JTAG-Booster for

Analog Devices ADSP-21xxx

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:	April 19, 2002
Author:	Dieter Fögele
Filename:	JTAG_ADSP-21xxxa.doc
Program Version:	4.xx

Table of Contents

1. General

The programs JTAG060.EXE, JTAG065.EXE and JTAG161.EXE use the JTAG port of the Analog Devices ADSP-21xxx DSP Microcomputer 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 an I²C Device
- 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.

The JTAG-BOOSTER' s software is highly optimized to the JTAG chain of a specific target CPU. To give support for all DSPs of the Analog Devices ADSP-21xxx family, there are three different programs on the distribution disk:

• JTAG060.EXE	: Tool for Analog Devices ADSP-21060/21061/21062
• JTAG065.EXE	: Tool for Analog Devices ADSP-21065
• JTAG161.EXE	: Tool for Analog Devices ADSP-21161

Please contact us, if you need support for other members of the Analog Devices ADSP-21xxx 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

- 940 JTAG-Booster Analog Devices ADSP-21xxx, 5V,
  DOS/Win9x/WinNT,
  delivered with adapter type 227

- 953 JTAG-Booster Analog Devices ADSP-21xxx, 3.3V,
  DOS/Win9x/WinNT,
  delivered with adapter type 285

1.2. System Requirements

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

- MSDOS, WIN3.x, WIN9x, WinME, WinNT or Win2000
  (WinNT/Win2000 is supported with an additional tool, see chapter 5)

- Intel 80386 or higher

- 205 kByte of free DOS memory

- Parallel Port

1.3. Contents of Distribution Disk

- JTAG060.EXE
JTAG060.OVL — Tool for Analog Devices ADSP-21060/21061/21062

- JTAG060.INI — Template configuration file for Analog Devices ADSP-21060/21061/21062. See chapter 1.10 "Initialization file JTAGxxx.INI"

- JTAG065.EXE
JTAG065.OVL — Tool for Analog Devices ADSP-21065

- JTAG065.INI — Template configuration file for Analog Devices ADSP-21065. See chapter 1.10 "Initialization file JTAGxxx.INI"

- JTAG161.EXE
JTAG161.OVL — Tool for Analog Devices ADSP-21060/21061/21062

- JTAG161.INI — Template configuration file for Analog Devices ADSP-21161. See chapter 1.10 "Initialization file JTAGxxx.INI"

- HEX2BIN.EXE — Converter program to convert Intel HEX and Motorola S-Record files to binary. See chapter 4 "Converter Program HEX2BIN.EXE"

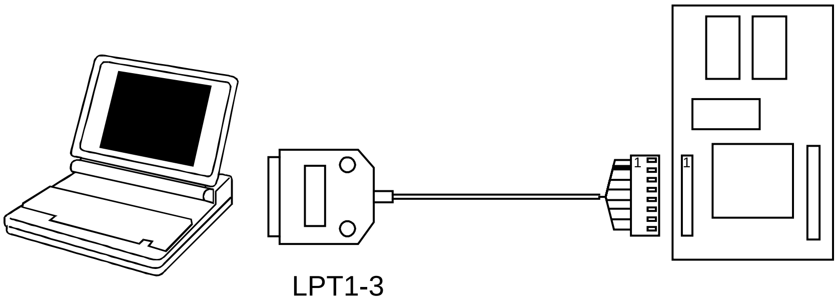
- WinNT.zip — Support for Windows NT and Windows 2000. See chapter 5 "Support for Windows NT and Windows 2000"

- 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 / +5V

*PIN 2 can be detected by the white 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.

There are two versions of the JTAG-Booster available: A 5V version (FS part number 227) and a 3.3V version (FS part number 285). **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 /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 JTAGxxx, press CTRL-Break-Key.

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

1.5. First Example with Analog Devices ADSP-21060/21061/21062

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

```
        JTAG060  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG060 --- JTAG utility for Analog Devices ADSP-21060/21061/21062
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAGxxx.INI
(2)     Target: Analog Devices SHARC EZ-KIT
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21060/21061/21062 is selected
(7)     Sum of instruction register bits  : 5
(8)     CPU position                      : 0
(9)     Instruction register offset       : 0
(10)    Length of boundary scan reg       : 363

        Looking for a known flash device. Please wait..
(11)    AMD 29F010 detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time          :        1.3 sec
        Programming Time    :      201.6 sec
```

(1) The initialization file JTAG060.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 Analog Devices ADSP-21060/21061/21062 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

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

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29F010 selected with BMS# was found.

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

(13) In this example one block must be erased.

1.6. First Example with Analog Devices ADSP-21065

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

```
JTAG065  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG065 --- JTAG utility for Analog Devices ADSP-21065
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG065.INI
(2)     Target: Generic Target with ADSP-21065
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21065 is selected
(7)     Sum of instruction register bits  : 5
(8)     CPU position                      : 0
(9)     Instruction register offset       : 0
(10)    Length of boundary scan reg       : 285

        Looking for a known flash device. Please wait..
(11)    AMD 29LV160B, 3.3V, Boot Block Bottom, byte mode detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0  1  2  3
(14)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time          :        0.9 sec
        Programming Time    :       81.5 sec
```

(1) The initialization file JTAG065.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 Analog Devices ADSP-21065 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

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

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29LV160B selected with BMS# was found.

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

(13) In this example four blocks must be erased.

(14) If the used Flash-EPROM supports the unlock bypass feature, it is used to speed up programming performance.

1.7. First Example with Analog Devices ADSP-21161

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

```
        JTAG161  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG161--- JTAG utility for Analog Devices ADSP-21161
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG161.INI
(2)     Target: Generic Target with Analog Devices ADSP-21161
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21161 is selected
(7)     Sum of instruction register bits   : 5
(8)     CPU position                       : 0
(9)     Instruction register offset        : 0
(10)    Length of boundary scan reg        : 481

        Looking for a known flash device. Please wait..
(11)    AMD 29LV160B, 3.3V, Boot Block Bottom, byte mode detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0  1  2  3
(14)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time          :         0.9 sec
        Programming Time    :        ??.0 sec
```

(1) The initialization file JTAG161.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 Analog Devices ADSP-21161 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

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

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29LV160B selected with BMS# was found.

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

(13) In this example four blocks must be erased.

(14) If the used Flash-EPROM supports the unlock bypass feature, it is used to speed up programming performance.

1.8. 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.

On very fast PCs there could be verify errors. To avoid this, watch for the ´IO recovery time´-switch in the BIOS Setup which must be turned on. Otherwise try to slow down your PC by setting the turbo switch off.

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 1.10 “Initialization file JTAGxxx.INI”). Also the address bits must be defined as output.

Use the option /NOWRSETUP to speed up flash programming.

1.9. Error Messages

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

- **Adapter not connected or target power fail**
  The JTAG-Booster wasn't found. Please check connection to parallel port and connection to target. Check target power. Check your BIOS-Setup.

- **Can't open x:\yyy\zzz\JTAGxxx.OVL**
  The overlay file JTAGxxx.OVL must be in the same directory as JTAGxxx.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 “JTAGxxx 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 “JTAGxxx 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 1.10 ”Initialization file JTAGxxx.INI”

- **illegal Flash Type:**
  The name specified with the option /DEVICE= must be one of the list of chapter 1.11 ”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 1.10 “Initialization file JTAGxxx.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.

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

- **missing option /I2CCLK=**
  Some functions need the option /I2CCLK= to be defined.

- **missing option /I2CDAT=**
  Some functions need the option /I2CDAT= or the options /I2CDATO= and /I2CDATI= to be defined.

- **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 TDI 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 TDI pin stuck at high level**
  A stream of 32 high bits was detected on the pin TDI. TDI 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 Analog Devices ADSP-21xxx**
  The option /CPUPOS= points to a part not a Analog Devices ADSP-21xxx

- **Pins specified with /I2CCLK= and /I2CDAT= must have different control cells**
  The pin specified with the option /I2CDAT= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.10 "Initialization file JTAGxxx.INI".

- **Pins specified with /I2CCLK= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.10 "Initialization file JTAGxxx.INI".

- **Pins specified with /I2CDATO= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CDATO= is an active output. See chapter 1.10 "Initialization file JTAGxxx.INI".

- **Specify only one of that 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 Analog Devices ADSP-21xxx**
  The sum of all instruction register bits in the JTAG chain does not fit to the Analog Devices ADSP-21xxx. 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 Analog Devices ADSP-21xxx in the JTAG chain**
  No Analog Devices ADSP-21xxx 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 Identifier (xxxx)**
  No valid identifier found. Check the specified chip select signal and the bus width. Try with the option /DEVICE= .

- **Wrong length of boundary scan register. Should be 363 for a Analog Devices ADSP-21060/21061/21062. (Should be 285 for a Analog Devices ADSP-21065/Should be 481 for a Analog Devices ADSP-21161.)**
  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 Analog Devices ADSP-21xxx. 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.

1.10. Initialization file JTAGxxx.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 Analog Devices ADSP-21xxx. The Target-Entry is used to identify your design which is displayed with most commands.

When the program JTAGxxx.EXE is started it scans the current directory for an existing initialization file named JTAGxxx.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 //.

Sample File JTAG060.INI:

```
// Description file for Analog Devices ADSP-21xxx
Target: Analog Devices SHARC EZ-KIT
// All chip select signals are set to output and inactive.
// All signals should be defined. Undefined signals are set to their defaults.
// Pin names are defined in upper case.
// Low active signals are signed with a trailing #.

// Group A:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L5CLK          Inp        // Link Port Clock
L5DAT0         Inp        // Link Port Data
L5DAT1         Inp        //
L5DAT2         Inp        //
L5DAT3         Inp        //

// Group B:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L4CLK          Inp        // Link Port Clock
L4DAT0         Inp        // Link Port Data
L4DAT1         Inp        //
L4DAT2         Inp        //
L4DAT3         Inp        //

// Group C:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L3CLK          Inp        // Link Port Clock
L3DAT0         Inp        // Link Port Data
L3DAT1         Inp        //
L3DAT2         Inp        //
L3DAT3         Inp        //

// Group D:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L2CLK          Inp        // Link Port Clock
L2DAT0         Inp        // Link Port Data
L2DAT1         Inp        //
L2DAT2         Inp        //
L2DAT3         Inp        //
```

```
// Group E:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L1CLK          Inp      // Link Port Clock
L1DAT0         Inp      // Link Port Data
L1DAT1         Inp      //
L1DAT2         Inp      //
L1DAT3         Inp      //

// Group F:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L0CLK          Inp      // Link Port Clock
L0DAT0         Inp      // Link Port Data
L0DAT1         Inp      //
L0DAT2         Inp      //
L0DAT3         Inp      //

// Group G:  All pins in this group must be set to the same direction
//           This pins are bidirectional
//           This group is switched between output/active and
//           input/tristate during programming of Flash-EPROMs
DATA0          Inp      // External Bus Data
DATA1          Inp      //
DATA2          Inp      //
DATA3          Inp      //
DATA4          Inp      //
DATA5          Inp      //
DATA6          Inp      //
DATA7          Inp      //
DATA8          Inp      //
DATA9          Inp      //
DATA10         Inp      //
DATA11         Inp      //
DATA12         Inp      //
DATA13         Inp      //
DATA14         Inp      //
DATA15         Inp      //
DATA16         Inp      // = Boot-ROM D0
DATA17         Inp      // = Boot-ROM D1
DATA18         Inp      // = Boot-ROM D2
DATA19         Inp      // = Boot-ROM D3
DATA20         Inp      // = Boot-ROM D4
DATA21         Inp      // = Boot-ROM D5
DATA22         Inp      // = Boot-ROM D6
```

```
DATA23         Inp        // = Boot-ROM D7
DATA24         Inp        //
DATA25         Inp        //
DATA26         Inp        //
DATA27         Inp        //
DATA28         Inp        //
DATA29         Inp        //
DATA30         Inp        //
DATA31         Inp        //
DATA32         Inp        //
DATA33         Inp        //
DATA34         Inp        //
DATA35         Inp        //
DATA36         Inp        //
DATA37         Inp        //
DATA38         Inp        //
DATA39         Inp        //
DATA40         Inp        //
DATA41         Inp        //
DATA42         Inp        //
DATA43         Inp        //
DATA44         Inp        //
DATA45         Inp        //
DATA46         Inp        //
DATA47         Inp        //

// Group H:  All pins in this group must be set to the same direction
//           DMAG1#, DMAG2# and ADRCLK are tristateable outputs but may
//           not be read back.
//           PAGE, RD# and WR# are bidirectional pins.
//           This group is switched to output/active during programming of
//           Flash-EPROMs.
PAGE           Out,Lo     // DRAM Page Boundary
DMAG1#         Out,Hi     // DMA Grant 1, DMA Channel 7
DMAG2#         Out,Hi     // DMA Grant 2, DMA Channel 8
WR#            Out,Hi     // Memory Write Strobe = WE# of Flash-EPROM
RD#            Out,Hi     // Memory Read Strobe = OE# of Flash-EPROM
ADRCLK         Out,Lo     // Clock Output Reference
```

```
// Group I:   All pins in this group must be set to the same direction
//            This pins are bidirectional
//            For Flash programming this pins must be set to output.
SW#               Out,Hi    // Synchronous Write Select
MS0#              Out,Hi    // Memory Select Line
MS1#              Out,Hi    // Memory Select Line
MS2#              Out,Hi    // Memory Select Line
MS3#              Out,Hi    // Memory Select Line
ADDR31            Out,Lo    //
ADDR30            Out,Lo    //
ADDR29            Out,Lo    //
ADDR28            Out,Lo    //
ADDR27            Out,Lo    //
ADDR26            Out,Lo    //
ADDR25            Out,Lo    //
ADDR24            Out,Lo    //
ADDR23            Out,Lo    //
ADDR22            Out,Lo    //
ADDR21            Out,Lo    //
ADDR20            Out,Lo    //
ADDR19            Out,Lo    //
ADDR18            Out,Lo    //
ADDR17            Out,Lo    //
ADDR16            Out,Lo    //
ADDR15            Out,Lo    //
ADDR14            Out,Lo    //
ADDR13            Out,Lo    //
ADDR12            Out,Lo    //
ADDR11            Out,Lo    //
ADDR10            Out,Lo    //
ADDR9             Out,Lo    //
ADDR8             Out,Lo    //
ADDR7             Out,Lo    //
ADDR6             Out,Lo    //
ADDR5             Out,Lo    //
ADDR4             Out,Lo    //
ADDR3             Out,Lo    //
ADDR2             Out,Lo    //
ADDR1             Out,Lo    //
ADDR0             Out,Lo    //
```

```
// The following pins are complete bidirectional pins.
// The direction of each pin can be set independent of the other pins.
// Each pin can be used as input.
L5ACK           Inp        // Link Port Acknowledge
L4ACK           Inp        // Link Port Acknowledge
L3ACK           Inp        // Link Port Acknowledge
L2ACK           Inp        // Link Port Acknowledge
L1ACK           Inp        // Link Port Acknowledge
L0ACK           Inp        // Link Port Acknowledge
BR1#            Inp        // Multiprocessing Bus Request
BR2#            Inp        // Multiprocessing Bus Request
BR3#            Inp        // Multiprocessing Bus Request
BR4#            Inp        // Multiprocessing Bus Request
BR5#            Inp        // Multiprocessing Bus Request
BR6#            Inp        // Multiprocessing Bus Request
ACK             Inp        // Memory Acknowledge
HBG#            Inp        // Host Bus Grant
RFS0            Inp        // Receive Frame Sync
RCLK0           Inp        // Receive Clock
TFS0            Inp        // Transmit Frame Sync
TCLK0           Inp        // Transmit Clock
RFS1            Inp        // Receive Frame Sync
RCLK1           Inp        // Receive Clock
TFS1            Inp        // Transmit Frame Sync
TCLK1           Inp        // Transmit Clock
BMS#            Out,Hi     // Boot Memory Select = CS# of Flash-EPROM
FLAG0           Out,Lo     // Flag Pin = LEx
FLAG1           Out,Hi     // Flag Pin = LEx
FLAG2           Out,Lo     // Flag Pin = LEx
FLAG3           Out,Hi     // Flag Pin = LEx

// The direction of each following pin can be set independent of the others,
// but this pins can not be read back.
REDY            Inp        // Host Bus Acknowledge
DT0             Out,Lo     // Data Transmit
DT1             Out,Lo     // Data Transmit
EMU#            Inp        // Emulation Status

// The following pin has an open drain output with a weak pull up.
// The pin is bidirectional.
CPA#            Inp        // Core Priority Access

// The following pins are output only pins.
// Setting to input (tristate) one of these pins results in an error.
ICSA            Out,Lo    // Reserved Output
TIMEXP          Out,Lo    // Timer Expired

// The following pins are input only.
// Setting to output of one of these pins results in an error.
// Declaration of the direction of these pins is optional.
IRQ0#           Inp       // Interrupt Request
IRQ1#           Inp       // Interrupt Request
IRQ2#           Inp       // Interrupt Request
EBOOT           Inp       // EPROM Boot Select, EBOOT=1 -> Boot from EPROM
RESET#          Inp       // Processor Reset
RPBA            Inp       // Rotating Priority Bus Arbitration Select
LBOOT           Inp       // Link Boot, is Low for EPROM boot
ID0             Inp       // Multiprocessing ID
ID1             Inp       // Multiprocessing ID
ID2             Inp       // Multiprocessing ID
CLKIN           Inp       // Clock In
CS#             Inp       // Chip Select Input
DR0             Inp       // Data Receive
DR1             Inp       // Data Receive
HBR#            Inp       // Host Bus Request
DMAR1#          Inp       // DMA Request 1, DMA Channel 7
DMAR2#          Inp       // DMA Request 2, DMA Channel 8
SBTS#           Inp       // Suspend Bus Three-State
```

Sample File JTAG065.INI:

```
// Description file for Analog Devices ADSP-21065
Target: Generic Target with ADSP-21065
// All chip select signals are set to output and inactive.
// All signals should be defined. Undefined signals are set to their defaults.
// Pin names are defined in upper case.
// Low active signal are signed with a trailing #.

// Group A:  All pins in this group must be set to the same direction
//           This pins are bidirectional
//           This group is switched between output/active and
//           input/tristate during programming of Flash-EPROMs
DATA0        Inp      // = Boot-ROM D0
DATA1        Inp      // = Boot-ROM D1
DATA2        Inp      // = Boot-ROM D2
DATA3        Inp      // = Boot-ROM D3
DATA4        Inp      // = Boot-ROM D4
DATA5        Inp      // = Boot-ROM D5
DATA6        Inp      // = Boot-ROM D6
DATA7        Inp      // = Boot-ROM D7
DATA8        Inp      //
DATA9        Inp      //
DATA10       Inp      //
DATA11       Inp      //
DATA12       Inp      //
DATA13       Inp      //
DATA14       Inp      //
DATA15       Inp      //
DATA16       Inp      //
DATA17       Inp      //
DATA18       Inp      //
DATA19       Inp      //
DATA20       Inp      //
DATA21       Inp      //
DATA22       Inp      //
DATA23       Inp      //
DATA24       Inp      //
DATA25       Inp      //
DATA26       Inp      //
DATA27       Inp      //
DATA28       Inp      //
DATA29       Inp      //
```

```
DATA30        Inp      //
DATA31        Inp      //

// Group B: All pins in this group must be set to the same direction
//          This pins are bidirectional
//          For Flash programming this pins must be set to output.
ADDR23        Out,Lo   //
ADDR22        Out,Lo   //
ADDR21        Out,Lo   //
ADDR20        Out,Lo   //
ADDR19        Out,Lo   //
ADDR18        Out,Lo   //
ADDR17        Out,Lo   //
ADDR16        Out,Lo   //
ADDR15        Out,Lo   //
ADDR14        Out,Lo   //
ADDR13        Out,Lo   //
ADDR12        Out,Lo   //
ADDR11        Out,Lo   //
ADDR10        Out,Lo   //
ADDR9         Out,Lo   //
ADDR8         Out,Lo   //
ADDR7         Out,Lo   //
ADDR6         Out,Lo   //
ADDR5         Out,Lo   //
ADDR4         Out,Lo   //
ADDR3         Out,Lo   //
ADDR2         Out,Lo   //
ADDR1         Out,Lo   //
ADDR0         Out,Lo   //

// Group C: All pins in this group must be set to the same direction
//          DMAG1# and DMAG2# are tristateable outputs but can't be
//          read back.
//          RD#, WR#, SW# and MSx# are bidirectional pins.
//          This group is switched to output/active during programming of
//          Flash-EPROMs.
DMAG1#        Out,Hi   // DMA Grant 1, DMA Channel 7
DMAG2#        Out,Hi   // DMA Grant 2, DMA Channel 8
WR#            Out,Hi  // Memory Write Strobe = WE# of Flash-EPROM
RD#            Out,Hi  // Memory Read Strobe = OE# of Flash-EPROM
MS0#          Out,Hi   // Memory Select Line
MS1#          Out,Hi   // Memory Select Line
```

```
MS2#            Out,Hi    // Memory Select Line
MS3#            Out,Hi    // Memory Select Line
SW#              Out,Hi   // Synchronous Write Select

// Group D: All pins in this group must be set to the same direction
//          DQM and SDA10 are tristateable outputs but can't be read back.
//          All others are bidirectional pins.
//          This group is switched to output/active during programming of
//          Flash-EPROMs.
SDCLK0          Out,Lo    // SDRAM clock output
RAS#            Out,Hi    // SDRAM row address strobe
CAS#            Out,Hi    // SDRAM column address strobe
SDWE#           Out,Hi    // SDRAM write enable
DQM             Out,Lo    // SDRAM data mask
SDCKE           Out,Hi    // SDRAM clock enable
SDA10           Out,Lo    // SDRAM A10 pin

// The following pins are complete bidirectional pins.
// The direction of each pin can be set independent of the other pins.
// Each pin can be used as input.
BR1#            Inp       // Multiprocessing Bus Request
BR2#            Inp       // Multiprocessing Bus Request
ACK             Inp       // Memory Acknowledge
HBG#            Inp       // Host Bus Grant
RFS0            Inp       // Receive Frame Sync
RCLK0           Inp       // Receive Clock
TFS0            Inp       // Transmit Frame Sync
TCLK0           Inp       // Transmit Clock
RFS1            Inp       // Receive Frame Sync
RCLK1           Inp       // Receive Clock
TFS1            Inp       // Transmit Frame Sync
TCLK1           Inp       // Transmit Clock
BMS#            Out,Hi    // Boot Memory Select = CS# of Flash-EPROM
FLAG0           Inp       // Flag Pin
FLAG1           Inp       // Flag Pin
FLAG2           Inp       // Flag Pin
FLAG3           Inp       // Flag Pin
FLAG4           Inp       // Flag Pin
FLAG5           Inp       // Flag Pin
FLAG6           Inp       // Flag Pin
FLAG7           Inp       // Flag Pin
FLAG8           Inp       // Flag Pin
FLAG9           Inp       // Flag Pin
```

```
FLAG10        Inp       // Flag Pin
FLAG11        Inp       // Flag Pin
PWM_EVENT1 Inp       // PWM output/event capture
PWM_EVENT0 Inp       // PWM output/event capture
SDCLK1        Out,Lo    // SDRAM clock output

// The direction of each following pin can be set independent of the others,
// but this pins can not be read back.
REDY          Inp       // Host Bus Acknowledge
DT0_A         Out,Lo    // Data Transmit
DT0_B         Out,Lo    // Data Transmit
DT1_A         Out,Lo    // Data Transmit
DT1_B         Out,Lo    // Data Transmit
EMU#          Inp       // Emulation Status

// The following pin has an open drain output with a weak pull up.
// The pin is bidirectional.
CPA#          Inp       // Core Priority Access

// The following pins are output only pins.
// Setting to input (tristate) one of these pins results in an error.
BMSTR         Out,Hi    // Bus Master Output

// The following pins are input only.
// Setting to output of one of these pins results in an error.
// Declaration of the direction of these pins is optional.
IRQ0#         Inp       // Interrupt Request
IRQ1#         Inp       // Interrupt Request
IRQ2#         Inp       // Interrupt Request
BSEL          Inp       // EPROM Boot Select, BSEL=1 -> Boot from EPROM
RESET#        Inp       // Processor Reset
ID0           Inp       // Multiprocessing ID
ID1           Inp       // Multiprocessing ID
CLKIN         Inp       // Clock In
CS#           Inp       // Chip Select Input
DR0_A         Inp       // Data Receive
DR0_B         Inp       // Data Receive
DR1_A         Inp       // Data Receive
DR1_B         Inp       // Data Receive
HBR#          Inp       // Host Bus Request
DMAR1#        Inp       // DMA Request 1, DMA Channel 7
DMAR2#        Inp       // DMA Request 2, DMA Channel 8
SBTS#         Inp       // Suspend Bus Three-State
```

Sample File JTAG161.INI:

```
// Description file for Analog Devices ADSP-21161
Target: Generic Target with ADSP-21161
// All chip select signals are set to output and inactive.
// All signals should be defined. Undefined signals are set to their defaults
// Pin Names are defined in upper case
// Low Active Signal are signed with a trailing #

// The following pins are complete bidirectional pins
// The direction of each pin can be set independent of the other pins
// Each pin can be used as an input
EBOOT           Inp       // EPROM Boot Select, EBOOT=1 -> Boot from EPROM
LBOOT           Inp       // Link Boot, should be low for EBOOT=1
WR#             Out,Hi    // Memory Write Strobe = WE# of Flash-EPROM
RD#             Out,Hi    // Memory Read Strobe = OE# of Flash-EPROM
BMS#            Out,Hi    // Boot Memory Select = CS# of Flash-EPROM
MS0#            Out,Hi    // Memory Select Lines
MS1#            Out,Hi    //
MS2#            Out,Hi    //
MS3#            Out,Hi    //
BMSTR           Out,Hi    // Bus Master Output
EMU#            Inp       // Emulation Status Output
RESET#          Inp       // Processor Reset Input

FLAG11          Inp       // Flag Pins
FLAG10          Inp       //
FLAG9           Inp       //
FLAG8           Inp       //
FLAG7           Inp       //
FLAG6           Inp       //
FLAG5           Inp       //
FLAG4           Inp       //
FLAG3           Inp       //
FLAG2           Inp       //
FLAG1           Inp       //
FLAG0           Inp       //

IRQ0#           Inp       // Interrupt Request Lines
IRQ1#           Inp       //
IRQ2#           Inp       //
ID0             Inp       // Multiprocessing ID
ID1             Inp       //
```

```
ID2           Inp       //
TIMEXP        Inp       // Timer Expired
SBTS#         Inp       // Suspend Bus & Three-State
PA#           Inp       // Priority Access
BR6#          Inp       // Multiprocessing Bus Request
BR5#          Inp       //
BR4#          Inp       //
BR3#          Inp       //
BR2#          Inp       //
BR1#          Inp       //
BRST          Inp       // Sequential Burst Access
SDCLK0        Out,Lo    // SDRAM Clock Output 0
SDCLK1        Out,Lo    // SDRAM Clock Output 1
SDA10         Out,Lo    // SDRAM A10 Pin
SDCKE         Out,Lo    // SDRAM Clock Enable
CLKOUT        Out,Lo    // Local Clock Out
CAS#          Out,Hi    // SDRAM Column Access Strobe
RAS#          Out,Hi    // SDRAM Row Acess Strobe
HBR#          Inp       // Host Bus Request
HBG#          Inp       // Host Bus Grant
REDY          Inp       // Host Bus Acknowledge
ACK           Inp       // Memory Acknowlwdge
CS#           Inp       // Chip Select ADSP-21161
CLKDBL        Inp       // Crystal Double Mode Enable
DQM           Out,Hi    // SDRAM Data Mask
SDWE#         Out,Hi    // SDRAM Write Enable
CLK_CFG1      Inp       // Core/CLKIN Ratio Control
CLK_CFG0      Inp       //
DMAR2#        Inp       // DMA Request 2
DMAG2#        Out,Hi    // DMA Grant 2
DMAR1#        Inp       // DMA Request 1
DMAG1#        Out,Hi    // DMA Grant 1
NC1           Inp       //

L0DAT0        Inp       // Link Port 0 Data
L0DAT1        Inp       //
L0DAT2        Inp       //
L0DAT3        Inp       //
L0DAT4        Inp       //
L0DAT5        Inp       //
L0DAT6        Inp       //
L0DAT7        Inp       //
L0ACK         Inp       // Link Port 0 Acknowledge
```

```
L0CLK         Inp      // Link Port 0 Clock

L1DAT0        Inp      // Link Port 1 Data
L1DAT1        Inp      //
L1DAT2        Inp      //
L1DAT3        Inp      //
L1DAT4        Inp      //
L1DAT5        Inp      //
L1DAT6        Inp      //
L1DAT7        Inp      //
L1CLK         Inp      // Link Port 1 Acknowledge
L1ACK         Inp      // Link Port 1 Clock

FS0           Inp      // Transmit/Receive Frame Sync
SCLK0         Inp      // Transmit/Receive Serial Clock
D0A           Inp      // Data Transmit/Receive Channel A
D0B           Inp      // Data Transmit/Receive Channel B

FS1           Inp      //
SCLK1         Inp      //
D1A           Inp      //
D1B           Inp      //

SFS2          Inp      //
SCLK2         Inp      //
D2A           Inp      //
D2B           Inp      //

SFS3          Inp      //
SCLK3         Inp      //
D3A           Inp      //
D3B           Inp      //

SPIDS#        Inp      // Serial Peripheral Interface Slave Device Select
SPICLK        Inp      // Serial Peripheral Interface Clock Signal
MOSI          Inp      // SPI Master Out Slave
MISO          Inp      // SPI Master In Slave Out
RPBA#         Inp      // Rotating Priority Bus Arbitration Select
```

```
// The following pins are complete bidirectional pins
// The direction of each pin can be set independent of the other pins
// For Flash Programming these pins must be set to output
ADDR0         Out,Lo    // External Bus Address
ADDR1         Out,Lo    //
ADDR2         Out,Lo    //
ADDR3         Out,Lo    //
ADDR4         Out,Lo    //
ADDR5         Out,Lo    //
ADDR6         Out,Lo    //
ADDR7         Out,Lo    //
ADDR8         Out,Lo    //
ADDR9         Out,Lo    //
ADDR10        Out,Lo    //
ADDR11        Out,Lo    //
ADDR12        Out,Lo    //
ADDR13        Out,Lo    //
ADDR14        Out,Lo    //
ADDR15        Out,Lo    //
ADDR16        Out,Lo    //
ADDR17        Out,Lo    //
ADDR18        Out,Lo    //
ADDR19        Out,Lo    //
ADDR20        Out,Lo    //
ADDR21        Out,Lo    //
ADDR22        Out,Lo    //
ADDR23        Out,Lo    //

// The following pins are complete bidirectional pins
// These pins are switched between output/active and input/tristate during
// programming of Flash-EPROMs
DATA16        Inp       // = Boot-ROM D0
DATA17        Inp       // = Boot-ROM D1
DATA18        Inp       // = Boot-ROM D2
DATA19        Inp       // = Boot-ROM D3
DATA20        Inp       // = Boot-ROM D4
DATA21        Inp       // = Boot-ROM D5
DATA22        Inp       // = Boot-ROM D6
DATA23        Inp       // = Boot-ROM D7
DATA24        Inp       //
DATA25        Inp       //
DATA26        Inp       //
DATA27        Inp       //
```

DATA28	Inp	//
DATA29	Inp	//
DATA30	Inp	//
DATA31	Inp	//
DATA32	Inp	//
DATA33	Inp	//
DATA34	Inp	//
DATA35	Inp	//
DATA36	Inp	//
DATA37	Inp	//
DATA38	Inp	//
DATA39	Inp	//
DATA40	Inp	//
DATA41	Inp	//
DATA42	Inp	//
DATA43	Inp	//
DATA44	Inp	//
DATA45	Inp	//
DATA46	Inp	//
DATA47	Inp	//

1.11. Supported flash devices

Type JTAGxxx /LIST [optionlist]

to get a online list of all flash types which could be used with the /DEVICE= option.

See separate file JTAG_V4xx_FLASHES.pdf to get a complete list of supported flash types.

2. JTAGxxx Parameter Description

When you start JTAGxxx.EXE without any parameters the following help screen with all possible functions and options is displayed:

```
JTAGxxx --- JTAG utility for Analog Devices ADSP-21xxx
Copyright © FS FORTH-SYSTEME GmbH, Breisach
Version 4.xx of mm/dd/yyyy

Programming of Flash-EPROMs and hardware tests on targets with the
Analog Devices ADSP-21xxx.

The JTAG-Booster is needed to connect the parallel port of the PC
to the JTAG port of the Analog Devices ADSP-21xxx.

Usage: JTAGxxx /function [filename] [/option_1] ... [/option_n]
Supported functions:
     /P          : Program a Flash Device
     /R          : Read a Flash Device to file
     /V          : Verify a Flash Device with file
     /DUMP       : Make a target dump
     /PI2C       : Program an I2C Device with file
     /RI2C       : Read an I2C Device to file
     /VI2C       : Verify an I2C Device with file
     /DUMPI2C    : Make a dump of an I2C Device
     /BLINK      : Toggle a CPU pin
     /PIN?       : Test a CPU pin
     /SAMPLE     : Test a CPU pin while the CPU is running
     /SNAP       : Test all CPU pins while CPU is running
     /LIST       : Print a list of supported Flash devices
```

Supported Options:

/BMS	/MS0	/MS1	/MS2	/MS3
/NOCS	/NOWRSETUP	/TOP	/BYTE-MODE	/BM
/PAUSE	/P	/NODUMP	/NOERASE	/LATTICE
/ERASEALL	/LPT1	/LPT2	/LPT3	/LPT-BASE=
/32BIT	/16BIT	/8BIT	/NOMAN	/LENGTH=
L=	/FILE-OFFSET=	/FO=	/OFFSET=	/O=
/DELAY=	/DEVICE-BASE=	/DB=	/DRIVER=	/DATA-MASK=
/DM=	/IROFFS=	/CPUPOS=	/DEVICE=	/PIN=
/I2CCLK=	/I2CDAT=	/I2CDATI=	/I2CDATO=	I2CBIG
/WATCH=	/OUT=	/INI=	/REP	

The following options are valid for most functions:

/DRIVER=x with x = 1,2,3,4

A driver for the interface to the JTAG-BOOSTER on the parallel port may be specified. /DRIVER=1 selects the fastest available driver, /DRIVER=4 selects the slowest one. Use a slower driver if there are problems with JTAG-BOOSTER.

Default: /DRIVER=3

/INI=file

An initialization file may be specified. By default the current directory is searched for the file JTAGxxx.INI. If this file is not found and no initialization file is specified in the command line, default initialization values are used (see also chapter 1.10 “Initialization file JTAGxxx.INI”).

Default: /INI=JTAGxxx.INI

/LATTICE

For demonstration purposes this software works with the Lattice ispLSI-Adapter, too. With the option /LATTICE you can simulate the speed achievable with the simple ispLSI-Adapter.

/LPT1 /LPT2 /LPT3

A printer port may be specified where the JTAG-Booster resides.

Default: /LPT1

/LPT-BASE
The physical I/O-Address of printer port may be specified instead of the logical printer name. Useful option, if you work with WinNT or Win2000, because the standard printer port is mapped as LPT2 here. Use the option /LPT-BASE=378 to get a command line which works independent of the operation system.

/OUT=file_or_device
All screen outputs are redirected to the specified file or device. Note that you can't redirect to the same parallel port where the JTAG-Booster resides.
Default: /OUT=CON

/PAUSE
With the option /PAUSE you can force the program to stop after each screen. Please do not use this option if you redirect the output to a file.
Abbreviation: /P

/WATCH=
With the option /WATCH= a pin can be specified, which is toggled twice per second, while the program is active. This pin may be the trigger of a watchdog. This pin must be specified as output in the initialization file.

/IROFFS=
Specifies the position of the Analog Devices ADSP-21xxx instruction register within the JTAG chain. In most cases this option is not needed.
Default: /IROFFS=0

/CPUPOS=
Specifies the position of the Analog Devices ADSP-21xxx within the JTAG chain.
Default: /CPUPOS=0

2.1. Program a Flash Device

Usage: JTAGxxx /P filename [optionlist]

The specified file is programmed into the flash memory. The flash status is polled after programming of each cell (cell=8, 16 or 32 bit, depending on current data bus width). In case of a programming error, the contents of the flash memory is written to a file with the extension DMP.

If you want a complete verify after programming, please use an additional command line with the verify function. See chapter 2.3 “Verify a Flash Device with file”. In most cases this additional verify step is not needed.

The type of the flash device is normally detected by the software. When autodetection fails you should use the /DEVICE= option together with /8BIT or /16BIT or /32BIT to set the right flash device and configuration. The known flash devices are shown in chapter 1.11 “Supported flash devices”.

Options:

/DEVICE=devicename
The flash device is detected automatically by switching to autoselect mode. In case of trouble you should select the flash device by using this parameter to avoid autodetection. Combine this option with one of the following options which specify the data bus width and the option /BYTE-MODE if applicable.

/8BIT /16BIT /32BIT
Specifies the data bus width to the target flash device. You can speed up autodetection, if you specify the correct data bus size. You need this option together with the option /DEVICE= to explicit specify a specific flash configuration.

/BYTE-MODE
If there is a flash device connected to the CPU which does have a byte mode pin (8 bit and 16/32 bit bus mode), you can force it to be used as 8 bit mode with the option /BYTE-MODE. In most cases this option will not be needed.

/NOMAN
If you use a flash device which is identical to one of the supported parts, but is from a different manufacturer, with this option you can suppress the comparison of the manufacturer identification code. We recommend to use this option together with the /DEVICE= option to avoid failures in autodetection.

/DEVICE-BASE=hhhhhh[1]
Here you can specify a flash device starting address. In most cases, where the flash device is selected with one of the CPUs chip select pins, this parameter is not needed. But if there is any decoding logic in your hardware, this option will be needed. Especially, if there are several flash banks connected to one chip select and a sub decoding logic generates chip selects for these flash banks, this option can be used to select a specific flash bank.

Default:	/DEVICE-BASE=0
Abbreviation:	/DB=

/OFFSET=hhhhhh
The programming starts at an offset of hhhhhh relative to the start address of the flash device. If the offset is negative, the offset specifies an address relative to the end of the flash device. See also option /TOP

Default:	/OFFSET=0
Abbreviation:	/O=

/TOP
If the option /TOP is used the option /OFFSET= specifies the address where the programming ends (plus one) instead of the starting address. This option is very important for Intel CPU architectures, because target execution always starts at the top of the address space.

/FILE-OFFSET=hhhhhh
If FILE-OFFSET is specified, the first hhhhhh bytes of the file are skipped and not programmed to target.

Default:	/FILE-OFFSET=0
Abbreviation:	/FO=

[1] hhhhhh=number base is hex

/LENGTH=hhhhhh
The number of programmed bytes may be limited to LENGTH. If no LENGTH is specified the whole file is programmed.
Default: /LENGTH=4000000 (64 MByte)
Abbreviation: /L=

/NODUMP
In case of a verify error the contents of the flash memory is written to a file with the extension .DMP. With /NODUMP you can suppress this feature.

/ERASEALL
Erase the whole flash device. If this option isn't set, only those blocks are erased where new data should be written to.

/NOERASE
This option prevents the flash device from being erased.

/BMS /MS0 /MS1 /MS2 /MS3
This options may be used to specify one or more chip select signals to the flash memory. The used chip selects must be defined as output and inactive in the initialization file. (See chapter 1.10 “Initialization file JTAGxxx.INI”.)
Default: /BMS

/NOCS
Use this option to switch off all chip select signals. This may be necessary if the device's chip select is generated via a normal decoder instead of using the Analog Devices ADSP-21xxx chip select unit.

/NOWRSETUP
By default write cycles to the Flash EPROM are realized with three steps: 1. set address/data 2. write strobe active 3. write strobe inactive. **In most cases** it is possible to set the write strobe coincident with setting of address and data by specifying the option /NOWRSETUP. **This increases the programming speed by 50%.**

Examples:

```
JTAGxxx  /P  ROMDOS.ROM  /L=20000  /TOP
```

This example programs up to 128 Kbytes of the file ROMDOS.ROM (with i.e. 512 Kbytes) to the top of the boot flash memory.

```
JTAGxxx  /P  CE.ROM  /32BIT  /MS0
```

This example programs the file CE.ROM to the 32 Bit Flash-EPROM connected to MS0#.

2.2. Read a Flash Device to file

Usage: JTAGxxx /R filename [optionlist]

The contents of a flash device is read and written to a file.

The type of the flash device is normally detected by the software. When autodetection fails you should use the /DEVICE= option together with /8BIT or /16BIT or /32BIT to set the right flash device and configuration. The known devices are shown in chapter 1.11 “Supported flash devices”.

Options:

/DEVICE=devicename
See function /P (Chapter 2.1)

/8BIT /16BIT /32BIT
See function /P (Chapter 2.1)

/BYTE-MODE
See function /P (Chapter 2.1)

/NOMAN
See function /P (Chapter 2.1)

/DEVICE-BASE=hhhhhh[2]
See function /P (Chapter 2.1)

/OFFSET=hhhhhh
Reading of the flash memory starts at an offset of hhhhhh relative to the start address of the flash device. If the offset is negative, the offset specifies a address relative to the end of the flash device.
See also option /TOP.
Default: /OFFSET=0
Abbreviation: /O=

[2] hhhhhh=number base is hex

/TOP
If the option /TOP is used the option /OFFSET= specifies the address where reading ends (plus one) instead of the starting address.

/LENGTH=hhhhhh
The number of read bytes may be limited to LENGTH. If no LENGTH is specified the whole flash device is read (if no offset is specified).

/BMS /MS0 /MS1 /MS2 /MS3 /NOCS
See function /P (Chapter 2.1)

/NOWRSETUP
See function /P (Chapter 2.1)
Please note: In the function /R write cycles are needed to detect the type of the flash memory.

Example:

```
JTAGxxx  /R  BIOS.ABS  /L=10000  /TOP
```

This example may be used to read the upper most 64 Kbyte of the flash memory to the file BIOS.ABS.

2.3. Verify a Flash Device with file

Usage: JTAGxxx /V filename [optionlist]

The contents of a flash device is compared with the specified file. If there are differences the memory is dumped to a file with the extension DMP.

The type of flash device is normally detected by the software. When autodetect fails you should use the /DEVICE= option together with /8BIT or /16BIT or /32BIT to set the right flash device and configuration. The known devices are shown in chapter 1.11 “Supported flash devices”.

Options:

/DEVICE=devicename
See function /P (Chapter 2.1)

/8BIT /16BIT /32BIT
See function /P (Chapter 2.1)

/BYTE-MODE
See function /P (Chapter 2.1)

/NOMAN
See function /P (Chapter 2.1)

/DEVICE-BASE=hhhhhh
See function /P (Chapter 2.1)

/OFFSET=hhhhhh
See function /P (Chapter 2.1)

/TOP
See function /P (Chapter 2.1)

/FILE-OFFSET=hhhhhh
See function /P (Chapter 2.1)

/LENGTH=hhhhhh
See function /P (Chapter 2.1)

/NODUMP
See function /P (Chapter 2.1)

/BMS /MS0 /MS1 /MS2 /MS3 /NOCS
See function /P (Chapter 2.1)

/NOWRSETUP
See function /P (Chapter 2.1)
Please note: In the function /V write cycles are needed to detect the type of the flash memory.

Example:

```
JTAGxxx  /V  ROMDOS.ROM  /L=20000  /TOP
```

This example may be used to verify the upper most 128 Kbytes of the flash memory with the file ROMDOS.ROM (with i.e. 512 Kbytes).

2.4. Dump target memory

Usage: JTAGxxx /DUMP [optionlist]

A Hex-Dump of the target memory is printed on the screen, if not redirected to file or device.

Options:

/8BIT /16BIT /32BIT
Default: /8BIT

/OFFSET=hhhhhh
The memory dump starts at an offset of hhhhhh plus the device start address (see option /DEVICE-BASE=).
Default: /OFFSET=0
Abbreviation: /O=

/DEVICE-BASE=hhhhhh[3]
The device start address is used as an additional offset. This gives the function /DUMP the same behavior as function /P /V and /R.
Default: /DEVICE-BASE=0
Abbreviation: /DB=

/TOP
If the option /TOP is used the option /OFFSET= specifies the address where the dump ends (plus one) instead of the starting address

/LENGTH=hhhhhh
Default: /LENGTH=100
Abbreviation: /L=

/BMS /MS0 /MS1 /MS2 /MS3 /NOCS
See function /P (Chapter 2.1)
Default: /BMS

[3] hhhhhh=number base is hex

Example:

```
JTAGxxx  /DUMP  /BMS
```

This example makes a memory dump of the first 256 bytes of the Boot-EPROM.

2.5. Program an I²C-Device

Usage: JTAGxxx /PI2C filename [/I2CBIG] [optionlist]

The specified file is programmed to an I²C-Device (i.e. a serial EEPROM) connected to pins of the CPU. Finally a complete verify is done. If the verify fails, the contents of the I²C-Device is written to a file with the extension DMP.

Two methods to connect the I²C-Device to the CPU are supported. The first method is to use two CPU pins, one pin for clock output (I2CCLK) and one pin for serial data input and output (I2CDAT). The second method is to use one pin for clock output (I2CCLK), one for serial data input (I2CDATI) and one for serial data output (I2CDATO).

Options:

/I2CBIG
Specify this option if there is a device which needs a three byte address instead of a two byte address.
This option must be the first option after the filename.

/DEVICE-BASE=hhhhhh
This option specifies an I²C device starting address. The default values are chosen to access an serial EEPROM.

Default:	/DEVICE-BASE=5000	(if option /I2CBIG omitted)
Default:	/DEVICE-BASE=500000	(if option /I2CBIG specified)

/OFFSET=hhhhhh
The programming starts at an offset of hhhhhh relative to the start address of the I²C-Device.

Default:	/OFFSET=0
Abbreviation:	/O=

/FILE-OFFSET=hhhhhh
If FILE-OFFSET is specified, the first hhhhhh bytes of the file are skipped and not programmed to target.

Default:	/FILE-OFFSET=0
Abbreviation:	/FO=

/LENGTH=hhhhhh
The number of programmed bytes may be limited to LENGTH. If no LENGTH is specified the whole file is programmed.
Abbreviation: /L=

/NODUMP
In case of a verify error the contents of the I^{2}C-Device is written to a file with the extension .DMP. With option /NODUMP you can suppress this feature.

/I2CCLK=pin_name
Specifies the CPU pin used for serial clock output.

/I2CDAT=pin_name
Specifies the CPU pin used for serial data input and output. Pin_name must specify a bidirectional pin otherwise an error message occurs. Instead of one bidirectional pin one pin for serial data input and one for serial data output may be used. See option /I2CDATO= and /I2CDATI= .

/I2CDATO=pin_name
Specifies the CPU pin used for serial data output. Pin_name must specify a output pin otherwise an error message occurs.

/I2CDATI=pin_name
Specifies the CPU pin used for serial data input. Pin_name must specify a input pin otherwise an error message occurs.

Example:

```
JTAGxxx  /PI2C  EEPROM.CFG /I2CCLK=FLAG0 /I2CDAT=FLAG1
```

This example loads the file EEPROM.CFG to a serial EEPROM connected to the pins FLAG0 and FLAG1 of the Analog Devices ADSP-21xxx

2.6. Read an I²C-Device to file

Usage: JTAGxxx /RI2C filename [/I2CBIG] /L=hhhhhh [optionlist]

The contents of an I²C-Device (i.e. a serial EEPROM) is read and written to a file. The option /LENGTH= must be specified.

Options:

/I2CBIG
This option must be the first option after the filename.
See function /PI2C (Chapter 2.5)

/DEVICE-BASE=hhhhhh
See function /PI2C (Chapter 2.5)

/OFFSET=hhhhhh
Reading of the I²C-Device starts at an offset of hhhhhh relative to the start address of the I²C-Device.
Default: /OFFSET=0
Abbreviation: /O=

/LENGTH=hhhhhh
The number of read bytes must be specified otherwise an error message occurs.
Abbreviation: /L=

/I2CCLK=pin_name
See function /PI2C (Chapter 2.5)

/I2CDAT=pin_name
See function /PI2C (Chapter 2.5)

/I2CDATO=pin_name
See function /PI2C (Chapter 2.5)

/I2CDATI=pin_name
See function /PI2C (Chapter 2.5)

Example:

```
JTAGxxx  /RI2C  EEPROM.CFG  /I2CCLK=GP26  /I2CDAT=GP27  /L=100
```

This example reads 256 bytes from a serial EEPROM to the file EEPROM.CFG. The serial EEPROM is connected to the pins CP26 and GP27 of the Analog Devices ADSP-21xxx.

2.7. Verify an I²C-Device with file

Usage: JTAGxxx /VI2C filename [/I2CBIG] [optionlist]

The contents of an I²C-Device (i.e. a serial EEPROM) is compared with the specified file. If there are differences the contents of the I²C -Device is written to a file with the extension DMP.

Options:

/I2CBIG
This option must be the first option after the filename.
See function /PI2C (Chapter 2.5)

/DEVICE-BASE=hhhhhh
See function /PI2C (Chapter 2.5)

/OFFSET=hhhhhh
See function /PI2C (Chapter 2.5)

/FILE-OFFSET=hhhhhh
See function /PI2C (Chapter 2.5)

/LENGTH=hhhhhh
See function /PI2C (Chapter 2.5)

/NODUMP
See function /PI2C (Chapter 2.5)

/I2CCLK=pin_name
See function /PI2C (Chapter 2.5)

/I2CDAT=pin_name
See function /PI2C (Chapter 2.5)

/I2CDATO=pin_name
See function /PI2C (Chapter 2.5)

/I2CDATI=pin_name
See function /PI2C (Chapter 2.5)

Example:

```
JTAGxxx /VI2C EEPROM.CFG /I2CCLK=GP26 /I2CDAT=GP27
```

This example verifies 256 bytes from a serial EEPROM with the file EEPROM.CFG. The serial EEPROM is connected to the pins CP26 and GP27 of the Analog Devices ADSP-21xxx.

2.8. Dump an I²C-Device

Usage: JTAGxxx /DUMPI2C [/I2CBIG] [optionlist]

A Hex-Dump of an I²C-Device is printed on the screen, if not redirected to file or device.

Options:

/I2CBIG
This option must be the first option.
See function /PI2C (Chapter 2.5)

/DEVICE-BASE=hhhhhh
See function /PI2C (Chapter 2.5)

/OFFSET=hhhhhh[4]
The memory dump starts at an offset of hhhhhh.
Default: /OFFSET=0
Abbreviation: /O=

/LENGTH=hhhhhh
Default: /LENGTH=100
Abbreviation: /L=

/I2CCLK=pin_name
Specifies the CPU pin used for serial clock output.

/I2CDAT=pin_name
Specifies the CPU pin used for serial data input and output. Pin_name must specify a bidirectional pin otherwise an error message occurs. Instead of one bidirectional pin one pin for serial data input and one for serial data output may be used. See option /I2CDATO= and /I2CDATI= .

/I2CDATO=pin_name
Specifies the CPU pin used for serial data output. Pin_name must specify a output pin otherwise an error message occurs.

[4] hhhhhh=number base is hex

/I2CDATI=pin_name
Specifies the CPU pin used for serial data input. Pin_name must specify a input pin otherwise an error message occurs.

Example:

```
JTAGxxx  /DUMPI2C  /I2CCLK=FLAG0  /I2CDAT=FLAG1
```

This example makes a memory dump of the first 100h bytes of a serial EEPROM connected to the CPU.

2.9. Toggle CPU pins

Usage: JTAGxxx /BLINK /PIN=pinname [optionlist]

This command allows to test the hardware by blinking with LEDs or toggling CPU signals. Faster signals can be generated by setting the delay option to zero. This can be a very helpful feature to watch signals on an oscilloscope.

The signal on the defined pin has an duty cycle of 1/2: The level is 67% high and 33% low.

Please Note: Not every pin of the Analog Devices ADSP-21xxx may be specified as an output pin.

Options:

/PIN=pin_name
CPU pin to toggle. If the option /PIN= is not specified an error message occurs. Most pins of the list in chapter 1.10 "Initialization file JTAGxxx.INI" can be used. If you type /PIN= without any pin declaration a list of the CPU pins is displayed.

/DELAY=dddddd[5]
Time to wait to next change of signal. This option can be adjusted to get optimum signals for measures with the oscilloscope.
Default: /DELAY=10000

Example:

JTAGxxx /BLINK /PIN=FLAG3 /DELAY=0
This example toggles the FLAG3 pin very fast which can be followed by the use of an oscilloscope.

[5] dddddd=number base is decimal

2.10. Polling CPU pins

Usage: JTAGxxx /PIN? /PIN=pinname [optionlist]

This command allows to test the hardware by polling CPU signals.

Please Note: Not every pin of the Analog Devices ADSP-21xxx may be specified as an input pin.

Options:

/PIN=pin_name
CPU pin to poll. If the option /PIN= is not specified an error message occurs. Most pins of the list in chapter 1.10 “Initialization file JTAGxxx.INI” can be used. If you type /PIN= without any pin declaration a list of the CPU pins is displayed.

Example:

JTAGxxx /PIN? /PIN=RESET#
This example samples the reset pin of the Analog Devices ADSP-21xxx.

2.11. Polling CPU pins while the CPU is running

Usage: JTAGxxx /SAMPLE /PIN=pinname [optionlist]

This command is similar to the function /PIN?. But with this function any pin can be observed, independent of the pin direction. Furthermore the CPU remains in normal operation.

Options:

/PIN=pin_name
CPU pin to poll. If the option /PIN= is not specified an error message occurs. All pins of the list in chapter 1.10 “Initialization file JTAGxxx.INI” can be used. If you type /PIN= without any pin declaration a list of the CPU pins is displayed.

Example:

JTAGxxx /SAMPLE /PIN=FLAG3
This example samples the state of the port pin FLAG3 while the Analog Devices ADSP-21xxx is running.

2.12. Show status of all CPU pins while the CPU is running

Usage: JTAGxxx /SNAP [optionlist]

This function is similar to the function /SAMPLE, but displays the status of all CPU pins on the screen. The CPU remains in normal operation.

The behavior of the function /SNAP depends on the option /REP: With this option specified, the JTAG-Booster samples and displays the state of the CPU pins repetitive. Without this option the status of the pins is displayed only once.

Options:

/PAUSE
Use this option to stop the output after each displayed screen. Don't use this option together with the option /REP or if the output is redirected to a file.
Abbreviation /P

/REP
If this option is specified the status of the pins is sampled and displayed repetitive. In case of many signals the display is separated into several screens. Therefor we recommend to use a video mode with 43 or 50 lines. Use the '+' and the '-' key to switch between different screens. Any other key terminates the program.

Sample output:
This is a sample output for a Analog Devices ADSP-21065

```
| 1 BSEL          | 1 BMS#          | 1 RESET#        | 1 ADDR23
| 1 ADDR22        | 1 ADDR21        | 1 ADDR20        | 1 ADDR19
| 1 ADDR18        | 1 ADDR17        | 1 ADDR16        | 1 ADDR15
| 1 ADDR14        | 1 ADDR13        | 1 ADDR12        | 1 ADDR11
| 1 ADDR10        | 0 ADDR9         | 1 ADDR8         | 1 ADDR7
| 0 ADDR6         | 1 ADDR5         | 0 ADDR4         | 1 ADDR3
| 1 ADDR2         | 1 ADDR1         | 1 ADDR0         | 0 FLAG0
| 1 FLAG1         | 0 FLAG2         | 0 FLAG3         | 1 IRQ0#
| 1 IRQ1#         | 1 IRQ2#         | 1 RFS0          | 1 RCLK0
| 1 DR0_A         | 1 DR0_B         | 1 TFS0          | 1 TCLK0
| 0 DT0_A         | 0 DT0_B         | 0 RFS1          | 1 RCLK1
| 1 DR1_A         | 1 DR1_B         | 0 TFS1          | 1 TCLK1
| 0 DT1_A         | 0 DT1_B         | 0 PWM_EVENT1    | 0 PWM_EVENT0
| 1 BR1#          | 1 BR2#          | 0 CLKIN         | 0 SDCLK1
| 0 SDCLK0        | 1 DMAR1#        | 1 DMAR2#        | 1 HBR#
| 1 RAS#          | 1 CAS#          | 1 SDWE#         | 1 DQM
| 1 SDCKE         | 1 SDA10         | 1 DMAG1#        | 1 DMAG2#
| 1 HBG#          | 1 BMSTR         | 1 CS#           | 1 SBTS#
| 0 CLKSEL        | 1 WR#           | 1 RD#           | 0 REDY
| 1 SW#           | 1 CPA#          | 1 ACK           | 1 MS0#
| 1 MS1#          | 1 MS2#          | 1 MS3#          | 0 FLAG11
| 1 FLAG10        | 1 FLAG9         | 1 FLAG8         | 0 DATA0
| 0 DATA1         | 0 DATA2         | 0 DATA3         | 0 DATA4
| 1 DATA5         | 0 DATA6         | 0 DATA7         | 1 DATA8
| 1 DATA9         | 0 DATA10        | 0 DATA11        | 0 DATA12
| 0 DATA13        | 0 DATA14        | 0 DATA15        | 0 DATA16
| 0 DATA17        | 0 DATA18        | 0 DATA19        | 0 DATA20
| 0 DATA21        | 1 DATA22        | 0 DATA23        | 0 DATA24
| 0 DATA25        | 0 DATA26        | 0 DATA27        | 0 DATA28
| 0 DATA29        | 0 DATA30        | 0 DATA31        | 1 FLAG7
| 1 FLAG6         | 1 FLAG5         | 1 FLAG4         | 0 ID1
| 0 ID0           | 0 EMU#          | 0 SPARE1        | 0 SPARE2
| 0 SPARE3        | 0 SPARE4        | 0 SPARE5        | 0 SPARE6
```

3. Implementation Information

This chapter summarizes some information about the implementation of the JTAG-Booster and describes some restrictions.

- The JTAG-Booster currently uses the EXTEST function of the JTAG-Interface to perform Flash programming.

- The JTAG interface of the Analog Devices ADSP-21xxx does not support an ICODE register. Therefor the analysis of the JTAG chain is not really possible. But we try to avoid unpredictable results by comparing the length of the boundary scan chain with the estimated length.

- Refer to the following table for connecting Flash-EPROMs to the Analog Devices ADSP-21xxx:

ADSP-21060 signal	**8 Bit Flash**	**16 Bit Flash**	**32 Bit Flash**
BMS# **1)**	CS#	-	-
MS0# or MS1# or MS2# or MS3#	CS#	CS#	CS#
RD#	OE#	OE#	OE#
WR#	WE#	WE#	WE#
DATA16..23	D0..7	-	-
DATA16..31	-	D0..15	-
DATA16..47	-	-	D0..31

ADSP-21065 signal	**8 Bit Flash**	**16 Bit Flash**	**32 Bit Flash**
BMS# **1)**	CS#	-	-
MS0# or MS1# or MS2# or MS3#	CS#	CS#	CS#
RD#	OE#	OE#	OE#
WR#	WE#	WE#	WE#
DATA0..7	D0..7	-	-
DATA0..15	-	D0..15	-
DATA0..31	-	-	D0..31

ADSP-21161 signal	**8 Bit Flash**	**16 Bit Flash**	**32 Bit Flash**
BMS# **1)**	CS#	-	-
MS0# or MS1# or MS2# or MS3#	CS#	CS#	CS#
RD#	OE#	OE#	OE#
WR#	WE#	WE#	WE#
DATA16..23	D0..7	-	-
DATA16..31	-	D0..15	-
DATA16..47	-	-	D0..31

1) Only a 8 bit Boot-Flash is supported by the Analog Devices ADSP-21xxx

4. Converter Program HEX2BIN.EXE

Since the JTAG-Booster software is not able to handle Intel-HEX or Motorola S-Record files, an separate converter tool is delivered with this product package.

Five types of HEX formats can be converted to BIN file:

- I : INTEL HEX format (BYTE oriented)
- D : Digital Research
- M : MOTOROLA S HEX format (BYTE oriented)
- T : TEKTRONICS HEX format (BYTE oriented)
- H : Intel HEX-32

Maximum conversion size is 256 kBytes. A 4th parameter for starting address can be specified to skip out the leading garbage and you will maintain a small size of output binary file.

If you start the HEX2BIN without any additional parameter all necessary parameters will be asked for in a prompt mode:

```
HEX2BIN
Input HEX file name: MYAPP.H86
Output BIN file name[MYAPP.BIN]:
HEX file format
<I>ntel  /<M>otorola  /<D>igital Research  /<T>ektronics  /[H] Intel HEX-32[I] : H
Input CODE segment start address[0000000]: 10000
Input CODE segment end address[FFFFFFF]:
Unused bytes will be <1>00  <2>FF   [1]  : 2
```

Instead of using the prompt mode, you can directly specify all necessary parameters in the command line. This is essential for making batch files:

```
HEX2BIN  MYAPP.H86  MYAPP.BIN  H  0010000  FFFFFFF  2
```

It is very important to fill unused bytes with 0xFF, because this are simply skipped by the JTAG-Boosters software and so it speeds up the programming performance.

Please Note: **"CODE segment start address"** is interpreted as a Intel x86 architecture segment address: You have to specify a start address of 10000 to start the conversion at 1 MByte.

This converter is a relatively old DOS tool and therefor it has problems with non DOS compliant file and directory names. Avoid names with spaces, limit names to eight characters. Otherwise the converter does not convert the input file, without any error message!!

5. Support for Windows NT and Windows 2000

A configured run time version of the "Kithara DOS Enabler, Version 5.1" is used to give support for some of our DOS based tools (like the JTAG-Booster) for Windows NT and Windows 2000. After installation of the "DOS Enabler" the accesses to the LPT or COM ports are allowed for the all programs listed in file Readme_WinNT.txt

Note: Accesses to the ports are only allowed for the programs listed in file Readme_WinNT.txt. If you rename one of our tools, the DOS Enabler does not work.

5.1. Installation on a clean system

If you have a clean system without having installed a previous version of the "Kithara Tool Center", this tool is really simple to install. Extract the ZIP file to a new folder and start KSETUP.EXE. Everything is done within a few seconds. No additional input is needed. Now reboot your PC.

5.2. Installation with already installed a previous version of Kithara

Important!! If you have already installed an older WinNT support, you have to deinstall it completely!!!

- Start kcenter
- Select Register "Einstellungen" (=Settings)
  and deactivate "VDD benutzen"
  and "speziellen seriellen Treiber benutzen".
- Stop Kernel
- exit the kcenter program
- Now you can deinstall the Kithara Package with:
  Settings - Control Panel.
  All unused parts must be removed.
- Reboot your PC
- Now you can install the Kithara 5.xx as described above.

5.3. De-Installation version 5.xx:

For deinstallation of the runtime version of the "Kithara DOS-Enabler Version 5.x":

- use: Settings - Control-Panel - Add/Remove Programs
  and remove the
  "WinNT support for JTAG-Booster and FLASH166"

- Reboot your PC

AMEYA360

Components Supply Platform

Authorized Distribution Brand：

ROHM SEMICONDUCTOR

Sunlord EXPERT IN PASSIVE PARTS

SSM SUSUMU Thin Film Specialist and Innovator

AVERLOGIC

NXP

Elprotronic

2Pai Semi

Ambarella

CanaanTek

Firstohm

GigaDevice

威兆半导体 Vanguard Semiconductor

Website：

Welcome to visit www.ameya360.com

Contact Us：

➢ Address：

401 Building No.5, JiuGe Business Center, Lane 2301, Yishan Rd
Minhang District, Shanghai , China

➢ Sales：

Direct	+86 (21) 6401-6692
Email	amall@ameya360.com
QQ	800077892
Skype	ameyasales1 ameyasales2

➢ Customer Service：

Email	service@ameya360.com

➢ Partnership：

Tel	+86 (21) 64016692-8333
Email	mkt@ameya360.com