

## TECH NOTE : catman Script – sending CAN (FD) messages

Version: 2020-02-03

Author: Manuel Schultheiß, Product Manager Test & Measurement, HBM Germany

Status: public

### Abstract

The TECH NOTE describes how to send CAN signals with catman AP via the embedded script functionality.

### Intro

CAN communication is a common standard in industry. QuantumX allows to send all incoming analog data to CAN bus in an easy way by just mapping inputs to outputs. But In some use cases it is interesting sending online calculated signals or alarms via CAN to another device.

The Script functionality of catman allows to send CAN signals.

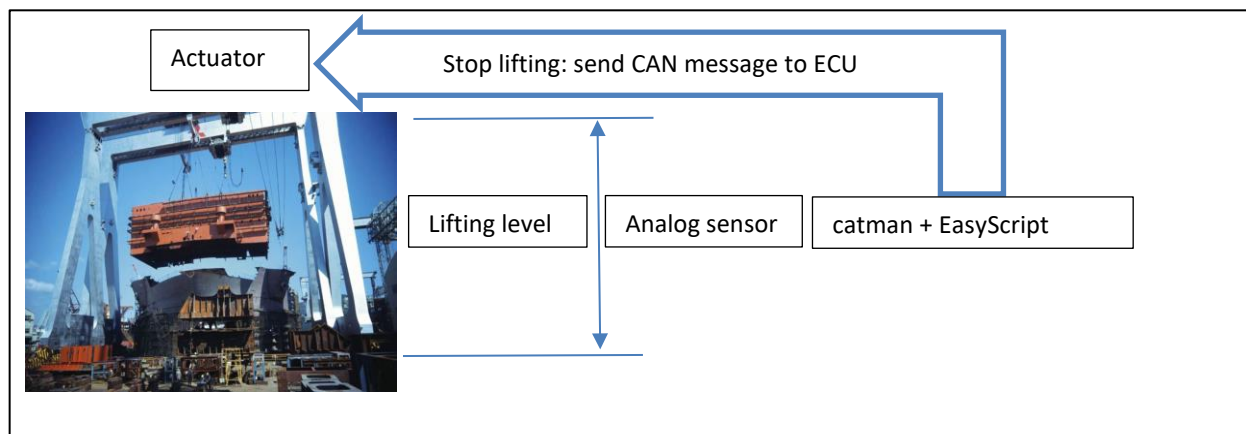
### Example

In the following example the lifting level of a harbor crane is measured by analog sensors.

catman is doing all online calculation and knows the limits of the overall test.

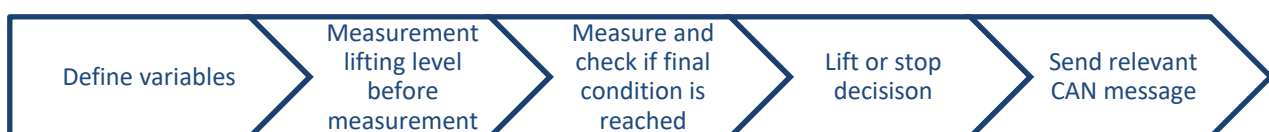
catman script shall send out a CAN message in the following cases:

1. The crane has not reached it's stop condition (lifting mechanism needs to be active)
2. The crane has reached stop or top level condition



### Approach 1: Using catman EasyScript

Two code examples show how CAN signals can be sent from catman AP.



The script example is divided into subroutines, so that the individual subs can be executed in CatmanAP at certain execution times.

### 1. Define variables

#### Sub Main

```
Dim Counter As Integer 'Counter Loop 1
Dim Counter2 As Integer 'Counter Loop 2
Dim h As Double 'Lifting_level at measurement start
Dim a As Double 'Current Lifting_level
Dim b As Long 'Statusbit measurement channel
Dim m(7) As Byte 'CAN-Telegram Array
```

#### End Sub

#### Sub Main

```
h = 0
m(0) = 255 'setpoint lifting function
m(1) = 0
m(2) = 0
m(3) = 0
m(4) = 16 'Direction lifting function "Lifting"
m(5) = 0
m(6) = 0
m(7) = 0
EA_Panel.SetValue(1, "TEXT_4", "n/a")
```

#### End Sub

### 2. Measurement lifting level before measurement

#### Sub Measure\_Lifting\_level\_before\_measurement\_start

```
EA_IO.Measure("Lifting_level", h, b)
EA_Panel.SetValue(1, "TEXT_4", h)
```

#### End Sub

### 3. Measurement of lifting level and calculation of difference and setting of counter

#### Sub Lifting\_level\_measure

##### Lifting\_Command

```
EA_DAO.CurrentReading("Lifting_level", a, DAQ_MAX)
Counter=a-h
EA_Panel.SetValue(1, "TEXT_1", Counter)
Counter > 100 Then
```

##### Stopp\_Command

##### End Sub

'Sende Lifting-Command

'Messe Lifting\_level

'Delta Lifting\_level

Counter > 100

'Stop when 100mm lifted

Counter <= 100

#### Sub Lifting\_Command

```
EA_IO.SendCANMessage (1, 0, 642, 8, m, 1) 'Sende Lifting-Command
EA_Panel.SetValue(1, "TEXT_1", Counter)
```

#### End Sub

#### Sub Stopp\_Command

```
m(0)=0
m(4)=0 'Sollwert und Richtung Hubfunktion 0
```

Do

```
EA_IO.SendCANMessage (1, 0, 642, 8, m, 1)
```

'Sende Stopp-Command

Counter2=Counter2+1

Loop Until Counter2 = 100

'Für 100 Telegramme

MsgBox "Stopp-Command 100 mal gesendet!"

EA.Terminate(TERMINATE\_SCRIPT)

End Sub

Script  
function  
to send  
CAN  
messages  
in catman

**EA\_IO.SendCANMessage**

Sends a CAN message from a MGCplus ML71 or a QuantumX MX840 or MX471 CAN connector.

```
EA_IO.SendCANMessage(ByVal Device As Variant, ByVal Channel As Variant, ByVal MsgID As Long, ByVal ByteCount As Integer, Bytes() As Byte, Optional ByVal Port As Integer) As Long
```

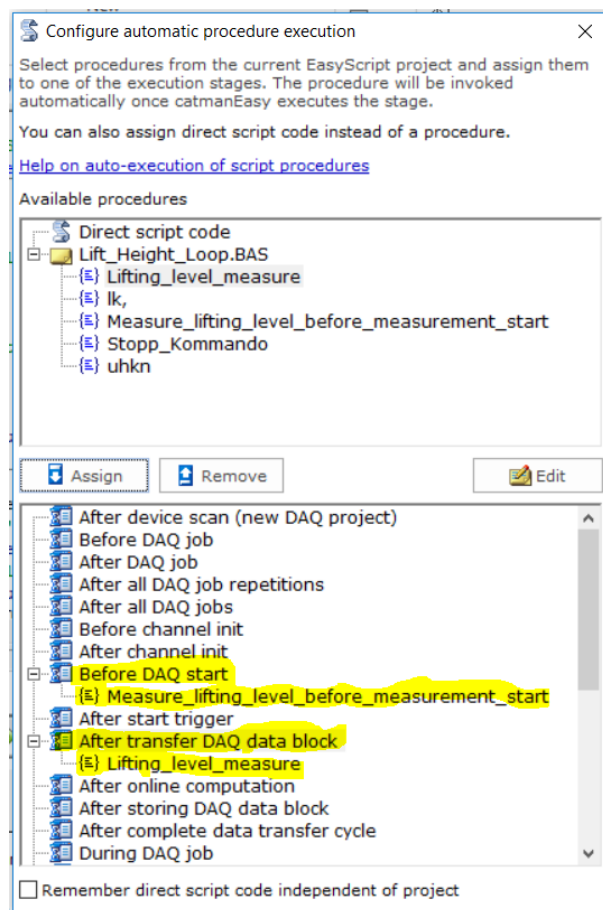
**Parameter Description**

Device	Either the name or the index of the device to which the command is to be sent. The first device in a DAQ project has the index 1.  Only required for MGCplus ML71. The channel is used to determine the slot position of the ML71 amplifier board. It therefore doesn't matter which sub-channel (CAN signal) of the connector you specify - catman will automatically determine this.
Channel	The channel can be specified via name or position. The first channel in a DAQ project has the position 1.  Specify 0 or empty string if used with MX840 or MX471
MsgID	CAN message ID. If the ID is larger than 2048, catman will set the Extended Frame Format bit ID automatically. To specify an ID in hexadecimal form use the VBA &H prefix, e.g. 255 = &HFF.
ByteCount	Number of bytes the message consists of.
Bytes()	Array (0...7) containing the bytes the message consists of.
Port	CAN port (connector) on which the command will be sent: ML71: 1 or 2 MX840: always 1 MX471: 1,2,3 or 4

**Return value**

0: No error  
<0: Error, e.g. not supported by module type.

Check that the subroutines are executed at the right time of the measurement. Use the “Auto execution”:



-- end

**Legal Disclaimer:** TECH NOTEs are designed to provide a quick overview. TECH NOTEs are continuously improved and so change frequently. HBM assumes no liability for the correctness and/or completeness of the descriptions. We reserve the right to make changes to the features and/or the descriptions at any time without prior notice.