www.triomotion.com

Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Ph: +1 412.968.9744
Fx: +1 412.968.9746

Doc No.: **TN20-50**
Version: **1.1**
Date: **21st April 2004**
Subject: **Reading and Writing CanOpen Parameters**

Technical Note

Overview:

CAN Open parameters can be read and written using a single CAN function (software version 1.63 and higher). The read/write functions use the CanOpen Service Data Object (SDO) protocol as defined in CiA Draft Standard 301.

Each SDO transfer requires 2 separate CAN identifiers or COB-IDs. One COB-ID is for the outgoing telegram(s) and the second one for the incoming telegram(s). Two CAN Buffers must be initialised in the Motion Coordinator prior to executing the CanOpen SDO commands, one for read and one CAN buffer for write. It is recommended to use the same buffers as Motion Perfect does, which are buffers 13 and 1.

In the following descriptions and examples;
ch is the slot number of the CAN card (0 to 3), or -1 for the built-in CAN port.
slave is the slave Node-ID.

Set SDO read/write buffers:

Two of the avalable 16 buffers must be allocated to be the read and write channels for the SDO transfers. The buffers are initialised to transfer 8 bytes per CAN telegram.

```
' set slave Node-ID
  slave = 5
' Buffer 13 is for writing "objects", buffer 1 is for reading response
  CAN(ch, 5, 13, $600+slave, 8)
  CAN(ch, 5, 1, $580+slave, 8)
```

Read CanOpen Object:

This function initiates a CanOpen SDO read sequence. If the read is successful the function returns TRUE and the data is stored as bytes in 8 VRs, otherwise a FALSE is returned.

`can_flag = CAN(ch, 8, transbuf, recbuf, object, subindex, vrstart)`

The object can be 8, 16, 32 or string type. The type is returned in the first VR(vrstart). Strings are returned as length, char1, char2..., 32 bit numbers are returned in 2 subsequent 16 bit words.

transbuf:	transmit buffer number, 13 in the above example
recbuf:	receive buffer, 1 in the above example
object:	CAN Open object number
subindex:	CAN Open object sub-index number
vrstart:	The VR location where the returned type and value is to be placed.
	8 bit numbers are returned in 2 VRs. Values: $4F, value of object.
	16 bit numbers are returned in 2 VRs. Values: $4B, value of object.
	32 bits are returned in 3 VRs. Values: $43, object value 16 lower bits, object value 16 upper bits.
	Strings are returned in N+1 VRs. Values: $41, N, char1, char2, ..charN.

Write CanOpen Object:

This function initiates a CanOpen SDO write sequence. If the write is successful the function returns TRUE, otherwise a FALSE is returned.

`can_flag = CAN(ch, 9, transbuf, recbuf, format, object, subindex, value, {valuems})`

Format can be 8,16 or 32 for 8,16, or 32 bits. *Strings cannot be written using this function.*

transbuf:	transmit buffer number, 13 in the above example
recbuf:	receive buffer, 1 in the above example
format:	Either 8, 16 or 32
object:	CAN Open object number
subindex:	CAN Open object sub-index number
value:	The value to be written (8 and 16 bit transfers) or the lower 16 bits of a 32 bit transfer.
valuems:	The upper 16 bits of a 32 bit write.

Example CanOpen Communication:

This is an example of communication setup for a typical CanOpen slave node. The read and write object commands are examples only and the exact sequence of objects that need to be set up will vary.

```
  IF CAN(0,2)<>1 THEN CAN(0,2,0) 'set baudrate to 1000kHz

' Initialise NMT buffer (cob_id = 0, data length = 2 bytes)
  CAN(ch, 5, 14, 0, 2)

' set slave Node-ID
  slave = 5

' Set buffer 13 for writing SD0s and buffer 1 for reading response
  CAN(ch, 5, 13, $600+slave, 8)
  CAN(ch, 5, 1, $580+slave, 8)

 ‘ open link to the slave
  CAN(ch, 7, 14, 01, slave)

  ' Set the drive mode using DS402 object $6060
  CAN(0,9 ,13,1,8,$6060,0,$0002,$0000)

  ‘ read the standard object “device type”
  can_ok = CAN(ch,8,13,1,$1000,0,vrbase)
  IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
    PRINT #5,"Device type = ";VR(vrbase+1)+VR(vrbase+2)*256[0]
  ELSE
    PRINT #5,"Error reading SDO 1000"
  ENDIF
```

A CanOpen Standard Object Reader:

This is a reader that gets a selection of the DS301 standard objects from the slave. It demonstrates the use of various data formats like byte, long word and string.

```
' Initialise some variables
ch = 0 '   Can port used on Motion coordinator
slave=1 ' Slave node address
vrbase=500

IF CAN(0,2)<>1 THEN CAN(0,2,1) 'set baudrate to 500kHz

CAN(0,5,7,0,2)'  NMT cob_id = 0

' Open connection to remote CanOpen Node
CAN(ch,7,7,01,slave)

' Set up the read and write buffers
CAN(ch,5,13,$600+slave,8)
CAN(ch,5,1,$580+slave,8)
WA(20)

' Read Standard Objects from CANopen Slave
  PRINT #5,""
```

```
' 32 bit read
can_ok = CAN(ch,8,13,1,$1000,0,vrbase)
IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
  PRINT #5,"Device type = ";VR(vrbase+1)+VR(vrbase+2)*256[0]
ELSE
  PRINT #5,"Error reading SDO 1000"
ENDIF

' byte read
can_ok = CAN(ch,8,13,1,$1001,0,vrbase)
IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
  PRINT #5,"Error register = ";(VR(vrbase+1))[0]
ELSE
  PRINT #5,"Error reading SDO 1001"
ENDIF

' String read
can_ok = CAN(ch,8,13,1,$1008,0,vrbase)
IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
  PRINT #5,"Device Name = ";
  FOR c=1 TO VR(vrbase+1)
    PRINT #5,CHR(VR(vrbase+1+c));
  NEXT c
  PRINT #5,""
ELSE
  PRINT #5,"Error reading SDO 1008"
ENDIF

can_ok = CAN(ch,8,13,1,$1009,0,vrbase)
IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
  PRINT #5,"Hardware version = ";
  FOR c=1 TO VR(vrbase+1)
    PRINT #5,CHR(VR(vrbase+1+c));
  NEXT c
  PRINT #5,""
ELSE
  PRINT #5,"Error reading SDO 1009"
ENDIF

can_ok = CAN(ch,8,13,1,$100a,0,vrbase)
IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
  PRINT #5,"Software version = ";
  FOR c=1 TO VR(vrbase+1)
    PRINT #5,CHR(VR(vrbase+1+c));
  NEXT c
  PRINT #5,""
ELSE
  PRINT #5,"Error reading SDO 100a"
ENDIF
```

```
  ' 32 bit read
  can_ok = CAN(ch,8,13,1,$100b,0,vrbase)
  IF (VR(vrbase)<>128) AND (can_ok=TRUE) THEN
    PRINT #5,"Node ID = ";VR(vrbase+1)+VR(vrbase+2)*256[0]
  ELSE
    PRINT #5,"Error reading SDO 100b"
  ENDIF

STOP
```