Nibjeppo Can Tool Sequences

Sequences for the 'Ninjeppo CAN Tool' are simple script to interact with CAN network.
The form of script is quite simple

SEQUENCE <name>
….
….
ENDSEQ

In order to perform different or repetitive tasks, sequences support LOOP / ENDLOOP and SELECT / CASE / ENDSEL statment

Example 1
SEQUENCE send one message
SENDMSG ID=1 DATA=1,2,3
ENDSEQ

This simple script send a CAN message with ID=1, length=3 and data [1,2,3] on the bus.

Example 2
SEQUENCE send my mesasges
SENDMSG ID=1 DATA=1,2,3
SENDMSG ID=2 DATA=1,2,3
SENDMSG ID=3 DATA=1,2,3
SENDMSG ID=4 DATA=1,2,3
ENDSEQ

This simple script send a sequence of 4 CAN message with ID 1, 2, 3 ,4, length=3 and data [1,2,3]

SEQUENCE select case
SELECT
CASE ID=1
CASE ID=2
SENDMSG ID=3 DATA=1,2,3
SENDMSG ID=4 DATA=1,2,3
BREAK
CASE ID=100
SENDMSG ID=103 DATA=1,2,3
SENDMSG ID=104 DATA=1,2,3
BREAK
DEFAULT
SENDMSG ID=203 DATA=1,2,3
SENDMSG ID=204 DATA=1,2,3
ENDSEQ

This simple script wait for a CAN message. If the massage has ID=1 or ID=2, regardless of payload the tool will send a sequence of two message (ID 2 and ID 3). If the message has ID=100, the tool will send a sequence of two message (ID 103 and ID 104). If the message has another ID script will fall into default condition an the tool will send two message (ID 203, ID 204).

we can run also some loops

```
SEQUENCE my loop
LOOP FOREVER
SENDMSG ID=1
ENDLOP
ENDSEQ
```

this script will send an infinite sequence of message length 0 and ID 1

```
SEQUENCE complex
LOOP FOREVER
SENDMSG ID=1
SELECT 100
CASE ID=2
EXITLOOP
ENDSEL
ENDLOOP
```

this script will send a messsage with ID=1 every 100ms until it will receive a message with ID=2. In this case it will exit form loop

Reference Guide

SEQUENCE <name>
Start of a sequence

SENDMSG ID=<id> D0=1 D1=2....
SENDMSG ID=<id> DATA=1,2,3,4
Send a message on the bus

DELAY <timeout millisecondi>
Generic delay

WAITMSG ID=<id> [MASK-ID=<mask>] D0=1 D1=2 D3=3
WAITMSG ID=<id> [MASK-ID=<mask>] DATA=1,2,3,4
Halt script waiting for a massage. You can specify also a MASK-ID= in the arguments.

LOOP <count> | FOREVER
Create a loop

EXITLOOP
Exit form a loop. Similar to BREAK in C language

ENDLOOP
End of loop

SELECT [Timeout]
Start a select case branch

CASE ID=<id> [MASK-ID=<mask>] D0=1 D1=2 D3=3
CASE ID=<id> [MASK-ID=<mask>] DATA=1,2,3,4
Start a branch for selected message. You can specify also a MASK-ID= in the arguments.

DEFAULT
Start a branch for other messagge

EXPIRED
Start a branch when timeout expires

BREAK
Terminate a branch

ENDSEL
End of SELECT statment

PRINT <text>
Display text on user interface

ENDSEQ
End of sequence