Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

JTAG-Booster for

AMD Alchemy Solutions Processors

Forth-Systeme GmbH

P.O: Box 1103	●	D-79200 Breisach, Germany
Kueferstrasse 8	●	D-79206 Breisach, Germany
Tel. +49 (7667) 908-0	●	Fax +49 (7667) 908-200
sales@fsforth.de	●	http://www.fsforth.de

FS FORTH-SYSTEME GmbH
Postfach 1103, D-79200 Breisach, Germany

Release of Document:	November 22, 2004
Author:	Dieter Fögele
Filename:	JTAG_alchemyb.doc
Program Version:	4.xx

Table of Contents

1. General

The programs JTAu1000.EXE, JTAu1100.EXE, JTAu1500.EXE and JTAu1550.EXE use the IEEE 1149.1 JTAG port of the AMD Alchemy Solutions Processors in conjunction with the small JTAG-Booster:

- to program data into flash memory
- to verify and read the contents of a flash memory
- to make a memory dump
- to access a serial device (I²C/SPI/MicroWire)
- to test CPU signals

All functions are done without any piece of software running in the target. No firmware or BIOS must be written. Bootstrap software may be downloaded to initially unprogrammed memories.

As this tool uses boundary scan, it is extremely simple and very powerful. It assists you in bringing-up new hardware. Even if there are essential bugs in the hardware (i.e. RAM not reliable working, soldering problems with the BGA package), in many cases you are able to load small test programs into flash, which helps you to analyze hardware problems. Or if you have a flash memory, which is not connected correctly to the CPU (i.e. CPU's A0 is connected to a 16 bit AMD flash), we can support you with a special adapted version of the JTAG-Booster.

The JTAG-BOOSTER' s software is highly optimized to the JTAG chain of a specific target CPU. To give support for all processors of the AMD Alchemy Solutions Processors family, there are three different programs on the distribution disk:

- JTAu1000.EXE : Tool for AMD Alchemy Solutions Au1000 Processor
- JTAu1100.EXE : Tool for AMD Alchemy Solutions Au1100 Processor
- JTAu1500.EXE : Tool for AMD Alchemy Solutions Au1500 Processor
- JTAu1550.EXE : Tool for AMD Alchemy Solutions Au1550 Processor

Please contact us, if you need support for other members of the AMD Alchemy Solutions Processors family.

For latest documentation please refer to the file README.TXT on the distribution disk.

1.1. Ordering Information

The following related products are available

- 9019 JTAG-Booster AMD Alchemy Solutions Processors, 3.3V,
  AMD Au1000, Au1100, Au1500, Au1550
  DOS/Win9x/WinNT/Win2000/WinXP
  delivered with adapter type 285
  and additional cable with single strands TK02206

1.2. System Requirements

To successfully run this tool the following requirements must be met:

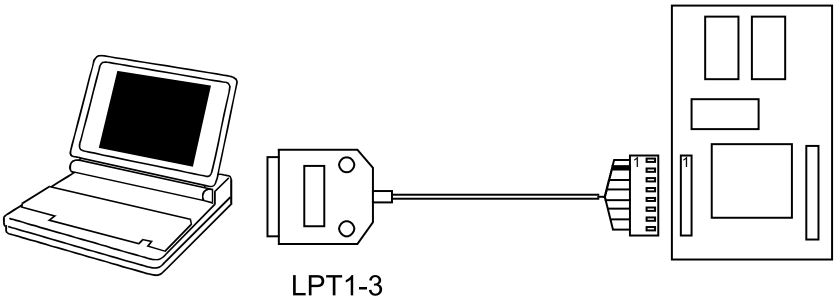
- MSDOS, WIN3.x, WIN9x, WinNT, Win2000 or WindowsXP (WinNT/Win2000/WindowsXP is supported with an additional tool, see chapter 5 “Support for Windows NT, Windows 2000 and Windows XP”)
- Intel 80386 or higher
- 205 kByte of free DOS memory
- Parallel Port

1.3. Contents of Distribution Disk

- JTAu1000.EXE JTAu1000.OVL Tool for AMD Alchemy Solutions Au1000 Processor
- JTAu1000.INI Template configuration file for AMD Alchemy Solutions Au1000 Processor. See chapter 1.11 "Initialization file JTAuxxxx.INI"
- JTAu1100.EXE JTAu1100.OVL Tool for AMD Alchemy Solutions Au1100 Processor
- JTAu1100.INI Template configuration file for AMD Alchemy Solutions Au1100 Processor. See chapter 1.11 "Initialization file JTAuxxxx.INI"
- JTAu1500.EXE JTAu1500.OVL Tool for AMD Alchemy Solutions Au1500 Processor
- JTAu1500.INI Template configuration file for AMD Alchemy Solutions Au1500 Processor. See chapter 1.11 "Initialization file JTAuxxxx.INI"
- JTAu1550.EXE JTAu1550.OVL Tool for AMD Alchemy Solutions Au1550 Processor
- JTAu1550.INI Template configuration file for AMD Alchemy Solutions Au1550 Processor. See chapter 1.11 "Initialization file JTAuxxxx.INI"
- WinNT This subdirectory contains the support for Windows NT, Windows 2000 and Windows XP. See chapter 5 "Support for Windows NT, Windows 2000 and Windows XP"
- JTAG_V4xx_FLASHES.pdf List of all supported Flash devices
- README.txt Release notes, new features, known problems

1.4. Connecting your PC to the target system

The JTAG-Booster can be plugged into standard parallel ports (LPT1-3) with a DB25-Connector.



The target end of the cable has the following reference:

1	2*	3	4	5	6	7	8
TCK	GND	TMS	TRST#	NC	TDI	TDO	+3.3V

*PIN 2 can be detected by the thick cable.

To connect your design to the JTAG-BOOSTER you need a single row berg connector with a spacing of 2.54mm on your PCB. The names refer to the target: Pin 7 is the target’s TDO pin and is connected to the JTAG-Booster’s TDI pin.

The 3.3V version of the JTAG-Booster (FS part number 285) is delivered together with this package. Don’t use the 5V version of the JTAG-Booster (FS part number 227) with a 3.3V target. **Don’t apply 5V to the 3.3V version of the JTAG-Booster!**

Your target must be able to power the JTAG-Booster, it draws about 100mA.

Before you start the program, the JTAG-BOOSTER must be plugged to a parallel interface of your PC and to the 8 pin JTAG connector on the target.

The utility is started with the general command line format: JTAGxxx

```
JTAuxxxx /function [filename] [/option_1] ... [/option_n].
```

Note that the function must be the first argument followed (if needed) by the filename.

If you want to cancel execution of JTAuxxxx, press CTRL-Break-Key.

On any error the program aborts with an MSDOS error level of one.

1.5. First Example with AMD Alchemy Solutions Au1000 Processor

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAu1000  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAu1000 --- JTAG utility for AMD Alchemy Solutions Au1000 Processor
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAu1000.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
        Device 0: IDCODE=003E828F  AMD Alchemy Au1000, Revision 0
(6)     Sum of instruction register bits  : 5
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : 569

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3,3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
(13)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time          :       0.8 sec
        Programming Time    :      xx.1 sec
```

(1) The initialization file JTAu1000.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here. With WinNT/Win2000/WinXP you must specify the option /LPT2 to access to the standard address 378h.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the AMD Alchemy Solutions Au1000 Processor are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the AMD Alchemy Solutions Au1000 Processor in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the AMD Alchemy Solutions Au1000 Processor is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a AMD Alchemy Solutions Au1000 Processor.

(10) A Flash AMD 29LV160B selected with RCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example 4 blocks must be erased.

(13) If available, the "Unlock Bypass Mode" is activated. In boundary scan mode, this increases the programming performance by 67..75%.

1.6. First Example with AMD Alchemy Solutions Au1100 Processor

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAu1100  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAu1100 --- JTAG utility for AMD Alchemy Solutions Au1100 Processor
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAu1100.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: IDCODE=0020228F  AMD Alchemy Au1100, Revision 0
(6)     Sum of instruction register bits  : 5
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : 672

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3.3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
(13)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time            :        0.8 sec
        Programming Time      :       xx.0 sec
```

(1) The initialization file JTAu1100.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the AMD Alchemy Solutions Au1100 Processor are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the AMD Alchemy Solutions Au1100 Processor in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the AMD Alchemy Solutions Au1100 Processor is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a AMD Alchemy Solutions Au1100 Processor.

(10) A Flash AMD 29LV160B selected with RCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example four block must be erased.

(13) If available, the "Unlock Bypass Mode" is activated. In boundary scan mode, this increases the programming performance by 67..75%.

1.7. First Example with AMD Alchemy Solutions Au1500 Processor

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
JTAu1500  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAu1500 --- JTAG utility for AMD Alchemy Solutions Au1500 Processor
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAu1500.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: IDCODE=0010228F  AMD Alchemy Au1500, Revision 0
(6)     Sum of instruction register bits  : 5
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : 744

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3.3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
(13)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time           :        0.8 sec
        Programming Time     :       xx.0 sec
```

(1) The initialization file JTAu1500.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the AMD Alchemy Solutions Au1500 Processor are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the AMD Alchemy Solutions Au1500 Processor in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the AMD Alchemy Solutions Au1500 Processor is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a AMD Alchemy Solutions Au1500 Processor.

(10) A Flash AMD 29LV160B selected with RCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example four block must be erased.

(13) If available, the "Unlock Bypass Mode" is activated. In boundary scan mode, this increases the programming performance by 67..75%.

1.8. First Example with AMD Alchemy Solutions Au1550 Processor

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
JTAu1550  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAu1550 --- JTAG utility for AMD Alchemy Solutions Au1550 Processor
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAu1550.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: IDCODE=0030228F  AMD Alchemy Au1550, Revision 0
(6)     Sum of instruction register bits  : 5
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : 773

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3.3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
(13)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time          :      0.8 sec
        Programming Time    :     xx.0 sec
```

(1) The initialization file JTAu1550.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the AMD Alchemy Solutions Au1550 Processor are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the AMD Alchemy Solutions Au1550 Processor in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the AMD Alchemy Solutions Au1550 Processor is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a AMD Alchemy Solutions Au1550 Processor.

(10) A Flash AMD 29LV160B selected with RCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example four block must be erased.

(13) If available, the "Unlock Bypass Mode" is activated. In boundary scan mode, this increases the programming performance by 67..75%.

1.9. Trouble Shooting

Avoid long distances between your Host-PC and the target. If you are using standard parallel extension cable, the JTAG-BOOSTER may not work. Don't use Dongles between the parallel port and the JTAG-BOOSTER.

Switch off all special modes of your printer port (EPP, ECP, ...) in the BIOS setup. Only standard parallel port (SPP) mode is allowed.

If there are problems with autodetection of the flash devices use the /DEVICE= option. To speed up autodetection specify one of the options /8BIT /16BIT or /32BIT.

Don't use hardware protected flash memories.

The used chip selects must be defined as output and inactive in the initialization file (see chapter 1.11 “Initialization file JTAuxxxx.INI”). Also the address bits must be defined as output.

Use the option /NOWRSETUP to speed up flash programming.

1.10. Error Messages

- **80386 or greater required**
  The JTAG-BOOSTER does not work on a 8088/8086 or a 80286 platform.

- **Cable not connected or target power fail**
  The JTAG-Booster (or one of the simple Parallel Port JTAG adapters selected with the options /LATTICE /WIGGLER /PLS) wasn't found. Please check connection to parallel port and connection to target. Check target power. Check the command line options. Check your BIOS-Setup. If you are using this program with WinNT, Win2000 or WinXP you must specify /LPT2 or /LPT-BASE=378 to get access to the standard printer port.

- **Can't open x:\yyy\zzz\JTAuxxxx.OVL**
  The overlay file JTAuxxxx.OVL must be in the same directory as JTAuxxxx.EXE.

- **Configuration file XYZ not found.**
  The file specified with the option /INI= wasn't found.

- **Device offset out of range**
  The value specified with the option /OFFSET= is greater than the size of the detected flash device.

- **Disk full**
  Writing a output file was aborted as a result of missing disk space.

- **Do not specify option /NOCS with any other chip select**
  There is a conflict in the command line.

- **Do not specify option /BYTE-MODE. *Flash device* does not have a byte mode pin.**
  The flash device specified with the option /DEVICE= does not support switching between 16 (or 32) bit mode and 8 bit mode. In practice it does not have a pin with the name BYTE#

- **Error creating file:**
  The output file could not be opened. Please check free disk space or write protection.

- **Error: *Pin-Name* is an output only pin**
  The specified pin cannot be sampled. Check the command line. Check the initialization file.

- **Error: *Pin-Name* is an input only pin**
  The specified pin cannot be activated. Check the command line. Check the initialization file.

- **Error: *Pin-Name* may not be read back**
  The specified pin can be switched to tristate, but cannot be read back. Check the command line.

- **illegal function:**
  The first parameter of the command line must be a valid function. See chapter 2 “JTAuxxxx Parameter Description” for a list of supported functions.

- **illegal number:**
  The specified number couldn’t be interpret as a valid number. Check the relevant number base.

- **illegal option:**
  See chapter 2 “JTAuxxxx Parameter Description” for a list of supported options.

- **illegal Pin Type:**
  The name specified with the option /PIN= must be one of the list of chapter 1.11 ”Initialization file JTAuxxxx.INI”

- **illegal Flash Type:**
  The name specified with the option /DEVICE= must be one of the list of chapter 1.12 “Supported flash devices”

- **Input file not found:**
  The specified file cannot be found

- **Input file is empty:**
  Files with zero length are not accepted

- **"   " is undefined**
  Please check the syntax in your configuration file. (See chapter 1.11 “Initialization file JTAuxxxx.INI”).

- **LPTx not installed**
  The LPT port specified with /LPTx cannot be found. Please check the LPT port or specify a installed LPT port. Check your BIOS setup. If you are using this program with WinNT, Win2000 or WinXP you 1st must install the WinNT support package as described in chapter 5"Support for Windows NT, Windows 2000 and Windows XP

- **missing filename**
  Most functions need a filename as second parameter.

- **missing option /SERCLK=**
  Some functions need the option /SERCLK= to be defined.

- **missing option /SERDAT=**
  Some functions need the option /SERDAT= or the options /SERDATO= and /SERDATI= to be defined.

- **missing option /SERCS=**
  Some functions need the option /SERCS= if the option /SPI or the option /MWIRE is specified.

- **missing option /LENGTH=**
  Some functions need the option /LENGTH= to be defined.

- **missing option /PIN=**
  Some functions need the option /PIN= to be defined.

- **More than 9 devices in the JTAG chain or TDO pin stuck at low level**
  The JTAG chain is limited to 9 parts. Check target power. Check the target’s TDO pin.

- **No devices found in JTAG chain or TDO pin stuck at high level**
  A stream of 32 high bits was detected on the pin TDO. TDO may stuck at high level. Check the connection to your target. Check the target power. Check the target’s TDO pin.

- **Option /CPUPOS= out of range**
  The number specified with the option /CPUPOS= must be less or equal to the number of parts minus 1.

- **Option /IROFFS= out of range**
  Please specify a smaller value

- **Part at specified position is not a AMD Alchemy Solutions Processors**
  The option /CPUPOS= points to a part not a AMD Alchemy Solutions Processors

- **Pins specified with /SERCLK= and /SERDAT= must have different control cells**
  The pin specified with the option /SERDAT= must be able to be switched to high impedance while the pin specified with option /SERCLK= is an active output. See chapter 1.11 “Initialization file JTAuxxxx.INI”.

- **Pins specified with /SERCLK= and /SERDATI= must have different control cells**
  The pin specified with the option /SERDATI= must be able to be switched to high impedance while the pin specified with option /SERCLK= is an active output. See chapter 1.11 “Initialization file JTAuxxxx.INI”.

- **Pins specified with /SERDATO= and /SERDATI= must have different control cells**
  The pin specified with the option /SERDATI= must be able to be switched to high impedance while the pin specified with option /SERDATO= is an active output. See chapter 1.11 “Initialization file JTAuxxxx.INI”.

- **Specify only one of these options:**
  Some options are exclusive (i.e. /8BIT and /16BIT). Don't mix them.

- **Sum of instruction register bits to low. Should be at least 5 bits for a AMD Alchemy Solutions Processors**
  The sum of all instruction register bits in the JTAG chain does not fit to the AMD Alchemy Solutions Processors. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

- **Target no longer connected**
  There is a cyclic check of the JTAG chain. Check target power. Check target connection.

- **There are unknown parts in the JTAG chain. Please use the option /IROFFS= to specify the instr. reg. offset of the CPU.**
  If there are unknown parts in the JTAG chain, the program isn't able to determine the logical position of the CPU's instruction register.

- **There is no AMD Alchemy Solutions Processors in the JTAG chain**
  No AMD Alchemy Solutions Processors was found in the JTAG chain. Check the target power. Try with option /DRIVER=4 again.

- **Value of option /FILE-OFFSET out of range**
  The value of the option /FILE-OFFSET= points behind end of file.

- **wrong driver #**
  The value specified with the option /DRIVER= is out of range.

- **Wrong Flash Identifier (xxxx)**
  No valid identifier found. Check the specified chip select signal and the bus width. Try with the option /DEVICE= . Use the option /8BIT or /16BIT or /32BIT to specify the correct data bus size.

- **Wrong length of boundary scan register. Should be 569 for a AMD Alchemy Solutions Au1000 Processor. (Should be 672 for a AMD Alchemy Solutions Au1100 Processor. Should be 744 for a AMD Alchemy Solutions Au1500 Processor Should be 773 for a AMD Alchemy Solutions Au1550 Processor.)**
  The length of the boundary scan register of the selected part (if there are more than one in the chain) does not fit to the AMD Alchemy Solutions Processors. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

1.11. Initialization file JTAuxxxx.INI

This file is used to define the default direction and level of all CPU signals. This file **must be carefully adapted** to your design with the AMD Alchemy Solutions Processors. The Target-Entry is used to identify your design which is displayed with most commands.

When the program JTAuxxxx.EXE is started it scans the current directory for an existing initialization file named JTAuxxxx.INI. If no entry is found the default values are used. You may also specify the initialization file with the option /INI= . If the specified file isn't found, the program aborts with an error message.

The CPU pins can also be used with the functions /BLINK (chapter 2.9), /PIN? (chapter 2.10) and /SAMPLE (chapter 2.11) to test the signals on your design.

The sample file below represents the values which are used for default initialization when no initialization file could be found in the current directory and no initialization file is specified with the option /INI=.

Changes to the structure of the file could result in errors. Remarks can be added by using //.