CAN Commands

```
CANOPEN speed
```

Configures the CAN controller for the specified communication speed and synchronizes to the CAN bus. The message reception acceptance filter is configured to receive all valid CAN message identifiers, both 11-bit STD (0..7FFh) and 29-bit EXT (0..1FFFFFFF).

The communication speed should be in the range 10 kbps to 1 Mbps.

Example:

```
CANOPEN 500000
```

```
CANCLOSE
```

Closes the connection with the CAN bus by placing the CAN controller back into its configuration mode.

Example:

```
CANCLOSE
```

```
CANSEND id, type, len, data(), ok
```

Places a CAN messages in the transmit queue and the moment a CAN transmit message slot becomes available, the CAN message is send onto the CAN bus.

Argument 'id' sets the message identifier and 'type' the identifier type. Set 'type' to 0 for a 11-bit STD message identifier (most commonly used) or to 1 for a 29-bit EXT message identifier. The number of data bytes in the messages is set by argument 'len'. This can be anywhere in the range 1..8. The actual message data is passed as a 1-dimensional array called 'data'.

Pass in a variable for argument 'ok' to determine if the message transmission was successful. In this case 'ok' holds the value 1. If it holds the value 0 then either the transmit queue is full so you can try a little later again or the connection with the CAN bus has not yet been opened (see command `CANOPEN`).

Example:

```
100 DIM txData(2) : DIM txOk
110 CANOPEN 500000
120 txData(0) = &H55
130 txData(1) = &HAA
140 CANSEND &H123,0,2,txData(0),txOk
150 IF txOk = 0 THEN PRINT "Could not send CAN message"
160 CANCLOSE
```

`CANRCV id, type, len, data(), ok`

Checks if a CAN message was received and is stored in the internal receptio queue.

The message identifier is stored in the variable that is passed as argument 'id'. The identifier type is stored in the variable that is passed as argument 'type'. A value of 0 means that it is a 11-bit STD identifier and a value of 1 means that it is a 29-bit EXT identifier. The number of data bytes in the messages is stored in the variable that is passed as argument 'len'. The actual received data bytes is stored in the 1-dimensional array that is passed as argument 'data'.

Pass in a variable for argument 'ok' to determine if a messages was received. In this case 'ok' holds the value 1. If it holds the value 0 then either there was no message present in the reception queue or the connection with the CAN bus has not yet been opened (see command `CANOPEN`).

Example:

```
100 DIM rxId : DIM rxType : DIM rxLen : DIM rxData(8) : DIM rxOk
110 CANOPEN 500000
120 DO
130 CANRCV rxId,rxType,rxLen,rxData(0),rxOk
140 IF rxOk = 1 THEN
150 PRINT "CAN message received"
160 EXIT
170 ENDIF
180 LOOP
190 CANCLOSE
```