

Console example – list of available commands

All references in this document refer to document *DPA Framework Technical Guide v.3.00*

List of commands:

- `rst` - print header of Console command processor
- `ls` - list of files in root directory of SD card
- `bond` - see chapter 3.2.6 *Bond node*
- `bond „RegAddr“ „BondingMask“` - see chapter 3.2.6 *Bond node* (parameters „RegAddr“ and „BondingMask“ are not obligatory)
- `bond get` - see chapter 3.2.4 *Get bonded nodes* (result of this command is map of bonded nodes. You can see it on the next picture)

```
cmd> bond get

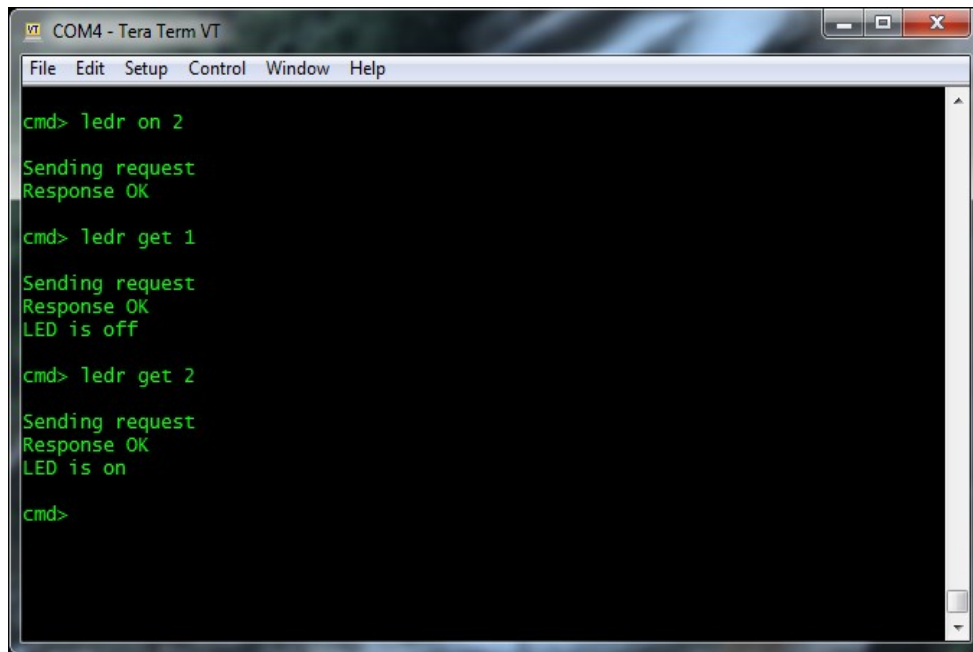
Sending request
Response OK

  0 1 2 3 4 5 6 7 8 9 A B C D E F
0 - . x x x x . . . . . . . .
1 - . . . . . . . . . . . . . .
2 - . . . . . . . . . . . . . .
3 - . . . . . . . . . . . . . .
4 - . . . . . . . . . . . . . .
5 - . . . . . . . . . . . . . .
6 - . . . . . . . . . . . . . .
7 - . . . . . . . . . . . . . .
8 - . . . . . . . . . . . . . .
9 - . . . . . . . . . . . . . .
A - . . . . . . . . . . . . . .
B - . . . . . . . . . . . . . .
C - . . . . . . . . . . . . . .
D - . . . . . . . . . . . . . .
E - . . . . . . . . . . . . . .
F - . . . . . . . . . . . . . .

cmd>
```

- `bond clear` - see chapter 3.2.5 *Clear all bonds*
- `unbondc „BondAddr“` - see chapter 3.2.7 *Remove bonded node*
- `rebondc „BondAddr“` - see chapter 3.2.8 *Re-bond node*
- `discovery` - see chapter 3.2.9 *Discovery*
- `discovery „TxPower“ „MaxAddr“` - see chapter 3.2.9 *Discovery* (parameters „TxPower“ and „MaxAddr“ are not obligatory)
- `discovery get` - see chapter 3.2.3 *Get discovered nodes*
- `ledr on „NADR“` - see chapter 3.9.2 *Set*
- `ledr off „NADR“` - see chapter 3.9.2 *Set*
- `ledg on „NADR“` - see chapter 3.9.2 *Set*
- `ledg off „NADR“` - see chapter 3.9.2 *Set*
- `ledr get „NADR“` - see chapter 3.9.3 *Get.*
- `ledg get „NADR“` - see chapter 3.9.3 *Get*
- `ledr pulse „NADR“` - see chapter 3.9.4 *Pulse*
- `ledg pulse „NADR“` - see chapter 3.9.4 *Pulse*

Example how to use some commands of peripherals LEDs, you can see on next picture.



A screenshot of a Tera Term VT window titled 'COM4 - Tera Term VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The terminal area shows the following commands and responses:

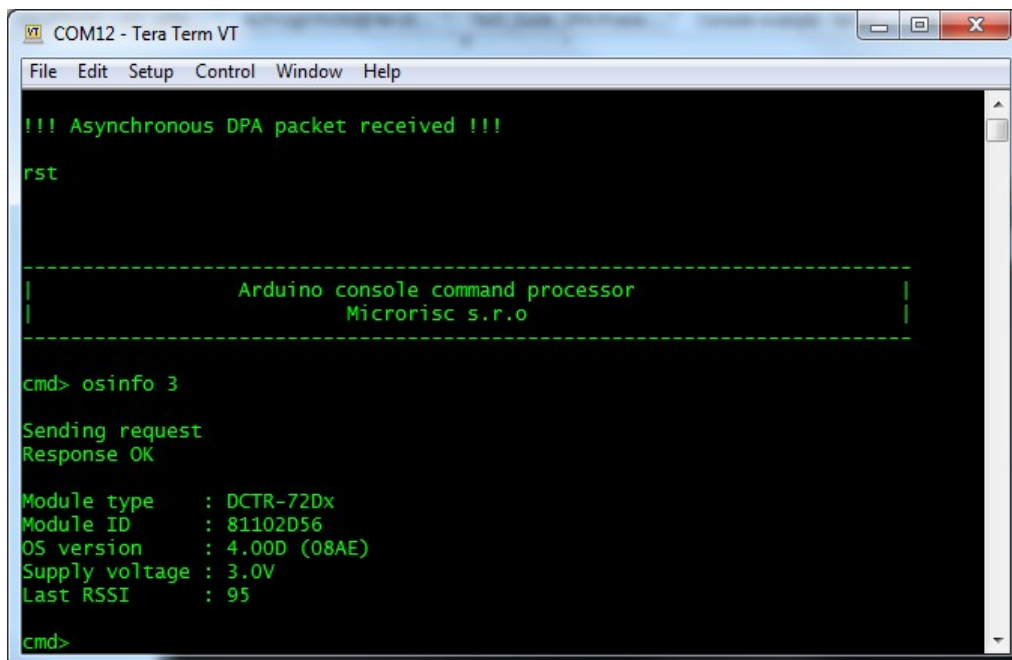
```
cmd> ledr on 2
Sending request
Response OK

cmd> ledr get 1
Sending request
Response OK
LED is off

cmd> ledr get 2
Sending request
Response OK
LED is on

cmd>
```

- unbondn „NADR” - see chapter 3.3.3 *Remove bond*
- osreset „NADR” - see chapter 3.4.3 *Reset*
- osrestart „NADR” - see chapter 3.4.4 *Restart*
- osinfo „NADR” - see chapter 3.4.2 *Read*. (example to this command, you can see on next picture)



A screenshot of a Tera Term VT window titled 'COM12 - Tera Term VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The terminal area shows the following commands and responses:

```
!!! Asynchronous DPA packet received !!!

rst

-----
|                      Arduino console command processor                      |
|                      Microrisc s.r.o                                         |
-----

cmd> osinfo 3

Sending request
Response OK

Module type   : DCTR-72Dx
Module ID    : 81102D56
OS version   : 4.00D (08AE)
Supply voltage : 3.0V
Last RSSI    : 95

cmd>
```

- loadcfg „filename“ „NADR“ - see chapter 3.4.6 *Write HWP configuration*

- Command is used for uploading new configuration to selected TR module. Configuration files must be prepared in root directory of SD card, which is connected to an Arduino board. To create a new configuration file, we use the IQRF IDE. The configuration file will have an extension *.trcnfg. As the library to work with SD card does not support long file names, we should rename the configuration file to match the format 8.3. For example *config1.cfg*, *conf.trc*

- Command parameters have the following meanings:

-filename - name of configuration file. The name should be in 8.3 format.

-NADR - address of destination TR module

The next set of commands is possible to use only, if the #define `__STORE_CODE_SUPPORT__` is enabled in *dpa_library.h* file.

- storecode „filename“ „eeeadr“ „NADR“

- Command is used for uploading the code image of *HEX* or *IQRF* file. Code image is uploaded to selected address in eeprom memory of TR module. *HEX* or *IQRF* files must be prepared in root directory of SD card. As the library to work with SD card does not support long file names, we should rename the *HEX* or *IQRF* files to match the format 8.3.

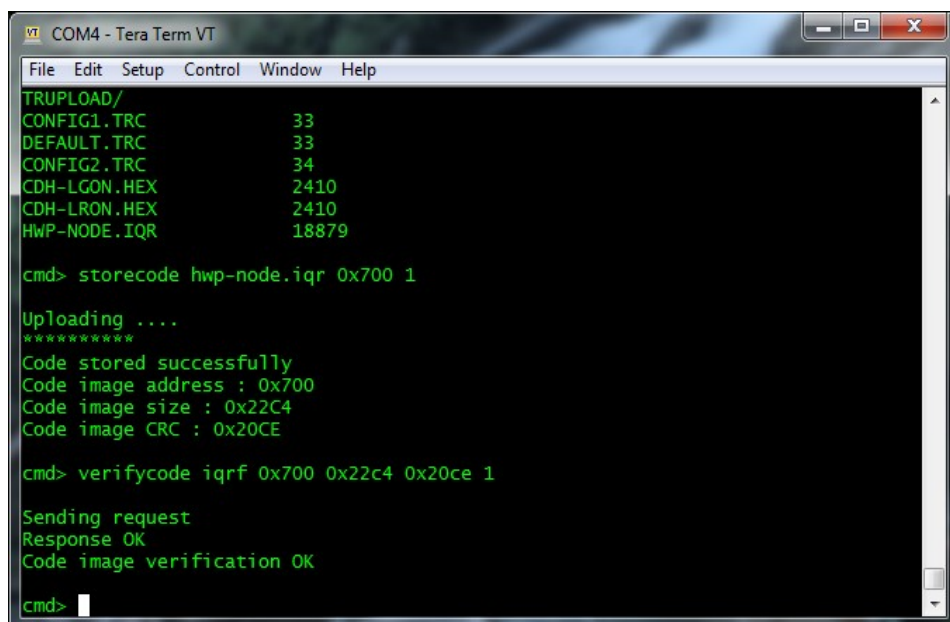
- Command parameters have the following meanings:

-filename - name of *HEX* or *IQRF* file. The name should be in 8.3 format.

-eeeadr - absolute address in eeprom memory of selected TR module. Address is entered in HEX format and must be a multiple of 64.

-NADR - address of destination TR module

Example how to use this command, you can see on next picture.



```
COM4 - Tera Term VT
File Edit Setup Control Window Help
TRUPLoad/
CONFIG1.TRC          33
DEFAULT.TRC          33
CONFIG2.TRC          34
CDH-LGON.HEX         2410
CDH-LRON.HEX         2410
HWP-NODE.IQR         18879

cmd> storecode hwp-node.iqr 0x700 1

Uploading ....
*****
Code stored successfully
Code image address : 0x700
Code image size : 0x22C4
Code image CRC : 0x20CE

cmd> verifycode iqr 0x700 0x22c4 0x20ce 1

Sending request
Response OK
Code image verification OK

cmd> 
```

- `verifycode „imgtype“ „eeeadr“ „imgsize“ „imgCRC“ „NADR“`

- Command is used for verifying the code image of *HEX* or *IQRf* file, in previous step uploaded to eeprom memory of TR module. See chapter 3.4.12 *LoadCode*

- Command parameters have the following meanings:

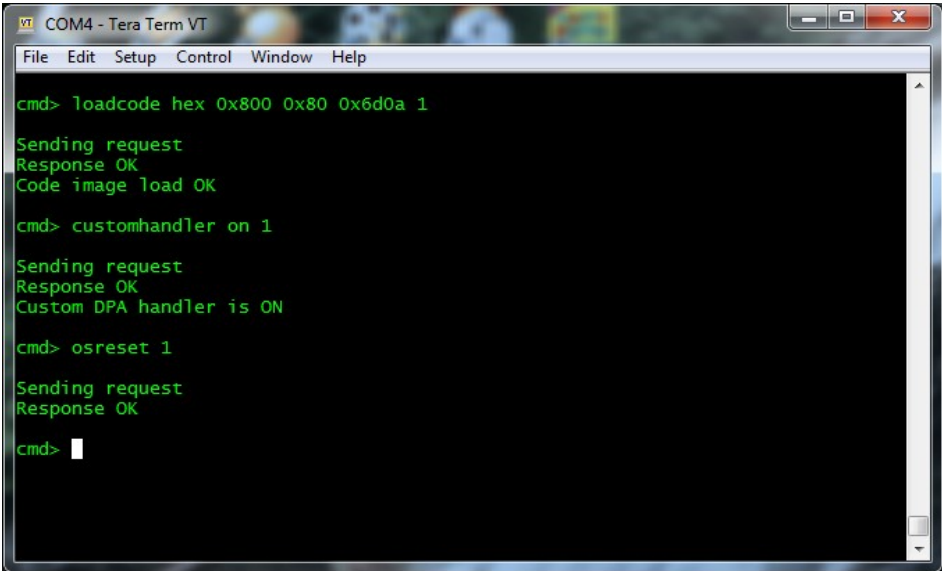
- `imgtype` - type of uploaded code image. Enter `hex` in case of *HEX* file, or `iqr` in case of *IQRf* file
- `eeeadr` - absolute address in eeprom memory of selected TR module. Address is entered in *HEX* format (e.g. `0x700`) and it is a result of `storecode` operation
- `imgsize` - size of code image stored in eeprom memory of TR module. Size is entered in *HEX* format (e.g. `0x22c4`) and it is a result of `storecode` operation
- `imgCRC` - checksum of code image stored in eeprom memory of TR module. Checksum is entered in *HEX* format (e.g. `0x20ce`) and it is a result of `storecode` operation
- `NADR` - address of destination TR module

- `loadcode „imgtype“ „eeeadr“ „imgsize“ „imgCRC“ „NADR“`

- Command is used to burning the code image of *HEX* or *IQRf* file, to *FLASH* memory of microcontroller in TR module. See chapter 3.4.12 *LoadCode*

- Command parameters have the same meaning as in case of `verifycode` command.

Example how to use this command, you can see on next picture.



```
COM4 - Tera Term VT
File Edit Setup Control Window Help

cmd> loadcode hex 0x800 0x80 0x6d0a 1
Sending request
Response OK
Code image load OK

cmd> customhandler on 1
Sending request
Response OK
Custom DPA handler is ON

cmd> osreset 1
Sending request
Response OK

cmd> 
```

- `customhandler on „NADR“`

- Command is used to enable the custom DPA handler in configuration of TR-7xD module. Configuration change is applied only if the module is reset by sending of `osreset` command.

- `customhandler off „NADR“`

- Command is used to disable the custom DPA handler in configuration of TR-7xD module. Configuration change is applied only if the module is reset by sending of `osreset` command.

Example of good practice to update the custom DPA handler

- Use the `storecode` command to upload the code image of new custom DPA handler to eeprom memory of TR-7xD module.

- Use the `verifycode` command to verify the integrity of code image in eeprom memory of TR-7xD module.

- Use the `customhandler off` and `osreset` commands to disable actual custom DPA handler in TR-7xD module.

- Use the `loadcode` command to burn the new custom DPA handler to FLASH memory of microcontroller in TR module.

- Use the `customhandler on` and `osreset` commands to enable new custom DPA handler in TR-7xD module.