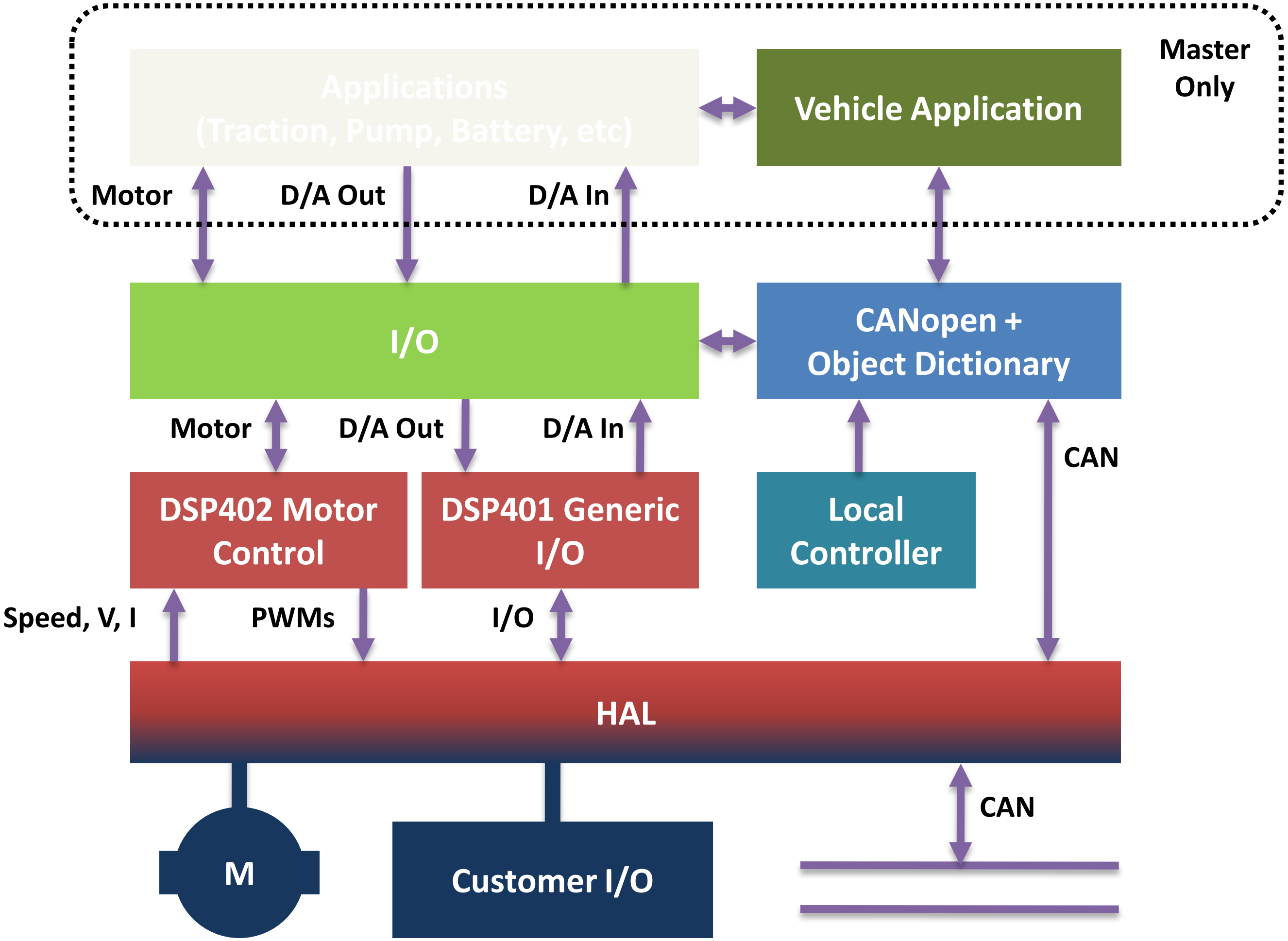
Gen4 Software Overview

Software Architecture



CANopen

- Application Layer and Communication Profile CiA DS-301
- ISO 11898 (CAN)
- 11 Bit Identifier field
- DSP-401 Generic IO modules
- DSP -402 Drives and Motion Control

Sevcon Object Dictionary

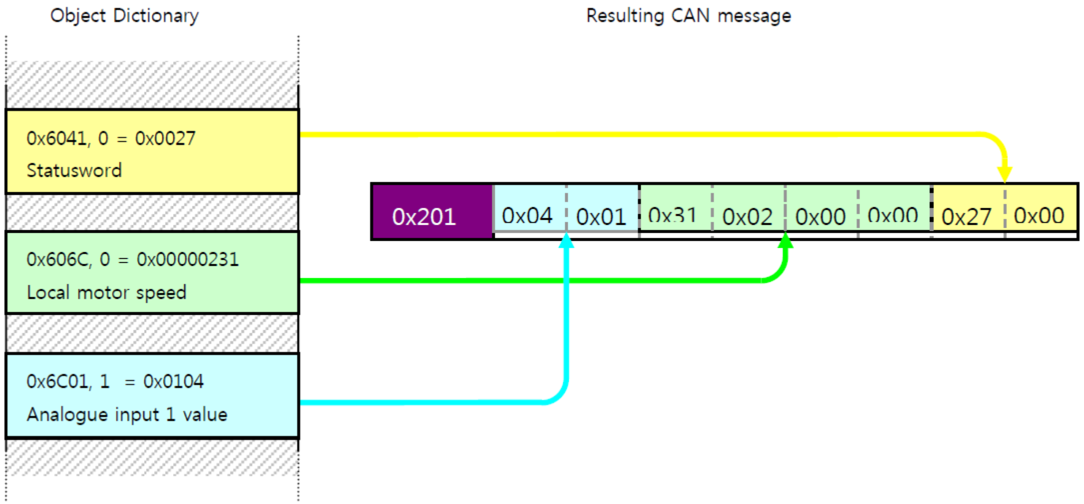
Data Types		0000h
Communication Profile Area		1000h
Manufacturer Specific Profile Area	Application Status	2000h
	Application Parameters	2800h
	Virtual PDO Mapping	3000h
	Customer Specific Area	3800h
	Logging	4000h
	Manufacturer Specific Device Profile Area	4600h
	Node Status	5000h
	Node Parameters	5800h
Standardised Device Profile Area	Profile 1 - AC Motor 1	6000h
	Profile 2 - Local I/O	6800h
	Profile 3 - AC Motor 2	7000h
	Profile 4 - DC Motor	7800h
	Profiles 5 - 8 (Unused)	8000h
Reserved		A000h

Object Dictionary

- Take a gentle look at the Object Dictionary....

PDOs and VPDOs

- Process Data Object (PDO)
  - Allow communication of real time data between devices on CANbus
  - 0 to 8 objects can be mapped into one PDO
  - Contains up to 8 bytes of data (eg 8 bytes, 4 bytes + 2 words, etc).



Object Dictionary

Resulting CAN message

0x6041, 0 = 0x0027
Statusword

0x606C, 0 = 0x00000231
Local motor speed

0x6C01, 1 = 0x0104
Analogue input 1 value

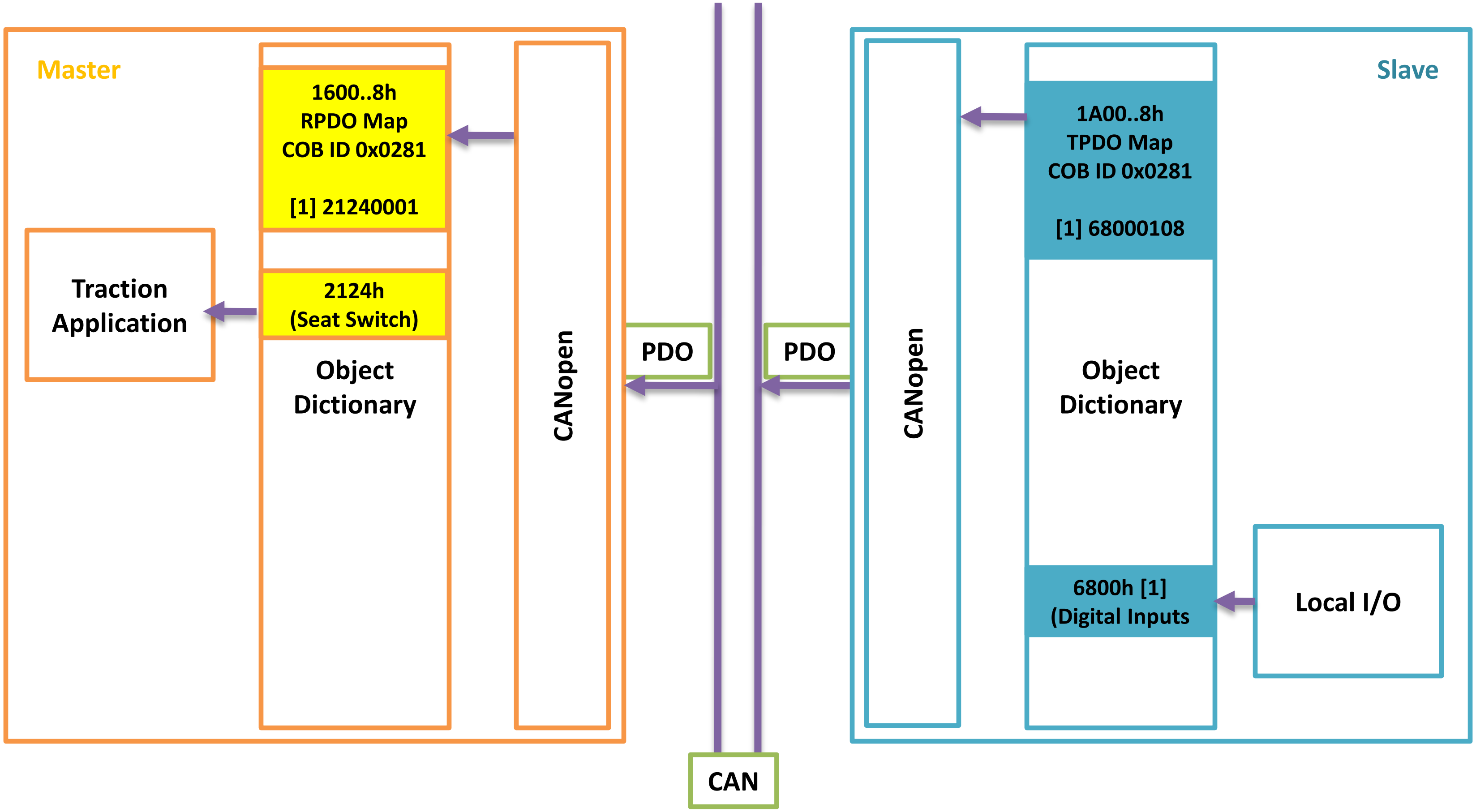
0x201 | 0x04 | 0x01 | 0x31 | 0x02 | 0x00 | 0x00 | 0x27 | 0x00

- Virtual PDOs
  - Do the same as PDOs for data internal to the Master node.
  - Are not CANopen. They are unique to Sevcon.

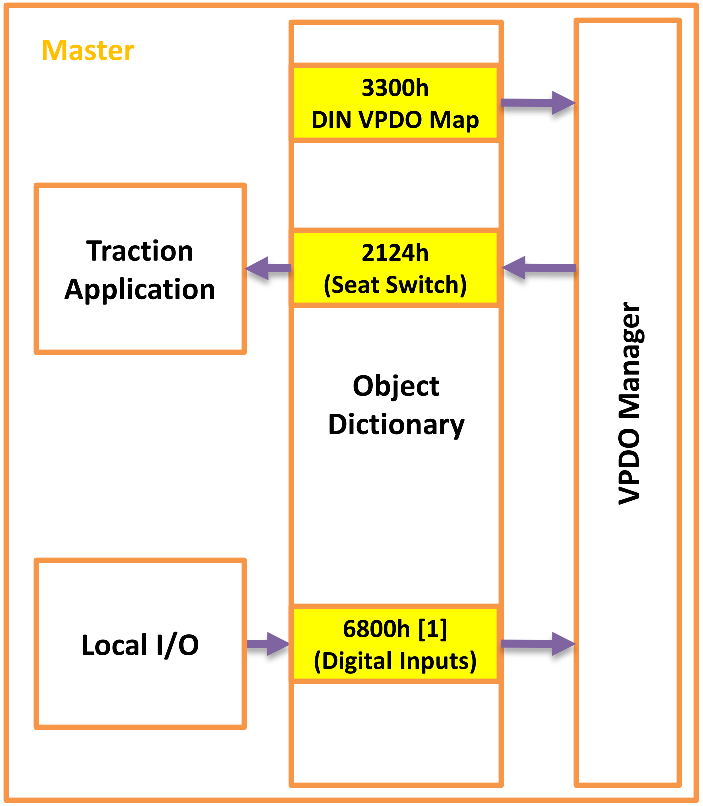
TPDO

- TPDO1 mapping at object 0x1A00
- Sub 0 – number of objects in TPDO
- Subs 1 to 8 – mapped object details
  - aaaabbcc
    - aaaa – mapped object number
    - bb – mapped object sub index
    - cc – number of bits in mapped object

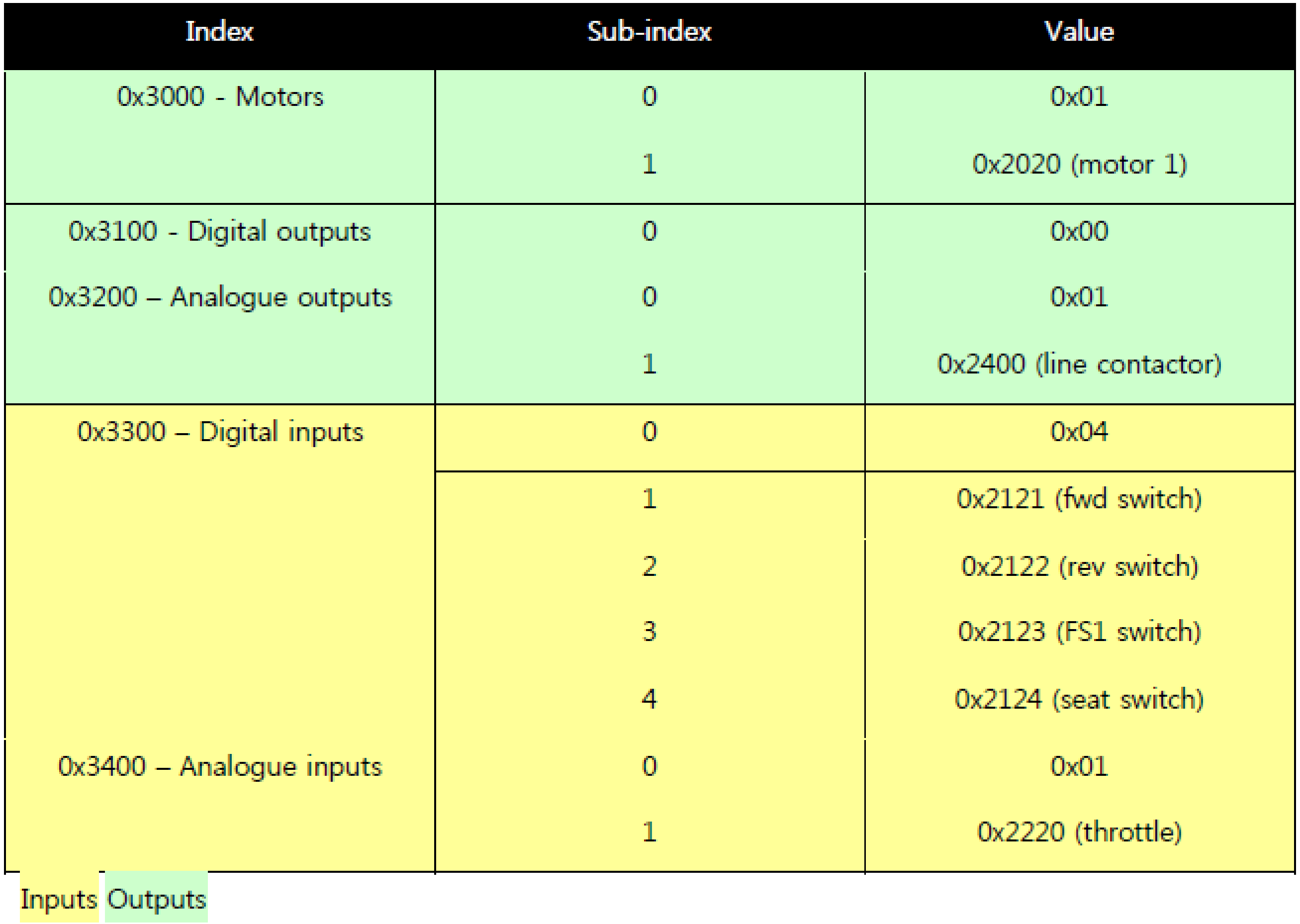
PDOs – Digital Input Example



VPDOs – Digital Input Example



Typical VPDOs



Index	Sub-index	Value
0x3000 - Motors	0	0x01
	1	0x2020 (motor 1)
0x3100 - Digital outputs	0	0x00
0x3200 – Analogue outputs	0	0x01
	1	0x2400 (line contactor)
0x3300 – Digital inputs	0	0x04
	1	0x2121 (fwd switch)
	2	0x2122 (rev switch)
	3	0x2123 (FS1 switch)
	4	0x2124 (seat switch)
0x3400 – Analogue inputs	0	0x01
	1	0x2220 (throttle)

Inputs Outputs

Device Profiles

- Defined in DS40x device profile.
  - DS401 – Generic I/O Profile.
  - DSP402 – Drives and Motion Control Profile. Supported modes are:
    - Profile Torque (tq)
    - Profile Velocity (pv)
- CANopen reserves space for 8 profile instances in the OD.
  - 6000h – 67FFh = AC Motor 1 Profile (DSP402).
  - 6800h – 6FFFh = Generic I/O Profile (DS401).
  - 7000h – 77FFh = AC Motor 2 Profile (DSP402).
  - 7800h – 7FFFh = DC Motor Profile (DSP402).
  - 8000h – 9FFFh = Unused

Software Features

- CANopen
- Motor Algorithms
- I/O functions
- Traction functions
- Pump and Power Steer functions
- Analogue Output (Contactor) functions
- Additional functions

I/O functions

- All digital and analogue I/O mappable via CANbus:
  - Standalone systems use VPDOs
  - Networked systems use PDOs
- Digital inputs (active high/low, polarity)
- Analogue inputs (2/3-wire inputs, wire-off, analog as digital)
- Analogue (Contactor) outputs (Voltage / current mode, error control)
- Motors (Encoders, PTC sensors)

Traction functions

- Safety Interlocks (FS1, seat, handbrake, electromechanical brake)
- Torque or speed mode.
- Throttle (proportional braking, directional throttle, characteristics)
- Footbrake and Economy Pot
- Dual motor steering algorithms
- Driveability Profiles
- Controlled roll-off and Hill Hold
- Inching

Application Subsystems

- Subsystems :
  - App_Mgr, BattApp, Display_App, FSM, TracApp, SSS, PumpApp, PSApp, VehApp
- Continuous (multiples of 20ms)
- Event driven
  - Digital input signals active/inactive/or changed.
  - Vehicle events (moving, stopped, etc).
- Each build has a customer application in the build folder (ie customer.c).
  - Customer specific functionality can be handled here.
  - This has 20ms continuous and event driven functions. Recommend that continuous code only is used.

Application Subsystems (contd)

- Vehicle Application
  - Co-ordinates all nodes in multi-node system via CANopen stack.
  - Ensure nodes boot up in correct manner and are operational/pre-operational.
  - Handles communication between applications.
- Traction (inc SSS), Pump and Power Steer Applications
  - Control of corresponding function, including FSMs. Each application takes digital and analogue inputs, and outputs motor targets and contactor states.
- Battery Application
  - Battery management
  - Line contactor.
- Display Application
  - Handles different display types.

I/O Subsystem + CANopen Stack

- Subsystems :
  - IO, Autoconfig, CANopen, OBD
- Handles all VPDO and PDO mapping.
- Interface between Applications layer and hardware (HAL) and CANopen interface.
- Input data to named input signals
  - Forward, Reverse, Throttle voltage, etc
- Named output signals to output data (voltage)
  - Line contactor, EBrake, etc
  - Chopping functionality

HAL (Hardware Abstraction Layer)

- Subsystems:
  - HAL.
- Implements generic hardware interface defined in hardware / software interface specification
- Only subsystem (excluding scheduler) which is platform specific.
- Handles:
  - Low level digital and analogue input acquisition.
  - Low level analogue output control.
  - CANbus
  - Motor I/O
  - Precharge

Debug Utilities (tracing)

- Vehicle interface trace
  - Used to record all objects mapped to TPDOs over a period of time.
  - Limited to fastest internal loop times (currently 20ms).
  - Use DVT to capture data and Vehicle Interface Viewer to graph it.
- Detailed Trace
  - Produces high resolution data down to 250us time base. Can also be used to record at 62.5us using additional code.
  - Up to 4 x 16-bit values can be stored, with a total of 905 points
  - Trace is buffered in internal 1k word buffer so has limited time period.
  - End of trace must be triggered manually
  - Trace has programmable trigger point typically half way through trace. Trace shows data leading up to and beyond trigger.
  - Uploaded using DVT and graphed using DSP Trace Viewer.

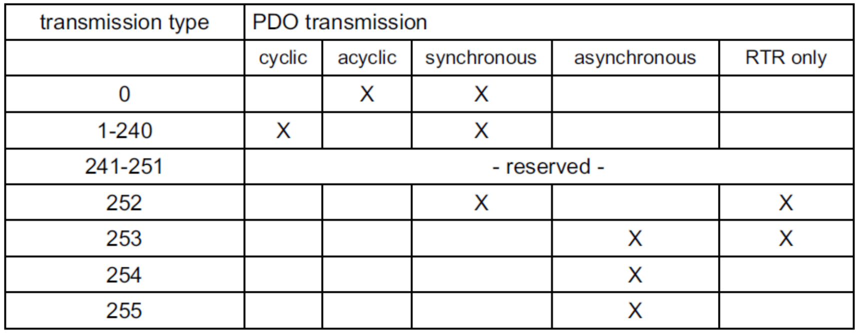
CANopen

- Application Layer and Communication Profile CiA DS-301
- ISO 11898 (CAN)
- http://www.can-cia.org/
- 11 Bit Identifier field

TPDO

- Transmit Process Data Object (TPDO)
- TPDO1 mapping at object 0x1A00
- Sub 0 – number of objects in TPDO
- Subs 1 to 8 – mapped object details
  - aaaabbcc
    - aaaa – mapped object number
    - bb – mapped object sub index
    - cc – number of bits in mapped object

- TPDO1 communication parameters at object 0x1800
- Sub 1 – Communication Object (COB) ID
  - 32 bits
  - must be a unique number across all nodes
  - Lower COB ID = higher priority
  - set to 0x80000000, which will disable the TPDO
- Sub 2 – Transmission Type
  - 8 bits
  - SEVCON - CYCLIC and SYCHRONOUS
  - Value between 1 and 240
  - Synchronous: tied to SYNC pulse
  - SYNC messages are produced by the master node. They usually have a COB-ID of 0x80, but not always.
  - The frequency at which SYNC messages are sent out can be adjusted in the Object Dictionary of the master node at index 0x1006.



transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

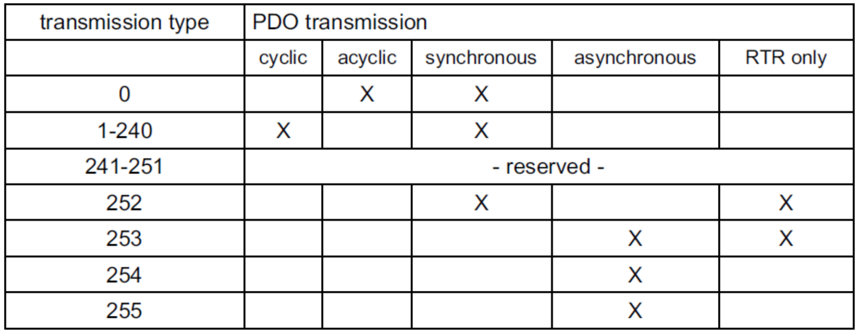
- Sub 3 – Inhibit time (optional)
- Minimum interval for PDO transmission (100us per bit)
- Sub 4 – reserved (not used)
- Sub 5 – event timer (optional)

RPDO

- Receive Process Data Object (RPDO)
- RPDO1 mapping at object 0x1600
- Sub 0 – number of objects in RPDO
- Subs 1 to 8 – mapped object details
  - aaaabbcc
    - aaaa – mapped object number
    - bb – mapped object sub index
    - cc – number of bits in mapped object

RPDO

- RPDO1 communication parameters at object 0x1400
- Sub 1 – Communication Object (COB) ID
  - 32 bits
- Sub 2 – Transmission Type
  - 8 bits
  - Synchronous: tied to SYNC pulse



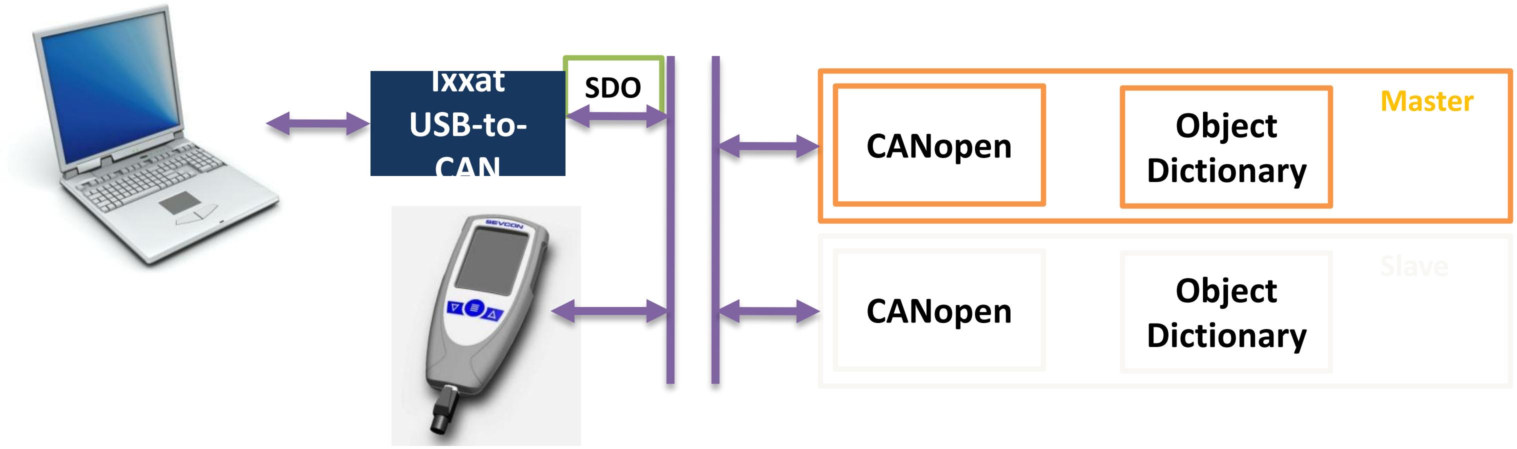
transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

- Sub 3 – Inhibit time (optional)
- Minimum interval for PDO transr
- Sub 4 – reserved (not used)
- Sub 5 – event timer (optional)
- RPDO Timeout function – object 0x5902

Twin motor Configuration

- 0bject 0x2020 for left motor (master)
- Object 0x2021 for right motor
- Object 0x2040 for pump motor
- Map these to TPDOs
  - Sub 1 – Control word
  - Sub 2 – Status word
  - Sub 3 – Target/Max Speed
  - Sub 4 – Actual Speed
  - Sub 5 – Target/Max Torque
  - Sub 6 – Actual Torque
- Unmapping RPDOs to slaves
  - 0x6040 – control word
  - 0x6071 – target torque/torque limit
  - 0x606B – target speed/speed limit

SDOs



- Used for:
  - Configuration (using DriveWizard or CANopen calibrator)
  - Accessing data between nodes at a low rate
- Only one object index + sub-index can be accessed per SDO.