

SMCAN

SMCAN is an easy way to utilize all the advantages with a CANmaster-master type transport layer. It is built into the servo motors and can be used to synchronize motion or share data between motors. It can be used between motors without the need of any external control system. Or it can coexist together with other CAN protocols on the same bus. SMCAN uses booth CAN 29 and 11bit frames on the same bus.

SMCAN 2.0B "29bit"

The purpose of this protocol is to be a simple and efficient way of communicating with Simplex Motion motors, and for motors themselves to exchange information in an efficient way.

The 29bit frame contain all the necessary information for write and read of registers in a SMCAN network, without the need of any pre configuration.

SMCAN 2.0A "11bit"

SMCAN use 11bit frames for motor-to-motor communication but can also be used to set and get values from motors in an SMCAN network.

1 SMCAN registers

| SMCAN registers | | | | |
|-----------------|-------|-------------------|--|------------------------------|
| No. | Type | Name | Description | Range |
| 5000 | uns16 | CANRx1ID | Lower 8bits of Arbitration Field. Set this to the TX id to listen for | 0..255 |
| 5001 | uns16 | CANRx1Parameter1 | Register to put received data in | Any writable register number |
| 5002 | uns16 | CANRx1Parameter2 | Register to put received data in | Any writable register number |
| 5003 | uns16 | CANRx1Parameter3 | Register to put received data in | Any writable register number |
| 5004 | uns16 | CANRx1Parameter4 | Register to put received data in | Any writable register number |
| 5500 | uns16 | CANTx1ID | Lower 8bits of Arbitration Field. Set this to a desired but unique value for the CAN network | 0..255 |
| 5501 | uns16 | CANTx1Parameter1 | Register to send | Any register number |
| 5502 | uns16 | CANTx1Parameter2 | Register to send | Any register number |
| 5503 | uns16 | CANTx1Parameter3 | Register to send | Any register number |
| 5504 | uns16 | CANTx1Parameter4 | Register to send | Any register number |
| 5505 | uns16 | CANTx1RefreshRate | How often to send the message in ms, 0=off, 1-65535ms | 0..65535 |
| 6000 | uns16 | CAN_HB_SETTINGS | Bit 0-1: Error Mode, 0= None, 1= QuickStop, 2= Shutdown, 3= Off. Bit 2: If edge or active, i.e. set mode on edge only or force mode if active Bit 15: Enable heartbeat function | |

| | | | | |
|------|-------|-----------------|--|--|
| 6001 | uns16 | CAN_HB_STATUS | Reflects the status of the CAN ID's 6010,6020... | |
| 6002 | uns16 | CAN_HB_RRATE | Timeout, i.e. time to wait for messages. | |
| 6010 | uns16 | CAN_HB1_CAN_ID | CAN id to listen for on the CAN bus | |
| 6011 | uns16 | CAN_HB1_CAN_CNT | Counter in ms since last received. | |

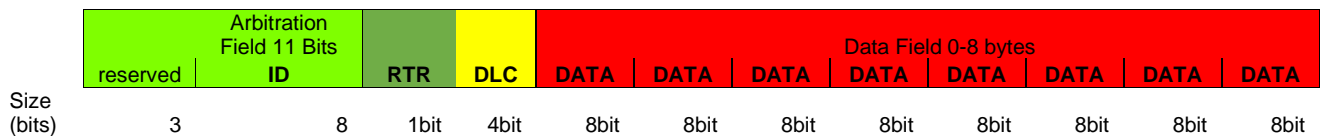
2 11bit frames "CAN 2.0A"

Used for pushing values

The Arbitration field is used as message ID.

Data is configured for each ID and sent as little endian.

Data is sent as 16bit values therefore DLC is valid if DLC is 2,4,6 or 8



Example: Set data

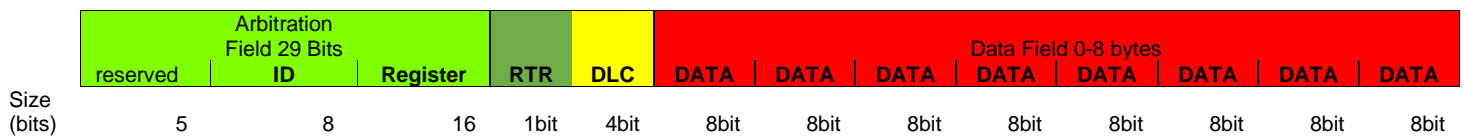
| | | | | | | | | | | | |
|---------------|---|---|---|---------|------------|----|----|----|----|----|----|
| Node1→ set | 2 | 0 | 2 | pay lsb | pay msb | Na | Na | Na | Na | Na | Na |
|---------------|---|---|---|---------|------------|----|----|----|----|----|----|

3 29Bit frame "CAN2.0B"

Used for exchanging parameters, RTR frames are used as read request.

The Arbitration field is used as unit ID and Register/parameter see below

- ID/Address: Every node has a unique id/address
- Register: Command or parameter to call/read/write
- RTR: Write=0, Read=1
- DLC: Number of bytes to write (when RTR=0) or number of bytes to read (when RTR=1).
- Data is sent and returned as little endian.



Example 1: Request data

| | | | | | | | | | | | | |
|------------------------|---|-----|---|---|---------|-----|-----|------------|----|----|----|----|
| Node1→Node2 request | 2 | 200 | 1 | 4 | Na | Na | Na | Na | Na | Na | Na | Na |
| Node2→Node1 reply | 2 | 200 | 0 | 4 | pay lsb | pay | pay | pay msb | Na | Na | Na | Na |

Example 2: Set data

| | | | | | | | | | | | | |
|----------------------|---|-----|---|---|---------|------------|----|----|----|----|----|----|
| Node1→Node2 set | 2 | 400 | 0 | 2 | pay lsb | pay msb | Na | Na | Na | Na | Na | Na |
| Node2→Node1 reply | 2 | 400 | 0 | 2 | pay lsb | pay msb | Na | Na | Na | Na | Na | Na |

4 Configuration/setup of SMCAN 11bit

Each motor has 8 receive and 8 transmit registers along with some other control registers,

CANTxnID, n=1-8:

This is the ID value of the message that will be sent. This is also the arbitration/address field of the CAN frame and should be a unique value. **i.e., don't set up the same value on motors in the same network.**

CANTxnParametry, n=1-8 & y=1-4:

Each TxID has the possibility of sending out 4 16bit parameters or 2 32bit or any mix of them.

Example, To synchronise the position of two motors you can send out the position and mode.

CANTx1Parameter1= 200 // position low bits "position is 32bit"
CANTx1Parameter2= 201 // position high bits "position is 32bit"
CANTx1Parameter3= 400 // mode

CANTxnRefreshRate, n=1-8:

This is the rate the message will be sent at, 0=off 1-65535ms

Example sending:

This will send position and mode at a rate of 10 frames/sec.

CANTx1ID= 1
CANTx1Parameter1= 200 // position low bits "position is 32bit"
CANTx1Parameter2= 201 // position high bits "position is 32bit"
CANTx1Parameter3= 400 // mode
CANTx1RefreshRate= 100

CANRxID, n=1-8:

This is the ID value of the message that you should receive, note that this can be the same on other motors in the network if they all need the same info.

CANRxParametry, n=1-8 & y=1-4:

Each RxID has the possibility of receiving 4 16bit parameters or 2 32bit or any mix of them.

Example, to synchronise the position of two motors you can send out the position and mode see TX example above,

The receiving parameters for that example would look like this.

CANRx1ID= 1 // listen for message with id1
CANRx1Parameter1= 450 // put position in parameter target input
CANRx1Parameter2= 451 // put position in parameter target input
CANRx1Parameter3= 620 // put the mode in parameter ApplData1, this way we can use events to set mode or something based on this info.

5 CAN Heartbeat

Heartbeat functionality can be used to monitor the existence of certain messages on the CAN network, it can also be used to set motors in a safe/known state if a message is not received.

CAN HB SETTINGS

Bit 0-1, ErrMode

The state to enter on a timeout valid modes are.

0= None, 1= QuickStop, 2= Shutdown, 3= Off.

If none mode is not changed on timeout but status is still monitored.

Bit 2: Edge

If edge or active, i.e. set mode on edge only or force mode if active

Bit 15: Enable heartbeat function

Enabled, Enable heartbeat function.

CAN HB STATUS

This register reflects the state of the enabled IDs to monitor. 0= good, 1= time out.

Bit0 = CAN_HB1

Bit15= CAN_HB16

CAN HB RRATE

Time 0-65535ms.

If a message is not received within this time the status bit in register

CAN_HB_STATUS for that CAN_ID will be set.

If mode is other than none then motor will enter that mode.

CAN HBx CAN ID

x= 1-16

The CAN message ID, 11bit CAN Arbitration Field value.

0= off, i.e. not checked.

CAN HBx CAN CNT

Time since last received, time in ms.