后续动作:**
根据 fault reaction option code(0x605E)的配置执行减速过程后停止. 需要注意的是外部供电未必能够支持 fault reaction 的动作.
- **是否能够关闭:**
能, 设置 following error window 为 0xFFFFFFFF 即可关闭.

0xFF01, wrong limit

- **描述:**
触碰到错误的 limit 传感器.
- **触发条件:**
往正方向运动碰到 NLS 传感器, 往负方向运动碰到 PLS 传感器.
- **后续动作:**
如果是回零模式下触发本告警, 伺服根据 0x605A quick-stop option 执行 quick-stop. 非回零模式下触发是按照 0x6085 减速后进入 fault 状态.
- **是否能够关闭:**
不能.
- **常见原因分析:**
 - PLS 传感器接到 NLS 接口.
 - NLS 传感器接到 PLS 接口.
 - 伺服的初始位置不再 NLS 和 PLS 之间.

0xFF02, negative limit switch reached

- **描述:**
到达负向极限.
- **触发条件:**
往负方向运动碰到 NLS 传感器
- **后续动作:**
伺服根据 0x6085 的快停速度执行快停, 停下后保持位置不动. 伺服依旧处于 enable 状态. 后续往正方向的运动会执行, 往负方向的运动会被忽略. NLS 传感器的位置会被记录到 0x3100_nls position 中, 只有当实际位置大于 0x3100_nls position 后才允许执行往负方向的运动. 注意, 本告警的触发不会导致伺服进入 fault 状态.
- **是否能够关闭:**
不能.
- **常见原因分析:**
 - 伺服往负方向运动出发了 NLS 传感器.

0xFF03, positive limit switch reached

- **描述:**
到达正向极限.
- **触发条件:**
往正方向运动碰到 PLS 传感器
- **后续动作:**
伺服根据 0x6085 的快停速度执行快停, 停下后保持位置不动. 伺服依旧处于 enable 状态. 后续往负方向的运动会执行, 往正方向的运动会被忽略. NLS 传感器的位置会被记录到 0x3101_pls position 中, 只有当实际位置小于 0x3101_pls position 后才允许执行往负方向的运动. 注意, 本告警的触发不会导致伺服进入 fault 状态.
- **是否能够关闭:**
不能.
- **常见原因分析:**
 - 伺服往正方向运动出发了 PLS 传感器.

0xFF04, wrong initial condition when homing

- **描述:**
回零状态的初始条件不对.
- **触发条件:**
详见章节 4.4.5 回零方法中的描述.
- **后续动作:**
伺服根据 0x605A quick-stop option 执行 quick-stop.
- **是否能够关闭:**
不能.
- **常见原因分析:**
伺服的初始位置错误.

5.3 SDO abort code

SDO abortcode 比较常见, 这里将可能碰到的 abort code 罗列一下并加以说明.:

abort code	name	description
0x05030000	toggle bit not alternated	
0x05040000	SDO protocol timed out	
0x05040001	client/server command specifier not valid or unknown	
0x05040002	invalid block size(block mode only)	
0x05040003	invalid sequence number(block mode only)	
0x05040004	CRC error(block mode only)	
0x05040005	out of memory	
0x06010000	unsupported access to an object	
0x06010001	attempt to read a write only object	
0x06010002	attempt to write a read only object	
0x06020000	object does not exist in the object dictionary	
0x06040041	Object cannot be mapped to the PDO	
0x06040042	The number and length of the objects to be mapped would exceed PDO length.	
0x06040043	General parameter incompatibility reason.	
0x06040047	General internal incompatibility in the device.	
0x06060000	Access failed due to an hardware error.	
0x06070010	Data type does not match, length of service parameter does not match	
0x06070012	Data type does not match, length of service parameter too high	
0x06070013	Data type does not match, length of service parameter too low	
0x06090011	Sub-index does not exist	
0x06090030	Invalid value for parameter (download only).	
0x06090031	Value of parameter written too high (download only).	
0x06090032	Value of parameter written too low (download only).	
0x06090036	Maximum value is less than minimum value.	
0x060A0023	Resource not available: SDO connection	
0x08000000	General error	
0x08000020	Data cannot be transferred or stored to the application.	
0x08000021	Data cannot be transferred or stored to the application because of local control	
0x08000022	Data cannot be transferred or stored to the application because of the present device state.	
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).	
0x08000024	No data available	

六、其它

6.1 LED

设备带有 2 个 LED 指示灯, 红灯和绿灯. 在同一时刻, 两个灯只使用一个, 红灯优先. 红灯用于输出告警, 绿灯表示状态.

6.1.1. 绿灯

绿灯表示伺服正常. 其中常亮的绿灯表示伺服状态正常且内置 CAN 总线的 120 欧姆电阻未被使能. 绿灯闪烁表示伺服状态正常且 CAN 总线的 120 欧姆电阻已经使能.

6.1.2. 红灯

详见 5.1 章节