

# **Configuring the DSACON32 using the built-in Shell Interface**

- Application Note -



## Contents

1	Introduction .....	3
2	Using HyperTerminal with configuration file.....	3
2.1	Using HyperTerminal with configuration file.....	3
2.2	Manual configuration of HyperTerminal .....	4
3	The Configuration Shell.....	5
4	Supported Commands.....	6
4.1	Example 1: Determine the Firmware Version .....	9
4.2	Example 2: Getting the current Communication Interface .....	9
4.3	Example 3: Setting the Interface to USB .....	9
4.4	Example 4: Reading the Sensor Configuration .....	10
4.5	Example 5: Reading the Configuration of Matrix 0.....	10
5	Appendix .....	11



## 1 Introduction

All sensor controllers of the DSACON32 family have a built-in configuration shell to allow a simple set up of the controller. This text-based interface can be accessed using any VT-100 compatible terminal program. On every start up, the DSACON32 automatically tests for a connected terminal at its RS232 interface. If present, it starts up in configuration mode providing a command prompt that can be used to get detailed information about the controller and the connected tactile transducer as well as to change various system parameters. The following settings of your terminal program are required for a proper communication:

- 115200 baud
- 8 data bits, 1 stop bit
- No parity
- No handshaking
- Emulation: VT-100 compatible

## 2 Using HyperTerminal with configuration file

If you are using Microsoft Windows, you can use the terminal emulator "HyperTerminal", that is included in every Windows XP and 2000 installation. The following two sections describe how to setup HyperTerminal to communicate with the DSACON32 family of sensor controllers.

### 2.1 Using HyperTerminal with configuration file

The Product CD contains the file "dsacon32.ht" that can be used in conjunction with HyperTerminal. It contains all necessary configurations for HyperTerminal to communicate with the DSACON32. Simply copy it to your desktop and open it by double-clicking. HyperTerminal opens up and tries to connect to the sensor controller connected to COM1 of your computer. If you are using a different port, please close the connection and change it using File|Settings...

Open the port by clicking "Call". Connect the DSACON32 to your PC (if not already done) by using the serial port you did set up with the steps before. Cycle the power supply of the sensor controller to force a reboot. The sensor controller will now recognize the connected terminal program and start in configuration mode.

## 2.2 Manual configuration of HyperTerminal

If you prefer to configure HyperTerminal by yourself, please follow the steps below:

1. Open HyperTerminal

Click Start | Programs | Accessories | Communications | HyperTerminal.

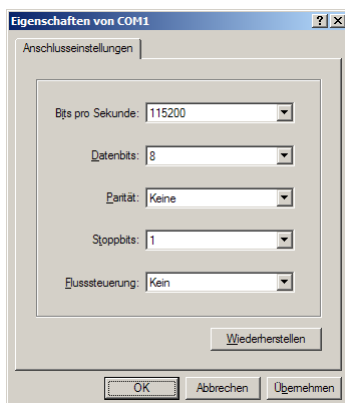
2. Create new connection.

Once HyperTerminal opens, it will automatically prompt you to create a new connection if none exist. If no connection(s) exists, you can click File | New Connection to create a new one. Specify a name for the connection, e.g. "DSACON32 at COM1", choose an icon, and click OK.



3. Select interface.

In the "Connect To" dialog box, choose the COM port where the DSACON32 is connected to from the Connect Using drop-down list and click OK. Into the port property sheet that appears, enter the configuration parameters from above:



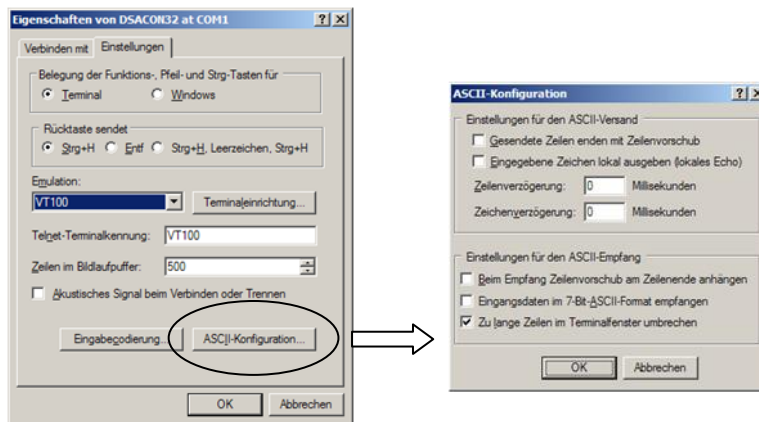
4. Set terminal configuration.

If the connection was opened automatically, close it by clicking "hang up".

Click Files | Properties to access additional connection settings. Use the tab "Settings" ( to set the emulation type to "VT100". Press the button ASCII configuration to enter the settings



as shown in the pictures below:

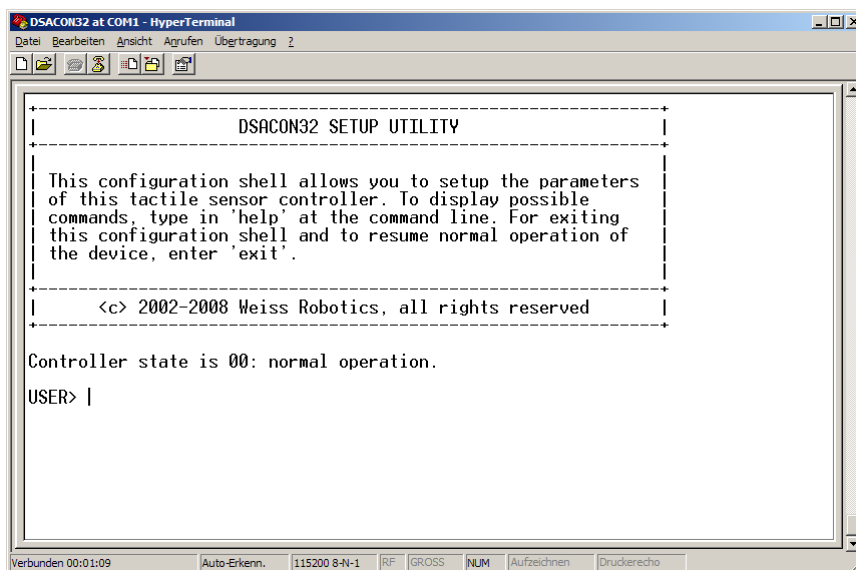


5. Enable the connection.  
Click "call" to open the connection.

Connect the DSACON32 to your PC (if not already done) by using the serial port you did set up with the steps before. Cycle the power supply of the sensor controller to force a reboot. The sensor controller will now recognize the connected terminal program and start in configuration mode.

### 3 The Configuration Shell

After the DSACON32 did boot up and detected the terminal connection successfully, you will see the welcome message inside the terminal window on success:





The DSACON32 is now waiting for commands. To get an overview over the supported commands, enter “help” at the configuration shell’s command prompt. To exit the command shell, enter “exit”. It depends on your changes, if the DSACON32 needs to restart to apply them.

## 4 Supported Commands

The following commands are currently supported by the DSACON32 configuration shell:

Command	Description
Acquire	<p>Acquires and displays the pressure profile from the connected tactile transducer.</p> <p><u>Syntax</u> acquire i= &lt;number&gt;</p> <p><u>Parameters</u> i:           Index of the matrix to be sampled</p>
Adjust	<p>Adjusts the pressure sensitivity of a matrix. Note, that this is not supported on all tactile transducer types.</p> <p><u>Syntax</u> adjust {i= &lt;number&gt;, all} {s= &lt;float&gt;, [factory]}</p> <p><u>Parameters</u> i:           Matrix index all:         Adjust all matrices to the same value s:           Sensitivity to be set. Use any value between 0% for the most insensitive and 100% for the most sensitive setting. factory:    Restore factory defaults</p>
Cls	<p>Clears the configuration shell screen.</p> <p><u>Syntax</u> cls</p>
Controller	<p>Displays the sensor controller’s serial number, revision, supported interfaces and some other basic configuration.</p> <p><u>Syntax</u> controller</p>
Descriptor	<p>Changes the descriptor string(s) of the sensor and its matrices. Each tactile transducer has a sensor descriptor describing the transducer itself and for every contained matrix an additional matrix descriptor.</p>



	<p><u>Syntax</u> descriptor {sensor, i=&lt;number&gt;} [set "&lt;string&gt;"]</p> <p><u>Parameters</u> sensor: If used, the descriptor of the sensor module is read/written i: Matrix index, if the descriptor string of a matrix should be read/written set: Set the given descriptor string for the sensor resp. the selected matrix. Passing an empty string will delete the descriptor. To set a descriptor string containing whitespaces, use quotation marks ("sample string") around the string.</p>
Exit	<p>Exits the setup menu. The controller may restart automatically, if it is necessary for the changes you made to take effect.</p> <p><u>Syntax</u> exit</p>
Grasping	<p>Setting up the Reactive Grasping Module. This command is only available on controllers with a Reactive Grasping Module installed (e.g. the DSACON32-H).</p> <p><u>Syntax</u> grasping {config, home, state, enable, disable}</p> <p><u>Parameters</u> config: Let you enter the configuration of the grasping module interactively home: Executes a reference drive for the gripper. Note: The gripper will move if you use this command. state: Returns the gripper's status word. enable: Enables the Reactive Grasping Module. disable: Disables the Reactive Grasping Module</p>
Help	<p>Displays a help screen</p> <p><u>Syntax</u> help</p>
Interface	<p>Sets up the communication interface. Depending on the platform, the user can select between several different interfaces.</p> <p><u>Syntax</u> interface [{serial, can, usb, auto}]</p> <p><u>Parameters</u> serial: Change to serial RS232 interface can: Change to CAN-Bus usb: Change to USB auto: Automatic detection of the communication interface. Note, that this feature is not available on all platforms.</p>



Mask	<p>Controls the mask of a sensor matrix. This speeds up data acquisition, if you only intend to sample a few points and not the whole matrix. Therefore, each matrix contains two matrices: one static and one dynamic. While the factory-set static mask is used to define the shape of non-rectangular matrices and is read-only, the dynamic mask can be read and written by the user with the built-in mask editor. A value of "1" indicates that the corresponding sensor cell is enabled. Static and dynamic masks are logically and-ed to determine, if a cell is sampled or not. The dynamic mask is lost on a power cycle.</p> <p><u>Syntax</u> mask i= &lt;number&gt; {static, dynamic}</p> <p><u>Parameters</u> i: Matrix index dynamic: Read/Write the dynamic mask using the built-in mask editor. static: Read the static mask.</p>
Matrix	<p>Display the sensor matrix configuration and changes the threshold of the matrix. Sensor values beneath the matrix threshold are treated as 0.</p> <p><u>Syntax</u> matrix i= &lt;number&gt; {config, threshold [{set, reset}]}</p> <p><u>Parameters</u> i: Matrix index config: Display the matrix configuration. threshold: Read the threshold for the matrix. threshold set: Sets the threshold for the matrix. threshold reset: Resets the threshold for the matrix to the factory setting.</p>
Measure	<p>Performs some measurements on the sensor system.</p> <p><u>Syntax</u> {speed, temp}</p> <p><u>Parameters</u> speed: Measures the maximum sampling speed of the current sensor configuration (i.e. using the currently set masks) without transferring frame data to a client temp: Displays the controller's board temperature. Please note, that not all controllers include an onboard temperature sensor.</p>
Restart	<p>Saves changes and restarts the controller.</p> <p><u>Syntax</u> restart</p>
Sensor	<p>Displays the configuration of the connected tactile transducer.</p> <p><u>Syntax</u> sensor</p>





Version	<p>Displays the firmware version and built date.</p> <p><u>Syntax</u> version</p>
---------	---------------------------------------------------------------------------------------

Some commands may have additional options with restricted access that are intended for factory use only. To get help on a specific command, please type in “[command name] help”. The usage of the configuration shell interface should be clarified by the following examples.

### 4.1 Example 1: Determine the Firmware Version

The user enters:

```
version
```

The sensor controller answers:

```
Firmware revision: 220, build on 2008/12/12 16:47:29
Running on RTOS kernel V5.0.0
DAQ ASIC revision: 0003
```

### 4.2 Example 2: Getting the current Communication Interface

The user enters:

```
interface
```

The sensor controller answers:

```
Current interface is: serial
```

### 4.3 Example 3: Setting the Interface to USB

The user enters:

```
interface USB
```

The sensor controller answers:

```
Changing interface to USB.
```



## 4.4 Example 4: Reading the Sensor Configuration

The user enters:

```
sensor
```

The sensor controller answers:

```
Sensor configuration:
```

```
Number of sensor matrices.....: 1
Sensor serial number.....: 12345678
Hardware revision.....: 0
Cfg. set up using firmware revision.: 227
Input signal inversion.....: no
Size of configuration memory.....: 32768 bytes
Descriptor string.....: "DSAMOD-5i"
```

## 4.5 Example 5: Reading the Configuration of Matrix 0

The user enters:

```
matrix config i=0
```

The sensor controller answers:

```
One wire unique identifier.....: 000000000000h
Matrix sensor type.....: 'DSAMOD-5i'
```

```
Matrix type specific:
```

```
Hardware revision.....: 0
Number of texels horizontal.....: 24
Number of texels vertical.....: 32
Texel width.....: 5.00 mm
Texel height.....: 5.00 mm
Sampling delay.....: 20 us
```

```
General parameters:
```

```
Sensitivity adjustment:
```

```
Sensitivity adjustment by user...: yes
User set sensitivity.....: 50.0%
Factory set sensitivity.....: 50.0%
X-coordinate of the matrix center.: 0.0 mm
Y-coordinate of the matrix center.: 0.0 mm
Z-coordinate of the matrix center.: 0.0 mm
Matrix rotation about x-axis.....: 0.0 deg
Matrix rotation about y-axis.....: 0.0 deg
Matrix rotation about z-axis.....: 0.0 deg
```



```
Analog input channel.....: 0  
Descriptor string.....: ""
```

## 5 Appendix

The following command syntax notation is used for this Application Note:

### Parameters

- |             |                                                                               |
|-------------|-------------------------------------------------------------------------------|
| a           | Denotes a mandatory parameter                                                 |
| [a]         | Denotes an optional parameter                                                 |
| {a, b, c}   | Denotes a selection of mandatory parameters (exactly one must be present)     |
| [{a, b, c}] | Selection of optional parameters (either exactly one or none must be present) |

### Values

- |          |                                                                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <string> | Has to be replaced by a string. If the string contains whitespaces, it has to be surrounded by quotation marks (e.g. "String with whitespaces") |
| <number> | Has to be replaced by an integer number (0..n)                                                                                                  |
| <float>  | Has to be replaced by a real number (e.g. 4.7 or 65.3)                                                                                          |



## Weiss Robotics

In der Gerste 2  
D-71636 Ludwigsburg, Germany  
e-mail: [info@weiss-robotics.com](mailto:info@weiss-robotics.com)

For further information and other products from Weiss Robotics, please visit our homepage at <http://www.weiss-robotics.de>.

---

© 2009 Weiss Robotics, all rights reserved.

All technical data mentioned in this data sheet can be changed to improve our products without prior notice. Used trademarks are the property of their respective trademark owners. Our products are not intended for use in life support systems or systems whose failure can lead to personal injury.