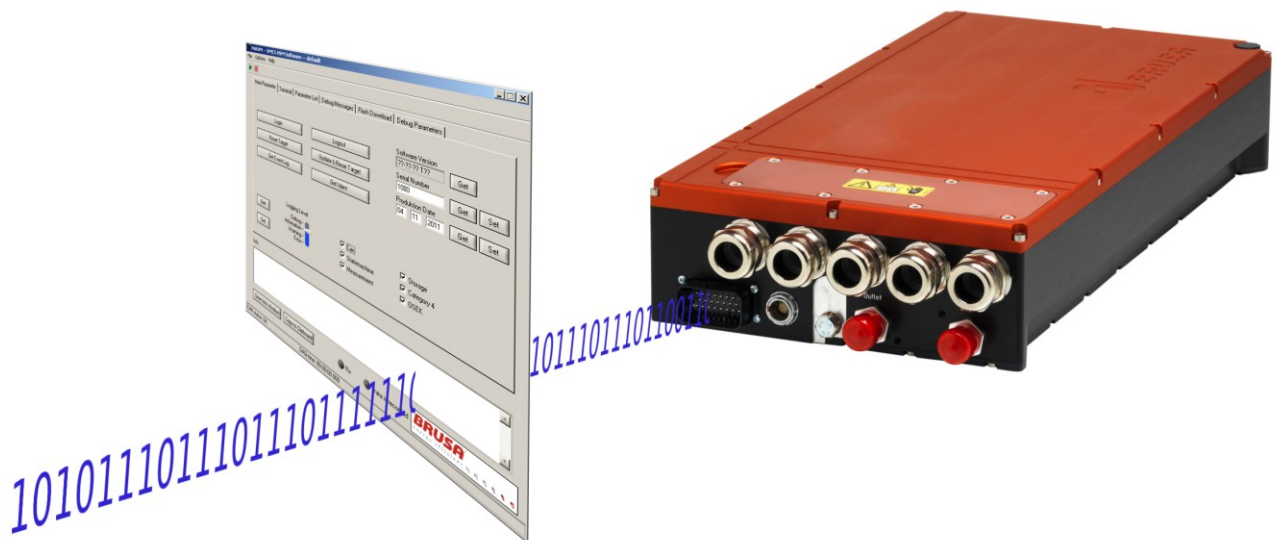


# PARAM Manual (Installation & Usage)

Version 0.1



## 1 Document History

Version	Date	Name	Comment
0.1	14.07.2011	Peter Oehry	New document

## 2 Introduction

This document gives an overview of how to install the PARAM tool and the associated drivers. Principle steps of using the PARAM is also subject of the document.

### 3 Table of contents

<b>1</b>	<b>Document History .....</b>	<b>2</b>
<b>2</b>	<b>Introduction .....</b>	<b>2</b>
<b>3</b>	<b>Table of contents .....</b>	<b>3</b>
<b>4</b>	<b>System requirements .....</b>	<b>4</b>
<b>5</b>	<b>Installation .....</b>	<b>4</b>
5.1	PCAN-USB OEM driver .....	4
5.2	PEAK Driver .....	5
5.3	Settings .....	5
5.4	PARAM tool installation .....	6
5.4.1	Problem with mesa.dll .....	7
5.5	Updating from old Versions .....	7
5.6	Setting up the PARAM .....	7
5.6.1	StartParam.cmd .....	8
5.6.2	XML file .....	8
<b>6</b>	<b>Principle operation .....</b>	<b>9</b>
6.1	Main Window .....	9
6.2	Check communication .....	10
6.3	Parameter List .....	11
6.3.1	Make changed parameters active .....	12
6.4	Terminal .....	13
6.5	Debug Messages – Data Logging .....	14
6.6	Flash Download (motor table for inverter) .....	15
6.6.1	Steps to program a new motor table .....	15
6.6.2	Identify motor table .....	16
6.7	Get diagnostic information for BRUSA .....	17
6.7.1	Serial number .....	17
6.7.2	CAN log .....	17
6.7.3	Software Version and Event Log .....	17
6.7.4	Get all parameters .....	18
<b>7</b>	<b>Appendix .....</b>	<b>19</b>
7.1	Command line options .....	19
7.2	Using PCAN tools to check if the CAN is working .....	19
7.2.1	PCAN Status Display .....	19
7.2.2	PCAN View .....	20
7.3	The PARAM tool does not start .....	20
7.4	Find target with different PARM CAN ID's .....	21
7.5	Change CAN ID or CAN baud rate in XML file .....	22

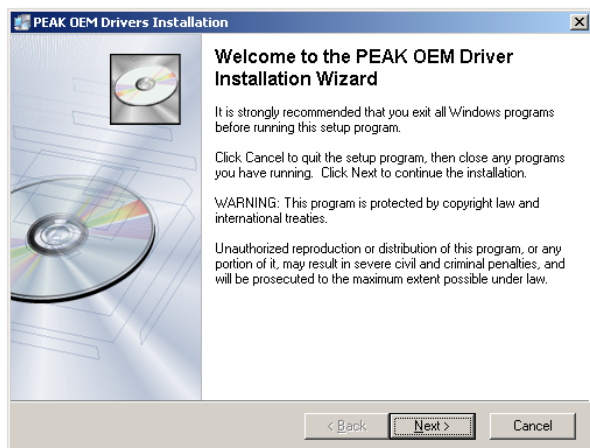
## 4 System requirements

The PARAM and the associated drivers are designed to work on Windows 7/Vista/XP (32-bit) computer. Make sure that your system meets these requirements. The communication is designed to work with a can adapter from PEAK-System Technik GmbH ([www.peak-system.com](http://www.peak-system.com)). The opto-decoupled PCAN-USB (IPEH-002022) is recommended.

## 5 Installation

### 5.1 PCAN-USB OEM driver

First install the driver for the PCAN-USB device. The driver is shipped on CD with the device or can be downloaded from the support area on the PEAK-System web site ([www.peak-system.com](http://www.peak-system.com)). Download the PCAN-USB-Package to get the latest release. The OEM driver provides the driver to connect the PCAN-USB to the PEAK subsystem. After extraction the PeakOemDrv.exe setup has to be started.



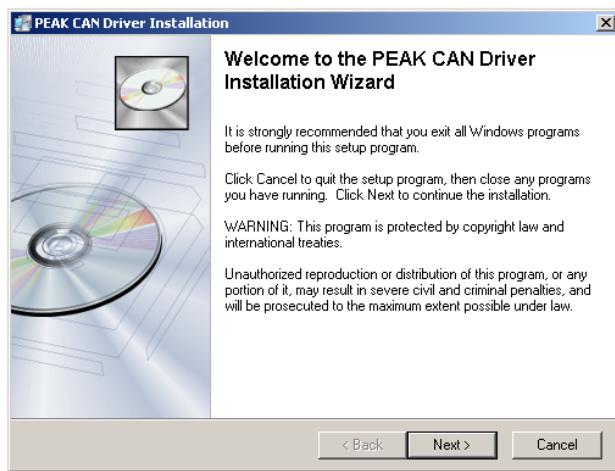
Where not noted the default settings are ok.



Select the PCAN-USB driver for installation. All other drivers and the PCAN-View application are not needed.

## 5.2 PEAK Driver

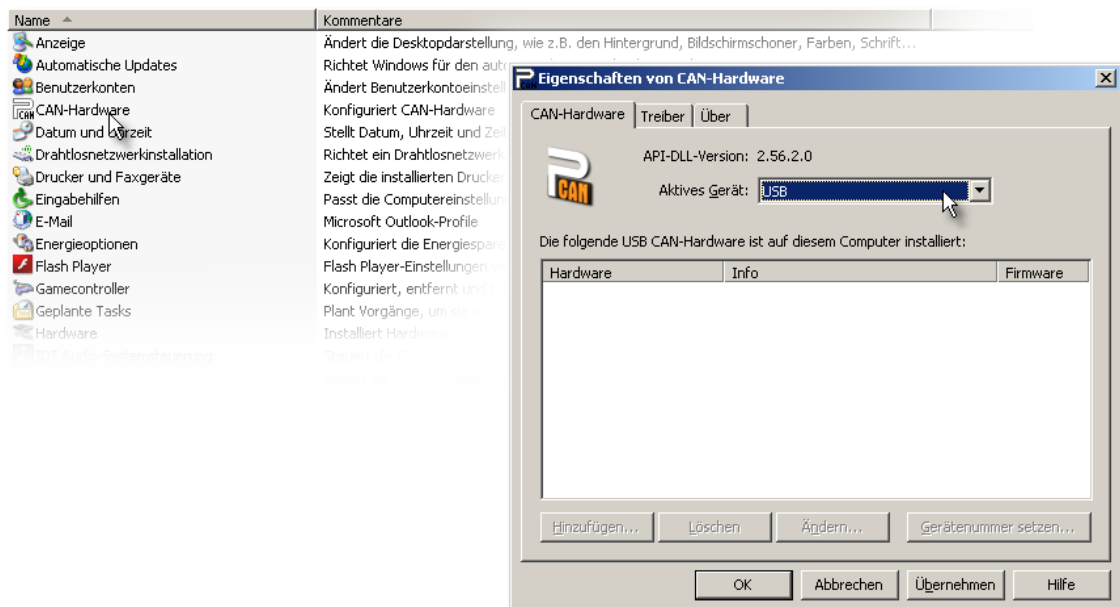
The PEAK Driver (PcanDrv.exe) is shipped with the BRUSA Software package. This is the driver that is needed by the PARAM tool to communicate with the PEAK subsystem. The driver provides the CanApi2.dll.



The package can be installed with all default settings.

## 5.3 Settings

After installing the drivers they have to be adjusted to use the USB subsystem. To do this setting a control was added to the windows control panel. Open the CAN-Hardware control and adjust it to USB as showed in the screen shot.



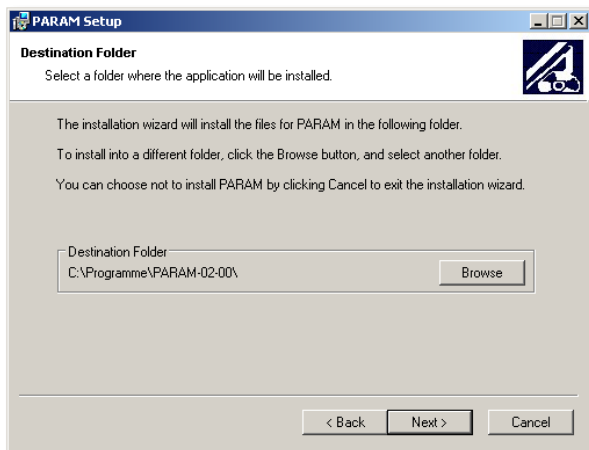
## 5.4 PARAM tool installation

The PARAM tool software is developed by BRUSA with LabWindows/CVI and relies on the Run-Time Engine from National.

To install the PARAM tool start the PARAM.msi.



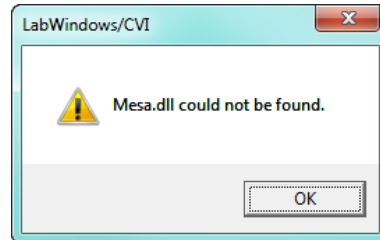
Where not noted the default settings are ok.



It is recommended to keep the default location for the destination Folder. If you change this location it might later be necessary to change some files distributed with the documentation package.

## 5.4.1 Problem with mesa.dll

On Windows 7 it is possible that the PARAM tool will not start. It complains about a missing Mesa.dll. The reason for that is an incompatibility of the included CVI Run-Time Engine with Windows 7. The PARAM has to be terminated with the Task Manager.



To solve this problem the newest Run-Time Engine has to be downloaded and installed. It is recommended to uninstall the PARAM tool first.

The latest Run-Time can be downloaded from the National website ([www.ni.com](http://www.ni.com)). Search the site for "LabWindows/CVI run time". The search result "LabWindows/CVI Run-Time Engine 2010 - Windows 7/Server 2003 R2 (32-bit)/XP x86/Vista x86/Server 2008 R2 (64-bit)" will be the right one. If in the meantime newer versions are available it should also be ok.

The PARAM tool can be reinstalled, when this Run-Time engine is installed.

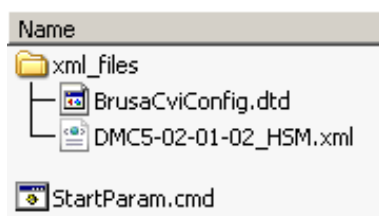
## 5.5 Updating from old Versions

If you want to update from older PEAK drivers and PARAM Tools it is highly recommended, that you remove the old versions first. PARAM tool version 01-00-xx can be installed in parallel to the actual 02-00-xx version.

For the peak drivers you can also download the "PCAN-SysInfo Tool" from the peak website. This tool helps in finding and removing old peak drivers.

## 5.6 Setting up the PARAM

The functionality of the PARAM tool is configured by an XML file. This XML file relies on the firmware of the target and will be update with every firmware release. The XML can only be valid if the file BrusaCviConfig.dtd is placed in the same directory as the XML file. It is best to us a cmd shell script to start the PARAM tool and place the XML file in a subdirectory. This gives the ability to have multiple XML configurations active in one single installation.



Example of the directory structure as it is normally shipped with the documentation package of a firmware release.

The BrusaCviConfig.dtd can also be found in the PARAM installation directory if it is missing.

### 5.6.1 StartParam.cmd

The StartParam.cmd is a short shell script with the following code. It is used to setup the location of the SW-CANAP-PARAM.exe executable and the used XML file. The XML file location is designed to be relative to the storage path of this script.

```
@echo OFF

REM -----
REM setting configuration
REM -----

set PARAM_LOCATION=%PROGRAMFILES%\PARAM-02-00
set XML_FILE=.\xml_files\DMC5-02-01-02_HSM.xml

REM -----
REM Starting PARAM...
REM -----
echo Starting PARAM from: %PARAM_LOCATION%

start %PARAM_LOCATION%\SW-CANAP-PARAM.exe XmlConfig="%XML_FILE%"

REM if you have problems you can uncomment the following line for debugging
REM pause
```

### 5.6.2 XML file

The XML file will change with every new software release and has a naming convention that identifies the product that is configured and also the corresponding firmware version.

For example DMC5-02-01-02\_HSM.xml:

DMC5	The file is for a DMC5 inverter
02-01-02	It is compatible to the firmware SW-FW-DMC5-02-01-02_HSM_BRUSA.mot
HSM	Optional additional information.

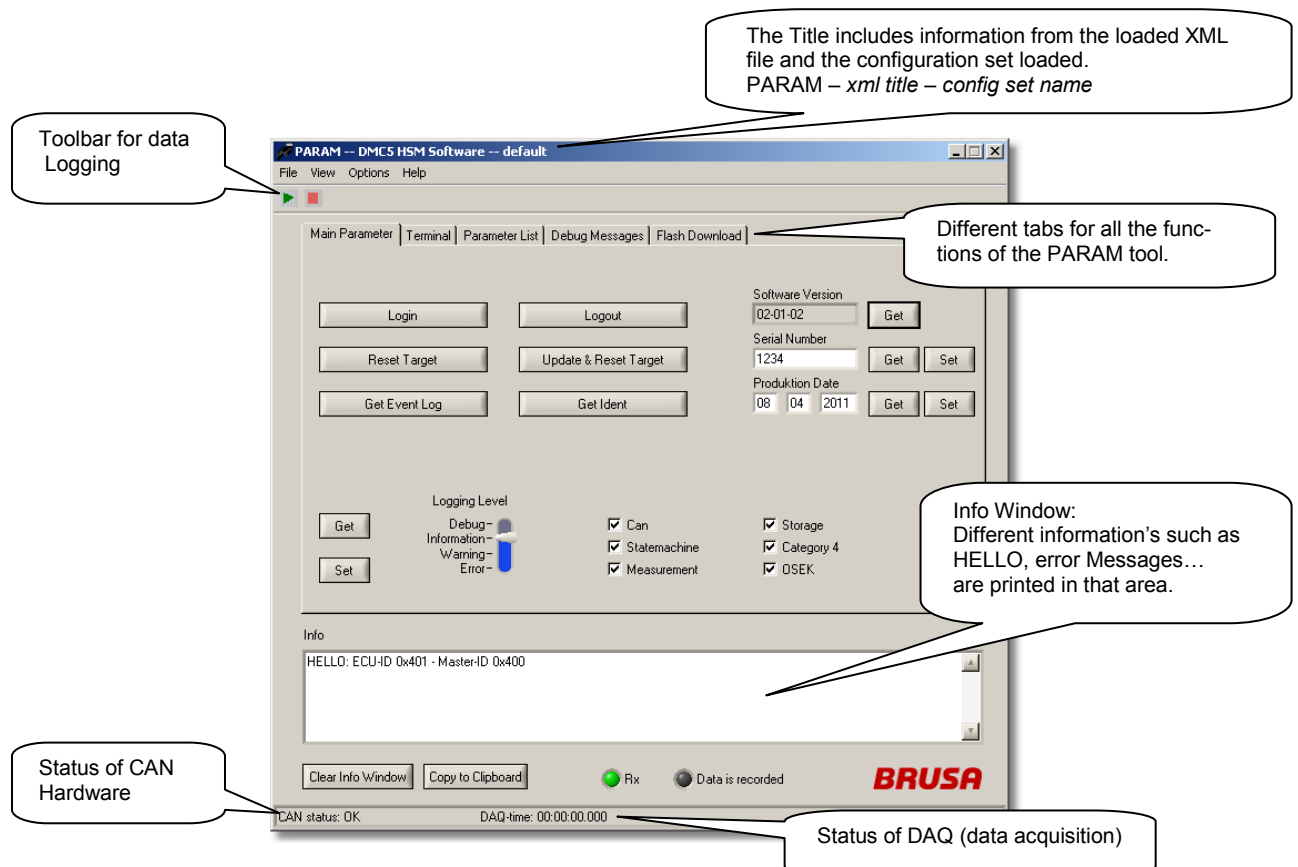


## 6 Principle operation

The PARAM tool is highly configurable (by XML file). This means also that the frontend might change depending on the configuration. Only buttons and tabs that are active are visible.

### 6.1 Main Window

The following picture gives an overview of the most important components from the PARAM tool.

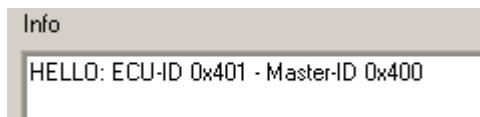


## 6.2 Check communication

First thing to do after starting the PARAM tool is to check if the communication can be established and if the XML corresponds to the software version.



Press the Button GET right to the Software Version. You get the proper version immediately if the BRUSA device is responding. The text will become red if a timeout occurs. If this is the case the communication might not work as expected. Check if the device is powered and ready for communication.



Most BRUSA devices are sending a HELLO CAN message when they start up. This message also includes the information about the proper CAN ID's for the PARAM tool communication.



In the status line you can see the status of the CAN hardware. If it is red there might be a caballing problem or the device is not responding at all.



If the status is BUSOFF the CAN hardware (on the computer) need's a reset. This can be done by one of the following actions:

1. restart the PARAM tool
2. Press STOP and GO in another BRUSA CVI application.
3. Reset the hardware in a PCAN tool.



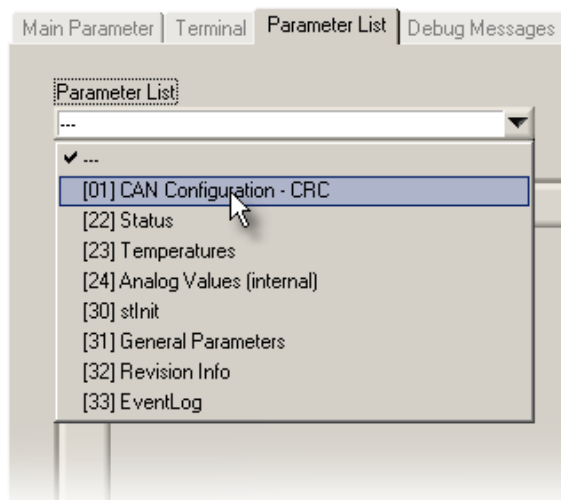
To get access to the parameters you have to be logged in into the BRUSA device. Press on the Login button and enter the password "monitor". If the password could be set you see two additional lines in the Info window below. This does only mean that the password could be written – it's not guaranteed that it was accepted to be correct.

STAT: Command OK (0)

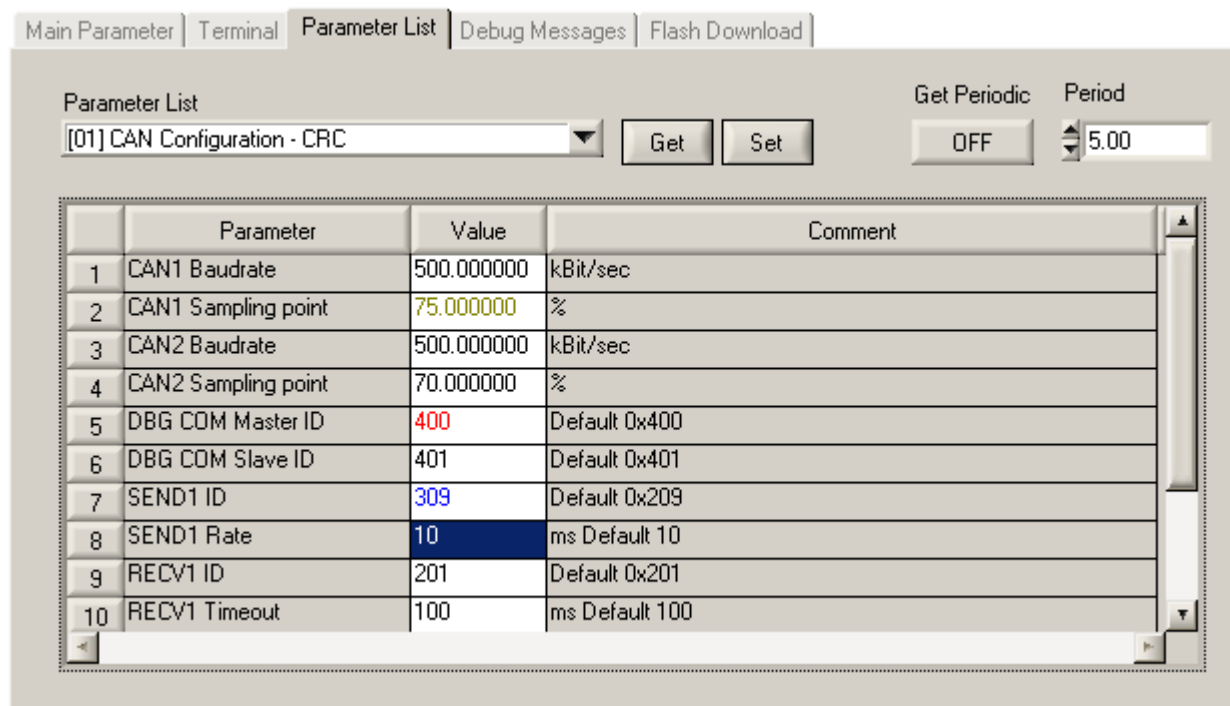
STAT: Command OK (0)

## 6.3 Parameter List

The Parameter List tab is the main access to the parameters of the BRUSA device.



Use the dropdown list to select the parameter list of interest. The lists are numbered for easier communication. The numbers are used to reference a certain parameter. The parameter [01:6] references to the 6th row in parameter list [01]. This is DBG COM Slave ID in this example.



The Get button initiates a read of all parameters in the list. This is also done when you select another list from the dropdown. You can switch on a periodic update of the list by enabling it on the top right of the list area. Don't forget to switch it off if you want to edit some parameters. Double click on a value to edit it. A changed parameter is marked blue which means that it has to be written. You can change more parameters in a list and set them together by pressing the Set button.

In the parameter list different colors are used to show the status of a parameter.

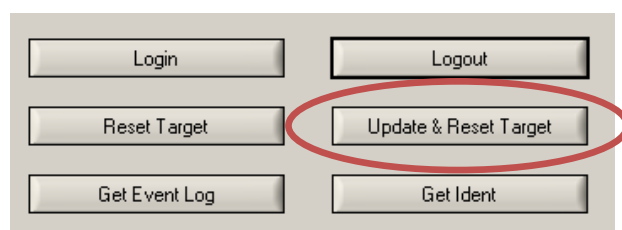
- black: the parameter was read normal
- red: the parameter could not be read
- yellow: the parameter was written, but it's value has been changed.
- blue: the parameter was changed and not yet written to the BRUSA device.
- the background is gray if a parameter is a read-only parameter.

Possible problems can be solved by looking at the info window. Different codes can be seen:

STAT: Command OK (0)	Parameter was written successfully.
STAT: No Access (4)	Read or write access was denied. Check if you are logged in or you might have now right to change the parameter.
STAT: Parameter data were changed (17)	Parameter was written but its value has been changed due to range check. The new value was sent back and the updated value is marked with a dark yellow color.
STAT: Parameter data out of range (16)	The parameter was not written because the value is out of the allowed range for that parameter. The affected value is still marked blue.

### 6.3.1 Make changed parameters active

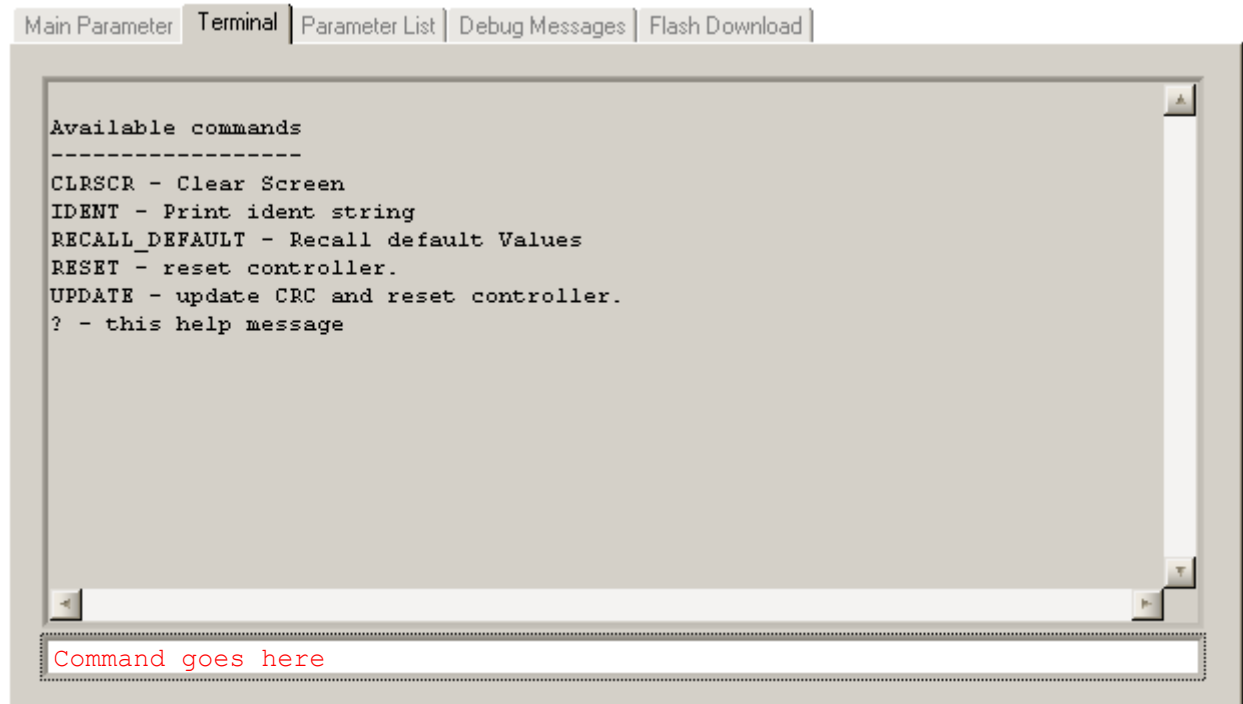
Parameters in the BRUSA device are protected by a CRC to prevent accidental changed parameters. Your changes of parameters will be ignored when the BRUSA device is restarted without forcing an update of the CRC.



By pressing the “Update & Reset Target” button the BRUSA device updates all CRC values and forces a reset. After this step the changed parameters are active.

## 6.4 Terminal

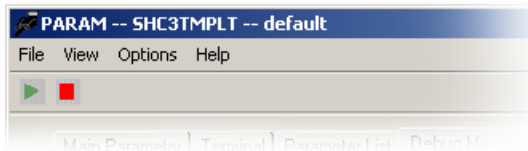
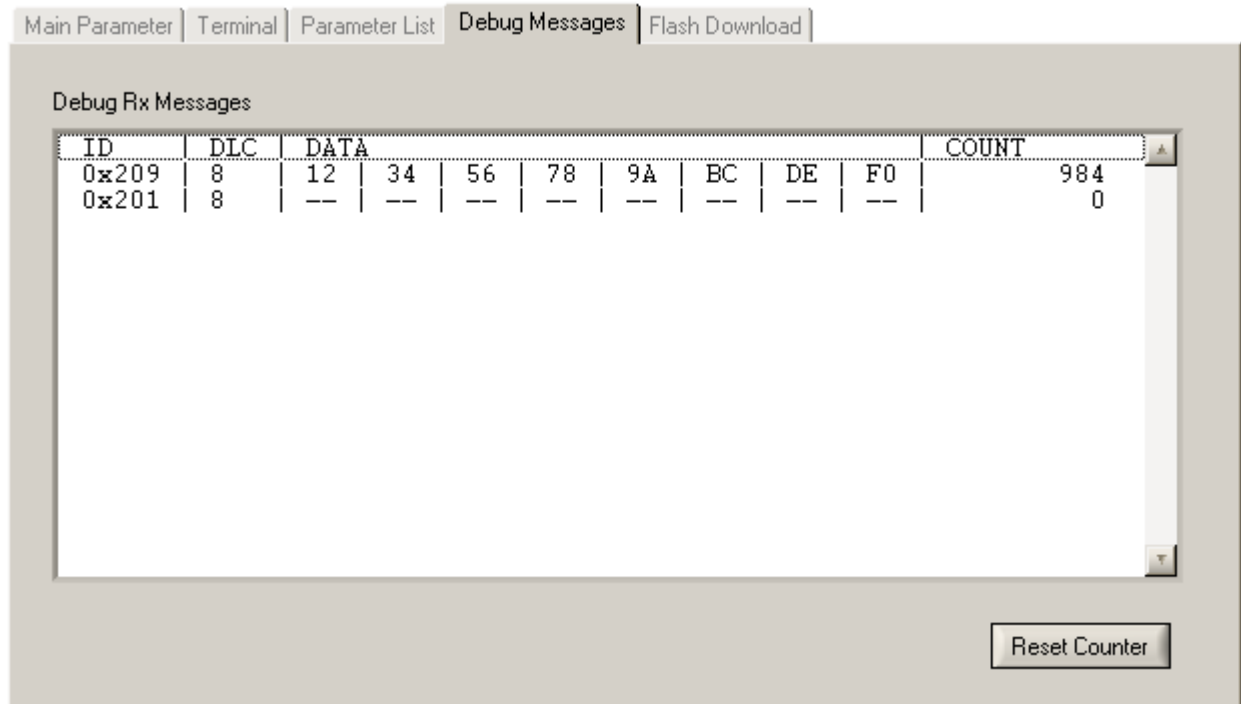
The Terminal tab is used as a text based interactive communication with the BRUSA device.



Type the command into the designated line and press the return key. If the command is returned it was not recognised by the target device. Enter a '?' to get a list of available commands.

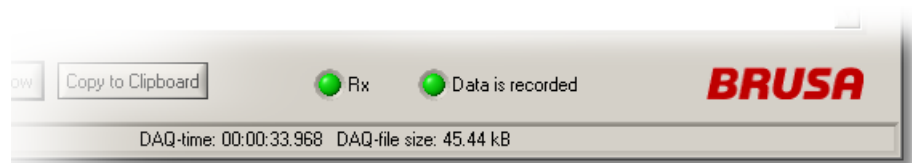
## 6.5 Debug Messages – Data Logging

The Debug Messages Tab lists all registered CAN messages. They are listed in the XML file. The view can be used to see if all expected messages are on the bus. The PARAM tool can save a log file (MDF-format) of all registered messages.



Press on the green start button to start the data acquisition. Select a filename and press Save. The filename will be appended by a date string. You'll see in the status area if the DAQ is running. The status bar shows the duration of the DAQ and the file size written so far. Press the stop button to

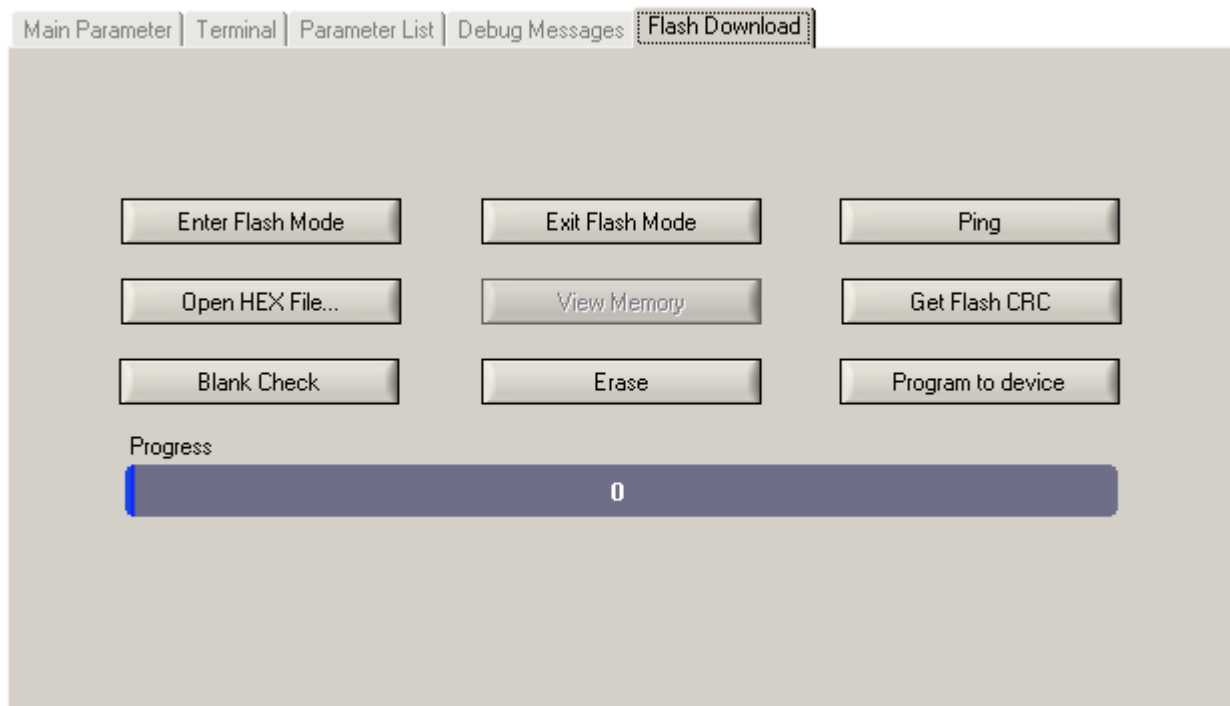
terminate the data logging. It is very important to stop the logging prior copying or opening the file. If you try to open a non-terminated file the data may be lost. MDF files can be opened with the CANgraph from Vector Informatik GmbH.



You can enable auto logging in the Setup dialog. This will automatically start DAQ whenever the PARAM tool is started. The logging will start a new file when the given Max DAQ file size is reached. A date string is added to the filename.

## 6.6 Flash Download (motor table for inverter)

This chapter is applicable for the BRUSA inverter (DMC). Other BRUSA devices will not use this flash download capability. The PARAM tool can program an external data flash on the BRUSA device. This is not the flash where the firmware is stored. Firmware updates are done with separate tools and are not subject of this document.



### 6.6.1 Steps to program a new motor table

Action: Make sure you are logged in as "monitor".  
Press "Enter Flash Mode"

Info: STAT: Command OK (0)  
HELLO: ECU-ID 0x401 - Master-ID 0x400

Action: You can check if you are in the flash mode by pressing on the Ping button.

Info: Respond to PING

Action: Press "Open HEX File..." and select the appropriate motor table. Motor tables are always named MotTable\_XXXX.mot. For example: MotTable\_HSM1-6.17.12-D01\_01-05-02\_110302.mot

Comment: This will take some time. The progress can be monitored in the progress bar. The mot file includes different lookup tables. Each table will print a short info line in the Info window. When the whole file is opened a CRC checksum is calculated and printed in the info window. The file is now in the intermediate buffer of the PARAM tool. Nothing was sent to the BRUSA device so far.

Info: Hex File info: <110302\_HSM\_control\_01\_FSB>  
Hex File info: <110302\_HSM\_control\_02\_GSB>  
Hex File info: <110302\_HSM\_control\_03\_LQ>  
Hex File info: <110302\_HSM\_control\_04\_LD>  
Hex File info: <110302\_HSM\_control\_05\_P\_mat>  
Hex File info: <110302\_HSM\_control\_06\_w\_mat>  
Hex File info: <110302\_HSM\_control\_07\_HSM\_TableInfo>  
Hex File info: <110302\_HSM\_control\_08\_Flash\_Header>  
Hex File info: <110302\_HSM\_control\_09\_ConfigAndParameter>  
Hex File info: <110302\_HSM\_control\_10\_TemperatureTable>  
File loaded successful! Caluculate CRC...  
-> CRC=0000

Action: Check if the flash is blank or not by pressing the “Blank Check” button.

Info: Device is not Blank @ 0x???????? → (has to be erased first)  
Device is Blank. → go on with program

Action: If the device is not blank press “Erase”

Comment: Watch the progress bar and wait for the info respond. The actual time needed by the device may vary – the progress bar shoes the time that is expected to wait for the respond.

Info: Erase Done.

Action: When ready press the “Program to device” button to do the actual programming of the flash device.

Info: Start programming...  
(look at the progress bar)  
File programmed successfully!

Action: To verify the programmed data press “Get Flash CRC”

Info: Get CRC from device...  
Got Flash CRC: 0000

Comment: The CRC has to be the same as it was calculated at the Open HEX File step.

Action: When finished press the “Exit Flash Mode” button.

Info: STAT: Command OK (0)  
HELLO: ECU-ID 0x401 - Master-ID 0x400

## 6.6.2 Identify motor table

To get the identification of the motor table got to the terminal tab and enter the command “Flash\_Ident”.

**Command:** Flash\_Ident  
**Answer:** Flash identification  
Motor Name: HSM1-6.17.12-GX-D03  
Datestring: 110526  
Version 01-05-01



## **6.7 Get diagnostic information for BRUSA**

If you have questions or problems regarding the BRUSA device it will for us always be helpful to know as much as possible on the situation. This chapter will show how you can get the information that is of interest for us.

### **6.7.1 Serial number**

When sending data to BRUSA support please add also the serial number of your device to the data sent. If other BRUSA devices are involved send also the type and serial of those components. For example the motor connected to a inverter.

### **6.7.2 CAN log**

If you are able to reproduce the situation of interest it is very helpful for us if you can provide us with CAN logging information that shows the situation. To do that you can use the logging feature of the PARAM tool or another CAN logger that can store \*.blf or \*.asc files.

### **6.7.3 Software Version and Event Log**

The BRUSA device can store an event log which has the capability to store the last 1024 events occurred in the device. Follow the following instructions to get detailed information on the actual firmware and the event log.

Action:      Make sure you are logged in as “monitor”.  
                 Press “Clear Info Window” to delete uninteresting information from the info window.  
                 Press “Get Ident” on the main parameter tab.

Comment:    Information about the firmware version is printed to the info window.

Action:      Press “Get Event Log”

Comment:    After some time a big list of events is printed to the info window.

Action:      Select File → Save Info text... on the menu bar.  
                 Chose a file and location where the info text should be stored.

Comment:    This will save a text file with the content of the info window.

Action:      Send this file together with the other information to BRUSA.

#### 6.7.4 Get all parameters

The PARAM tool has a feature to read all parameters that are defined by an XML file into an XML file. The following instruction will explain how to do that.

Action:      Make sure you are logged in as “monitor”.  
                 Select Options → Read all parameters... from the menu bar.

Comment:    A dialog appears to select the source XML file.

Action:      If you got a special XML file from BRUSA select that file as source file. If you don't have a special file select the XML that you use for the PARAM tool. You should be in the right directory for that file.

Comment:    A new dialog appears to ask you for the target XML file.

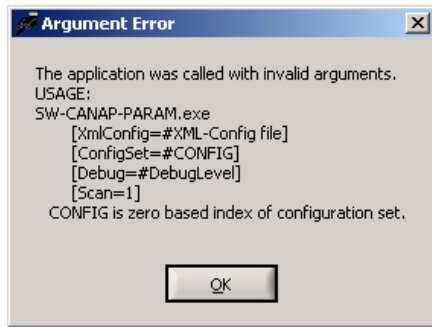
Action:      Chose an appropriate file name and location to store the parameter data. A date string will be automatically appended to the given file name.

Comment:    A CMD window opens and reads all parameters to store them in the target XML file. When ready send this file together with the other information to BRUSA.

## 7 Appendix

### 7.1 Command line options

The PARAM tool can be started with different command line options. They can be added after creating a short cut to the executable. If you start the PARAM with invalid options a dialog will show the valid options.



XmlConfig= (selects a specified XML file)

ConfigSet= (If different configurations are present this option select a specified configuration)

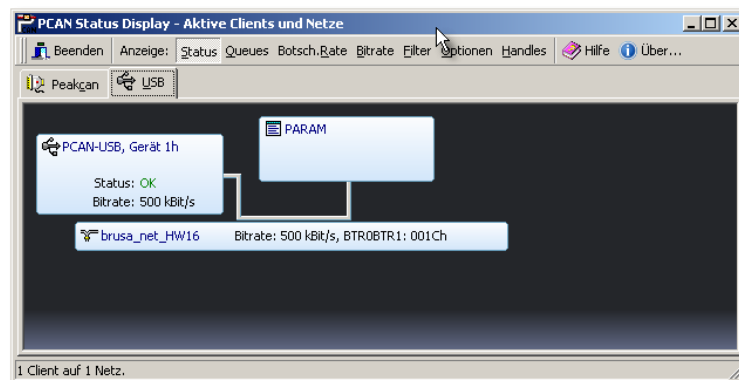
Debug= (not used by customer without advice from BRUSA)

Scan=1 (starts up in scan mode)

### 7.2 Using PCAN tools to check if the CAN is working

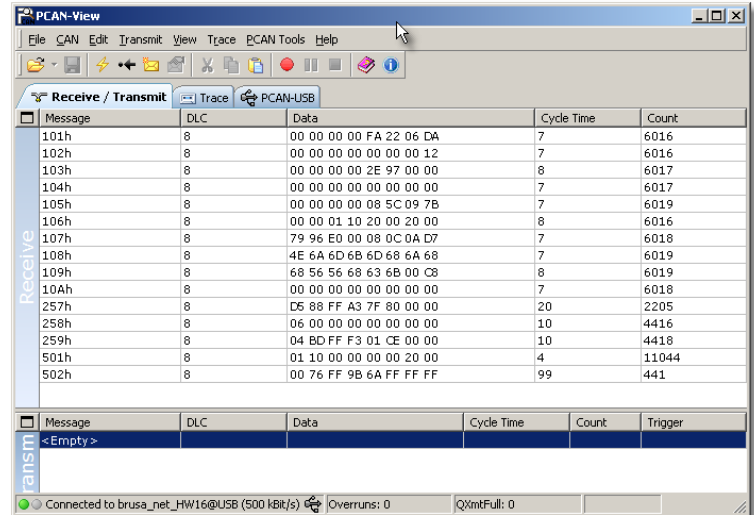
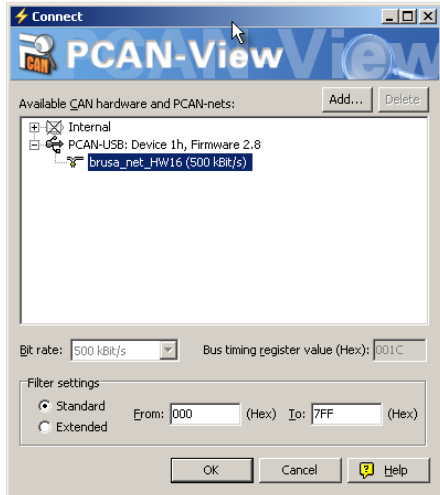
#### 7.2.1 PCAN Status Display

Open the PCAN Status Display tool to see if the CAN hardware was properly connected to the PARAM tool. If everything is as expected you can see the PARAM client, the PCAN-USB hardware and the net connected to each other.



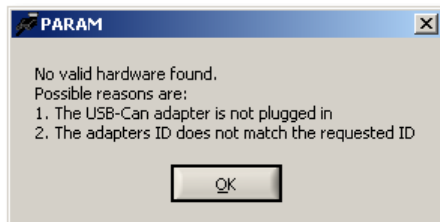
## 7.2.2 PCAN View

When you start the PCAN View tool you have to select the net that corresponds to the connected hardware. Now you can see all messages that are received by the PCAN-USB dongle. Looking at the ID's that are present might give some information if the BRUSA device is running and sending its own CAN messages.

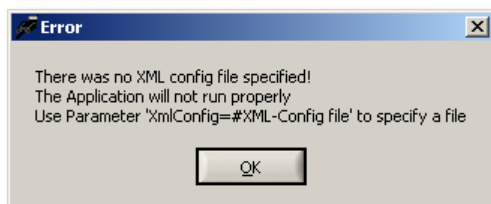


## 7.3 The PARAM tool does not start

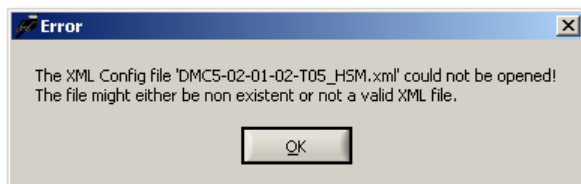
In the following section you will see some common error messages from the PARAM tool.



The PCAN-USB adapter could not be found. Check if the device is plugged into an USB port and if the driver was loaded successfully.



The PARAM tool was started without any specification of an XML file and there wasn't an XML file found in the current working directory. Read the section StartParam.com in the chapter installation.



The given XML file could not be opened. There are two possible reasons for this error.

1. The BrusaCviConfig.dtd file is missing. This file has to be present in the same directory as the XML file is.
2. The XML file was modified and has invalid or incomplete tags.

## 7.4 Find target with different PARM CAN ID's

The HELLO message can be used to detect a BRUSA device when the PARAM ID's are misconfigured or lost. To find such a device and configure it back to default values (or other wanted ID's) follow these instructions:

Action: Open the PARAM tool as usual.  
Select Options → Scan BRUSA Device... from the menu bar.

Comment: The PARAM tool opens the System Information window which prints the text Scan for BRUSA devices.

Action: Toggle KL15 of the appropriate BRUSA device. It is best if only one device is active at a time.

Comment: If a BRUSA device starts up it sends a HELLO message which is recognised in Scan mode.

Info: Got a HELLO: TxID=0x401 RxID=0x400

Action: Press OK if you see such a HELLO message.

Comment: The PARAM tool is now using these recognised ID's to communicate with the BRUSA device. You can now look at the configured ID's and enter the correct ones.

## 7.5 Change CAN ID or CAN baud rate in XML file

Usually it's not necessary to change anything within the XML file. But there might be still some reasons to look into the file – especially when you do some custom configurations such as changing CAN id's or baud rate. Do not change anything you don't know what it does.

The most important information is just located in the header of the XML file

```
<header>
  <device>Title TEXT</device>

  <version>XML-Version</version>

  <date>Date string</date>

  <componentlist>
    :
    :
  </componentlist>

  <canconfig>
    <rxid>0x401</rxid>

    <txid>0x400</txid>

    <baudrate baudrate="500KBit" />
  </canconfig>
</header>
```

rxid	RX CAN ID (received by the PARAM tool)
txid	TX CAN ID (transmitted by the PARAM tool)
baudrate	String for the desired baud rate. Valid options are: 125KBit 250KBit 500KBit 1MBit

Canmessages active for the logging ability are configured at the end of the XML file in the <canmsglist> section. You might also want to adapt changed CAN id's in that section.

```
<canmsglist>
  <canmsg>
    <name>xxx</name>
    <canid>0x123</canid>
    <dlc>x</dlc>
    :
  </canmsg>
```