P3 CANopen Manual

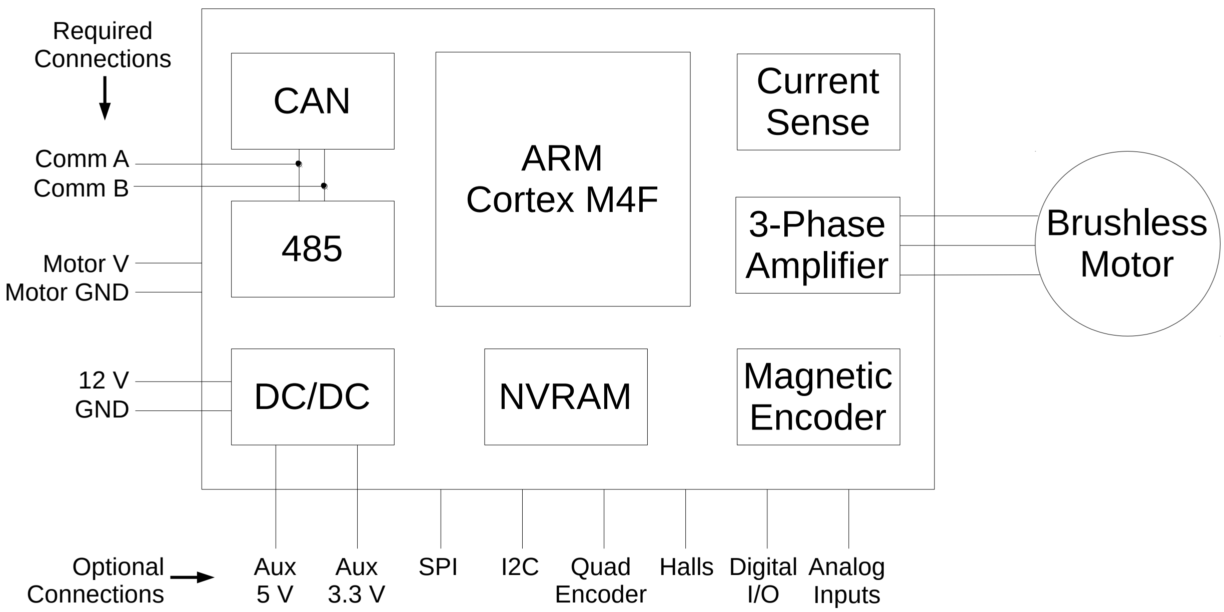
Table of Contents

Introduction

Overview

The Barrett Puck, version 3, (P3) is an ultraminiature brushless motor controller with a built-in encoder. Designed to be mounted directly to the motor, P3 simplifies system architecture by replacing home-run wiring between motors, controllers, and sensors with a convenient bus for power and CANopen communications.

Block Diagram



Hardware Features

- 120 MHz 32-bit ARM Cortex M4F DSP
- Integrated current sensing
- Space-vector commutation
- Built-in encoder:
    - 4096 cts/rev
    - Absolute within a single revolution
    - Impervious to dust & debris
- External encoder support
    - SPI
    - Quadrature
- 3.5 A peak current
- 12-48 V motor bus
- ISO 11898-2 CAN physical layer
- RS-485 physical layer
- Software-enabled termination resistor
- 3.3 and 5 V auxiliary outputs
- Internal temperature sensing
- In-system upgradeable firmware
- PWM frequency adjustable up to 100 kHz
- Digital Hall-effect feedback
- Dual analog inputs
- Up to 4 digital I/O
- SPI/I2C master for external sensors

Supported CANopen Features

- Network Management (NMT) messages
- Heartbeat producer
- Expedited Service Data Objects (SDOs)
- Byte-level Receive/Transmit Process Data Objects (RPDOs/TPDOs)
- Sync messages for RPDOs/TPDOs
- Up to 4 RPDOs and 4 TPDOs
- Up to 4 mappable objects per PDO
- Dynamic RPDO/TPDO configuration
- SDO abort message generation
- Optional "Boot to Operational State"
- Single pair of static SDO Connection Object Identifiers (COB IDs), 0x600/0x580 + NodeID
- 11-bit CAN 2.0A identifiers
- Control word (0x6040), except halt
- Status word (0x6041), except remote
- Drive modes:
  - Idle (0)
  - Profile Position (1)
  - Profile Velocity (3)
  - Profile Torque (4)
  - Homing Immediate (6)
  - Cyclic Synchronous Position (8)
  - Cyclic Synchronous Velocity (9)
  - Cyclic Synchronous Torque (10)
  - Cyclic Synchronous Torque with Angle (11)
  - Phase Voltage with Angle (12)
- Emergency (EMCY) messages
- Error registers (0x1001, 0x1002, 0x1003, 0x603F)
- Object Dictionary "dummy" entries

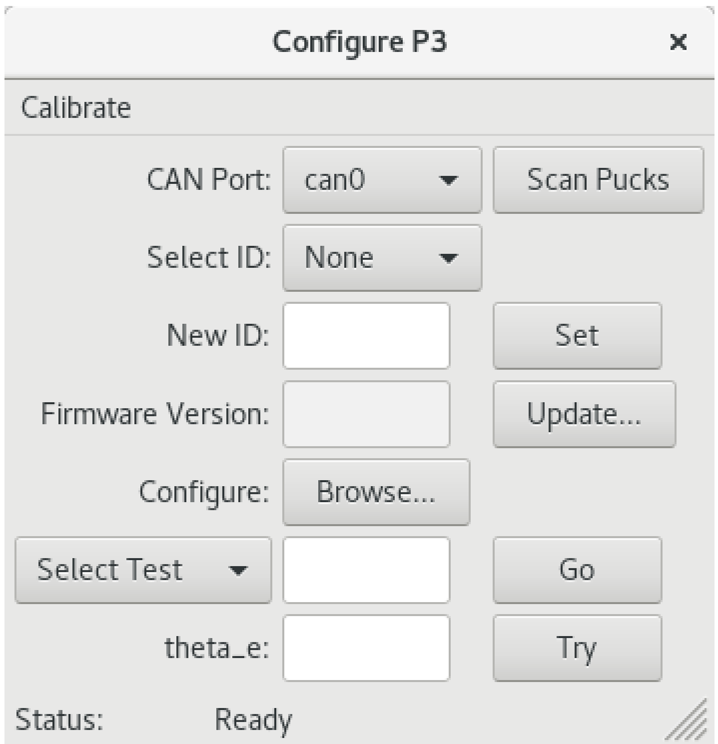
CANopen Features Not Yet Supported

- Heartbeat consumption
- Timestamps
- Segmented SDOs
- Block transfer SDOs
- SDO size indication (all Tx SDOs are 8 bytes regardless of payload type)
- Bit-level RPDOs/TPDOs
- Drive modes:
  - Velocity (2)
  - Hard-stop, Limit-switch Homing (6)
  - Interpolated Position (7)
- Remote Transmit Requests (RTR)
- Multiple/dynamic SDO COB IDs
- NMT master discovery
- TPDO inhibit timers
- TPDO event timers (tx on data change)
- 29-bit CAN 2.0B identifiers
- CAN bitrates other than 1 Mbps
- Deceleration ramps when switching from Operation Enabled
- Digital input triggers
- Analog torque/speed control
- Safe Torque Off (STO) input

Software Installation

Barrett's wxp3_app.py is an open-source wxPython-based application which provides basic configuration and operation, including:

- Enumerate the CAN bus (find and select pucks on a bus)
- Change CAN ID
- In-system firmware updates over CAN
- CANopen Object Dictionary configuration from a CSV file
- Profile Torque, Velocity, and Position control tests



Note: While any CAN hardware should work fine, Barrett uses the Peak PCAN-USB adapter for development and testing.

Install wxp3 under Windows

[TBD]

Install wxp3 under Linux

[TBD]

Usage

[TBD]

Manual Configuration

Enumerate

MsgID	DLC	Data	Description
0x000	2	81 00	NMT Reset All Nodes
0x701	*1*	*00*	*Boot-up message from ID = 1*
0x702	*1*	*00*	*Boot-up message from ID = 2*

Update Node ID

Write a new ID to the Network Configuration (0x21B0,0). The new ID is usable immediately.

Caution: if you accidentally set a node ID that already exists on the CAN bus, those nodes will go offline. You will then need to unplug one of the conflicting nodes from the network and give the other one a unique ID before reconnecting the node.

Save the new ID to non-volatile storage by setting 0x1010,4 to 0x21B0,0.

MsgID	DLC	Data	Description
0x601	6	22 B0 21 00 02 00	Change from ID = 1 to ID = 2
0x602	8	22 10 10 04 00 B0 21 00	Save the new ID to non-volatile memory

Update PDOs, if necessary for your application. It is common, but not required, for the MsgIDs of RPDOs and TPDOs to be based on the node ID. See the section, Configure PDOs, below.

Calibrate Current Sensor

The P3 needs to know its current sensor's internal bias voltage when no motor current is flowing.

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On
0x601	6	22 40 60 00 0F 00	Go to Operation Enabled
0x601	5	22 60 60 00 00	Set Mode: Idle
			(Wait 1 second)
0x601	4	40 08 30 01	Read filtered current sense
0x581	*8*	*42 08 30 01 96 72 00 00*	*Response: 0x7296*
			Divide by 16 to remove the 4-bit fractional part: 0x7296 / 16 = 0x0729
0x601	6	22 09 30 01 29 07	Write the result to the calibration entry
0x601	8	22 10 10 04 01 09 30 00	Save the new calibration

Calibrate Electrical Zero

Because the encoder magnet is attached at an arbitrary angle with respect to the rotor magnets, the P3 needs to record the raw encoder value when the rotor is stalled under Phase A. The P3 uses this calibration value to set the electro-magnetic field orientation for optimal torque generation. This calibration is performed using Mode 12: Phase Voltage with Angle. To calculate an appropriate Phase Voltage (0x3010,4) for stalling the motor, try using 50% of the Continuous Current rating for your motor:

Phase Voltage (0x3010,4) = 2^15 * (Continuous_current * 0.5 * Rt / 2) / (Bus_voltage / sqrt(3))

Where:

- Continuous_current = value from motor's data sheet, in Amps
- Bus_voltage = nominal motor bus voltage, in Volts
- Rt = motor winding's terminal resistance value, in Ohms

Example for the Maxon EC-Max 22, model 283860:

Parameter	Value
Continuous_current	0.716 A
Bus_voltage	48 V
Terminal resistance	13.1 Ohms

Phase Voltage (0x3010,4) = 2^15 * (0.716 * 0.5 * 13.1 / 2) / (48 / sqrt(3)) = 2218 (0x08AA)

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On
0x601	6	22 40 60 00 0F 00	Go to Operation Enabled
0x601	5	22 60 60 00 0C	Set Mode: Phase Voltage with Angle
0x601	6	22 EA 60 00 00 00	Set Electrical Angle = 0 (Phase A stall under D-axis)
0x601	6	22 10 30 04 AA 08	Set the Phase Voltage
			(Wait 1 second)
0x601	4	40 12 30 01	Read the raw encoder value
0x581	*8*	*42 12 30 01 F8 0D 00 00*	*Response: 0x0DF8*
0x601	5	22 60 60 00 00	Set Mode: Idle
0x601	6	22 11 30 01 F8 0D	Write the result to the calibration entry
0x601	8	22 10 10 04 01 11 30 00	Save the new calibration

It is important to ensure that the motor shaft is free to spin during the calibration. Any load will invalidate the calibration. And if the

Calibrate Encoder Lag

The commutation code requires an accurate estimation of the motor's electrical angle in order to orient its electro-magnetic field for optimum torque. However, there are delays between reading the encoder, calculating the sine and cosine of the electrical angle, and generating the electro-magnetic field. Accounting for these delays requires predicting where the rotor will be at the moment the electro-magnetic field is updated. This predicted angle is a function of the most recent encoder position and the present velocity.

[TBD]

Configure PDOs

Receive Process Data Objects (RPDOs)

P3 can listen for up to 4 RPDOs, and each RPDO can contain up to 4 object dictionary entries (up to the CAN frame limit of 8 bytes). Each RPDO can be configured to apply its data immediately or on the nth SYNC message.

Transmit Process Data Objects (TPDOs)

P3 can transmit up to 4 TPDOs, and each TPDO can contain up to 4 object dictionary entries (up to the CAN frame limit of 8 bytes). Each TPDO can be configured to transmit on the nth SYNC message. The ability to transmit on data change is not yet supported.

Example 1: Torque Control with Position Feedback

MsgID	DLC	Data	Description
0x601	5	22 00 16 00 00	Set RPDO1 number of mapped objects to 0
0x601	5	22 01 16 00 00	Set RPDO2 number of mapped objects to 0
0x601	5	22 02 16 00 00	Set RPDO3 number of mapped objects to 0
0x601	5	22 03 16 00 00	Set RPDO4 number of mapped objects to 0
0x601	8	22 00 14 01 01 02 00 00	Set RPDO1 COB ID to 0x0201
0x601	5	22 00 14 02 FF	Set RPDO1 Rx type = Async (apply upon receipt)
0x601	8	22 00 16 01 10 00 71 60	Set first RPDO1 entry to 0x6071,0 (16-bit torque)
0x601	5	22 00 16 00 01	Set RPDO1 number of mapped objects to 1
0x601	5	22 00 1A 00 00	Set TPDO1 number of mapped objects to 0
0x601	5	22 01 1A 00 00	Set TPDO2 number of mapped objects to 0
0x601	5	22 02 1A 00 00	Set TPDO3 number of mapped objects to 0
0x601	5	22 03 1A 00 00	Set TPDO4 number of mapped objects to 0
0x601	8	22 00 18 01 81 01 00 00	Set TPDO1 COB ID to 0x0181
0x601	5	22 00 18 02 00	Set TPDO1 to transmit on every SYNC
0x601	8	22 00 1A 01 20 00 64 60	Set first TPDO1 entry to 0x6064,0 (32-bit pos)
0x601	5	22 00 1A 00 00	Set TPDO number of mapped objects to 1

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On
0x601	6	22 40 60 00 0F 00	Go to Operation Enabled
0x601	5	22 60 60 00 04	Set Mode: Profile Torque
0x201	2	64 00	Set Torque: 0x0064 = 100 or 1/10 of Peak Torque
0x080	0		SYNC
0x181	*4*	*01 02 03 04*	*Response: Position = 0x04030201*

Example 2: Cyclic Synchronous Velocity Control with Error, Status, and Position Feedback

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On

Example 3: Get Temperature Feedback at a Lower Rate

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On

Example 4: Using Dummy Entries

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On

Set Limits

Hardware vs. Application

The P3's amplifier and the actuator have absolute maximum hardware ratings which must not be exceeded in order to avoid permanent damage. Your application may also impose its own limits.

Hardware

[Table of Hardware Limits]

Application

[Table of Application Limits]

Tune Gains

[TBD]

Operate

Network Management (NMT) Messages

A CANopen device can be in one of four states:

Boot-up (0x00)

- Does not consume any messages
- Emits a single boot-up (heartbeat) message
- Automatically enters Pre-op or Operational state after initialization
  - Next state depends on the value of the device's Startup entry (0x1F80,0). 0 = Boot to Operational, 4 = Boot to Pre-Op.
  - If periodic heartbeats are configured (0x1017,0 > 0), then it will begin sending heartbeats with the new state.

Stopped (0x04)

- Communication limited to NMT and heartbeats only

Pre-operational (0x7F)

- All messages except RPDOs are allowed
- Send TPDO responses to SYNC messages

Operational (0x05)

- All messages are allowed

The device state is reported in the device's heartbeat.

NMT messages have a message ID of 0x000 and are heard by all nodes on the bus. To command a device to enter a new state, generate a CAN frame with:

- MsgID = 0x000 (NMT)
- DLC = 2
- Payload = [Commanded_State] [NodeID]
  - Commanded_State
    - 0x01 = Operational
    - 0x02 = Stopped
    - 0x80 = Pre-operational
    - 0x81 = Reset Node
    - 0x82 = Reset Communications
  - NodeID
    - 0 = All nodes
    - >0 = Single node specified by NodeID

Reset Node

- Performs a power-on reset (full reset)

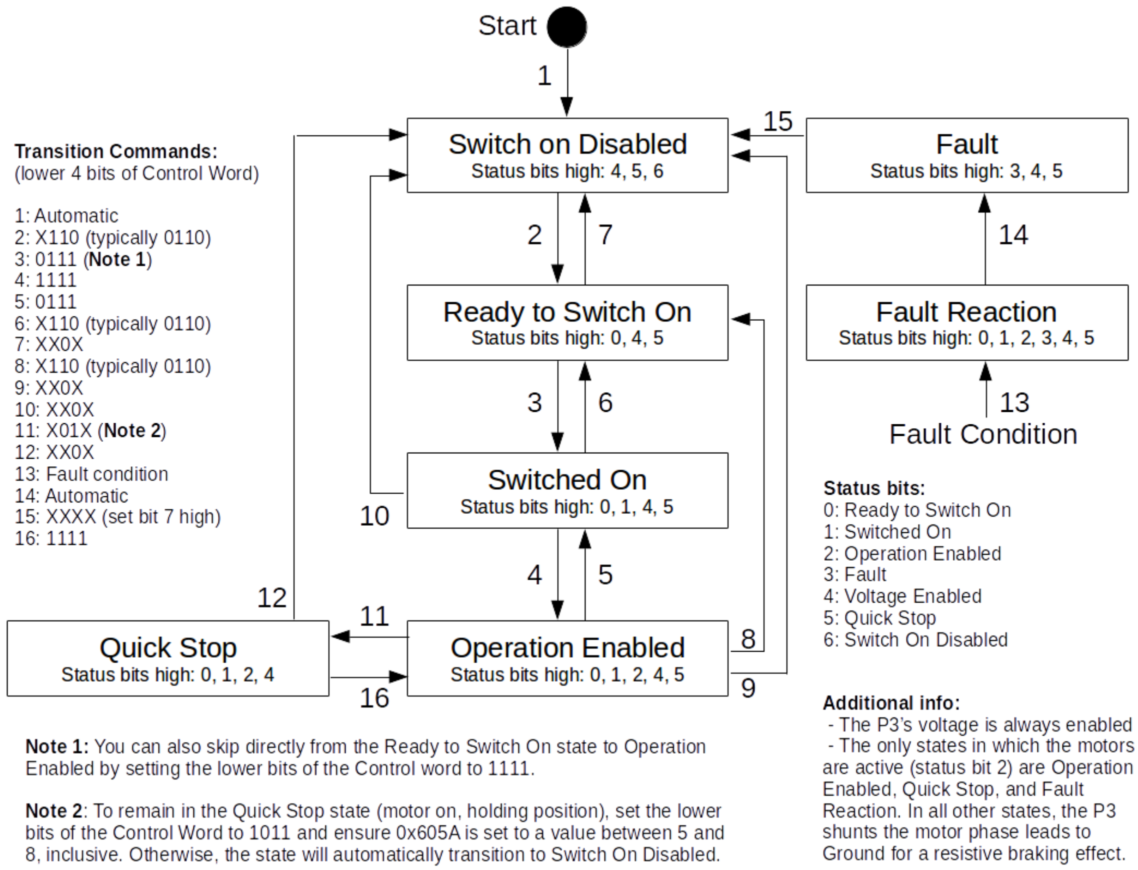
Reset Communications

- Resets the CAN device then enters Boot-up state

Note that the commanded state codes do not match the reported state codes- this is in accordance with the CiA-301 specification.

Navigating the State Machine

The CANopen drive control Finite State Automaton (FSA) is a state machine designed to provide safe and consistent operation across all drives in a system. Transitions between states are governed by the Control Word (0x6040,0) and any active faults (0x603F,0). The present state of the drive is available in the Status Word (0x6041,0).



CANopen Finite State Automaton (FSA)

Control Word

Control Word (0x6040,0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS					R	OMS	H	FR	OMS			EO	QS	EV	SO

Key	Meaning
MS	Manufacturer-Specific
R	Reserved (0)
OMS	Operation Mode Specific
H	Halt
FR	Fault Reset
EO	Enable Operation
QS	Quick Stop
EV	Enable Voltage
SO	Switch On

Status Word

Status Word (0x6041,0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS		OMS		ILA	TR	RM	MS	W	SOD	QS	VE	F	OE	SO	RTSO

Key	Meaning
MS	Manufacturer-Specific
OMS	Operation Mode Specific
ILA	Internal Limit Active
TR	Target Reached
RM	Remote
W	Warning
SOD	Switch On disabled
QS	Quick Stop
VE	Voltage Enabled
F	Fault
OE	Operation enabled
SO	Switched On
RTSO	Ready To Switch On

The status word (0x6041) is mappable to a PDO and contains bits (13,12,10) indicating the homing status while the Control Mode Display (0x6061) is Homing (6).

- 000 = homing

- 001 = interrupted
- 011 = success

Changing the control mode clears the mode-specific status bits.

Example: Use SDOs to Command a Torque for Node ID = 1

MsgID	DLC	Data	Description
0x601	6	22 40 60 00 86 00	Clear Faults, and go to Ready To Switch On
0x601	6	22 40 60 00 0F 00	Go to Operation Enabled
0x601	5	22 60 60 00 04	Set Mode: Profile Torque
0x601	6	22 71 60 00 64 00	Set Torque: 0x0064 = 100 or 1/10 of Peak Torque
0x601	6	22 71 60 00 00 00	Set Torque: 0
0x601	5	22 60 60 00 00	Set Mode: Idle

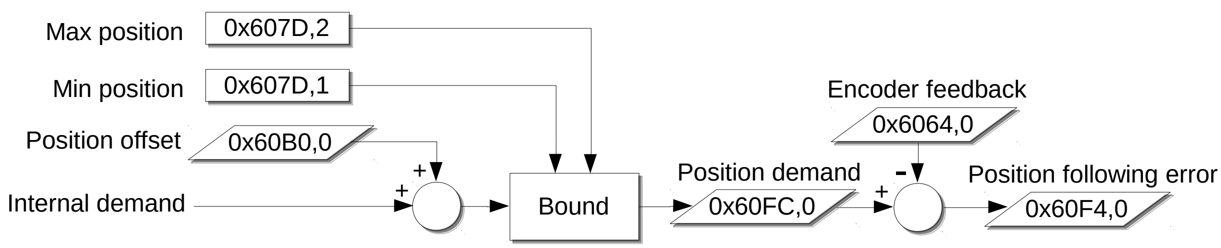
Drive Modes

The Control Mode Display (0x6061,0) is cleared to zero upon entering the state Operation Enabled.

The Control Mode (0x6060,0) can only be set to a non-zero value when the state is Operation Enabled.

(1) Profile Position

[CiA-402 diagram]



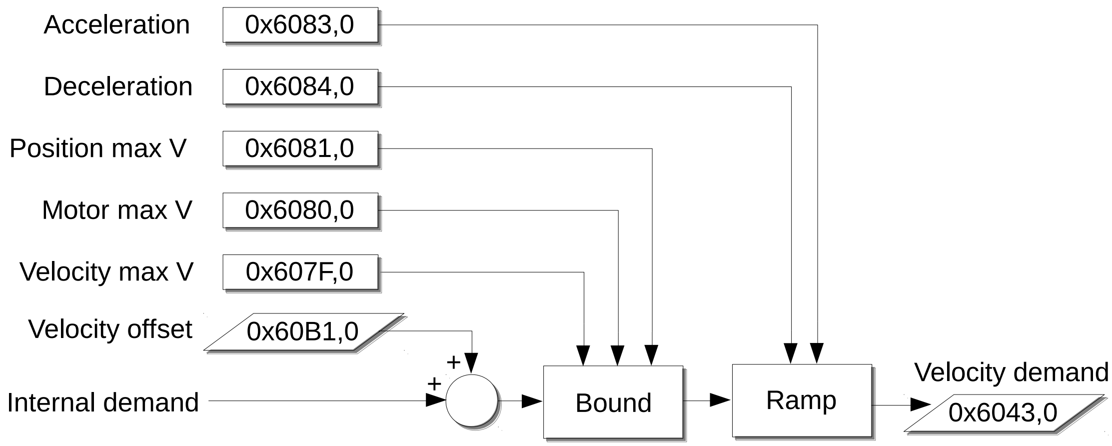
Following Error Calculation

[PDO configuration]

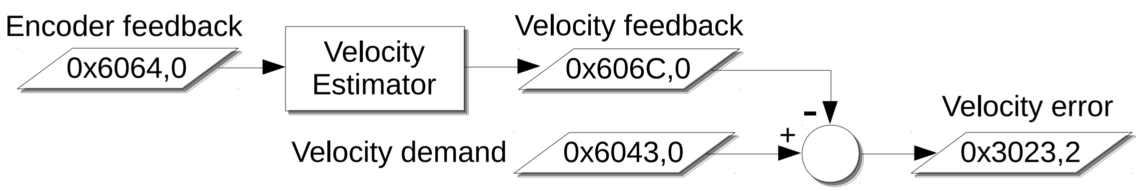
[Example Operation]

(3) Profile Velocity

[CiA-402 diagram]



Velocity Demand Limiting



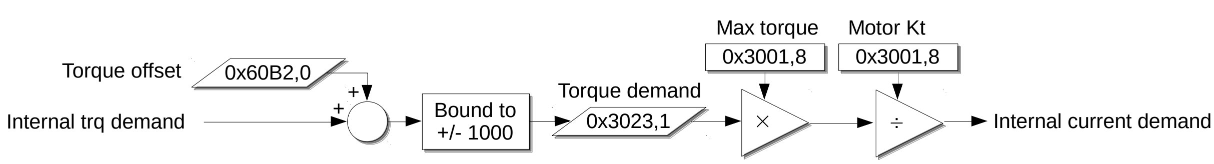
Velocity Error

[PDO configuration]

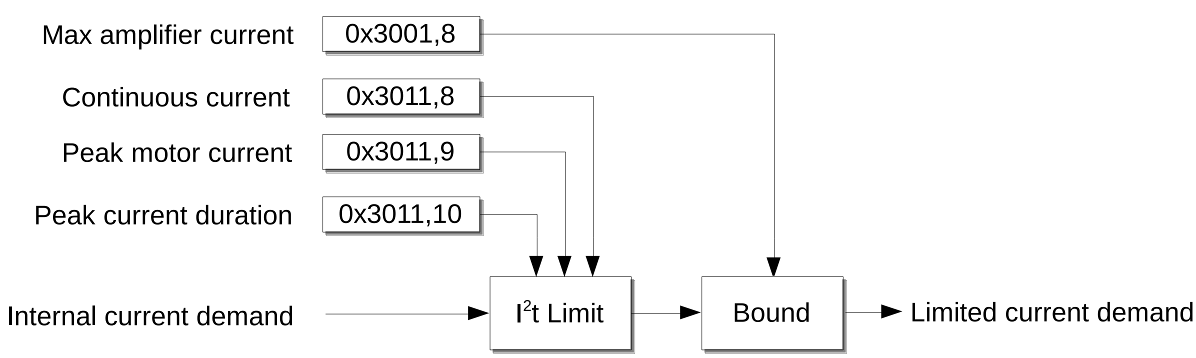
[Example Operation]

(4) Profile Torque

[CiA-402 diagram]



Torque-to-Current Conversion



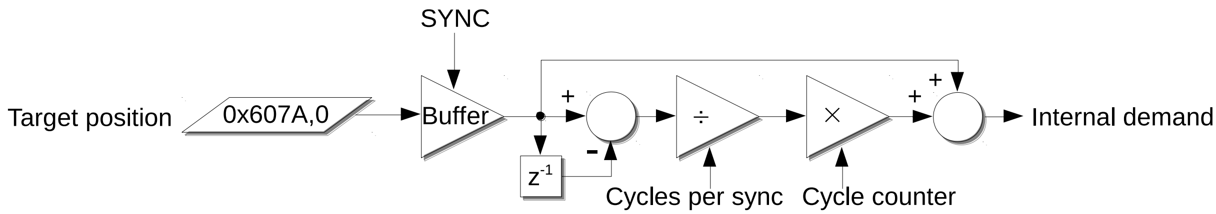
Current Limiting

[PDO configuration]

[Example Operation]

(6) Homing

(8) Cyclic Synchronous Position

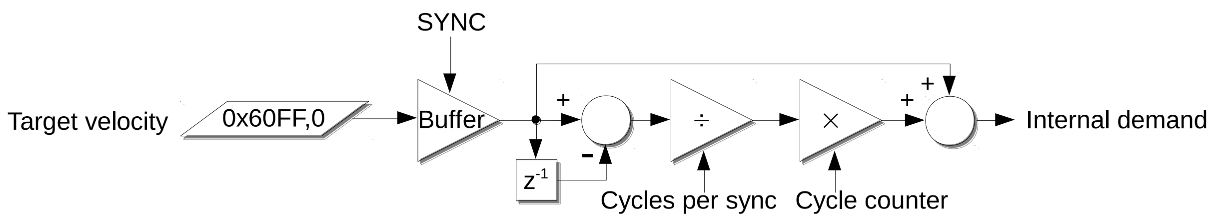


Cyclic Synchronous Position (CSP), Linear Interpolation

[PDO configuration]

[Example Operation]

(9) Cyclic Synchronous Velocity

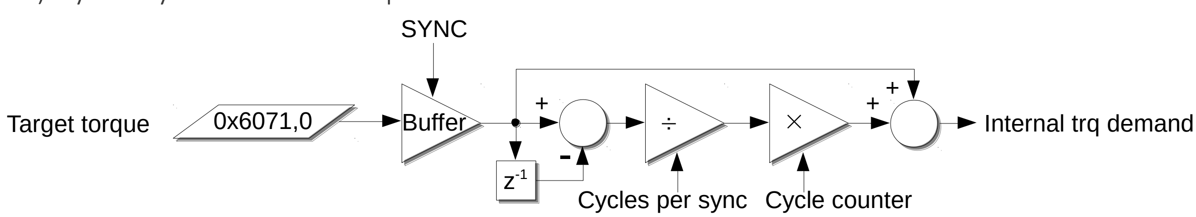


Cyclic Synchronous Velocity (CSV), Linear Interpolation

[PDO configuration]

[Example Operation]

(10) Cyclic Synchronous Torque



Cyclic Synchronous Torque (CST), Linear Interpolation

[PDO configuration]

[Example Operation]

(11) Cyclic Synchronous Torque with Angle

[CiA-402 diagram]

[PDO configuration]

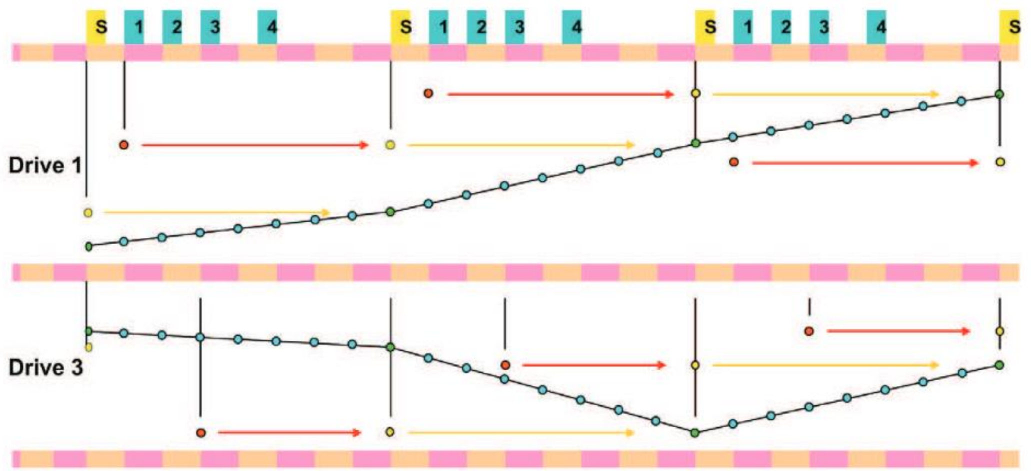
[Example Operation]

(12) Phase Voltage with Angle

[See Encoder Calibration section]

Understanding Cyclic Synchronous Control

CANmotion Bus Principle for Axis Synchronization



- (red dot) New setpoint (calculated by the controller) is sent to the drive (TPDO)
- (yellow dot) All drives take into account the new setpoint at reception of the Synchronous signal
- (green dot) New drive actual value is sent to the controller (RPDO)
- (blue dot) Intermediate setpoints are calculated inside the drive every 250 µs (Linear interpolation)

S — Synchronous signal

1 — Data exchange with drive 1

Warnings & Faults (EMCY)

Error bitfield (0x1001,0):

Bit	Meaning
0	Generic error. Plus, if any higher bit is set, this bit is also set.
1	Motor current error
2	Motor voltage error
3	Temperature error
4	Communication error
5	Unused
6	Unused
7	Motion control error

Manufacturer Status Register (0x1002,0)

Error history (0x1003,n)

Last error (0x603F,0)

In-Depth

I2T Power Limiting

The puck uses an I2t algorithm to ensure that the integral of the power dissipated by the motor in the form of thermal energy does not exceed its thermal limits. The algorithm uses three parameters in its thermal model:

1) Continuous Current – this nominal current that can flow through the motor is determined by the power it can dissipate continuously without exceeding its thermal limits.
2) Peak Current – a transient current above the Continuous Current level that can be tolerated.
3) Peak Current Time – the amount of time the system can tolerate operating at the Peak Current.

It is important to note that these parameters are heavily dependent on the thermal conductivity and thermal capacity of the frame in which the actuator is mounted and the surrounding air temperature.

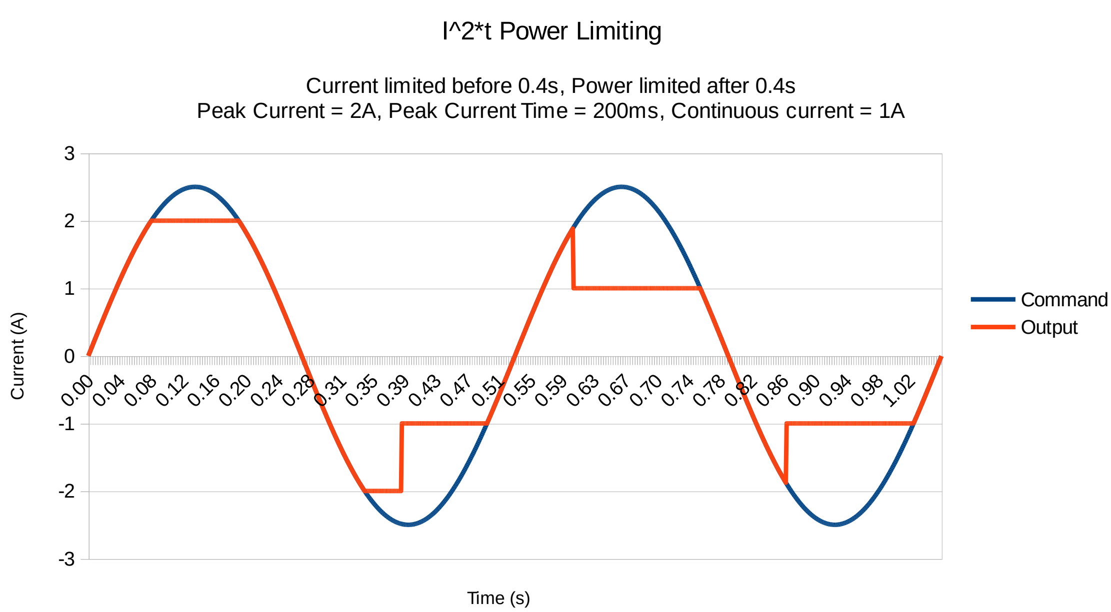
The excess energy limit of the system is calculated by:

$$Energy\_Limit = Peak\_Current * Peak\_Current\_Time$$

If the commanded current exceeds the Peak Current parameter, it is limited at Peak Current. The controller keeps a running sum of the system's excess energy by:

$$Energy = Energy + (Command^2 - Continuous^2) * dt$$

Whenever this Energy exceeds the Energy_Limit, the commanded current is limited to the Continuous Current parameter. This results in the following behavior:



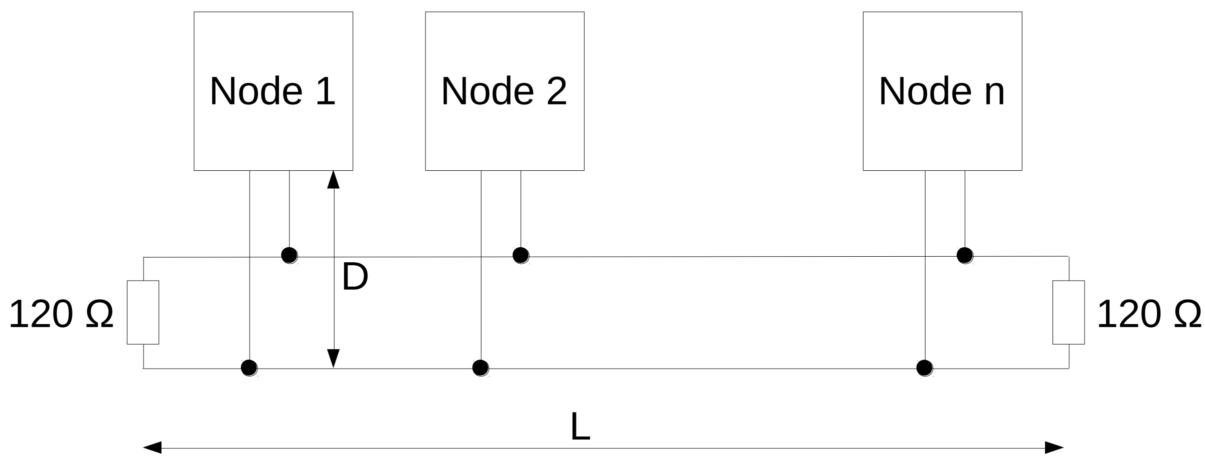
Velocity Calculation

The velocity feedback is derived from the encoder position feedback. Due to flicker and discretization of the encoder data and the relatively large unit of time for the velocity (cts/s) when compared to the encoder read rate (measured in µs), the Actual Velocity (0x606C,0) is both noisy and discretized. A filter is applied to the Actual Velocity with a cutoff frequency (Fc) that is proportional to the present velocity. This helps minimize discretization at low velocities while providing low control lag at higher velocities. Here is a table of Actual Velocity accuracy at various speeds, using independent measurements taken with a photo-tachometer:

Measured V (RPM)	Measured V (cts/s)	Actual Velocity (cts/s)	% Error
1000	25	1	8

CAN

Bus Topology and Termination



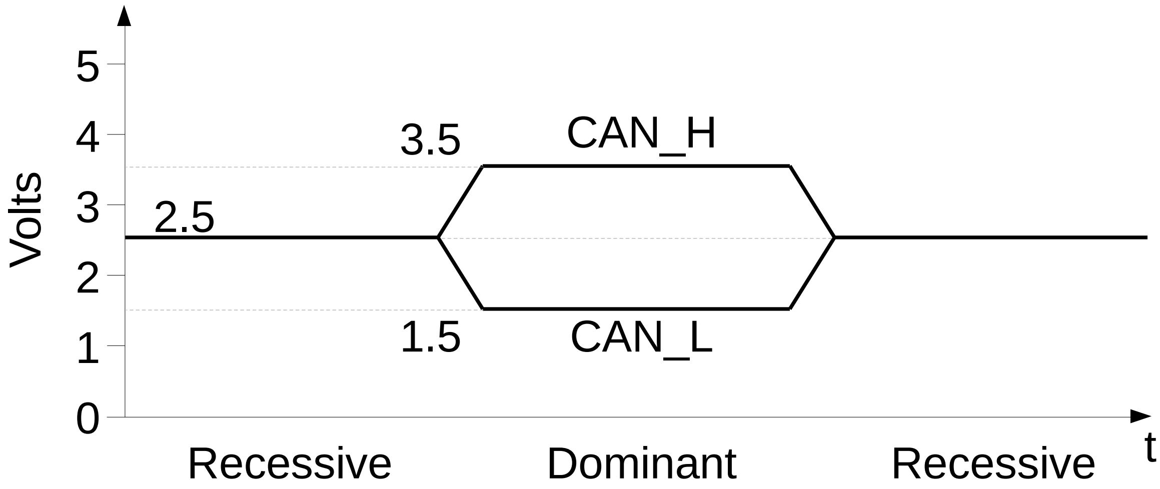
Bus Topology and Termination

The CAN bus must be terminated at each end with a 120-Ohm resistor.

The maximum bus length (L) for a 1 Mbps bus is 25 meters.

The maximum drop length (D) for a 1 Mbps bus is 30 cm.

Nominal Bus Voltage Levels



Nominal Bus Voltage Levels

Bit Rates vs. Line Lengths

Use the following recommended settings to configure the timing parameters of your CAN bus.

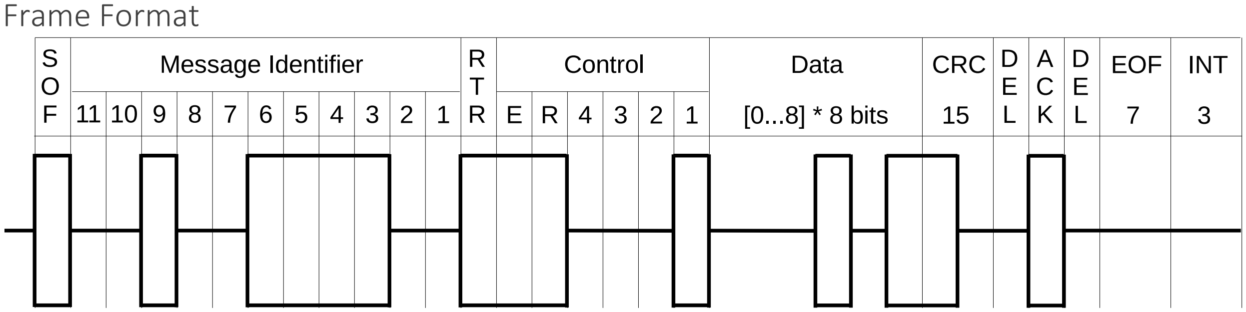
Rate (kbps)	Max L (m)	Bit Time (us)	TQ	Q (ns)	Sample Point (tq)
1000	25	1	8	125	6
800	50	1.25	10	125	8
500	100	2	16	125	14
250	250	4	16	250	14
125	500	8	16	500	14

TQ = Time Quanta

Q = Quantum duration

SJW = Synchronization Jump Width: 1 TQ

Frame Format



Frame Format

SOF = Start-of- Frame bit, dominant

Message Identifier = 11 bits (CAN 2.0A)

RTR = Remote Transmission Request bit: dominant = Data Frame, recessive = Request Frame

Control Field

E = Extended Frame: dominant = 11 bit identifier, recessive = 29-bit identifier

R = Reserved, always dominant

[4 3 2 1] = Data Length Code (DLC), specifies number of bytes in Data Field [0...8]

Data Field = 0 to 8 bytes of data

CRC = Cyclic Redundancy Check: 15-bit checksum of SOF, Message ID, RTR, Control, and Data fields

DEL = Acknowledgement Delimiter, always recessive

ACK = Acknowledgement bit, transmitted recessively. Any node which receives this message correctly will assert a dominant value here to indicate a successful transmission to the sending node.

EOF = End-of-Frame, a series of 7 recessive bits

INT = Intermission, a minimum of 3 bit times between successive frames

CANopen

CANopen Message Format

- Network Management (NMT)
  - MsgID = 0x000 (heard by all nodes)
  - DLC = 2
  - Payload = [Commanded_State] [NodeID]
    - Commanded_State
      - 0x01 = Operational
      - 0x02 = Stopped
      - 0x80 = Pre-operational
      - 0x81 = Reset Node
      - 0x82 = Reset Communications
    - NodeID
      - 0 = All nodes
      - >0 = Single node
- Heartbeat
  - MsgID = 0x700 | NodeID
  - DLC = 1
  - Payload = [State]
    - State
      - 0x00 = Boot-up
      - 0x04 = Stopped
      - 0x05 = Operational
      - 0x7F = Pre-operational
- SYNC
  - MsgID = 0x080
  - DLC = 0
- Emergency
  - MsgID = 0x080 | NodeID

    - DLC = 8
    - Payload = [ErrLow] [ErrHigh] [ErrRegister] [Mfg-Specific]…
        - [ErrLow] [ErrHigh] = 16-bit CANopen error code
        - [ErrRegister] = copy of OD 1001,00
        - [Mfg-Specific] = 0-5 bytes of error data (optional)
- Timestamp
    - MsgID = 0x100
    - DLC = 4
    - Payload = 32-bit timestamp value (LSB)
- SDO Write (Expedited) – Write this data to the Object Dictionary (OD)
    - MsgID = 0x600 | NodeID
    - DLC = 8
    - Payload = [0010 nn e s] [lowIdx] [highIdx] [subIdx] [data]…
        - 0010 = SDO Write
        - nn = number of bytes w/o data (iff s == 1)
        - e = Expedited
        - s = size indicated in nn
        - [lowIdx] [highIdx] = 16-bit OD index
        - [subIdx] = OD entry sub-index
        - [data] = 1-4 bytes of data (little-endian)
- SDO Write Ack – Data was written to the OD
    - MsgID = 0x580 | NodeID
    - DLC = 8
    - Payload = [0110 0000] [lowIdx] [highIdx] [subIdx] 0x00 0x00 0x00 0x00
- SDO Read – Read data from OD
    - MsgID = 0x600 | NodeID
    - DLC = 8
    - Payload = [0100 0000] [lowIdx] [highIdx] [subIdx] 0x00 0x00 0x00 0x00
- SDO Read Response (Expedited) – Here is the data you requested
    - MsgID = 0x580 | NodeID
    - DLC = 8
    - Payload = [0100 nn e s] [lowIdx] [highIdx] [subIdx] [data]…
        - 0100 = SDO Read
        - nn = number of bytes w/o data (iff s == 1)
        - e = Expedited
        - s = size indicated in nn
        - [lowIdx] [highIdx] = 16-bit OD index
        - [subIdx] = OD entry sub-index
        - [data] = 1-4 bytes of data (little-endian)
- Receive PDO – process data object to be received by the device
    - MsgID = 0x200 | NodeID, 0x300 | NodeID, etc.
    - DLC = n (byte count)
    - Payload = [data]…
        - [data] = little-endian data
- Transmit PDO – process data object to be transmitted by the device

- MsgID = 0x180 | NodeID, 0x280 | NodeID, etc.
- DLC = n (byte count)
- Payload = [data]…
  - [data] = little-endian data

Object Dictionary

Object Dictionary Key

Term	Definition
U08	Unsigned 8-bit integer
U16	Unsigned 16-bit integer
U32	Unsigned 32-bit integer
I08	Signed 8-bit integer
I16	Signed 16-bit integer
I32	Signed 32-bit integer
RO	Read-Only
RW	Read-Write

Communication

Index	SubIdx	Type	Access	Description
0x0002	0	I08	RW	Dummy entry, for PDO padding
0x0003	0	I16	RW	Dummy entry, for PDO padding
0x0004	0	I32	RW	Dummy entry, for PDO padding
0x0005	0	U08	RW	Dummy entry, for PDO padding
0x0006	0	U16	RW	Dummy entry, for PDO padding
0x0007	0	U32	RW	Dummy entry, for PDO padding
0x1000	0	U32	RO	Device Type: [16-bit extra info | 16-bit device profile]
0x1001	0	U08	RO	Error bitfield (see below)
0x1002	0	U32	RO	Manufacturer Status Register
0x1003	0	U08	RO	Error history, record count
0x1003	1-8	U32	RO	Error history, most recent error is at 0x1003,1
0x1005	0	U32	RW	Sync COB ID
0x1006	0	U32	RO	Sync period in microseconds (for producer)
0x1008	0	STR	RO	Mfg device name (4 chars)
0x1009	0	STR	RO	Mfg hardware version (4 chars)
0x1010	1	U32	RW	Save All, write 0x65766173 “SAVE”
0x1010	2	U32	RW	Save Communication parameters, write 0x65766173
0x1010	3	U32	RW	Save Application parameters, write 0x65766173
0x1010	4	U32	RW	Save single entry, write [Idx << 8 | SubIdx]
0x100A	0	STR	RO	Mfg software version (4 chars)
0x1014	0	U32	RW	Emergency COB ID (default = 0x80 + NodeID)
0x1015	0	U16	RW	Emergency inhibit time
0x1017	0	U16	RW	Heartbeat period in milliseconds
0x1018	1	U32	RO	Vendor ID

0x1018	2	U32	RO	Product code
0x1018	3	U32	RO	Revision number
0x1018	4	U32	RO	Serial number
0x1200	1	U32	RO	SDO receive COB ID, default = 0x600 + NodeID
0x1200	2	U32	RO	SDO transmit COB ID, default = 0x580 + NodeID
0x1F80	0	U32	RW	NMT startup (0 = Boot to Operational, 4 = Boot to Pre-op)

PDO Configuration

Index	SubIdx	Type	Access	Description
0x1400-0x1403	1	U32	RW	RPDO COB ID
0x1400-0x1403	2	U08	RW	RPDO Rx type: 0-240 = Apply on nth Sync, 254 = Mfg specific, 255 = Async (Apply upon receipt)
0x1600-0x1603	0	U08	RW	RPDO mapping, number of mapped objects (0-4)
0x1600-0x1603	1-4	U32	RW	RPDO mapping, bits 16-31: Index, bits 8-15: SubIdx, bits 0-7 bit length (8, 16, or 32 bits)
0x1800-0x1803	1	U32	RW	TPDO COB ID
0x1800-0x1803	2	U08	RW	TPDO Tx type: 0-240 = Tx on nth Sync, 254 = Mfg specific, 255 = Async
0x1800-0x1803	3	U16	RW	TPDO inhibit time in multiples of 100 uS (not implemented)
0x1800-0x1803	4	U08	RO	Unused
0x1800-0x1803	5	U16	RW	TPDO event timer in ms (not implemented)
0x1A00-0x1A03	0	U08	RW	TPDO mapping, number of mapped objects (0-4)
0x1A00-0x1A03	1-4	U32	RW	TPDO mapping, bits 16-31: Index, bits 8-15: SubIdx, bits 0-7 bit length (8, 16, or 32 bits)

States and Modes

Index	SubIdx	Type	Access	Description
0x6040	0	U16	RW	Control Word
0x6041	0	U16	RO	Status Word
0x605A	0	U16	RW	Quick stop option code. If value = [5, 6, 7, 8] when Quick Stop is commanded, remain in Quick Stop State (enabled, holding position), else transition to Switch On Disabled state.
0x6060	0	U08	RW	Set mode of operation (0 = Idle, 1 = Profile Position, 2 = Velocity, 6 = Homing, 8 = Cyclic Position, 9 = Cyclic Velocity)

0x6061	0	U08	RO	Read mode of operation
0x6502	0	U32	RO	Bitfield of supported drive modes

General feedback

Index	SubIdx	Type	Access	Description
0x603F	0	U16	RO	Last error (EMCY code), also copied to 0x1003 history
0x6041	0	U16	RO	Status Word
0x6064	0	I32	RW	Actual position (cts)
0x606C	0	I32	RO	Actual velocity (cts/sec), calculated from encoder history
0x6078	0	I16	RO	Actual current (mA), Q-axis current derived from sensor

Motor parameters

Index	SubIdx	Type	Access	Description
0x6076	0	I32	RW	Rated torque (mNm)
0x607E	0	I08	RW	User polarity (-1 or +1), sets the rotational direction for a positive torque/velocity
0x6080	0	I32	RW	No-load speed (cts/sec)
0x60C5	0	I32	RW	Max acceleration (cts/sec^2), to prevent damage
0x60C6	0	I32	RW	Max deceleration (cts/sec^2), to prevent damage
0x60EA	0	I16	RW	Electrical angle (-2^15 to 2^15-1), corresponding to -180 to +180 degrees. Writeable only in modes 11 & 12.
0x3011	1	I16	RW	Electrical zero (cts), the raw encoder reading when the rotor is stalled under Phase A (calibrated)
0x3011	2	I08	RW	Electrical polarity (-1 or +1), phase order calibration
0x3011	3	I08	RW	Pole count (pole pairs * 2)
0x3011	4	U16	RW	Torque constant (mNm/A)
0x3011	5	U16	RW	Terminal resistance (Ohms * 100)
0x3011	6	U16	RW	Terminal inductance (mH * 100)
0x3011	7	U32	RW	Rotor inertia (gcm^2 * 2^15)
0x3011	8	I16	RW	Max continuous current (mA)
0x3011	9	I16	RW	Peak current (mA)
0x3011	10	I16	RW	Peak current duration (ms)
0x3012	1	I16	RO	Raw encoder reading (cts)
0x3013	1	U32	RW	Encoder resolution (cts/rev)
0x3013	5	U32	RW	Encoder lag factor (calibrated gain), adjusts magnetic field to account for rotation since last encoder update

Amplifier parameters

Index	SubIdx	Type	Access	Description
0x607E	0	I08	RW	User polarity (-1 or +1), sets the rotational direction for a positive torque/velocity
0x6078	0	I16	RO	Actual current (mA), Q-axis current derived from sensor

0x3000	1	U16	RO	Sensed bus voltage (units TBD)
0x3000	2	U16	RO	Temperature (units TBD)
0x3001	1	U32	RW	PWM frequency (Hz), control rate = PWM freq / 5
0x3001	2	U16	RW	Dead time (ns)
0x3001	3	U16	RW	Minimum gate driver propagation delay (ns)
0x3001	4	U16	RW	Delta propagation delay (ns), max delay – min delay
0x3001	5	U16	RW	Maximum settling time of current sensor (ns), calibrated
0x3001	6	U16	RW	ADC sampling time (ns)
0x3001	7	U16	RW	ADC conversion time (ns)
0x3001	8	I16	RW	Maximum current (mA)
0x3001	9	I16	RW	Maximum phase voltage (mV), bus voltage / sqrt(3)
0x3022	1	U16	RO	Aux ADC 1 (0-4095, corresponding to 0-3.3 V)
0x3022	2	U16	RO	Aux ADC 2 (0-4095, corresponding to 0-3.3 V)

Current control parameters

Index	SubIdx	Type	Access	Description
0x3019	1	U16	RW	D-axis control bandwidth (Hz * 10)
0x3019	2	U16	RW	D-axis damping term (zeta * 256), default = 1.0 * 256
0x3019	3	U16	RW	Q-axis control bandwidth (Hz * 10)
0x3019	4	U16	RW	Q-axis damping term (zeta * 256), default = 1.0 * 256

Torque parameters (control/feedback)

Index	SubIdx	Type	Access	Description
0x6071	0	I16	RW	Target torque (-1000 to +1000), modes 4 & 10, per thousand of motor rated torque
0x3023	1	I16	RO	Internal torque demand (-1000 to +1000), after min/max limit, per thousand of motor rated torque
0x6077	0	I16	RO	Actual torque (-1000 to +1000), per thousand of motor rated torque, calculated from sensed current

Torque parameters (configuration)

Index	SubIdx	Type	Access	Description
0x60B2	0	I16	RW	Torque offset (-1000 to +1000), per thousand of motor rated torque, default = 0
0x6072	0	I16	RW	Max application torque (-1000 to +1000), per thousand of motor rated torque
0x6080	0	I32	RW	Torque controller max velocity (cts/sec)

Velocity parameters (control/feedback)

Index	SubIdx	Type	Access	Description
0x60FF	0	I32	RW	Target velocity (cts/sec), modes 3 & 9
0x6043	0	I32	RO	Internal velocity demand (cts/sec), after min/max limit

				and accel/decel ramp limit
0x606C	0	I32	RO	Actual velocity (cts/sec), calculated from encoder history
0x3023	2	I32	RO	Velocity error (cts/sec), demand - actual

Velocity parameters (configuration)

Index	SubIdx	Type	Access	Description
0x60B1	0	I32	RW	Velocity offset (cts/sec), default = 0
0x60B2	0	I16	RW	Torque offset (-1000 to +1000), default = 0
0x607F	0	I32	RW	Velocity controller max velocity (cts/sec)
0x6083	0	U32	RW	Max acceleration (cts/sec^2)
0x6084	0	U32	RW	Max deceleration (cts/sec^2)
0x3019	10	U16	RW	Control bandwidth (Hz * 10)
0x3019	11	U16	RW	Damping term (zeta * 256), default = 1.0 * 256
0x3019	12	I32	RW	Proportional gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]
0x3019	13	I32	RW	Integral gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]
0x3019	14	I32	RW	Derivative gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]

Position parameters (control/feedback)

Index	SubIdx	Type	Access	Description
0x607A	0	I32	RW	Target position (cts), modes 1 & 8
0x60FC	0	I32	RO	Internal position demand (cts), after min/max limit
0x6064	0	I32	RO	Actual position (cts)
0x60F4	0	I32	RO	Position following error (cts), demand - actual

Position parameters (configuration)

Index	SubIdx	Type	Access	Description
0x60B0	0	I32	RW	Position offset (cts), default = 0
0x60B1	0	I32	RW	Velocity offset (cts/sec), default = 0
0x60B2	0	I16	RW	Torque offset (-1000 to +1000), default = 0
0x607D	1	I32	RW	Position minimum (cts)
0x607D	2	I32	RW	Position maximum (cts)
0x6081	0	I32	RW	Position controller max velocity (cts/s)
0x6065	0	U32	RW	Position tracking warning window (cts)
0x6066	0	U16	RW	Position tracking warning timeout (ms)
0x2120	0	U32	RW	Position tracking fault window (cts)
0x2121	0	U16	RW	Position tracking fault timeout (ms)
0x3019	5	U16	RW	Control bandwidth (Hz * 10)
0x3019	6	U16	RW	Damping term (zeta * 256), default = 1.0 * 256
0x3019	7	I32	RW	Proportional gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]
0x3019	8	I32	RW	Integral gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]
0x3019	9	I32	RW	Derivative gain (acc32_t), [Sign | 16-bit Int | 15-bit Frac]

Cyclic synchronous configuration (applies to cyclic synchronous Torque/Velocity/Position)

Index	SubIdx	Type	Access	Description
0x60C2	1	U08	RW	Interpolation time period value
0x60C2	2	I08	RW	Interpolation time period scale (10^n), use -3 for ms

Faults and Warnings

Object Dictionary Entries

Error Description	EMCY Code	OD Warn	OD Fault	Type	Units	0x1001 bit	0x1002 bit	States	Notes
Watchdog reset	0x6010		x	U08	Boolean	0	0	All	Emitted after boot-up message if reset was due to watchdog
Parameter error	0x6320		x	U08	Boolean	0	1	All	Checks for out-of-range and valid configuration hash
Encoder feedback error	0x7320		x	U08	Boolean	0	2	All	Invalid encoder feedback. Set to 1 to generate EMCY message & fault when error occurs.
Encoder magnet distance	0x7321		0x2403,0	U08	Boolean	0	3	All	Set to 1 to generate EMCY message & fault when error occurs
Current limit active	0x2310	0x2111,0		U16	0.01 A	1	4	OpEn	User Continuous Current. Also requires 0x2110,0 U16 User Peak Current (0.01 A) and 0x2112,0 U16 User Peak Current Time (ms)
Short-circuit detected	0x2320		x	U08	Boolean	1	5	OpEn	Peak Current exceeded. Set to 1 to generate EMCY message & fault when error occurs.
Bus over voltage	0x3210		0x2384,6	U16	0.1 V	2	6	All	Amplifier Max Voltage
Bus under voltage	0x3220		0x2384,7	U16	0.1 V	2	7	All	Amplifier Min Voltage
Amplifier over temperature	0x4210		0x2384,9	U08	Dec C	3	8	OpEn	Amplifier Max Temperature
Motor over temperature	0x4310		0x220A,0	U08	Deg C	3	9	OpEn	Motor Max Temperature
CAN overrun (frames lost)	0x8110		0x2404,0	U08	Boolean	4	10	All	Set to 1 to generate EMCY message & fault
SYNC loss fault timeout	0x8180		0x2406,0	U16	ms	4	11	All	The drive will fault if it does not hear periodic SYNCs
Recovered from bus-off	0x8140		0x2405,0	U08	Boolean	4	12	All	Set to 1 to generate EMCY message & fault
Motor phasing error	0x7122		x	U08	Boolean	7	13	OpEn	Set to 1 to generate EMCY message & fault
Positive limit switch	0x7380		x	U08	Boolean	7	14	OpEn	Set to 1 to generate EMCY message & fault. Ignored in Idle/Homing modes.
Negative limit switch	0x7381		x	U08	Boolean	7	15	OpEn	Set to 1 to generate EMCY message & fault. Ignored in Idle/Homing modes.
Homing switch	0x7382		x	U08	Boolean	7	16	OpEn	Set to 1 to generate EMCY message & fault. Ignored in Idle/Homing modes.
Positive soft stop	0x7383	x		U08	Boolean	7	17	OpEn	Set to 1 to generate EMCY message

Negative soft stop	0x7384	x		U08	Boolean	7	18	OpEn		Set to 1 to generate EMCY message
Position wrapped	0x73A0		x	U08	Boolean	7	19	All		Set to 1 to generate EMCY message & fault
Velocity tracking warning	0x8411	0x606D,0		U16	cts/sec	7	20	OpEn		Velocity Tracking Warning Window. Also requires 0x606E,0 U16 Velocity Tracking Warning Timeout (ms)
Velocity tracking fault	0x8414		0x2104,0	U16	cts/sec	7	21	OpEn		Velocity Tracking Fault Window. Also requires 0x2105,0 U16 Velocity Tracking Fault Timeout (ms)
Velocity limit active	0x8418	0x6046,1		U32	cts/sec	7	22	OpEn		Max Velocity Limit. Use 0x6046,2 U32 for Min Velocity Limit (same EMCY message)
Acceleration limit active	0x8480	0x60C5,0		U32	cts/sec/sec	7	23	OpEn		Acceleration Limit. Use 0x60C6,0 U32 for Deceleration Limit (same EMCY message)
Position tracking warning	0x8611	0x6065,0		U32	cts	7	24	OpEn		Position Tracking Warning Window. Also requires 0x6066,0 U16 Position Tracking Warning Timeout (ms)
Homing error	0x8613		x	U08	Boolean	7	25	OpEn		Failed to achieve minimum homing range. Set to 1 to generate EMCY message & fault when error occurs.
Position tracking fault	0x8614		0x2120,0	U32	cts	7	26	OpEn		Position Tracking Fault Window. Also requires 0x2121,0 U16 Position Tracking Fault Timeout (ms)

Important Notes

Only the **BOLD** errors have been implemented.

Warnings are auto-reset when the condition clears.

Faults are latched, amplifier transitions to fault state, motor voltage is disabled.

Transitioning from Fault to Switch On Disabled clears all latched faults, but if the fault condition still exists, it will re-trigger the fault.

EMCY message format: MsgID=0x80|NodeID, DLC=8, [CodeLow, CodeHigh, 0x1001, 0, 0, 0, ResetCodeLow, ResetCodeHigh].

The last emitted (non-zero) EMCY code is stored in 0x603F,0 U16 and 0x1003,1 U32 (lower 16 bits), and its corresponding bit is set in 0x1002,0 and 0x1001,0.

When an error condition is cleared, an EMCY message is emitted with code = 0x0000, an updated 0x1001, and the code of the corresponding cleared error.

To disable a warning/fault, set its value to zero.

Detailed Descriptions

Encoder magnet distance

If enabled (set to 1), and if the magnetic field generated by encoder magnet is too weak or too strong for the encoder sensor, a fault will occur. This may be due to improper spacing between the encoder magnet and the sensor, or the magnet may have lost its field strength due to excessive heat, or the magnet may be the wrong type.

Current limit active

When the i2t power limit is active, this warning will be emitted. This warning will be reset automatically when the i2t power limit becomes inactive.

Bus over voltage

If enabled (set > 0), and if the bus voltage exceeds the amplifier max voltage for more than 450 ms, a fault will occur.

Bus under voltage

If enabled (set > 0), and if the bus voltage falls below the amplifier min voltage for more than 450 ms, a fault will occur.

Amplifier over temperature

If enabled (set > 0), and the 18 Hz low-pass filtered temperature exceeds the max temperature, a fault will occur.

CAN overrun (frames lost)

If enabled (set to 1), and the CAN input buffer of the actuator is overrun, a fault will occur. The actuator will automatically attempt to re-initialize its CAN subsystem to re-enable the receipt of CAN messages, but one or more CAN message may be lost.

SYNC loss fault timeout

If enabled (set > 0), and the time elapsed since the last SYNC message exceeds the timeout, a fault will occur.

Recovered from bus-off

If enabled (set to 1), and the actuator recovers from a bus-off condition, a fault will occur. The actuator will automatically attempt to re-initialize its CAN subsystem to re-enable the receipt of CAN messages, but one or more CAN message may be lost.

Velocity tracking warning

If enabled (set > 0), and the velocity tracking error exceeds the warning threshold for longer than the warning timeout, this warning will be emitted. This warning will be reset automatically when the velocity tracking error falls below the warning threshold.

Velocity tracking fault

If enabled (set > 0), and the velocity tracking error exceeds the fault threshold for longer than the fault timeout, a fault will occur.

Velocity limit active

If enabled (set either min or max limit > 0), and the actuator velocity does not fall between the min/max limits, this warning will be emitted. This warning will be reset automatically when the actuator velocity is between the min/max limits.

Position tracking warning

If enabled (set > 0), and the velocity tracking error exceeds the warning threshold for longer than the warning timeout, this warning will be emitted. This warning will be reset automatically when the velocity tracking error falls below the warning threshold.

Position tracking fault

If enabled (set > 0), and the velocity tracking error exceeds the fault threshold for longer than the fault timeout, a fault will occur.

PID Control

Units, scaling, acc32_t, step response, direct vs. bandwidth & zeta

[TBD]

Homing

Support modes: 37 (home on immediate position)

[TBD]

Other Resources

Beckhoff TwinCAT

[CANopen Electronic Data Sheet (EDS) file for TwinCAT]

Data Sheet

[Link to Marketing Data Sheet]

Dimensional Drawing

[Link to Dimensional Drawing]

Electrical Pinout

[Link to electrical pinout]

Electrical Specifications & Limits

Designing an Interface Board

[Link to interface board document]

System Grounding

4-wire: [CAN_H, CAN_L, Motor+, GND] DC/DC @ interface board, reverse voltage diode, tie grounds @ DC/DC

5-wire: [CAN_H, CAN_L, Logic+, Motor+, GND] Keep logic/encoder alive when motor power is cut

6-wire: [CAN_H, CAN_L, Logic+, Logic GND, Motor+, Motor GND] Tie GNDs at power supply

Earth/frame bypass cap, braid across bearings, Earth at wall

CAN galvanic/opto-isolation, ground reference, different phases at wall

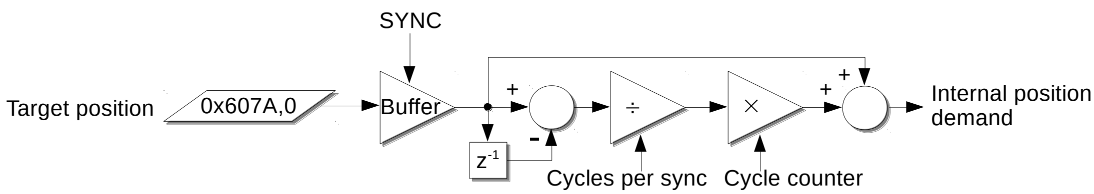
Heat Dissipation

Sil-pad between P3 and metal frame. Monitor the amplifier temperature.

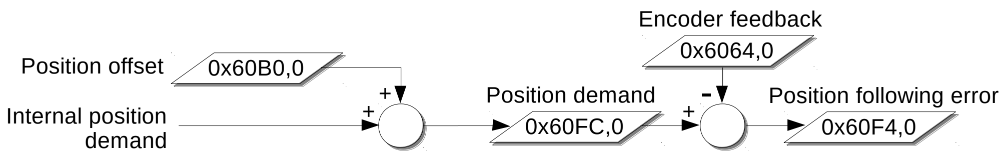
Environmental Operating Conditions

Thermal, Humidity, Dust, Liquid, Radiation, Vibration

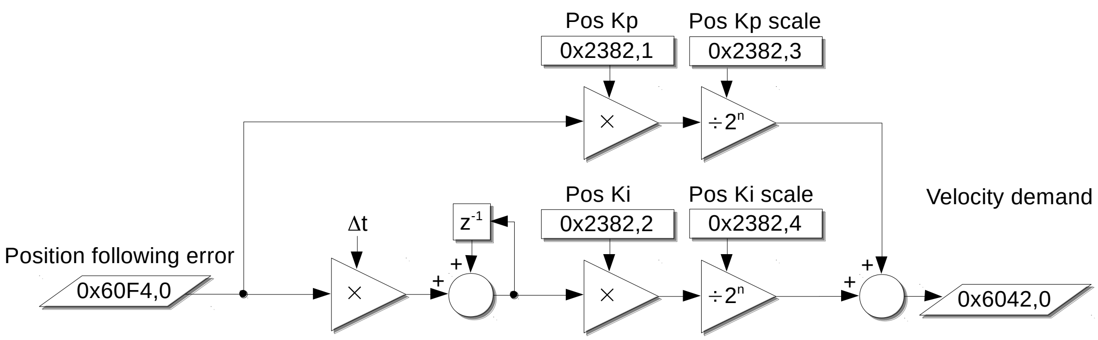
[TBD]



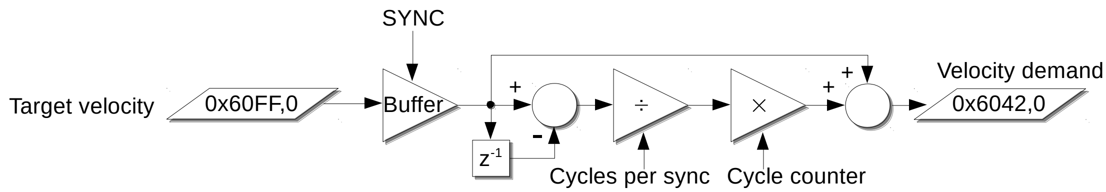
Cyclic Synchronous Position (CSP), Linear Interpolation



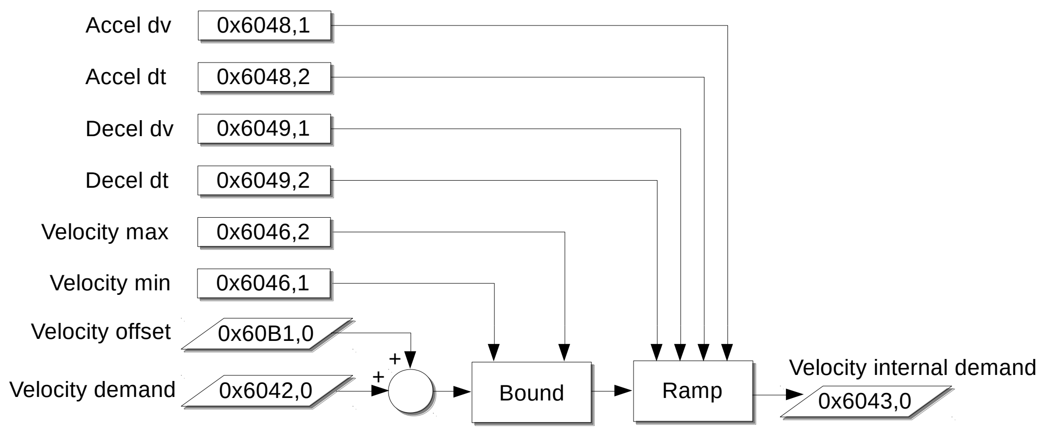
Following Error Calculation



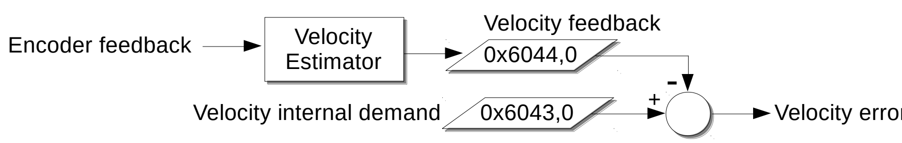
Position PI Control



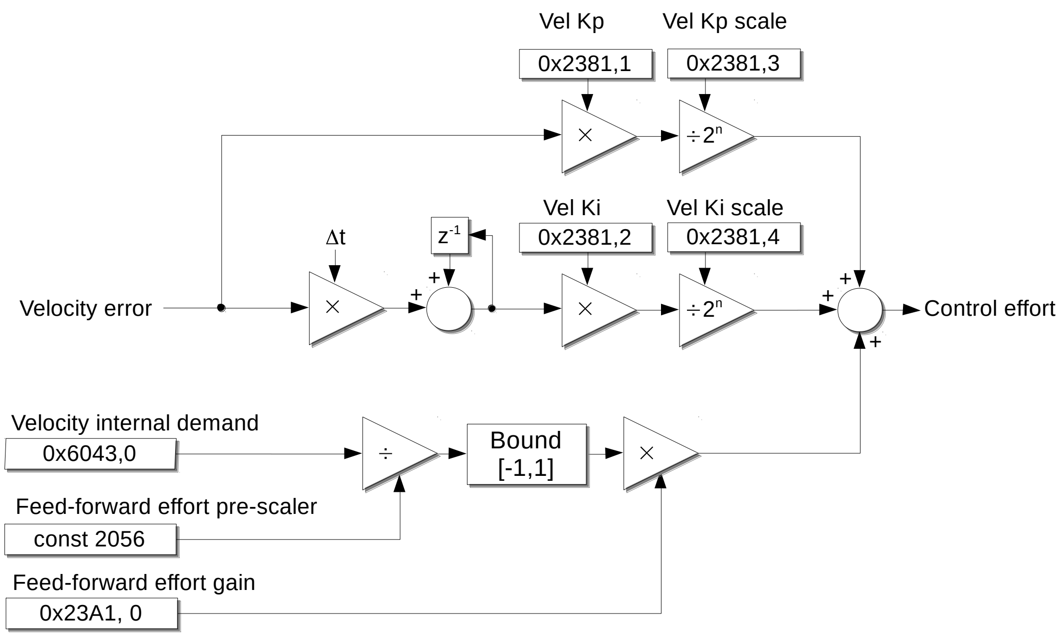
Cyclic Synchronous Velocity (CSV), Linear Interpolation



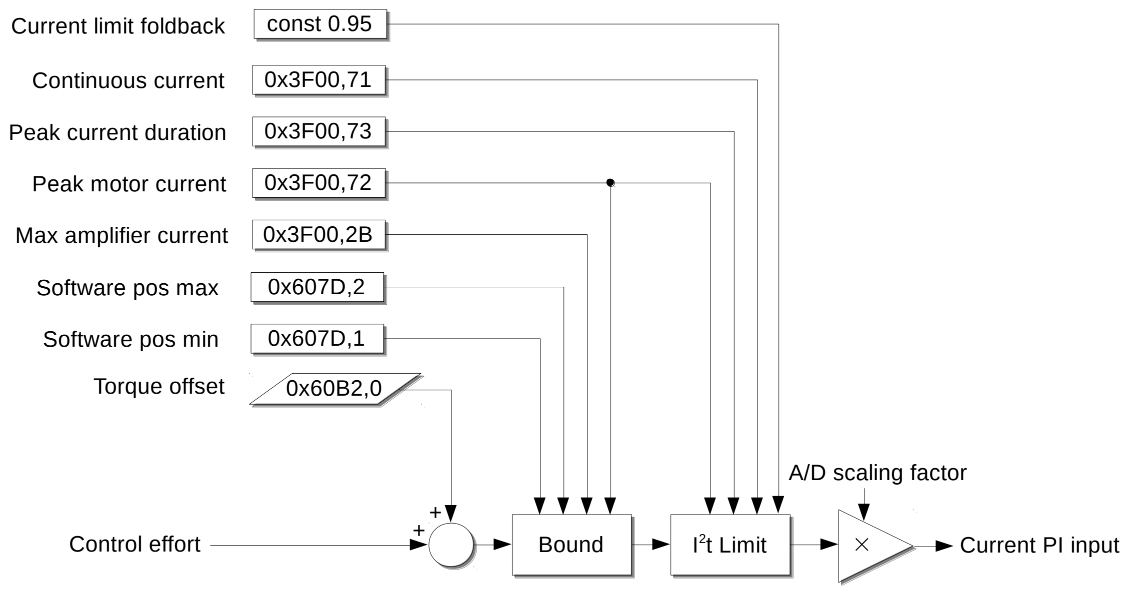
Velocity Demand Limiting



Velocity Error



Velocity PI Control
with Proportional Feed-Forward



Current Input Limits

Document Change History

Revision	Date	Description	Originator
AA	2016-01-18	Original Issue	B. Zenowich
AB	2016-01-18	Fixed Scaled Feedback OD Index	B. Zenowich
AC	2016-01-19	Added more detail to Example Operation, added change history.	B. Zenowich
AD	2016-01-19	Restored change history after accidental deletion.	B. Zenowich
AE	2016-02-10	Added documentation for the "set mode" entry in the OD (index = 0x6060).	C. Woodall
AF	2016-02-24	Added documentation for the "Error" entry in the OD (index = 0x1001) Added example of how to check for an over-temperature fault.	C. Woodall
AG	2016-03-01	Added information on CANopen OD entry 0x3F00 for accessing BarrettCAN properties Added examples for using property 0x3F00	C. Woodall
AH	2016-04-13	Added Appendix B on supporting multi-field TPDOs	C. Woodall
AI	2016-04-22	Added safe position information.	C. Woodall
AJ	2016-05-31	Added Trapezoidal Trajectory Mode. Added over current error to Error register.	C. Woodall
AK	2016-06-21	Updated some fields which were marked with the wrong Read-Write permissions	C. Woodall
AL	2017-03-24	Added Target Velocity, Target Position, and more documentation on the BarrettCAN Link including an example of writing to Auto-Homing and Saving the Change To EEPROM	C. Woodall

AM	2017-07-19	Added LOAD, control word, status word, quick stop option code, raw position command, raw position feedback, actual current feedback, bitfield of supported drive modes. Updated 0x6060 modes to match DSP-402 standard. Corrected velocity target units. Added a Notes section, Finite State Automaton (FSA) flow chart, and a "How it was tested" section. Reorganized Appendix B: Examples	B. Zenowich
AN	2017-07-20	Added Motor Is Homed flag (0x2402,0)	B. Zenowich
AO	2017-09-21	Added new Velocity, CSV, CSP modes. Added Faults/Warnings/EMCY codes.	B. Zenowich
AP	2017-09-22	Corrected Control Mode data type	B. Zenowich
AQ	2017-09-27	Added SYNC loss fault timeout	B. Zenowich
AR	2017-11-03	Added velocity feedforward parameters	C. Woodall
AS	2017-12-08	Updated control diagrams to match firmware r147. Current feedback range changed from rated (peak) current to continuous current. Removed Appendix B- Example CAN messages.	B. Zenowich C. Woodall
AT	2018-01-09	Added detailed fault descriptions, a note about temperature faults, and Appendix B- Algorithm Descriptions. Matches firmware r147.	B. Zenowich
AU	2018-11-20	Updated to match rewritten P3 firmware.	B. Zenowich
AV	2019-10-22	Added position control and wxp3 software.	B. Zenowich