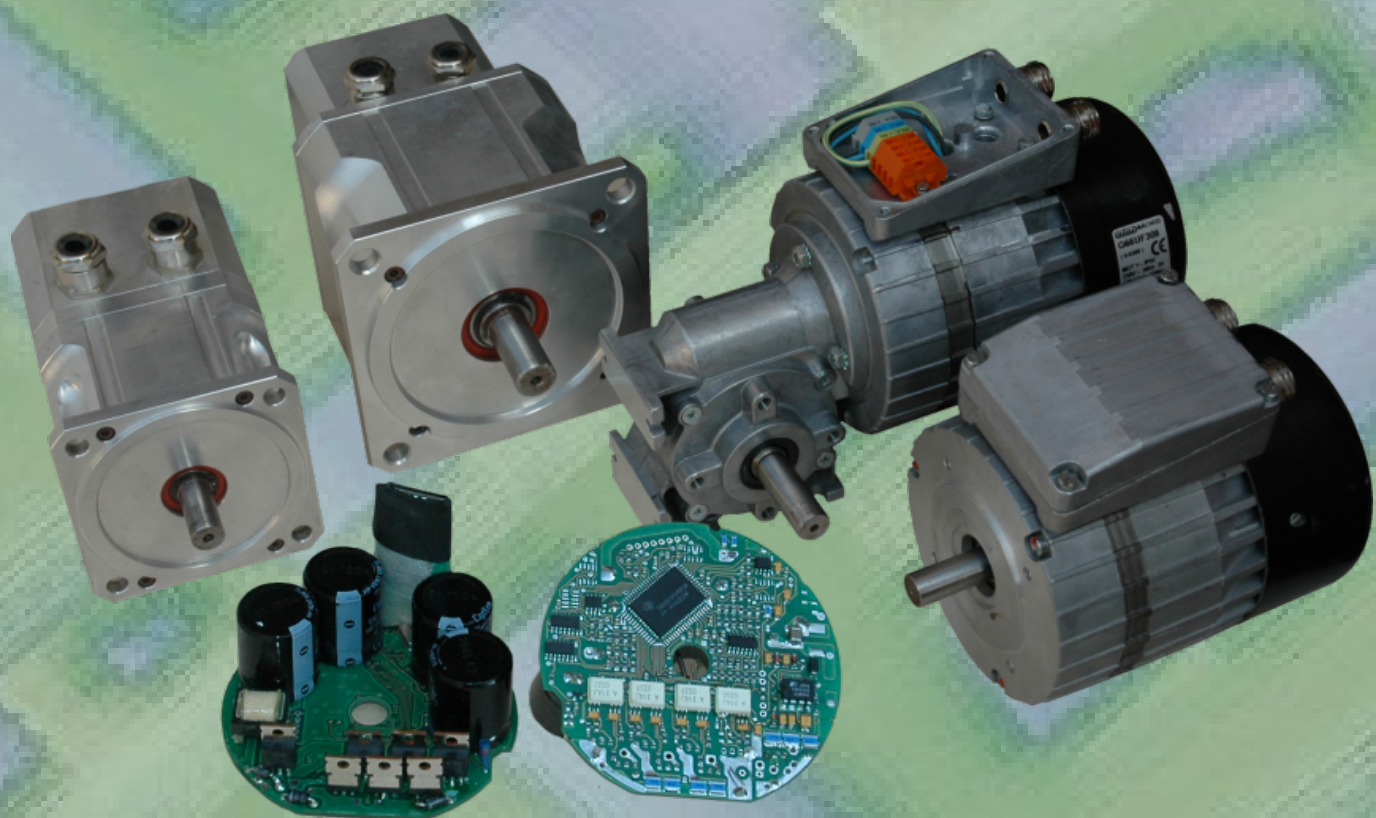


CAN communication profile

version 4



PMControl s.r.o.

CAN – Communication profile

version 4

Table of contents

CAN Communication profile.....	3
CAN communication.....	3
EC motor settings for CAN communication.....	3
Implemented communication objects.....	4
NMT – Network Management.....	4
EMCY – Emergency.....	4
TPDO1 Proces Data Object	4
SDO Service Data Object.....	4
SDO 2000h-203Fh: Motor and user variables.....	5
SDO 2000h-200Fh: Bit-accessible objects.....	8
SDO 2040h – 2045h: Functions for motor control.....	11
2040h Read from physical addresses.....	11
2041h Read firmware version number.....	11
2042h Set motor working mode.....	12
2043h User program task 1 management.....	13
2044h User program task 2 management.....	13
2045h Clear functions.....	13

PMControl s.r.o., www.pmcontrol.cz,

revision 2006-12-14

CAN Communication profile

EC motor can be controlled by:

- RS232/RS485 communication
- **CAN communication**
- user program

CAN communication

- based on **CANopen**
- keeps default types of communication objects **COB**
- using **11** bit message identifier **COB_ID**
format: type COB = 4b module address = 7b (see table bellow)
- factory definition of objects **SDO**
index 2000h...5FFFh -> "Manufactuter Specific Profile Area"
- data types

I16	16bit	signed
U16	16bit	unsigned
B16	16bit	binary
I32	32bit	signed
- data flows in order "**low to high**"
for example 1234h sent as: 34h 12h
for example 1234567h sent as: 67h 45h 23h 01h

Communication object COB		Message indentificator COB ID			
mnemo	description	hex	dec.	bin	direction
NMT	Network Management	000h	0	000 0000 0000	MA->EC
SYNC	Synchronizace	080h	128	000 1000 0000	
EMCY	Emergency	080h+ aa	128 + aa	000 1.aa aaaa	EC->MA
TIME	Time Stamp	100h	256 + aa	001 0000 0000	
TPDO1	Transmit Proces Data	180h+ aa	364 + aa	001 1.aa aaaa	EC->MA
RPDO1	Receive Proces Data	200h + aa	512 + aa	010 0.aa aaaa	MA->EC
TPDO2	Transmit Proces Data	280h + aa	640 + aa	010 1.aa aaaa	
RPDO2	Receive Proces Data	300h + aa	768 + aa	011 0.aa aaaa	
TPDO3	Transmit Proces Data	380h + aa	896 + aa	011 1.aa aaaa	
RPDO3	Receive Proces Data	400h + aa	1024 + aa	100 0.aa aaaa	
TPDO4	Transmit Proces Data	480h + aa	1152 + aa	100 1.aa aaaa	
RPDO4	Receive Proces Data	500h + aa	1280 + aa	101 0.aa aaaa	
TSDO	Transmit Servis Data	580h + aa	1408 + aa	101 1.aa aaaa	EC->MA
RSDO	Receive Servis Data	600h + aa	1536 + aa	110 0.aa aaaa	MA->EC
NODG	NodeGuard	700h + aa	1792 + aa	111 0.aa aaaa	

aa,aa aaaa node address of motor v **c_set.adr0** (object 200Dh)
MA master (PLC, PC with CAN interface, ...)
EC EC motor

EC motor settings for CAN communication

- only with EC113 interface (see ECI113 manual...)
- firmware version **ECA19** or later
- connect motor to PC via RS232 and switch the motor on
- run the program "EC Motor Control" with motor address = 0
- in menu SETUP, choose SETUP, folder "Basic" ... ,Communication settings c_set (see object 200Dh)
 - set the **address of motor** **1-15**
 - set the **CAN communication speed** **1-7**
 - save it to EEPROM (press Ctrl + Enter or press the button EEPROM)
- make HW reset of motor (switch off for 30s)



- program "EC Motor Control" can communicate with any motor by sending data with address 0, i.e. motor can be set to different address in c_set
- for correct function of CAN bus you should connect termination resistor 120R between CANL and CANH wires

Implemented communication objects:

NMT – Network Management

NMT – motor reset		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	do reset	NMT	000h	2	82h	x						
EC:	without answer											

82h code reset U8 code CANopen
x motor address v **c_set.adr0** U8 (object 200Dh)

EMCY – Emergency

EMCY is emitted in the following cases:

- activated SW or HW protection of motor
- rejected communication command
- by set bit **c_sss.bcse** (object 200Ch)
- user program command UPRG

EMCY error announcement		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
EC:	error message	EMCY	08xh	8	10h	00h	m_sss		c_sss		m_ssm	

0010h unspecified error U16 code CANopen
m_sss error states of motor B16 (object 200Eh)
c_sss error states of communication B16 (object 200Ch)
m_ssm working status word B16 (object 200Fh)

TPDO1 Proces Data Object

TPDO1 is emitted in the following cases:

- answer to RPDO1 - request for state of MA
- by set bit **c_sss.bcscs** (object 200Ch)
- user program command UPRG

PDO1 motor state		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send state	RPDO1	20xh	0								
EC:	sending state	TPDO1	18xh	8			h_poz			m_eci		m_ssm

h_poz actual position I32 (object 2012h)
m_eci input / output word B16 (object 200Ah)
m_ssm working state of motor B16 (object 200Fh)

SDO Service Data Object

SDO objects are used to setup all parameters of motor. SDO is already initiated by master (MA) and slave (EC) always send confirmation.

SDO index		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	command / query	RSDO	60xh	4..8	cs	index		subi.				data
EC:	confirmation / answer	TSDO	58xh	4..8	cs	index		subi.				data

ID message id
x address of motor 1-15 in **c_set.adr0** (object 202Fh)
DLC message length 0..8
 EC receipt: not checked (shorten by data type or fill in 00h to DLC=8)
 EC transmission: shorten by data type
cs "command specifier"
 use short transfer type "expedited transfer" => max 4byte of data
 write **w** MA: **22h** "write" EC: **60h** "confirmation"
 read **r** MA: **40h** "send" EC: **40h** "sending"
index object index
 2000h ... 203Fh **motor and user objects**
 2040h ... 2045h **special functions**
subi. object subindex
 access modify to objects item (length, function..)
data data in byte order "**low to high**"

SDO 2000h-203Fh: Motor and user variables

CAN		UPRG	bit-accessible objects and status words				
index	subi.	address	type	r/w	class	mnemo	description
2000h	x	00h/00h	x		x	URAM12	user memory
2001h	x	01h/04h	x	x	x	URAM13	user memory
2002h	x	02h/08h	x	x	x	URAM14	user memory
2003h	x	03h/0Ch	x	x	x	URAM15	user memory
2004h	x	04h/10h	x	x	x	URAM16	user memory
2005h	x	05h/14h	x	x	x	URAM17	user memory
2006h	x	06h/18h	x	x	x	URAM18	user memory
2007h	x	07h/1Ch	x	x	x	URAM19	user memory
2008h	x	08h/20h	x	x	x	URAM20	user memory
2009h	x	09h/24h	–	–	–	----	
200Ah	0h	0Ah/28h	B16	rw	–	m_eci	Input / output word
200Bh	0h	0Bh/2Ch	B16	rw	–	S_err	Definition of sources for SW protection
200Ch	0h	0Ch/30h	B16	r	–	c_sss	Communication status word
200Dh	0h	0Dh/34h	B16	rw	–	m_set	Control word
200Eh	0h	0Eh/38h	B16	r	–	m_sss	Error status word
200Fh	0h	0Fh/3Ch	B16	r	–	m_ssm	Working status word

CAN		UPRG	motor variables – read only				
index	subi.	address	type	r/w	class	mnemo	description
2010h	1h	10h/40h	I32	r	P	p_pox	Profile gen. - position
2011h	0h	11h/44h	I16	r	V	p_vex	Profile gen. - velocity
2012h	1h	12h/48h	I32	r	P	h_poz	Actual position
2013h	1h	13h/4Ch	I32	r	P	h_tet	Sensor angle
2014h	0h	14h/50h	I16	r	V	h_vel	Actual velocity
2015h	0h	15h/54h	I16	r	C	iq_ref	Actual current Iq
2016h	0h	16h/58h	U16	r	V	udc	DC Line Voltage
2017h	0h	17h/5Ch	U16	r	T	tsn	Temperature of electronics
2018h	0h	18h/60h	U16	r	T	tcu	Motor utilisation
2019h	0h	19h/64h	I16	r	P	p_err	Position error
201Ah	0h	1Ah/68h	–	–	–	----	
201Bh	0h	1Bh/6Ch	–	–	–	----	
201Ch	0h	1Ch/70h	–	–	–	----	
201Dh	0h	1Dh/74h	–	–	–	----	
201Eh	0h	1Eh/78h	–	–	–	----	
201Fh	0h	1Fh/7Ch	–	–	–	----	

CAN		UPRG	motor variables – write only				
index	subi.	address	type	r/w	class	mnemo	description
2020h	1h	20h/80h	U32	rw	–	u_tim	System time
2021h	1h	21h/84h	I32	rw	P	P_lil	Low position limit
2022h	1h	22h/88h	I32	rw	P	P_lih	Hihg position limit
2023h	1h	23h/8Ch	I32	rw	P	p_pon	Requested position
2024h	0h	24h/90h	I16	rw	V	p_ven	Requested velocity
2025h	0h	25h/94h	U16	rw	A	p_acc	Requested acceleration
2026h	0h	26h/98h	U16	rw	A	p_dec	Requested deceleration
2027h	0h	27h/9Ch	I16	rw	M	p_mon	Requested torque limit
2028h	0h	28h/A0h	–	–	–	----	
2029h	0h	29h/A4h	–	–	–	----	
202Ah	0h	2Ah/A8h	–	–	–	----	
202Bh	0h	2Bh/ACh	–	–	–	----	
202Ch	0h	2Ch/B0h	–	–	–	----	
202Dh	0h	2Dh/B4h	–	–	–	----	
202Eh	0h	2Eh/B8h	–	–	–	----	
202Fh	0h	2Fh/BCCh	U16	rw	–	c_set	Communication settings

CAN		UPRG	UPRG initialized variables				
index	subi.	address	type	r/w	class	mnemo	description
2030h	x	30h/C0h	x	x	x	URAM00	User memory
2031h	x	31h/C4h	x	x	x	URAM01	User memory
2032h	x	32h/C8h	x	x	x	URAM02	User memory
2033h	x	33h/CCh	x	x	x	URAM03	User memory
2034h	x	34h/D0h	x	x	x	URAM04	User memory
2035h	x	35h/D4h	x	x	x	URAM05	User memory
2036h	x	36h/D8h	x	x	x	URAM06	User memory
2037h	x	37h/DCh	x	x	x	URAM07	User memory
2038h	x	38h/E0h	x	x	x	URAM08	User memory
2039h	x	39h/E4h	x	x	x	URAM09	User memory
203Ah	x	3Ah/E8h	x	x	x	URAM10	User memory
203Bh	x	3Bh/ECh	x	x	x	URAM11	User memory
203Ch	x	3Ch/F0h	-	-	-	----	
203Dh	x	3Dh/F4h	-	-	-	----	
203Eh	x	3Eh/F8h	-	-	-	----	
203Fh	x	3Fh/FCh	-	-	-	----	

---- free index – don't write or read -> error EMCY
URAMxx User program's (user task) variable array
x Can be used in user program
type variable format (u16, i16, b16, i32)
r/w type of access w – write, r – read, rw – read/write
class variable class for conversion to physical quantity, see table below

Class	Name	Conversion const.		Unit	Minimum	Maximum
V	Velocity	0.1875	3/16	RPM	-32768	32767
A	Acceleration	0.0146484375		RPM/ms	0	4096
P	Position	0.1171875	360/3072	DEG	-2 ³¹	2 ³¹
M	Torque	motor type dependent		Nm	0	16384
C	Current	motor type dependent		A		
V	Voltage	0.01	1/100	V	8000	35000
T	Temperature	0.01	1/100	°C	-20000	20000
K	Time	0.8	8/10	ms	0	32768
--	machine unit	1	1	-	0	65535

SDO write I16,U16,B16		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	write	RSDO	60xh	6/8	22h	ii	20h	00h	d0	d1	0	0
EC:	confirm	TSDO	58xh	4	60h	ii	20h	00h	--	--	--	--

SDO write I32		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	write	RSDO	60xh	6/8	22h	ii	20h	01h	d0	d1	d2	d3
EC:	confirm	TSDO	58xh	4	60h	ii	20h	01h	--	--	--	--

SDO send I16,U16,B16		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me	RSDO	60xh	4/8	40h	ii	20h	00h	0	0	0	0
EC:	sending	TSDO	58xh	6	40h	ii	20h	00h	d0	d1	--	--

SDO send I32		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me	RSDO	60xh	4/8	40h	ii	20h	01h	0	0	0	0
EC:	sending	TSDO	58xh	8	40h	ii	20h	01h	d0	d1	d2	d3

ii low byte of index (index = 2000h+ ii)
-- data aren't send
0 data aren't send or can be filled up by value 00h to DLC=8



Example 1: Set requested velocity p_ven = 1234h
 motor address c_set.adr0 = 03h **(used in all examples !!!)**
 requested velocity p_ven : index= 2024h subi.= 00h

SDO write I16,U16,B16	COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA: p_ven=1234h	RSDO	603h	8	22h	24h	20h	00h	34h	12h	00h	00h
EC: confirm	TSDO	583h	4	60h	24h	20h	00h				



Example 2: Set requested position p_pon = 1234567h
 requested position p_pon : index= 2023h subi.= 01h

SDO write I32	COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA: p_pon = 1234567h	RSDO	603h	8	22h	23h	20h	01h	67h	45h	23h	01h
EC: confirm	TSDO	583h	4	60h	23h	20h	01h				



Example 3: Read requested velocity p_ven
 requested velocity p_ven : index= 2024h subi.= 00h (previously writed p_ven = 1234h)

SDO read I16,U16,B16	COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA: send me p_ven	RSDO	603h	8	40h	24h	20h	00h	00h	00h	00h	00h
EC: sending p_ven	TSDO	583h	6	40h	24h	20h	00h	34h	12h		



Example 4: Read requested position p_pon
 requested position p_pon : index= 2023h subi.= 01h (previously writed p_pon = 1234567h)

SDO read I32	COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA: send me p_pon	RSDO	603h	4	40h	23h	20h	01h				
EC: sending p_pon	TSDO	583h	8	40h	23h	20h	01h	67h	45h	23h	01h

SDO 2000h-200Fh: Bit-accessible objects

200Dh		m_set		Control word										
bit		--	cgr	cbi	cdi	function of ECI200	--	--	--	--	cen	--	clp	cdn
dec.	hex.	mask	mnemo	description										
15	0Fh	8000h												
14	0Eh	4000h	car	dec is acc. Vmi is Vma for GR MAX										
13	0Dh	2000h	cbi	Block BIO input in potentiometer										
12	0Ch	1000h	cdi	DIR - rotation direction										
11	0Bh	0800h		ECI200 function selector 0-15										
10	0Ah	0400h												
09	09h	0200h												
08	08h	0100h												
07	07h	0080h												
06	06h	0040h												
05	05h	0020h												
04	04h	0010h												
03	03h	0008h	cen	Enable input activation										
02	02h	0004h												
01	01h	0002h	clp	SW position limit - activation										
00	00h	0001h	cdn	After reach a destination position switch to mode 03 (Power Stage ON)										

200Fh		m_ssm			Working tatus								
bit		ber	bwa	bxx	bio	bpz	bru	bpo	bup	bci	bco	bex	mod motoru
dec.	hex.	mask	mnemo	description									
15	0Fh	8000h	ber	Error (HW.SW.Reset)									
14	0Eh	4000h	bwa	Warning see m_sss									
13	0Dh	2000h	bxx	User Flag									
12	0Ch	1000h	bio	Input BIO									
11	0Bh	0800h	bpz	Power Stage ON									
10	0Ah	0400h	bru	Motor Run									
09	09h	0200h	bpo	On Position									
08	08h	0100h	bup	User Task Run									
07	07h	0080h	bci	Input 2 (CAN=IO)									
06	06h	0040h	bco	Output 1 (CAN=IO)									
05	05h	0020h	bex	Oscilloscope start									
04	04h	0010h	mod	Working mode - see T_MOD									
03	03h	0008h											
02	02h	0004h											
01	01h	0002h											
00	00h	0001h											

200Eh		m_sss			Error status word												
bit		bid	brs	bhw	bsw	buh	bul	bih	bep	bts	btm	btb	bek	ben	bee	bll	blh
dec.	hex.	mask	mnemo	description													
15	0Fh	8000h	bid	Data integrity error in FLASH or EEPROM													
14	0Eh	4000h	brs	After Switch On or Reset													
13	0Dh	2000h	bhw	HW Protection – error set in													
12	0Ch	1000h	bsw	SW Protection – error set in													
11	0Bh	0800h	buh	Max. power supply (over voltage)													
10	0Ah	0400h	bul	Min. power supply (under nominal voltage)													
09	09h	0200h	bih	Max. torque / current Iq													
08	08h	0100h	bep	Max. of position error													
07	07h	0080h	bts	Max. sensor temperature													
06	06h	0040h	btm	Max. utilization of motor													
05	05h	0020h	btb	Max. utilization of brake													
04	04h	0010h	bek	Communication Error (from c_sss)													
03	03h	0008h	ben	Disable BIO (BIO=0 enable by m_set.cen)													
02	02h	0004h	bee	Error ECI module													
01	01h	0002h	bll	Min position limit													
00	00h	0001h	blh	Max position limit													



- bits are set when error is detected
- clear bits by command (for example: SDO index=2045h subi.=0)

202Fh		c_set		Communication control word					
bit		mapm		adr1		cfk	cfc		adr0
dec.	hex.	mask	mnemo	description					
15	0Fh	8000h	mapm	Monitor command mask 0-15					
14	0Eh	4000h							
13	0Dh	2000h							
12	0Ch	1000h							
11	0Bh	0800h	adr1	CAN group address 1-15 (preliminary)					
10	0Ah	0400h							
09	09h	0200h							
08	08h	0100h							
07	07h	0080h	cfk	RS232 speed: 0 = 9.6kBd 1 = 38kBd					
06	06h	0040h	ccc	CAN speed: 000 = CAN_OFF -> IO 001 = 10kBd 010 = 20kBd 011 = 50kBd 100 = 125kBd 101 = 250kBd 110 = 500kBd 111 = 1MBd					
05	05h	0020h							
04	04h	0010h							
03	03h	0008h							
02	02h	0004h	adr0	Motor address CAN / RS232 1-15					
01	01h	0002h							
00	00h	0001h							

200Ch		c_sss		Communication status word											
bit		bcer	bcef					bcse	bcss			bcok	bcep		
dec.	hex.	mask	mnemo	description											
15	0Fh	8000h	bcer	CAN communication format error											
14	0Eh	4000h	bcef	CAN relected command											
13	0Dh	2000h													
12	0Ch	1000h													
11	0Bh	0800h													
10	0Ah	0400h													
09	09h	0200h	bcse	CAN send message "error", object EMCY											
08	08h	0100h	bcss	CAN send message "state", object TPDO1											
07	07h	0080h													
06	06h	0040h													
05	05h	0020h	bcok	RS232 packet receive OK											
04	04h	0010h	bcep	RS232 relected command											
03	03h	0008h													
02	02h	0004h													
01	01h	0002h													
00	00h	0001h													

200Ah		m_eci		ECI200 Input / output word – preliminary											
bit															
dec.	hex.	mask	mnemo	description											
15	0Fh	8000h	brfc	Read function											
14	0Eh	4000h													
13	0Dh	2000h													
12	0Ch	1000h													
11	0Bh	0800h													
10	0Ah	0400h													
09	09h	0200h	in9	INPUT 9											
08	08h	0100h	in8	INPUT 8											
07	07h	0080h	in7	INPUT 7											
06	06h	0040h	io6	INPUT / OUTPUT 6											
05	05h	0020h	io5	INPUT / OUTPUT 5											
04	04h	0010h	io4	INPUT / OUTPUT 4											
03	03h	0008h	io3	INPUT / OUTPUT 3											
02	02h	0004h	ou2	OUTPUT 2											
01	01h	0002h	ou1	OUTPUT 1											
00	00h	0001h	ou0	OUTPUT 0											



Bit access

For bit-accessible objects type B16 (2000h-200Fh) you can use ordinary SDO for read or write U16 or use logical functions:

subi.= **02h** **SET** set bits from bit mask
 subi.= **03h** **CLR** clear bits from bit mask
 subi.= **04h** **NEG** complement bits from bit mask

SDO B16 set bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	set bits	RSDO	60xh	6/8	22h	ii	20h	02h	b0	b1	0	0
EC:	confirm	TSDO	58xh	4	60h	ii	20h	02h	--	--	--	--

SDO B16 clear bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	clear bits	RSDO	60xh	6/8	22h	ii	20h	03h	b0	b1	0	00h
EC:	confirm	TSDO	58xh	4	60h	ii	20h	03h	--	--	--	

SDO B16 complement bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	complements bits	RSDO	60xh	6/8	22h	ii	20h	04h	b0	b1	0	00h
EC:	confirm	TSDO	58xh	4	60h	ii	20h	04h	--	--	--	--

b1b0 bit mask see SDO 2000h-200Fh bit accesible objects



Example 5: Set, clear and complement bit **m_ssm.bxx** (user flag)

working status word **m_ssm**: index = 200Fh, bit **bxx** - mask = 2000h

subindex depend on function: subi. = 02h-> SET 03h->CLR 04h->NEG

SDO B16 set bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	set bits	RSDO	603h	8	22h	0Fh	20h	02h	00h	20h	00h	00h
EC:	confirm	TSDO	583h	4	60h	0Fh	20h	02h				

SDO B16 clear bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	clear bits	RSDO	603h	8	22h	0Fh	20h	03h	00h	20h	00h	00h
EC:	confirm	TSDO	583h	4	60h	0Fh	20h	03h				

SDO B16 complement bits		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	complements bits	RSDO	603h	8	22h	0Fh	20h	04h	00h	20h	00h	00h
EC:	confirm	TSDO	583h	4	60h	0Fh	20h	04h				

SDO 2040h – 2045h: Functions for motor control

T_CAN		parameter			CAN – motor control functions						
index	subi	data	type	r/w	mnemo	description					
2040h	0h	a1a0	U16	r		read U16 data from physical address a1a0					
2040h	1h	a1a0	U16	r		read U32 data from physical address a1a0					
2041h	0h	----	--	r		read version firmware SW (type U16)					
2042h	0h	m0	U16	w		set motor mode - see table T_MOD					
2043h	0h	0000h	U16	w		stop user program task 1					
2043h	0h	u1u0	U16	w		start user program task 1 from address u1u0					
2044h	0h	0000h	U16	w		stop user program task 2					
2044h	0h	u1u0	U16	w		start user program task 2 from address u1u0					
2045h	0h	----	--	w		clear error states (warning m_sss , m_ssm , c_sss)					
2045h	1h	----	--	w		clear position when motor stops					
2045h	2h	----	--	w		clear position when motor run (only in mode=05 velocity!!!)					
2045h	3h	----	--	w		clear position absolutely + (follow absolute sensor position)					
2045h	4h	----	--	w		clear position absolutely - (follow absolute sensor position)					
2046h	x		x	x	----	not used					
2047h	x		x	x	----	not used					

2040h Read from physical addresses

- only for debugging or advanced functions

SDO send U16 from address		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me	RSDO	60xh	6/8	40h	40h	20h	00h	a0	a1	0	00h
EC:	sending	TSDO	58xh	6	40h	40h	20h	00h	d0	d1	--	--

SDO send U32 from address		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me	RSDO	60xh	6/8	40h	40h	20h	01h	a0	a1	0	00h
EC:	sending	TSDO	58xh	8	40h	40h	20h	01h	d0	d1	d2	d3

a1a0 - physical address U16
d1d0 - data U16
d3d2d1d0 - data U32

Example 6: Read from physical address – only for debugging - user don't know physical address of parameters



index: 2040h subi.: 00h condition: p_ven: a1a0 = 367h p_ven = 1234h
 index: 2040h subi.: 01h condition: p_pon: a1a0 = 365h- p_pon = 1234567h

SDO send U16		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me (367h)	RSDO	603h	8	40h	40h	20h	00h	67h	03h	00h	00h
EC:	sending (367h)	TSDO	583h	6	40h	40h	20h	00h	34h	12h		

SDO send U32		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me (365h)	RSDO	603h	8	40h	40h	20h	01h	65h	03h	00h	00h
EC:	sending (365h)	TSDO	583h	8	40h	40h	20h	01h	67h	45h	23h	01h

2041h Read firmware version number

SDO send version		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me	RSDO	60xh	4/8	40h	41h	20h	00h	0	0	0	0
EC:	sending	TSDO	58xh	6	40h	41h	20h	00h	v0	v1	--	--

v1v0 - firmware version



Example 7: Read firmware version number

index: 2041h subi.: 00h

interpretation: **00BEh** -> 00h = ECA BEh = 190₁₀ => SW = **ECA19.0**

SDO send version		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	send me version num.	RSDO	603h	4	40h	41h	20h	00h				
EC:	sending	TSDO	583h	6	40h	41h	20h	00h	BEh	00h		

2042h Set motor working mode

SDO working mode		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	set mode	RSDO	60xh	6/8	22h	42h	20h	00h	m0	00h	0	0
EC:	confirm	TSDO	58xh	4	60h	42h	20h	00h	--	--	--	--

m0 - motor working mode - see table **T_MOD**

T_MOD		table of motor working modes									
mod		Main working modes				mod		Customer working modes			
dec.	hex.	mnemo	description			dec.	hex.	mnemo	description		
0	00h	MON	service mode (debug)			16	00h	ARRP	ARROW velocity profile		
1	01h	OFF	Power stage OFF			17	01h	ETY1	Mavet old labeler		
2	02h	STP	Decelarate to STOP			18	02h	ETY2	Mavet new labeler		
3	03h	PON	Power stage ON (standstill)			19	03h	VYSR	Vyškov SX_VYS		
4	04h	MOM	Torque mode			20	04h	VYSL	Vyškov SX_VYL		
5	05h	VEL	Velocity mode			21	05h				
6	06h	POZ	Position mode			22	06h				
7	07h					23	07h				
8	08h					24	08h				
9	09h					25	09h				
10	0Ah					26	0Ah				
11	0Bh					27	0Bh				
12	0Ch					28	0Ch	M24	Old debug mode		
13	0Dh					29	0Dh	ERR	Error state		
14	0Eh					30	0Eh	JMP	Debug mode		
15	0Fh	VELP	Potentiometer velocity control			31	0Fh	RES	HW reset of motor		



- selected mode is possible to read from m_ssm.mod (object 200Fh)
- Error state mod=29 is canceled by Power stage off mode mod=1
when error causes passes off you can continue selecting any other mode
- switch to power stage on is not possible when motor is rotating
- modes 28-31 are internal working modes – don't use
- do not switch to customers modes 16-20 without correspond data structures in URAM !!!
- after motor stop in mode 2 motor automatically switch to mode 3 - power stage ON



Example 8: Switch to velocity mode

index: 2042h subi.: 00h

velocity mode = 05h

SDO select mode		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	select mode	RSDO	603h	6	22h	42h	20h	00h	05h	00h		
EC:	confirm	TSDO	583h	4	60h	42h	20h	00h				

2043h User program task 1 management

SDO user program – task 1		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	run program	RSDO	60xh	6/8	22h	43h	20h	00h	u0	u1	0	0
EC:	confirm	TSDO	58xh	4	60h	43h	20h	00h	--	--	--	--

u1u0 - addresses are of user program are allocated by compilation

SDO user program – task 1		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	stop program	RSDO	60xh	6/8	22h	43h	20h	00h	00h	00h	0	0
EC:	confirm	TSDO	58xh	4	60h	43h	20h	00h	--	--	--	--

2044h User program task 2 management

SDO user program – task 2		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	run program	RSDO	60xh	6/8	22h	44h	20h	00h	u0	u1	0	0
EC:	confirm	TSDO	58xh	4	60h	44h	20h	00h	--	--	--	--

SDO user program – task 2		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	stop program	RSDO	60xh	6/8	22h	44h	20h	00h	00h	00h	0	0
EC:	confirm	TSDO	58xh	4	60h	44h	20h	00h	--	--	--	--



Example 9: Stop UPRG2 a run UPRG1 from address 0FF42h (addresses are allocated by compilation)

UPRG2 index: 2044h subi.: 00h data: 0000h -> stop
UPRG1 index: 2043h subi.: 00h data: FF42h -> start

SDO user program – task 2		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	stop task 2	RSDO	603h	8	22h	44h	20h	00h	00h	00h	00h	00h
EC:	confirm	TSDO	583h	4	60h	44h	20h	00h				

SDO user program – task 1		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	run task 1	RSDO	603h	8	22h	43h	20h	00h	42h	FFh	00h	00h
EC:	confirm	TSDO	583h	4	60h	43h	20h	00h				

2045h Clear functions

SDO clear function		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	clear	RSDO	60xh	4/8	22h	45h	20h	ff	0	0	0	0
EC:	confirm	TSDO	58xh	4	60h	45h	20h	ff				

ff - function (subindex) clear see **T_CAN** table



Example 10: Clear position - motor is in standstill
index 2045h subi.: 01h

SDO clear function		COB	ID	DLC	data1	data2	data3	data4	data5	data6	data7	data8
MA:	clear position	RSDO	603h	4	22h	45h	20h	01h				
EC:	confirm	TSDO	583h	4	60h	45h	20h	01h				



PMControl s.r.o.
Krásnohorské 994
547 01 Náchod
tel.: +420 491 424 821
fax.: +420 491 424 821
e-mail: info@pmcontrol.cz
<http://www.pmcontrol.cz>