

# P3 CANopen JPL Manual

## 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	2016-08-30	Adding better support for DS-402	C. Woodall
AM	2016-09-01	Correction to Object Dictionary Table adding entry for 0x607A and 0x60FF	C. Woodall

### Supports:

- Network Management (NMT) messages
- Heartbeat producer
- Expedited Service Data Object (SDO) messages
- Byte-level Receive/Transmit Process Data Objects (RPDOs/TPDOs)
- Sync messages for RPDOs/TPDOs
- Up to 4 RPDOs and 4 TPDOs
- Up to 4 configurable 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

### Does not yet support:

- Emergency messages
- 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
- Remote Transmit Request (RTR) messages
- Multiple/dynamic SDO COB IDs
- NMT master discovery
- TPDO inhibit timers
- TPDO event timers
- 29-bit CAN 2.0B identifiers
- Object Dictionary “dummy” entries

### Boot Sequence:

- Initialize DSP.
- Operates in Barrett CAN mode.
- Wait for NMT state message (typ. Pre-op or Operational).
- Initialize CANopen subsystem.
- Transmit heartbeat message.
- Enter specified state.

### States

- Boot-up
  - Does not consume any messages
  - Only able to produce the boot-up message
  - Enters BarrettCAN mode for configuration and waits for NMT messages.
  - ~~Automatically enters Pre-op state after initialization and boot-up transmission~~
- Stopped

- Communication limited to NMT and heartbeats only
- Pre-operational
  - All messages except PDOs are allowed
  - Note: How should we respond to SYNC messages in this state?
- Operational
  - All messages are allowed
- Reset Node
  - Performs a power-on reset (full reset)

See Figure 1 for a state transition Diagram

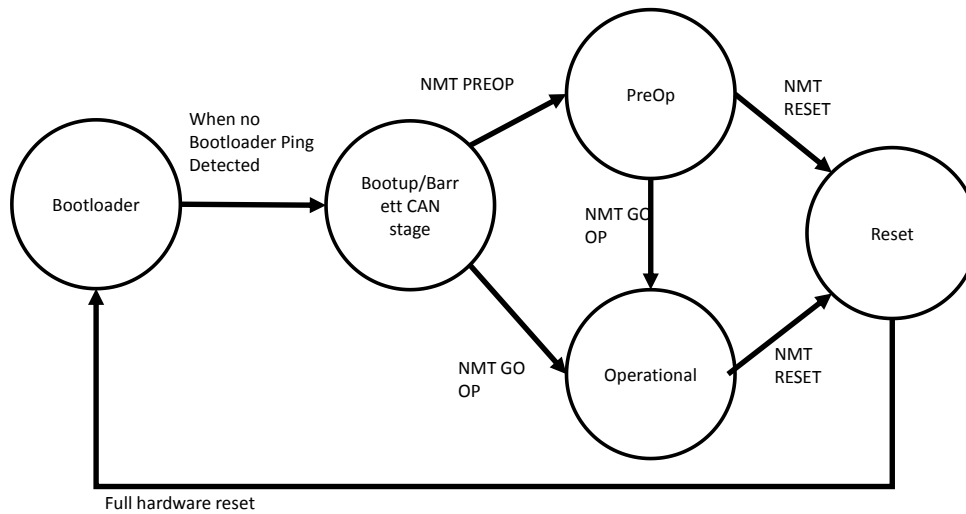


Figure 2:State transition diagram

**Object Dictionary:**

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

Index	SubIdx	Type	Access	Description
0x1000	0	U32	RO	Device Type low16=profile, high16=adddt'l info
0x1001	0	U08	RO	Error (Bit 0 = Generic, Bit 1 = Current, Bit 3 = Temperature)
0x1005	0	U32	RW	Sync COB ID
0x1006	0	U32	RO	Sync period in microseconds (iff producer)
0x1008	0	CHAR[4]	RO	Mfg device name (4 chars)
0x1009	0	CHAR[4]	RO	Mfg hardware version (4 chars)
0x1010	0	U08	RO	Save, number of entries = 1
0x1010	1	U32	RW	Save All (write 0x65766173 "SAVE")
0x100A	0	CHAR[4]	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	0	U08	RO	Identity, number of entries = 6
0x1018	1	U32	RO	Vendor ID
0x1018	2	U32	RO	Product code
0x1018	3	U32	RO	Revision number
0x1018	4	U32	RO	Serial number
0x1018	5	U32	RW	Machine code
0x1018	6	U32	RW	Role code
0x1200	0	U08	RO	SDO parameters, number of entries = 2
0x1200	1	U32	RO	SDO receive COB ID, default = 0x600 + NodeID
0x1200	2	U32	RO	SDO transmit COB ID, default = 0x580 + NodeID
0x1400-0x1403	0	U08	RO	RPDO parameters, number of entries = 2
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	0	U08	RO	TPDO parameters, number of entries = 5
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
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)
0x2201	0	I16	RO	Bus voltage returned as a Q8.8 signed integer in units of Volts.
0x2202	0	I16	RO	Ambient temperature returned as a Q8.8 signed integer in units of deg. C.
0x2A01	0	U16	RO	Read ADC1 as a 16-bit unsigned integer. NOT IMPLEMENTED YET
0x2A01	1	U16	RO	Read ADC2 as a 16-bit unsigned integer. NOT IMPLEMENTED YET
0x2A10	0	U8	RW	Set DIO[0..1] Input or Output. NOT IMPLEMENTED YET
0x2A10	1	U8	RW	Set DIO0 Input or Output masks bit 0 NOT IMPLEMENTED YET
0x2A10	2	U8	RW	Set DIO1 Input or Output masks bit 1 NOT IMPLEMENTED YET
0x2A11	0	U8	RO	Read Set DIO[0..1] NOT IMPLEMENTED YET
0x2A11	1	U8	RO	Read DIO0 masks bit 0 NOT IMPLEMENTED YET
0x2A11	2	U8	RO	Read DIO1 masks bit 1 NOT IMPLEMENTED YET
0x2A12	0	U8	RW	Set DIOs. NOT IMPLEMENTED YET (only works on outputs)
0x2A12	1	U8	RW	Set DIO0 masks bit 0. NOT IMPLEMENTED YET (only works on outputs)
0x2A12	2	U8	RW	Set DIO1 masks bit 1. NOT IMPLEMENTED YET (only works on outputs)
0x3F63	0	I32	RW	Set actual position. Write to 0 to zero position. Changes the actual position to the value written to 0x2280.
0x3F60	0	I8	RW	Default mode to boot to after going operational.
0x3F00	0-127	U32	RO	BarrettCAN Property Access. The sub-index maps to the BarrettCan Property. For example, sub-index 20 (0x14) will read the THERM property and sub-index 50 (0x32) will read the DP parameter. This can be used to read limits which have not been moved to direct CANOpen support yet.
0x603F	0	U16	RO	Error Codes. See Table ? NOT IMPLEMENTED YET
0x6040	0	U16	RW	Control Word. See Table ? NOT IMPLEMENTED YET
0x6041	0	U16	RO	Status Word. See Table ? NOT IMPLEMENTED YET
0x6060	0	I08	RW	Set mode of operation (0 = IDLE, 1 = POSITION, 3 = profile velocity mode, 4 = Torque mode, 5 = RESERVED, 6= Homing Mode, 7 = Barrett Trapezoidal Trajectory Mode
0x6061	0	I08	RO	Read mode of operation
0x6062	0	I32	RO	Position Demand Value , internal units (encoder counts at output shaft)
0x6063	0	I32	RO	Position actual internal value, internal units (encoder

				counts at output shaft)
0x6064	0	I32	RO	Position actual value, internal units (encoder counts at output shaft)
0x6069	0	I32	RO	Velocity sensor actual value. Increments per second
0x606B	0	I32	RO	Velocity demand value.mjn Radians per second Q16.16 at output shaft.
0x606C	0	I32	RO	Velocity sensor actual value. Radians per second Q16.16
0x606F	0	U32	RW	Digital Outputs (mapping undefined at the moment. NOT IMPLEMENTED YET.
0x6074	0	I16	RW	Demanded torque (motor effort).
0x607A	0	I32	RW	Profile target position (encoder count increments)
0x60FF	0	I32	RW	Velocity target position (radians per second)
0x6083	0	I32	RW	Profile acceleration
0x6084	0	I32	RW	Profile deceleration
0x1F80	0	U32	RW	NMT startup (0 = Boot to Operational, 4 = Boot to Pre-op)

### Example configuration for Node 1:

<MessageID> [Data Length Code] byte0 byte1... (all in hexadecimal)

# Go Pre-op (switches from BarrettCAN to CANopen,

# PDOs not processed in Pre-op)

<0x000> [2] 80 01 (NMT)

<0x701> [1] 7f (heartbeat)

# Set Sync COB ID to 0x080

<0x601> [8] 22 05 10 00 80 00 00 00 (set)

<0x581> [8] 60 05 10 00 80 00 00 00 (ack)

# Set Heartbeat MS to 0x03E8 = 1000 ms

<0x601> [6] 22 17 10 00 E8 03 (set)

<0x581> [8] 60 17 10 00 E8 03 00 00 (ack)

# Set RPDO\_1 COB ID (0x1400,1) to 0x201

<0x601> [8] 22 00 14 01 01 02 00 00 (set)

<0x581> [8] 60 00 14 01 01 02 00 00 (ack)

# Set RPDO\_1 Rx type (0x1400,2) to 0xFF (async, process immediately,

# do not wait for SYNC)

<0x601> [5] 22 00 14 02 FF (set)

<0x581> [8] 60 00 14 02 FF 00 00 00 (ack)

# Set RPDO\_1 first mapping (0x1600,1) to 0x607A,0 (position command)

<0x601> [8] 22 00 16 01 10 00 7A 60 (set)

<0x581> [8] 60 00 16 01 10 00 7A 60 (ack)

# Set RPDO\_1 mapping count (0x1600,0) to 1

<0x601> [5] 22 00 16 00 01 (set)

<0x581> [8] 60 00 16 00 01 00 00 00 (ack)

# Set TPDO\_1 COB ID (0x1800,1) to 0x181

<0x601> [8] 22 00 18 01 81 01 00 00 (set)

```

<0x581> [8] 60 00 18 01 81 01 00 00 (ack)
21
# Set TPDO_1 Tx type (0x1800,2) to 0 (Tx on every SYNC)
<0x601> [5] 22 00 18 02 00 (set)
<0x581> [8] 60 00 18 02 00 00 00 00 (ack)

# Set TPDO_1 Inhibit time (0x1800,3) to 0 (unused)
<0x601> [6] 22 00 18 03 00 00 (set)
<0x581> [8] 60 00 18 03 00 00 00 00 (ack)

# Set TPDO_1 Event timer (0x1800,5) to 0 (unused)
<0x601> [6] 22 00 18 05 00 00 (set)
<0x581> [8] 60 00 18 05 00 00 00 00 (ack)

# Set TPDO_1 first mapping (0x1A00,1) to 0x6063,0 (position feedback)
<0x601> [8] 22 00 1A 01 10 00 63 60 (set)
<0x581> [8] 60 00 1A 01 10 00 63 60 (ack)

# Set TPDO_1 mapping count (0x1A00,0) to 1
<0x601> [5] 22 00 1A 00 01 (set)
<0x581> [8] 60 00 1A 00 01 00 00 00 (ack)

# Save object dictionary changes to EEPROM (write "SAVE" to 0x1010,1)
<0x601> [8] 22 10 10 01 73 61 76 65 (set)
<0x581> [8] 60 10 10 01 01 00 00 00 (ack, reading this object returns 1 =
manual write support)

# Reboot node 1
<0x000> [2] 81 01 (NMT)

```

#### Example operation:

```

# Apply power
# Wait at least 5 ms for P3 to boot up
# Go Operational (switches from BarrettCAN to CANopen, RPDOs and TPDOs
enabled)
<0x000> [2] 01 00 NMT - Go Operational, all nodes

# Initial heartbeats
<0x701> [1] 05 Puck 1 responds with its initial heartbeat
<0x702> [1] 05 Puck 2 responds with its initial heartbeat

# Set control mode to position (0x6060, 1 = POSITION control mode)
# NOTE: This step can be bypassed by setting and saving 0x3F60 to the mode you
# want to automatically boot to after go-op.
<0x601> [5] 22 60 60 00 01

# First control loop iteration, RPDOs sent from master to pucks
<0x201> [4] 34 12 00 01 # Master sends Puck 1 a position command
<0x202> [4] 56 34 00 02 # Master sends Puck 2 a position command

# Master asks for puck TPDOs
<0x080> [0] Master sends a Sync to get position feedback from all
nodes

```

```

# Pucks respond with their TPDOs
<0x181> [2] 20 12 Puck 1 responds with its scaled position feedback = 0x1220
<0x182> [2] 50 34 Puck 2 responds with its scaled position feedback = 0x3450

# Second control loop iteration...
<0x201> [2] 56 12 Master sends Puck 1 a scaled position command = 0x1256
<0x202> [2] 89 34 Master sends Puck 2 a scaled position command = 0x3489
<0x080> [0] Master sends a Sync to get position feedback from all
nodes
<0x181> [2] 33 12 Puck 1 responds with its scaled position feedback = 0x1233
<0x182> [2] 85 34 Puck 2 responds with its scaled position feedback = 0x3485

# ... N iterations

```

#### Example Retrieving an Error:

```

# In the GO-OP state

# Check Puck 1s Error Register (0x1001,0)
<0x601> [5] 40 01 10 00 00 (read)

# IF there is a temperature error
# NOTE: To clear this error, a power cycle or reset of the node is
necessary.
<0x581> [8] 42 01 10 00 09 00 00 00 # Bit 3 (Temperature) and Bit 0
(Generic) are set (0x8 + 0x1)

# OTHERWISE
<0x581> [8] 42 01 10 00 00 00 00 00 # No bits are set

```

#### Example operation, reading BarrettCAN parameters:

```

# In The GO-OP State

# Get the Default Position (DP) property 50.
<0x601> [4] 40 00 3F 32
<0x581> [8] 42 00 3F 32 00 80 00 00

# Get the Thermistor Temperature (THERM) property 20.
<0x601> [4] 40 00 3F 14
<0x581> [8] 42 00 3F 32 00 1E 00 00

```

#### Appendix A – 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

## Appendix B – CANopen Multi-Field TPDO Example of the setup procedure.

```
# Set TPDO_1 COB ID (0x1800,1) to 0x181
<0x601> [8] 22 00 18 01 81 01 00 00 (set)
<0x581> [8] 60 00 18 01 81 01 00 00 (ack)

# Set TPDO_1 Tx type (0x1800,2) to 0 (Tx on every SYNC)
<0x601> [5] 22 00 18 02 00 (set)
<0x581> [8] 60 00 18 02 00 00 00 00 (ack)

# Set TPDO_1 Inhibit time (0x1800,3) to 0 (unused)
<0x601> [6] 22 00 18 03 00 00 (set)
<0x581> [8] 60 00 18 03 00 00 00 00 (ack)

# Set TPDO_1 Event timer (0x1800,5) to 0 (unused)
<0x601> [6] 22 00 18 05 00 00 (set)
<0x581> [8] 60 00 18 05 00 00 00 00 (ack)

# Set TPDO_1 first mapping (0x1A00,2) to 0x2201,0 (Bus voltage, 16-bits)
<0x601> [8] 22 00 1A 01 10 00 01 22 (set)
<0x581> [8] 60 00 1A 01 10 00 01 22 (ack)

# Set TPDO_1 second mapping (0x1A00,3) to 0x2202,0 (Amplifier temperature, 16-bits)
<0x601> [8] 22 00 1A 02 10 00 02 22 (set)
<0x581> [8] 60 00 1A 02 10 00 02 22 (ack)

# Set TPDO_1 third mapping (0x1A00,4) to 0x1001,0 (Error register, 8-bits)
<0x601> [8] 22 00 1A 03 08 00 01 10 (set)
<0x581> [8] 60 00 1A 03 08 00 A1 60 (ack)

# Set TPDO_1 mapping count (0x1A00,0) to 4
<0x601> [5] 22 00 1A 00 3 (set)
<0x581> [8] 60 00 1A 00 3 00 00 00 (ack)

# Save object dictionary changes to EEPROM (write "SAVE" to 0x1010,1)
<0x601> [8] 22 10 10 01 73 61 76 65 (set)
<0x581> [8] 60 10 10 01 01 00 00 00 (ack, reading this object returns 1 = manual write support)
```

### Test:

```
<0x080> [0] (Send SYNC)
<0x181> [7] 89 18 83 2b 00 (PDO response)
# Parse data as
# Voltage = 0x1889 = 6281/256 = 24.5 V
# Temp = 0x2B83 = 11139/256 = 43.5 deg. C
# Error = 0 (no error)
```