AxiCat Command Line Tool

v1.3.1

User Manual

September 2016

Table of Contents

Revision History

Date	Authors	Description
2014-09-05	Peter S'heeren	Initial release.
2014-09-17	Peter S'heeren	New software version.
2014-09-28	Peter S'heeren	New software version.
2014-12-12	Peter S'heeren	New software version.
2016-09-13	Peter S'heeren	New software version.

1 Program

Overview

The program talks directly with the AxiCat at the level of the serial interface. This is the lowest level of communication possible and is defined as the AxiCat serial protocol.

The program applies the serial protocol in a very simple and minimalistic way. It takes user commands and translates them into command packets that are sent to the AxiCat. A number of user commands read back data bytes from the AxiCat.

Other user commands provide features such as reading data bytes from the AxiCat and waiting for a certain amount of milliseconds.

The program prints transmitted and received data bytes in hexadecimal format to standard output. This feature allows you to monitor the data bytes transferred over the FT245 serial interface. This is very useful for learning the serial protocol or to use the program as a reference when you're creating your proper implementation of the serial protocol.

The program offers two main operation modes: interactive and batch processing.

Interactive Mode

In interactive mode, the program displays a prompt and the user enters a command to be executed. The program only quits when the user says so with the 'q' command.

```
$ ./axicatcl -axicat /dev/ttyUSB0

Enter command 'h' for help.

> io 10 in
Tx: 04h 0Ah
> io 10 1
Tx: 05h 8Ah
> io 10 rd
Tx: 06h 0Ah
Rx: 06h CAh
> q
$
```

In the above example, the program is run on a Linux system. GPIO10 is set as input with pull-activated. Next the state of GPIO10 is read. Finally, the 'q' command quits the program.

Batch Processing Mode

In batch processing mode, the program is invoked with an input file that contains user commands to be executed. After the input file has been processed, the program quits. This mode is useful for running your own scripts. The program allows comments and empty lines to be included in the input files so you can document your scripts.

The next example shows how you can initialize UART0 using a script file. The program is run on a Linux system.

The script file contains the commands to initialize UART0:

```
# Initialize UART0
ser 0 br 9600        # Baudrate
ser 0 db 8           # Data bits
ser 0 sb 1           # Stop bits

# Enable UART 0
ser 0 ena
```

The script file is passed as input file to the program:

```
$ ./axicatcl -axicat /dev/ttyUSB0 -f uart0_init.txt
```

The program runs non-interactively and quits as soon as it has processed all lines of the input file. It prints out the lines from the input file and transferred data bytes. If you like you can redirect the output to a file.

Command Line

Parameter	Description
-h	Display help and exit.
-axicat PATH	Specify the serial path of the AxiCat. Example PATH in Linux: /dev/ttyUSB0 Example PATH in Windows: \\.\COM4
-f FILE	Specify a file to read commands from. This parameter enables batch processing mode.

If no parameters are provided, the program displays help and exits.

Commands

Enter 'h' at the prompt to display all available commands.

Many commands require parameters. If you omit a parameter, the program explains the parameter it's expecting.

Some commands expect one or more data words for composing a data payload. You can enter a combination of values and strings to build up the payload. Values can be specified in binary, decimal and hexadecimal format, though they must always start with a decimal digit (0..9).

Command	Word Size	Example
`tx`	8	`tx 5 1101b 0F8h 48d`
`ser 0 tx` `ser 1 tx`	8	`ser 0 tx "A line of text" 13 10`
`ser 0 txw` `ser 1 txw`	16	`ser 1 txw 20 511 "A" 280 "B"`
`twi mtx`	8	`twi mtx 05h 20h`
`twi stx`	8	`twi stx 34 "AxiCat" 34 last`
`spi xfr`	8	`spi xfr 2 03h 20h 0 0 0 0 0 0 0 0 last`
`ow bytes`	8	`ow bytes 0CCh 44h spu`

Some commands can be concluded with an optional specifier:

Command	Specifier	Example
`twi mrx`	`last`	`twi mrx 10 last`
`twi sena`	`gca`	`twi sena 68h gca`
`twi stx`	`last`	`twi stx 1 2 3 4 last`
`twi srx`	`last`	`twi srx 5 last`
`spi xfr`	`last`	`spi xfr 0 "I/O Card" last`
`ow bytes`	`spu`	`ow bytes 0CCh 44h spu`
`ow bits`	`spu`	`ow bits 1 spu`

1-Wire Enumeration

The first parameter of command **ow enum** must be **first** or **next**. Parameter **first** starts a new enumeration procedure and allows further parameters to follow. Parameter **next** searches for the next 1-Wire slave in the current enumeration procedure.

Command **ow enum first** accepts a combination of search criteria in this order:

- Alarm search: Specify **alarm** to enable alarm search.
- Family search: Specify a family code (**0..255**) to enable searching 1-Wire slaves of the specified family.
- DS2409 smart ON: Enumerate 1-Wire slaves behind the main or auxiliary port of a DS2409. Specify **main** or **aux** followed by a string containing the ROM code of the target DS2409. The ROM code can be specified with or without a CRC value.

Examples: **"1F-56BA7"**, **"1F-56BA7-80"**.

Here are some example commands:

Command	Description
`> ow enum first`	Search of all 1-Wire slaves.
`> ow enum first alarm`	Search slaves with alarm condition.
`> ow enum first 28h`	Search DS18B20 slaves.
`> ow enum first main "1F-56AB7"`	Search behind the main port.
`> ow enum first aux "1F-56AB7-80"`	ROM code also specified.
`> ow enum first alarm 28h main "1F-56AB7"`	All search criteria specified.
`> ow enum next`	Search the next 1-Wire slave.

2 Software Revision History

Version	Description
1.0.0	▪ Initial release of AxiCat Command Line Tool.
1.1.0	▪ Added 1-Wire commands. ▪ Replaced GEN BUF INFO command with GEN INFO command. ▪ Based on AxiCat serial protocol v1.1.0.
1.2.0	▪ Added 1-Wire enumeration command. ▪ Based on AxiCat serial protocol v1.2.0.
1.3.0	▪ Added 1-Wire probing command. ▪ Extended output of INFO command. ▪ Based on AxiCat serial protocol v1.3.0.
1.3.1	▪ Added version information to Windows executable. ▪ Renamed command line argument -s to -axicat. ▪ Released as open source.

3 Software License

The license is stated in the source code.

4 Legal Information

Disclaimer

Axiris products are not designed, authorized or warranted to be suitable for use in space, nautical, space, military, medical, life-critical or safety-critical devices or equipment.

Axiris products are not designed, authorized or warranted to be suitable for use in applications where failure or malfunction of an Axiris product can result in personal injury, death, property damage or environmental damage.

Axiris accepts no liability for inclusion or use of Axiris products in such applications and such inclusion or use is at the customer's own risk. Should the customer use Axiris products for such application, the customer shall indemnify and hold Axiris harmless against all claims and damages.

Trademarks

All product names, brands, and trademarks mentioned in this document are the property of their respective owners.

5 Contact Information

Official website: http://www.axiris.eu/