Trenz Electronic GmbH - TE0300-01IBMLP Datasheet

Welcome to E-XFL.COM

Understanding **Embedded - Microcontroller, Microprocessor, FPGA Modules**

Embedded - Microcontroller, Microprocessor, and FPGA Modules are fundamental components in modern electronic systems, offering a wide range of functionalities and capabilities. Microcontrollers are compact integrated circuits designed to execute specific control tasks within an embedded system. They typically include a processor, memory, and input/output peripherals on a single chip. Microprocessors, on the other hand, are more powerful processing units used in complex computing tasks, often requiring external memory and peripherals. FPGAs (Field Programmable Gate Arrays) are highly flexible devices that can be configured by the user to perform specific logic functions, making them invaluable in applications requiring customization and adaptability.

Applications of **Embedded - Microcontroller,**

Details	
Product Status	Not For New Designs
Module/Board Type	FPGA, USB Core
Core Processor	Spartan-3E
Co-Processor	Cypress FX2 USB 2.0
Speed	100MHz
Flash Size	4MB
RAM Size	64MB
Connector Type	B2B
Size / Dimension	1.6" x 1.9" (40.5mm x 47.5mm)
Operating Temperature	0°C ~ 70°C
Purchase URL	https://www.e-xfl.com/product-detail/trenz-electronic/te0300-01ibmlp

Table of Contents

Applications

- IP (intellectual property) development
- Digital signal processing
- Image processing
- Cryptography
- Industrial control
- Low-power design
- General-purpose prototyping platform

Description

The FPGA industrial micromodule integrates a leading edge Xilinx Spartan-3E FPGA, an USB 2.0 microcontroller, configuration Flash, DDR SDRAM and power supplies on a tiny footprint. A large number of configurable I/Os are provided via B2B mini-connectors.

The module is intended to be used as an OEM board, or to be combined with our carrier boards. It is a powerful system widely used for educational and research activities.

Boards with other configurations, larger FPGA's or equipped with industrial temperature grade parts are available on request.

Software for SPI flash programming over USB as well as reference designs for high speed data transfer over USB are included.

Physical Features

Board Dimensions

The module measures 40.50 mm by 47.50 mm.

Board-to-Board Connectors

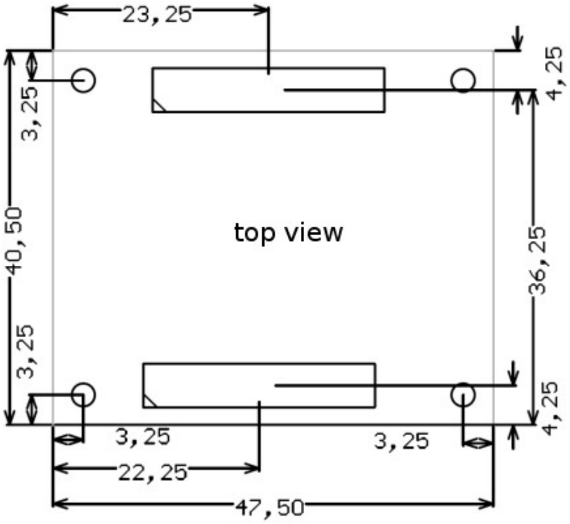


Figure 3: module dimensions in mm (top view).

The module has two B2B (board-to-board) receptacle connectors (J4 and J5) for a total of 160 contacts (Figure 5).

Figure 5: micromodule receptacle.

The ordering numbers of the connector receptacles are given in Table 1.

supplier	header
Digikey	H11113CT-ND H11113TR-ND H11113DKR-ND
Hirose	DF17(3.0)-80DS-0.5V(57)
Trenz Electronic	22684

Table 1: equivalent part numbers of the receptacle connectors J4 and J5.

The on-board receptacles mate with their corresponding headers on the carrier board (Figure 6).

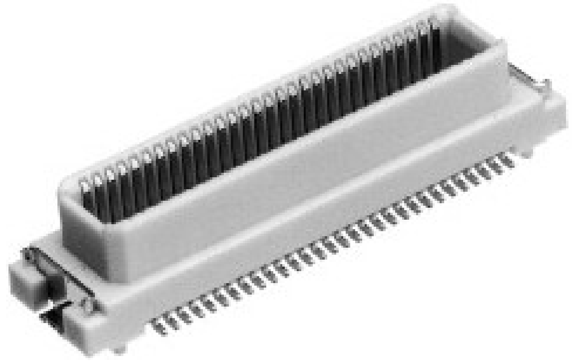


Figure 6: mating header.

The ordering number of the headers is given in Table 2.

supplier	header
Digikey	H11148DKR-ND H11148TR-ND H11148CT-ND
Hirose	DF17(4.0)-80DP-0.5V(57)
Trenz Electronic	22938

Table 2: equivalent part numbers of the mating connectors.

Figure 7 shows the definition of stacking height featured by the combination of the TE0300 receptacle with its corresponding header.

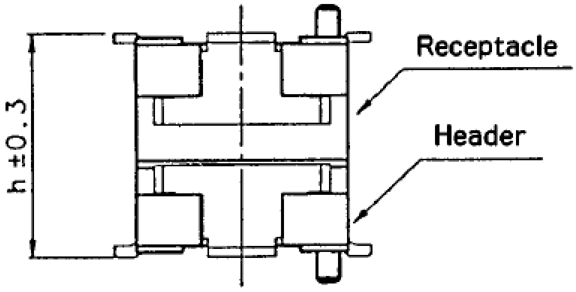


Figure 7: stacking height (h).

The stacking height of the TE0300 B2B connectors is 7 (seven) mm. The stacking height does not include the solder paste thickness.

USB Connector

The micromodule uses a mini-USB (B type) receptacle connector.

Figure 8: mini-USB (B type) receptacle connector.

Power Supply

The module can be powered by the B2B connector or the USB connector. If both power supplies are available, the B2B connector power supply takes precedence, disabling the USB power supply automatically.

B2B Connector Power Supply

The B2B connector power supply requires a single nominal 5 V DC power supply. The power is usually supplied to the module through the 5 V contacts (5Vb2b) of the B2B connectors J5 (see Appendix). The recommended minimum supply voltage is

4 V. The maximum supply voltage is 5.5 V. The recommended maximum continuous supply current is 1.5 A.

USB Power Supply

The module is powered by the USB connector if the following conditions are met:

- the module is equipped with an USB connector,
- the module is connected to a USB bus,
- no power supply is provided by the B2B connectors.

In this case, other components (e.g. extension or carrier boards) may also be powered by the corresponding 5 Volt line (5V) of the B2B connector J5.

On-board Power Rails

Three on-board voltage regulators provide the following power supply rails needed by the components on the micromodule:

- 1.2 V, 1 A max
- 2.5 V, 1 A max
- 3.3 V, 1 A max

The power rails are available for the FPGA and can be shared with a baseboard by the **corresponding** lines of the B2B connectors J4 and J5. Please note that the **power consumption of the FPGA is highly dependent on the design** actually loaded. So please use a tool like Xilinx Xpower to determine the expected power consumption.

Even if the provided voltages of the module are not used on the baseboard, it is recommended to bypass them to ground with 10 nF - 100 nF capacitors.

I/O Banks Power Supply

The Spartan-3E architecture organizes I/Os into four I/O banks (see Table 3).

Bank	Supply Voltage (V)	Min (V)	Max (V)
B0	VccIO	1.2	3.3
B1	2,5	-	-
B2	3,3	-	-
B3	3,3	-	-

Table 3: I/O banks power supply.

Voltage for banks B1, B2 and B3 is fixed respectively to 2,5 V, 3,3 V and 3,3 V.

Voltage VccIO for bank B0 shall span from 1.2 V to 3.3 V. VccIO can be supplied either externally or internally to the micromodule.

Warning! Spartan-3 I/Os are not 5 V tolerant. Applying more than the recommended operating voltages at any pin, results in a damaged FPGA (see Xilinx Answer AR#19146).

Externally Supplied VccIO

VccIO can be externally supplied over the B2B connector J4. If bank B0 is not used, then VccIO can be left open.

Internally Supplied VccIO

If VccIO is **not** externally supplied, it can be internally supplied by **one** of the internal power rails of 2.5 V and 3.3 V. This is possible by short-circuiting **one** of the two pad pairs placed on the right of connector J4 at the top right corner of the bottom side of the micromodule.

Figure 9 shows how to short-circuit VccIO to internal 3.3 V power rail.

Figure 10 shows how to short-circuit VccIO to internal 2.5 V power rail.

Signal	FPGA pin	FPGA ball
PB	IP (bank 2)	V16

Table 6: user button signal details.

The input is normally low. The input is pulled up when pressed.

Configuration Switches

The micromodule hosts 4 DIP switches on the top side: S1; S2, S3 and S4.

For customers requesting a sufficient amount of units, the micromodules can be manufactured replacing the switches by fixed connections.

DIP Switch S1

S1 enables / disables the communication between the Cypress EZ-USB FX2 microcontroller and the I2C CMOS Serial EEPROM.

Turn S1 off when programming the USB EEPROM (storing the USB vendor ID and device ID). This will force the USB microcontroller to provide its default vendor ID and device ID.

S1	position
EEPROM (on)*	EEPROM <u>en</u>abled
Off (off)	EEPROM <u>dis</u>abled

Table 7: S1 (* default: EEPROM).

For further information, please read paragraph "Software Configuration".

DIP Switch S2

S2 enables / disables the reset line. The reset line (available also on 2 contacts of the B2B connector) resets the USB microcontroller and the FPGA.

S2 has to be turned off (*Reset*) if the user wants to program the SPI Flash memory in direct mode. For programming the SPI Flash memory in indirect mode over JTAG, S2 has to be turned on (*Run*).

S2	position
Run (on)*	system running
Reset (off)	system reset

Table 8: S2 (*default: *Run*).

For further information, please read paragraph "Software Configuration".

DIP Switch S3

S3 conditionally / unconditionally enables the 1.2 V and 2.5 V power rails.

When S3 is turned on, the 1.2 V and 2.5 V power rails are controlled by the USB microcontroller. At start-up, the USB microcontroller switches off the 1.2 V and 2.5 V power rails and starts up the module in low-power mode. After enumeration, the USB microcontroller firmware switches the 1.2 V and 2.5 V power rails on, if enough current is available from the USB bus.

When S3 is turned off, the 1.2 V and 2.5 V power rails are always enabled.

S3	position
FX2 PON (on)*	rails controlled by FX2
PON (off)	rails always enabled

Table 9: S3 (* default: FX2 PON).

Warning! When S3 is turned on (*FX2 PON*), make sure that no signals are applied to the input pins when power-rails are disabled by the USB microcontroller.

The 3.3 V power-rail though is out of the control of the USB-microcontroller and is supplied down-converting the 5 V power supply provided by either the USB-bus or the B2B receptacle connector. In this case, signals that are applied to the 3.3 V I/O

TE0300 modules with the following part numbers

- TE0300-00
- TE0300-00-4I5C
- TE0300-00B
- TE0300-01
- TE0300-01B
- TE0300-01BLP

are assembled with

Qimonda HYB25DC512160CF-6

512Mb DDR SDRAM components. When developing DDR SDRAM designs with Xilinx tools, you should select the following product type:

HYB25D512160BF-6.

SPI Flash

TE0300 modules have a

STMicroelectronics M25P32

32-Mbit, low voltage, serial Flash memory with 75 MHz SPI bus interface for configuration and operating storage accessible through USB or SPI.

Serial EEPROM

TE0300 modules have a

Micron Technology 24LC128

128K I2C CMOS Serial EEPROM. It used for EZ-USB FX2 firmware, vendor ID and device ID storage. EEPROM accessible through the EZ-USB FX2 microcontroller.

Module Configuration

This section describes how to configure the TE0300 module and access some of its resources.

The JTAG interface allows a fast, frequent but volatile configuration of the TE0300 module. However, only through the JTAG interface it is possible to develop and debug with Xilinx tools (e.g. Xilinx ChipScope, Xilinx Microprocessor Debugger.

The SPI interface allows a fast, frequent and non-volatile configuration of the TE0300 module.

Configuration of the TE0300 module through a USB host is recommended for occasional, non-volatile on-site operations such as firmware upgrade.

System Requirements

TE0300 modules can be configured through a host computer with the following system requirements:

- Operating system: Microsoft Windows 2000, Microsoft Windows XP, Microsoft Vista;
- Xilinx ISE 10.1 or later for indirect SPI in-system programming (see Xilinx Answer AR #25377);
- Xilinx EDK for some reference designs;
- Interface: USB host;
- JTAG/SPI USB cable with flying leads.

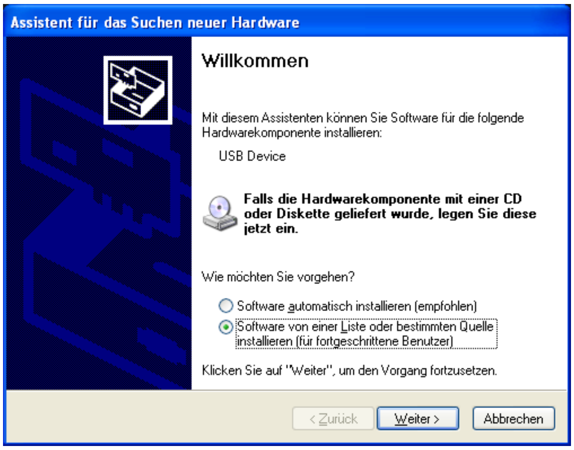
EZ-USB FX2 Microcontroller Firmware

If the EEPROM has never been programmed before (virgin board), S1 can be switched to **EEPROM**. The USB microcontroller will detect an empty EEPROM and will provide its default vendor ID and device ID to the USB host.

DIP switch	on (left)	off (right)
S1	EEPROM	-
S2	Run	-
S3	X	X
S4	X	X

If the EEPROM has been programmed before (EEPROM not empty), S1 must be switched to **Off**. The USB microcontroller will detect a missing EEPROM and will provide its default vendor ID and device ID to the USB host.

DIP switch	on (left)	off (right)
S1	-	Off
S2	Run	-
S3	X	X
S4	X	X



Willkommen

Mit diesem Assistenten können Sie Software für die folgende Hardwarekomponente installieren:

USB Device

Falls die Hardwarekomponente mit einer CD oder Diskette geliefert wurde, legen Sie diese jetzt ein.

Wie möchten Sie vorgehen?

○ Software automatisch installieren (empfohlen)

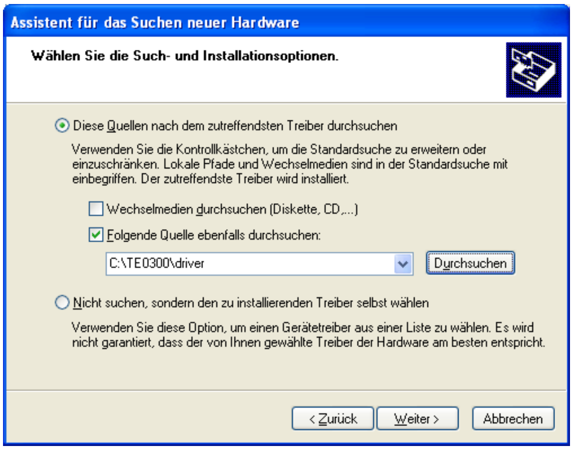
● Software von einer Liste oder bestimmten Quelle installieren (für fortgeschrittene Benutzer)

Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

Generic USB Microcontroller Driver installation

If the USB microcontroller (Cypress EZ-ESB FX2) driver is not installed on the host computer, then the easiest way to do it is the following:

- disconnect the micromodule or leave the micromodule unconnected;
- configure the micromodule such that the USB microcontroller will provide its default vendor ID and device ID to the USB host (i.e. S1 = OFF -- see paragraph "EZ-USB FX2 Microcontroller Firmware");
- connect the micromodule to the host computer through the USB interface;
- wait until the operating system detects new hardware and starts the hardware assistant;
- if S1 is not already switched to EEPROM, do it now;
- answer the hardware assistant questions as shown in the following example.



Wählen Sie die Such- und Installationsoptionen.

● Diese Quellen nach dem zutreffendsten Treiber durchsuchen

Verwenden Sie die Kontrollkästchen, um die Standardsuche zu erweitern oder einzuschränken. Lokale Pfade und Wechselmedien sind in der Standardsuche mit einbegriffen. Der zutreffendste Treiber wird installiert.

☐ Wechselmedien durchsuchen (Diskette, CD,...)

☑ Folgende Quelle ebenfalls durchsuchen:

C:\TE0300\driver

○ Nicht suchen, sondern den zu installierenden Treiber selbst wählen

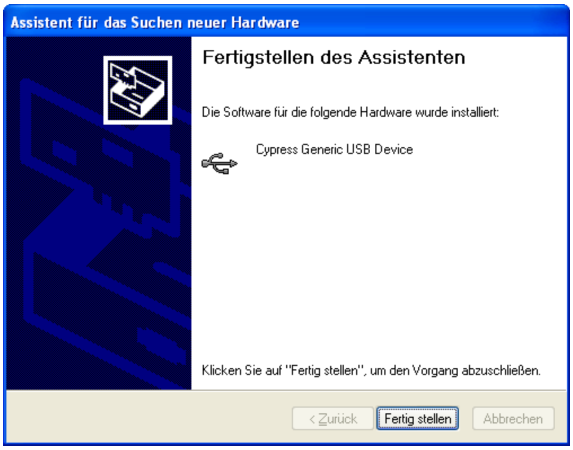
Verwenden Sie diese Option, um einen Gerätetreiber aus einer Liste zu wählen. Es wird nicht garantiert, dass der von Ihnen gewählte Treiber der Hardware am besten entspricht.

Die Software, die für diese Hardware installiert wird:

Cypress Generic USB Device

hat den Windows-Logo-Test nicht bestanden, der die Kompatibilität mit Windows XP überprüft. (Warum ist dieser Test wichtig?)

Das Fortsetzen der Installation dieser Software kann die korrekte Funktion des Systems direkt oder in Zukunft beeinträchtigen. Microsoft empfiehlt strengstens, die Installation jetzt abzubrechen und sich mit dem Hardwarehersteller für Software, die den Windows-Logo-Test bestanden hat, in Verbindung zu setzen.



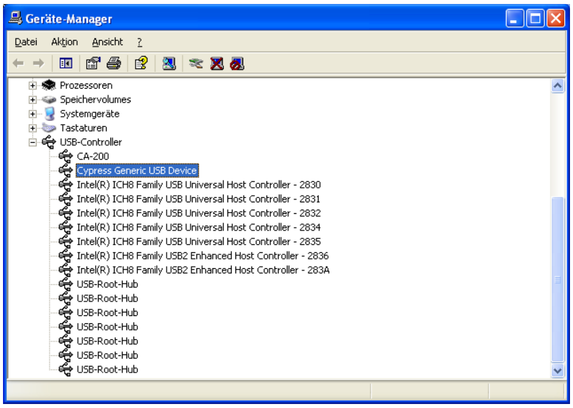
Fertigstellen des Assistenten

Die Software für die folgende Hardware wurde installiert:

Cypress Generic USB Device

Klicken Sie auf "Fertig stellen", um den Vorgang abzuschließen.

Check that in the "Device Manager" under "USB-Controller" the "Cypress Generic USB Device" has been added.



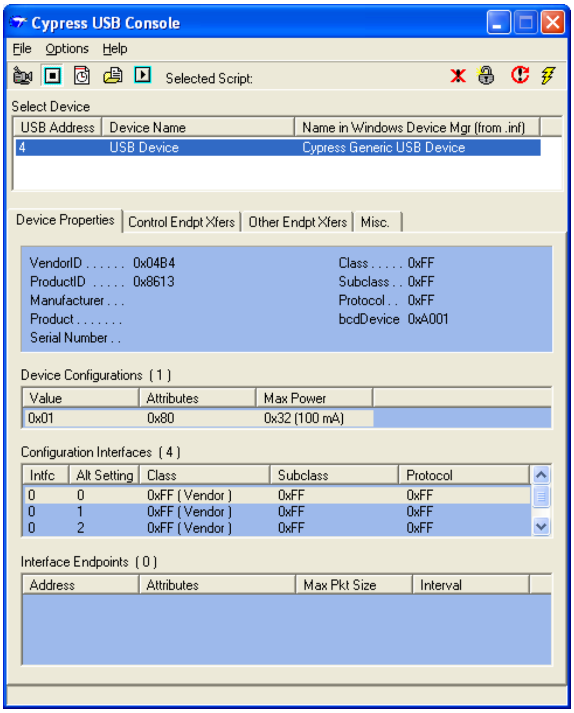
- Prozessoren
- Speichervolumes
- Systemgeräte
- Tastaturen
- USB-Controller
  - CA-200
  - Cypress Generic USB Device
  - Intel(R) ICH8 Family USB Universal Host Controller - 2830
  - Intel(R) ICH8 Family USB Universal Host Controller - 2831
  - Intel(R) ICH8 Family USB Universal Host Controller - 2832
  - Intel(R) ICH8 Family USB Universal Host Controller - 2834
  - Intel(R) ICH8 Family USB Universal Host Controller - 2835
  - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836
  - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 283A
  - USB-Root-Hub
  - USB-Root-Hub
  - USB-Root-Hub
  - USB-Root-Hub
  - USB-Root-Hub
  - USB-Root-Hub
  - USB-Root-Hub

Now the USB microcontroller can be accessed from the host computer through dedicated software.

EZ-USB FX2 EEPROM Programming

First of all, check that S1 is actually switched to EEPROM.

The USB EEPROM can be programmed by opening the dedicated software "Cypress USB Console" (double click the "CyConsole.exe" file in the "1st_program\CyConsole" folder).



Select Device

USB Address	Device Name	Name in Windows Device Mgr (from .inf)
4	USB Device	Cypress Generic USB Device

VendorID 0x04B4
ProductID 0x8613
Manufacturer . . .
Product . . . .
Serial Number . .
Class 0xFF
Subclass . . 0xFF
Protocol . . 0xFF
bcdDevice 0xA001

Device Configurations (1)

Value	Attributes	Max Power
0x01	0x80	0x32 (100 mA)

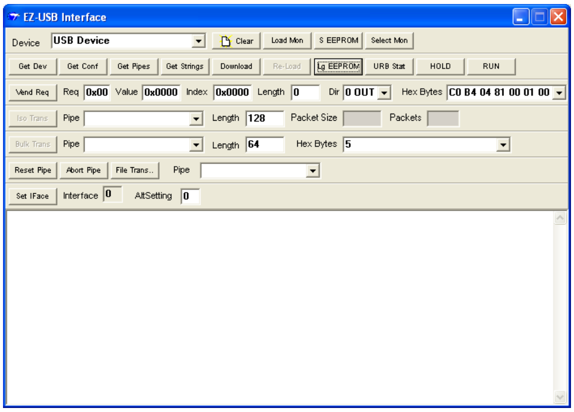
Configuration Interfaces (4)

Intfc	Alt Setting	Class	Subclass	Protocol
0	0	0xFF (Vendor)	0xFF	0xFF
0	1	0xFF (Vendor)	0xFF	0xFF
0	2	0xFF (Vendor)	0xFF	0xFF

Interface Endpoints (0)

Address	Attributes	Max Pkt Size	Interval

Click "Options > EZ-USB Interface" to Open EZ-USB Interface window.



Device USB Device

Req 0x00 Value 0x0000 Index 0x0000 Length 0 Dir 0 OUT Hex Bytes C0 B4 04 81 00 01 00

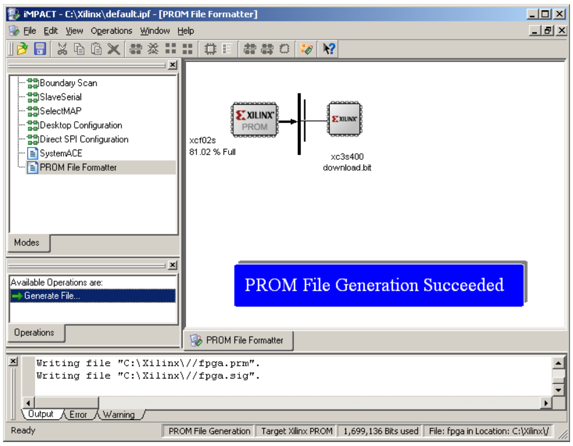
Pipe Length 128 Packet Size Packets

Pipe Length 64 Hex Bytes 5

Pipe

Interface 0 AltSetting 0

"S EEPROM" button refers to the small EEPROM (256 bytes) whereas the "Lg EEPROM" refers to the large EEPROM (64 KB). Press the "Lg_EEPROM" button, select the "USB.iic" file and press the "Open" button to start writing to EEPROM.

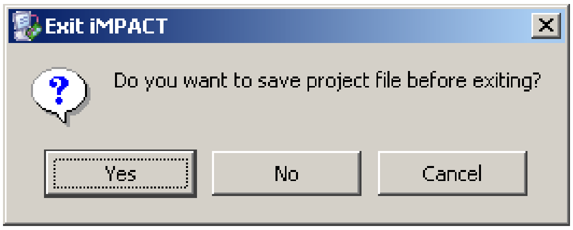


PROM File Generation Succeeded

```
Writing file "C:\Xilinx\//fpga.prm".
Writing file "C:\Xilinx\//fpga.sig".
```

PROM File Generation | Target Xilinx PROM | 1,699,136 Bits used | File: fpga in Location: C:\Xilinx\/

Don't forget to save your project for further use



Do you want to save project file before exiting?

Once you have got your *fpga.in* file, you can proceed and generate your FWU file. The FWU file is a ZIP file containing 3 files:

- *Bootload.ini* – booting settings
- *fpga.bin* – FPGA programming file
- *usb.bin* – FX2 firmware

To create your FWU file, you need to

- replace the existing *USBFWUTool\FWUs\fpga.bin* with the latest *fpga.bin* (*Bootload.ini* and *usb.bin* are always unchanged)
- zip the 3 files
- change the "*zip" file extension to "fwu"*
- upload the file as explained in the next section (Micromodule Configuration).

Warning! file and path names are given and must NOT be changed!

Micromodule Configuration

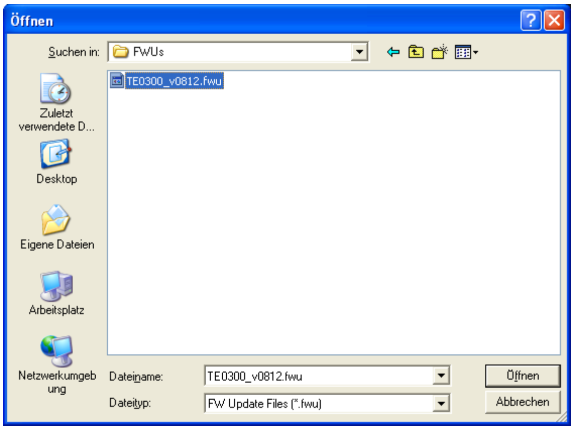
The micromodule can now be programmed with its dedicated firmware upload tool. Turn S1, S2, S3 and S4 on. Open the dedicated firmware upgrade tool "USB Firmware Upgrade Tool" (double click the "USBFirmwareUpgradeTool.exe" file in the "USBFWUTool" folder).

Device: USB Device

File name:

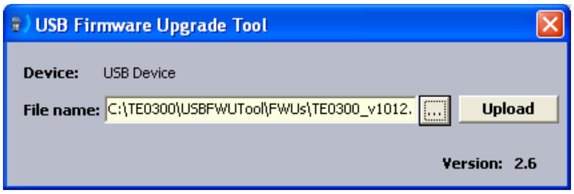
Version: 2.6

Press the "..." button corresponding to the "File name" field and select for instance the sample firmware upload file "TE0300_v1012.fwu" in the "USBFWUTool\FWUs" folder.



Dateiname: TE0300_v0812.fwu

Dateityp: FW Update Files (*.fwu)

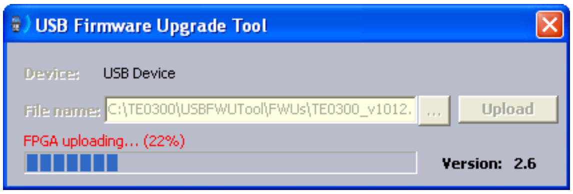


Device: USB Device

File name: C:\TE0300\USBFWUTool\FWUs\TE0300_v1012.

Version: 2.6

Press the “Upload” button to upload the micromodule firmware and check the “FPGA uploading…” progress bar.



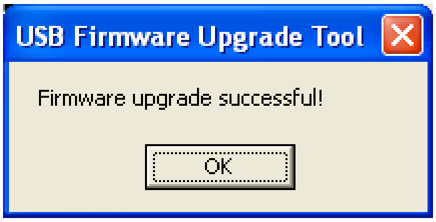
Device: USB Device

File name: C:\TE0300\USBFWUTool\FWUs\TE0300_v1012.

FPGA uploading... (22%)

Version: 2.6

After successful completion of the firmware upload procedure, the following message should pop up.



Firmware upgrade successful!

Reboot the micromodule with the new firmware by disconnecting and reconnecting the USB cable. You may want to test the sample application “TE0300_API_Example.exe” in the “TE0300_API_Example\Debug” folder.

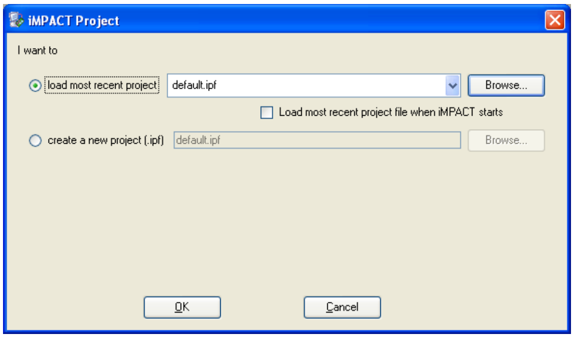
To generate your own firmware upload file, please read the document “Generating_FWU_file.doc” in the “USB-FWUTool” folder.

SPI Direct In-System Programming (ISP)

Make sure S2 is switched to “Reset” (off) during programming.

Connect the host computer to the micromodule through both the SPI flying leads cable and the USB cable.

Start Xilinx ISE iMPACT. The following example shows the case of iMPACT 9.2. If the “iMPACT Project” window pops up, press the “Cancel” button.



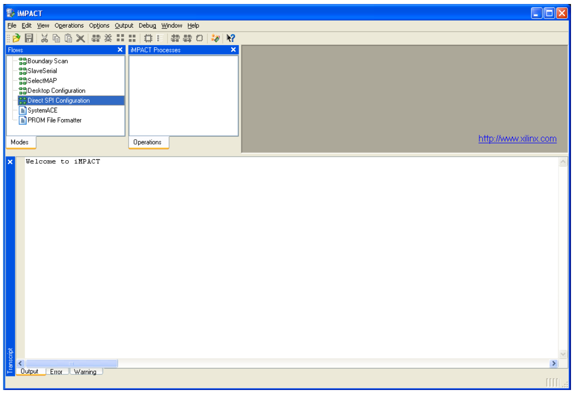
I want to

● load most recent project: default.ipf

☐ Load most recent project file when iMPACT starts

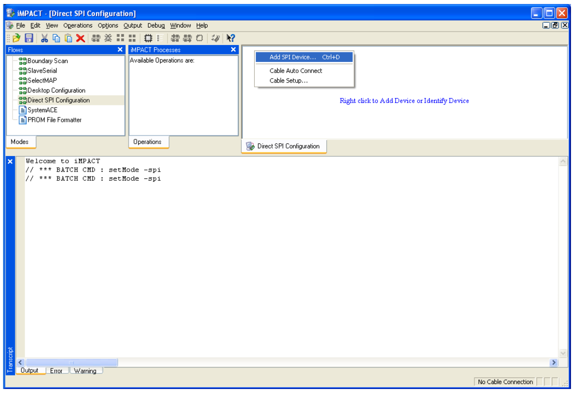
○ create a new project (.ipf): default.ipf

Double click the “Direct SPI Configuration” option in the “Modes” panel.



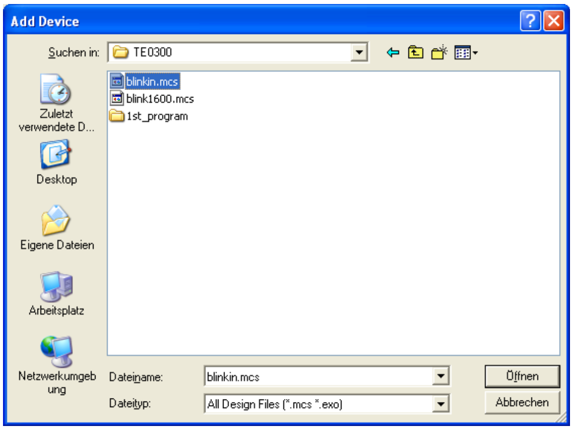
```
Welcome to iMPACT
```

Right click the “Direct SPI Configuration” panel to add a device and select “Add SPI Device”.



```
Welcome to iMPACT
// *** BATCH CMD : setMode -spi
// *** BATCH CMD : setMode -spi
```

You can now select the file corresponding to your device. In the following example, we will show how to select the micromodule reference device “blinking.mcs” in the “TE0300” folder.



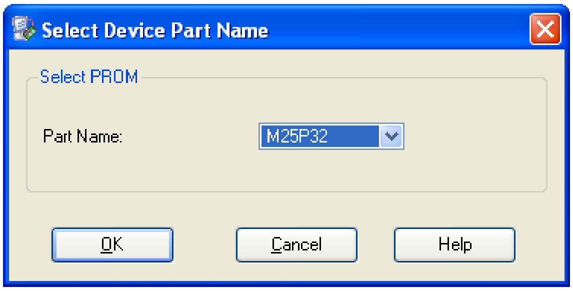
Suchen in: TE0300

- blinkin.mcs
- blink1600.mcs
- 1st_program

Dateiname: blinkin.mcs

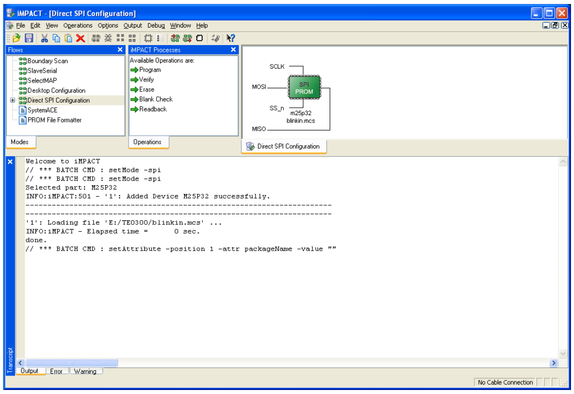
Dateityp: All Design Files (*.mcs *.exo)

Select the part name corresponding to the SPI flash present on the module (STMicroelectronics M25P32, a 32 Mbit (4M x 8) Serial Flash memory).



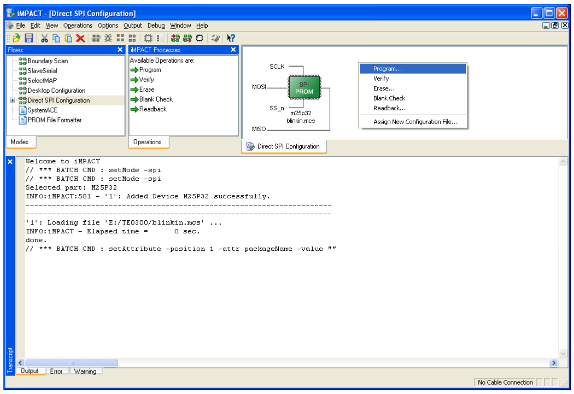
Part Name: M25P32

iMPACT should now look like this.



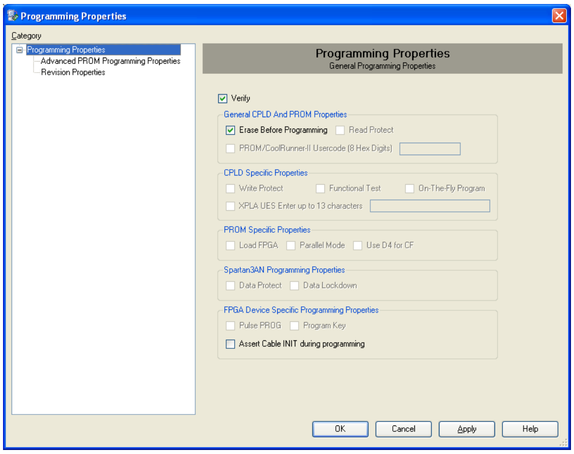
```
Welcome to iMPACT
// *** BATCH CMD : setMode -spi
// *** BATCH CMD : setMode -spi
Selected part: M25P32
INFO:iMPACT:501 - '1': Added Device M25P32 successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
'1': Loading file 'E:/TE0300/blinkin.mcs' ...
INFO:iMPACT - Elapsed time =      0 sec.
done.
// *** BATCH CMD : setAttribute -position 1 -attr packageName -value ""
```

Right click the SPI PROM device and select the “Program” operation.



```
Welcome to iMPACT
// *** BATCH CMD : setMode -spi
// *** BATCH CMD : setMode -spi
Selected part: M25P32
INFO:iMPACT:501 - '1': Added Device M25P32 successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
'1': Loading file 'E:/TE0300/blinkin.mcs' ...
INFO:iMPACT - Elapsed time =      0 sec.
done.
// *** BATCH CMD : setAttribute -position 1 -attr packageName -value ""
```

In the “Programming Properties” window, just leave the default settings and press the “OK” button.



☑ Verify

General CPLD And PROM Properties

☑ Erase Before Programming ☐ Read Protect

☐ PROM/CoolRunner-II Usercode (8 Hex Digits)

CPLD Specific Properties

☐ Write Protect ☐ Functional Test ☐ On-The-Fly Program

☐ XPLA UES Enter up to 13 characters

PROM Specific Properties

☐ Load FPGA ☐ Parallel Mode ☐ Use D4 for CF

Spartan3AN Programming Properties

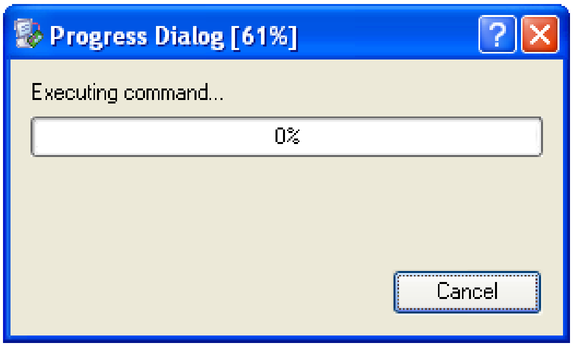
☐ Data Protect ☐ Data Lockdown

FPGA Device Specific Programming Properties

☐ Pulse PROG ☐ Program Key

☐ Assert Cable INIT during programming

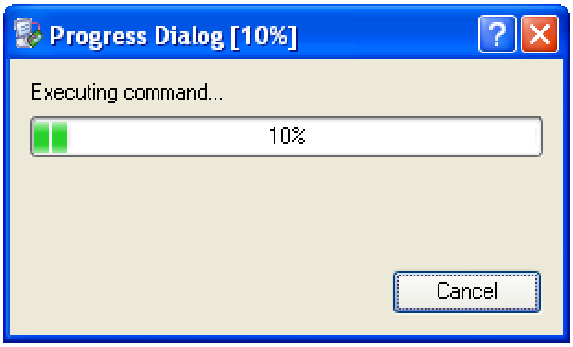
iMPACT will first erase the memory (notice the mismatch between the two progress indicators)


Progress Dialog [61%]

Executing command...

0%

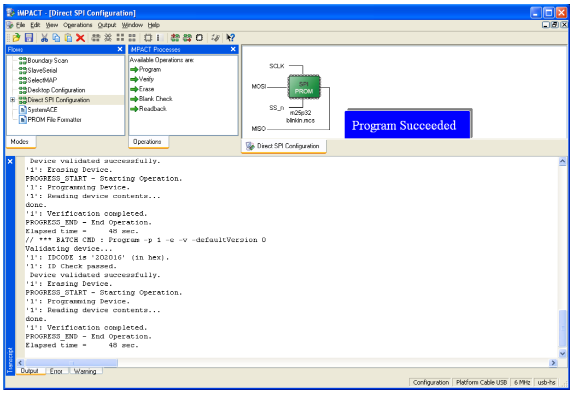
and then write it (notice the match between the two progress indicators).


Progress Dialog [10%]

Executing command...

10%

After successful programming, you should read the message "Program Succeeded" popping up for a few seconds in the "Direct SPI Configuration" panel.



Available Operations are:

Program Succeeded

```
 Device validated successfully.
'1': Erasing Device.
PROGRESS_START - Starting Operation.
'1': Programming Device.
'1': Reading device contents...
done.
'1': Verification completed.
PROGRESS_END - End Operation.
Elapsed time =     48 sec.
// *** BATCH CMD : Program -p 1 -e -v -defaultVersion 0
Validating device...
'1': IDCODE is '202016' (in hex).
'1': ID Check passed.
 Device validated successfully.
'1': Erasing Device.
PROGRESS_START - Starting Operation.
'1': Programming Device.
'1': Reading device contents...
done.
'1': Verification completed.
PROGRESS_END - End Operation.
Elapsed time =     48 sec.
```

Switch S2 back to the "Run" position. In case you uploaded the test design, you should see the on-board led blinking at 0.5 Hz.

For further information about direct (pure SPI) in-system programming of SPI Flash memories, please see Xilinx Application Note XAPP951 "Configuring Xilinx FPGAs with SPI Serial Flash".

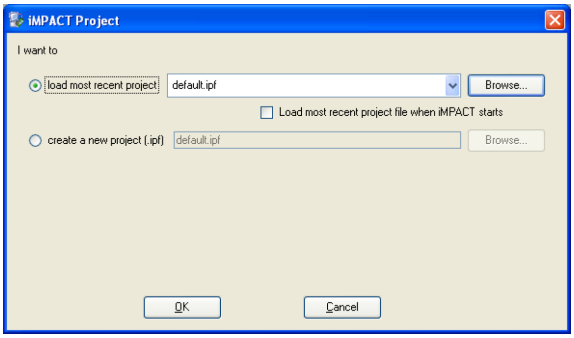
SPI Indirect In-System Programming (ISP)

Check the configuration switches against the following table:

DIP switch	on (left)	off (right)
S1	X	X
S2	Run	
S3	-	PON
S4	X	X

Connect the host computer to the micro-module through both the SPI flying leads cable and the USB cable.

Start Xilinx ISE iMPACT. The following example shows the case of iMPACT 10.1. If the "iMPACT Project" window pops up, press the "Cancel" button.


iMPACT Project

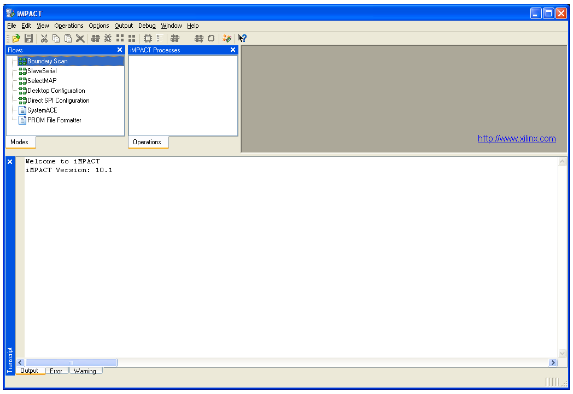
I want to

● load most recent project   default.ipf

☐ Load most recent project file when iMPACT starts

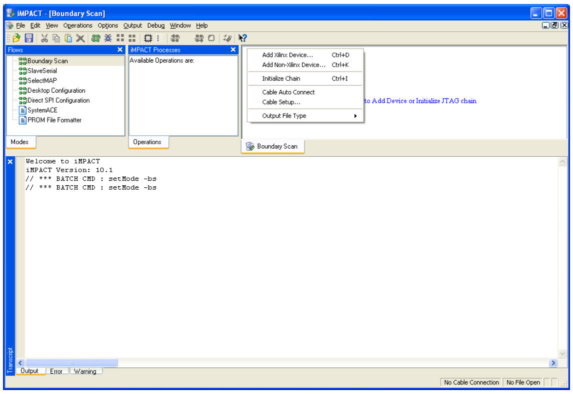
○ create a new project (.ipf)   default.ipf

Double click the "Boundary Scan" option in the "Modes" panel.



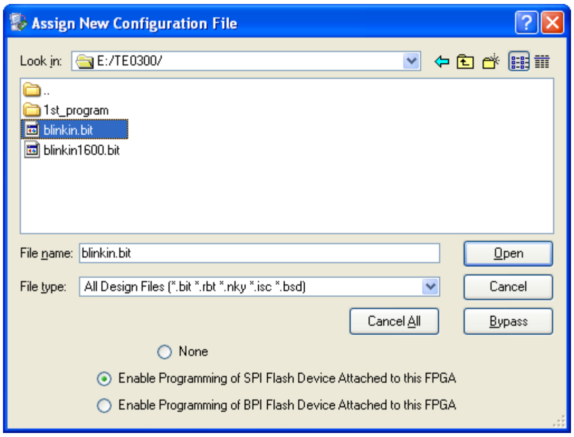
```
Welcome to iMPACT
iMPACT Version: 10.1
```

Right click the "Boundary Scan" to initialize the chain and select "Initialize Chain".



```
Welcome to iMPACT
iMPACT Version: 10.1
// *** BATCH CMD : setMode -bs
// *** BATCH CMD : setMode -bs
```

An "Assign New Configuration File" dialog window should pop up automatically. You can now select the file corresponding to your design. In the following example, we will show how to select the micromodule reference design "blinking.bit" in the "TE0300" folder. Do not forget to select the "Enable Programming of SPI Flash Device Attached to this FPGA" option in the same window.



Look in: E:/TE0300/

File name: blinkin.bit

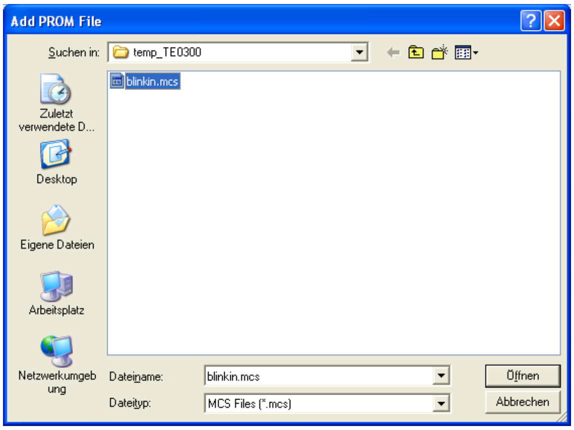
File type: All Design Files (*.bit *.rbt *.nky *.isc *.bsd)

○ None

● Enable Programming of SPI Flash Device Attached to this FPGA

○ Enable Programming of BPI Flash Device Attached to this FPGA

An "Add PROM File" dialog window should pop up automatically. You can now select the file corresponding to your design. In the following example, we will show how to select the micromodule reference design "blinking.mcs" in the "TE0300" folder.

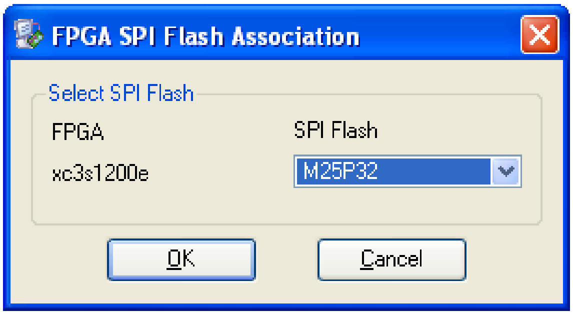


Suchen in: temp_TE0300

Dateiname: blinkin.mcs

Dateityp: MCS Files (*.mcs)

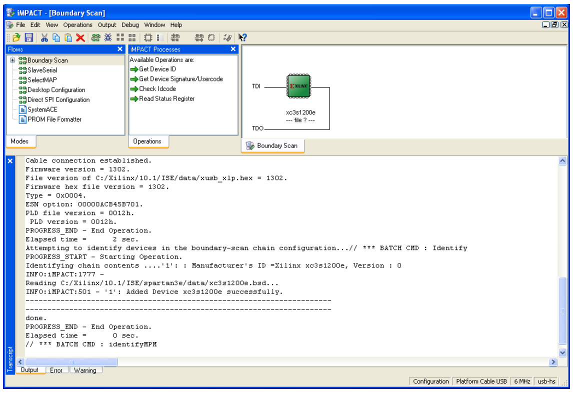
Select now the SPI Flash corresponding to the one present on the module (STMicroelectronics M25P32 in the example, a 32 Mbit (4M x 8) Serial Flash memory).



FPGA: xc3s1200e

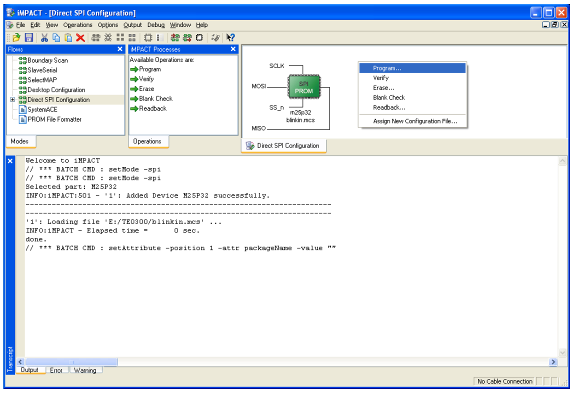
SPI Flash: M25P32

iMPACT should now look like this.



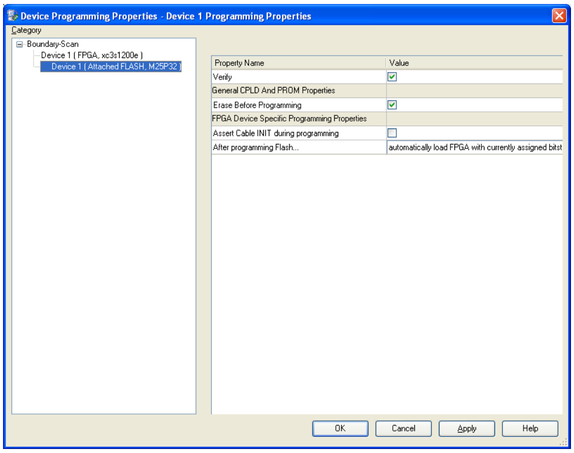
```
Cable connection established.
Firmware version = 1302.
File version of C:/Xilinx/10.1/ISE/data/xusb_xlp.hex = 1302.
Firmware hex file version = 1302.
Type = 0x0004.
ESN option: 00000ACB45B701.
PLD file version = 0012h.
 PLD version = 0012h.
PROGRESS_END - End Operation.
Elapsed time =      2 sec.
Attempting to identify devices in the boundary-scan chain configuration...// *** BATCH CMD : Identify
PROGRESS_START - Starting Operation.
Identifying chain contents ....'1': : Manufacturer's ID =Xilinx xc3s1200e, Version : 0
INFO:iMPACT:1777 -
Reading C:/Xilinx/10.1/ISE/spartan3e/data/xc3s1200e.bsd...
INFO:iMPACT:501 - '1': Added Device xc3s1200e successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
done.
PROGRESS_END - End Operation.
Elapsed time =      0 sec.
// *** BATCH CMD : identifyMPM
```

Right click the "Flash" device and select the "Program" operation.



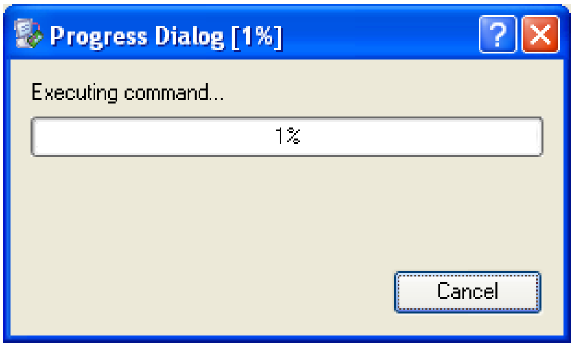
```
Welcome to iMPACT
// *** BATCH CMD : setMode -spi
// *** BATCH CMD : setMode -spi
Selected part: M25P32
INFO:iMPACT:501 - '1': Added Device M25P32 successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
'1': Loading file 'E:/TE0300/blinkin.mcs' ...
INFO:iMPACT - Elapsed time =      0 sec.
done.
// *** BATCH CMD : setAttribute -position 1 -attr packageName -value ""
```

In the "Device Programming Properties" window, just leave the default settings and press the "OK" button.



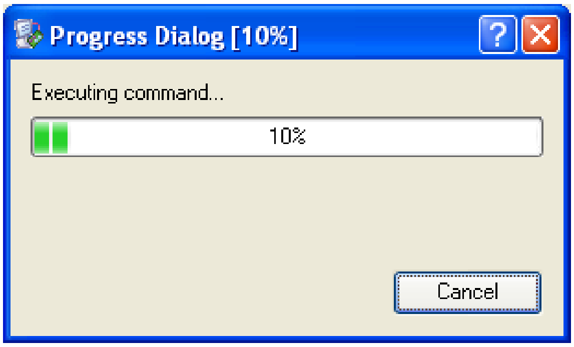
Property Name	Value
Verify	☑
General CPLD And PROM Properties	
Erase Before Programming	☑
FPGA Device Specific Programming Properties	
Assert Cable INIT during programming	☐
After programming Flash...	automatically load FPGA with currently assigned bitst

iMPACT will first erase the memory



Executing command...

and then write it.



Executing command...

After successful programming, you should read the message "Program Succeeded" popping up for a few seconds in the "Boundary Scan" panel.

tion into a foreign language) without prior agreement and written consent from Trenz Electronic.

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH (Registration, Evaluation, Authorisation and Restriction of Chemicals) compliance statement

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet.

According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products.

Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS (Restriction of Hazardous Substances) compliance statement

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE (Waste Electrical and Electronic Equipment)

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment.

Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Appendix

The following tables reports pin-out information of B2B (board-to-board) receptacle connectors J4 and J5 respectively.

The reference design summaries report the resources needed by the reference design on TE0300 modules of different dimensions (1200 and 1600 versions).

pin	B2B name	FPGA pin	FPGA name	bank	dir	supply	diff	diff	supply	dir	bank	FPGA name	FPGA pin	B2B name	pin
1	VccIO	-	VccO	0	I	-	-	-	-	I	0	VccO		VccIO	2
3	VccIO	-	VccO	0	I	-	-	-	-	I	0	VccO	-	VccIO	4
5	B3_L01_P	C1	IO_L01P	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L07P	G6	B3_L07_P	6
7	B3_L01_N	C2	IO_L01N	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L07N	G5	B3_L07_N	8
9	B3_L02_P	D1	IO_L02P	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L03N	E1	B3_L03_N	10
11	B3_L02_N	D2	IO_L02N / VREF	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L03P	E2	B3_L03_P	12
13	GND													GND	14
15	B0_IO_C3	C3	IO_L25P	0	IO	VccIO	N	Y	VccIO	IO	0	IO_L19_P	F7	B0_L19_P	16
17	B0_L24_N	B4	IO_L24_N	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L19N / VREF	E7	B0_L19_N	18
19	B0_L24_P	A4	IO_L24P	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L21N	E6	B0_L21_N	20
21	B0_IO_C4	C4	IO	0	IO	VccIO	N	Y	VccIO	IO	0	IO_L21P	D6	B0_L21_P	22
23	GND													GND	24
25	B0_L23_N	D5	IO_L23N / VREF	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L18N / VREF	D7	B0_L18_N	26
27	B0_L23_P	C5	IO_L23P	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L18P	C7	B0_L18_P	28
29	B0_L20_P	B6	IO_L20P	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L17N	F8	B0_L17_N	30
31	B0_L20_N	A6	IO_L20N	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L17P	E8	B0_L17_P	32
33	3.3 V													3.3 V	34
35	B0_IO_A7	A7	IO	0	IO	VccIO	N	N	VccIO	IO	0	IO	A8	B0_IO_A8	36
37	B0_IO_G9	G9	IO	0	IO	VccIO	N	Y	VccIO	IO	0	IO_L14N / GCLK11	D9	GCLK_L14_N	38
39	GCLK_L13_P	B8	IP_L13P / GCLK8	0	I	VccIO	Y	Y	VccIO	IO	0	IO_L14P / GCLK10	C9	GCLK_L14_P	40
41	GCLK_L13_N	B9	IP_L13N / GCLK9	0	I	VccIO	Y	Y	VccIO	IO	0	IO_L11N / GCLK5	E10	GCLK_L11_N	42
43	GND							Y	VccIO	IO	0	IO_L11P / GCLK4	D10	GCLK_L11_P	44
45	GCLK_L12_P	B10	IO_L12P / GCLK6	0	IO	VccIO	Y							GND	46
47	GCLK_L12_N	A10	IO_L12N / GCLK7	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L09N	D11	B0_L09_N	48
49	B0_L15_P	E9	IO_L15P	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L09P	C11	B0_L09_P	50
51	B0_L15_N	F9	IO_L15N	0	IO	VccIO	Y	N	VccIO	IO	0	IO	A11	B0_I0_A11	52
53	2.5 V													2.5 V	54
55	B0_L08_P	E11	IO_L08P	0	IO	VccIO	Y	N	VccIO	IO	0	IO/VREF	B11	B0_IO_B11	56
57	B0_L08_N	F11	IO_L08N	0	IO	VccIO	Y	N	VccIO	IO	0	IO	A12	B0_IO_A12	58
59	B0_L05_P	A13	IO_L05P	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L06P	F12	B0_L06_P	60
61	B0_L05_N	B13	IO_L05N / VREF	0	IO	VccIO	Y	Y	VccIO	IO	0	IO_L06N	E12	B0_L06_N	62
63	GND													GND	64
65	B0_L04_N	A14	IO_L04N	0	IO	VccIO	Y	N	VccIO	IO	0	IO	D13	B0_IO_D13	66
67	B0_L04_P	B14	IO_L04P	0	IO	VccIO	Y	N	VccIO	IO	0	IO	E13	B0_IO_E13	68
69	B0_L03_N	C14	IO_L03N / VREF	0	IO	VccIO	Y		3.3 V	I	2	TDI	A2	TDI	70
71	B0_L03_P	D14	IO_L03P	0	IO	VccIO	Y		3.3 V	O	2	TDO	C16	TDO	72
73	1.2 V													1.2 V	74
75	B0_L01_N	A16	IO_L01N	0	IO	VccIO	Y		3.3 V	I	2	TCK	A17	TCK	76
77	B0_L01_P	B16	IO_L01P	0	IO	VccIO	Y		3.3 V	I	2	TMS	D15	TMS	78
79	GND													GND	80

receptacle connector J4 pinout information

pin	B2B name	FPGA pin	FPGA name	bank	dir	supply	diff	diff	supply	dir	bank	FPGA name	FPGA pin	B2B name	pin
1	5Vb2b				I					I				5Vb2b	2
3	5Vb2b				I					I				5Vb2b	4
5	5V				O					I				/MR	6
7	B2B_D_P				IO		Y			O				/RESET	8
9	B2B_D_N				IO		Y			O				RESET	10
11	GND													GND	12
13	B3_L22_P	P3	IO_L22P	3	IO	3.3 V	Y	N	3.3 V	IO	3	IO_L24P	T2	B3_IO_T2	14
15	B3_L22_N	P4	IO_L22N	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L21N	P1	B3_L21_N	16
17	B2_IP_V4	V4	IP_L02P	2	I	3.3 V	N	Y	3.3 V	IO	3	IO_L21P	P2	B3_L21_P	18
19	B3_L20_P	N4	IO_L20P	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L23N	R2	B3_L23_N	20
21	B3_L20_N	N5	IO_L20N	3	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L23P	R3	B3_L23_P	22
23	GND													GND	24
25	B2_L04_N	T5	IO_L04N	2	IO	3.3 V	Y	N	3.3 V	IO	3	IO_L18N	M3	B3_IO_L18N	26
27	B2_L04_P	R5	IO_L04P	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO_L03P / DOUT / BUSY	U4	B2_IO_L03	28
29	B2_L05_P	R6	IO_L05P	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO / VREF	U5	B2_IO_U5	30
31	B2_L05_N	P6	IO_L05N	2	IO	3.3 V	Y	Y	3.3 V	IO	2	IO_L06P	V5	B2_L06_P	32
33	B2_IO_V7	V7	IO	2	IO	3.3 V	N	Y	3.3 V	IO	2	IO_L06N / VREF	V6	B2_L06_N	34
35	3.3 V													3.3 V	36
37	B2_L07_N	P7	IO_L07N	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO	U6	B2_IO_U6	38
39	B2_L07_P	N7	IO_L07P	2	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L17N / VREF	L5	B3_L17_N	40
41	B2_GCLK12	M9	IO_L12N / D6 / GCLK12	2	I	3.3 V	N	Y	3.3 V	IO	3	IO_L17P	L6	B3_L17_P	42
43	GND													GND	44
45	B2_L10_N	T8	IO_L10N	2	IO	3.3 V	Y	N	3.3 V	I	2	IP_L08P	T7	B2_IP_T7	46
47	B2_L10_P	R8	IO_L10P	2	IO	3.3 V	Y	N	3.3 V	I	2	IP_L11P	U8	B2_IP_U8	48
49	B2_GCLK_L13_N	V9	IO_L13N / D3 / GCLK15	2	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L19P	M5	B3_L19_P	50
51	B2_GCLK_L13_P	U9	IO_L13P / D4 / GCLK14	2	IO	3.3 V	Y	Y	3.3 V	IO	3	IO_L19N	M6	B3_L19_N	52
53	2.5 V					3.3 V								2.5 V	54
55	B2_L18_N	N11	IO_L18N	2	IO	3.3 V	Y	Y	3.3 V	IO	2	IO_L09P	P8	B2_L09_P	56
57	B2_L18_P	P11	IO_L18P	2	IO	3.3 V	Y	Y	3.3 V	IO	2	IO_L09N	N8	B2_L09_N	58
59	B2_L20_N	R12	IO_L20N	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO	P9	B2_IO_P9	60
61	B2_L20_P	T12	IO_L20P	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO	R11	B2_IO_R11	62
63	GND													GND	64
65	B2_L19_N	V13	IO_L19N / VREF	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO_L15N / D1 / GCLK3	P10	B2_IO_P10	66
67	B2_L19_P	V12	IO_L19P	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO / D5	R9	B2_IO_R9	68
69	B2_L22_N	R13	IO_L22N / A22	2	IO	3.3 V	Y	Y	3.3 V	IO	2	IO_L21N	P12	B2_L21_N	70
71	B2_L22_P	P13	IO_L22P / A23	2	IO	3.3 V	Y	Y	3.3 V	IO	2	IO_L21P	N12	B2_L21_P	72
73	1.2 V													1.2 V	74
75	B2_L24_P	T14	IO_L24P / A21	2	IO	3.3 V	Y	N	3.3 V	I	2	IP_L23P	V14	B2_IP_V14	76
77	B2_L24_N	R14	IO_L24N / A20	2	IO	3.3 V	Y	N	3.3 V	IO	2	IO	U13	B2_IO_U13	78
79	GND													GND	80

receptacle connector J5 pinout information

Reference Design Summary: Xilinx Spartan-3E 1200

```
----------------------------------------------------------------------

platgen -p xc3s1200efg320-4 -lang vhdl -lp x:/xxx/  system.mhs
Release 11.5 - platgen Xilinx EDK 11.5 Build EDK_LS5.70 (nt)
Copyright (c) 1995-2009 Xilinx, Inc.  All rights reserved.

----------------------------------------------------------------------

Command Line: platgen -p xc3s1200efg320-4 -lang vhdl -lp

Running post-placement packing...

Design Summary:
Number of errors:      0
Number of warnings:   16
Logic Utilization:
  Number of Slice Flip Flops:         5,885 out of  17,344   33%
  Number of 4 input LUTs:             8,041 out of  17,344   46%
Logic Distribution:
  Number of occupied Slices:          6,622 out of   8,672   76%
    Number of Slices containing only related logic:   6,622 out of   6,622 100%
    Number of Slices containing unrelated logic:         0 out of   6,622   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:       8,424 out of  17,344   48%
    Number used as logic:             6,066
    Number used as a route-thru:        383
    Number used for Dual Port RAMs:   1,812
      (Two LUTs used per Dual Port RAM)
    Number used as Shift registers:     163

  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.

  Number of bonded IOBs:                 74 out of     250   29%
    IOB Flip Flops:                      36
    IOB Master Pads:                      1
    IOB Slave Pads:                       1
  Number of ODDR2s used:                 22
    Number of DDR_ALIGNMENT = NONE       22
    Number of DDR_ALIGNMENT = C0          0
    Number of DDR_ALIGNMENT = C1          0
  Number of RAMB16s:                     25 out of      28   89%
  Number of BUFGMUXs:                     5 out of      24   20%
  Number of DCMs:                         1 out of       8   12%
  Number of BSCANs:                       1 out of       1  100%
  Number of MULT18X18SIOs:                3 out of      28   10%

  Number of RPM macros:            2
Average Fanout of Non-Clock Nets:                3.57

----------------------------------------------------------------------
```

Reference Design Summary: Xilinx Spartan-3E 1600

```
----------------------------------------------------------------------

platgen -p xc3s1600efg320-4 -lang vhdl -lp x:/xxx/  system.mhs

Release 11.5 - platgen Xilinx EDK 11.5 Build EDK_LS5.70 (nt)
Copyright (c) 1995-2009 Xilinx, Inc.  All rights reserved.

----------------------------------------------------------------------

Command Line: platgen -p xc3s1600efg320-4 -lang vhdl

Running post-placement packing...

Design Summary:
Number of errors:      0
Number of warnings:   16
Logic Utilization:
  Number of Slice Flip Flops:         5,885 out of  29,504   19%
  Number of 4 input LUTs:             8,038 out of  29,504   27%
Logic Distribution:
  Number of occupied Slices:          6,424 out of  14,752   43%
    Number of Slices containing only related logic:   6,424 out of   6,424 100%
    Number of Slices containing unrelated logic:          0 out of   6,424   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:       8,421 out of  29,504   28%
    Number used as logic:             6,063
    Number used as a route-thru:        383
    Number used for Dual Port RAMs:   1,812
      (Two LUTs used per Dual Port RAM)
    Number used as Shift registers:     163

  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.

  Number of bonded IOBs:                 74 out of     250   29%
    IOB Flip Flops:                      36
    IOB Master Pads:                      1
    IOB Slave Pads:                       1
  Number of ODDR2s used:                 22
    Number of DDR_ALIGNMENT = NONE       22
    Number of DDR_ALIGNMENT = C0          0
    Number of DDR_ALIGNMENT = C1          0
  Number of RAMB16s:                     25 out of      36   69%
  Number of BUFGMUXs:                     5 out of      24   20%
  Number of DCMs:                         1 out of       8   12%
  Number of BSCANs:                       1 out of       1  100%
  Number of MULT18X18SIOs:                3 out of      36    8%

  Number of RPM macros:            1
Average Fanout of Non-Clock Nets:                3.57

----------------------------------------------------------------------
```