Trizeps6 Getting Started with Windows Embedded CE6 R3 2010Q2

Version 1.0

This document guides you through the first steps of using the Trizeps6 running on the ConXS, uConXS or iPAN7-Board.

1.0 Introduction

This Board-Support-Package (BSP) is intended for use with following boards:

- ConXS with Trizeps6.
- i-PAN7 with Trizeps6.
- uConXS with Trizeps6.
- TC-Box with Trizeps6.
- Customer-boards using Trizeps6.

The Trizeps6 is a small and complete SODIMM-size computer. Everything for a fully working system, like Flash-Storage, RAM and more is already included and only a single power-supply of 3.3V is needed to run it. There is an option to run the Trizeps6 directly from a LiPo-battery-cell or +5V power-supply; contact Keith & Koep for more information.

The (u)ConXS-board is designed as motherboard for SODIMM200 Trizeps modules. It contains several interfaces and is a good starting point for your first experiences with embedded computing.

The i-PAN7 is a 7“ WVGA(800x480) display, combined with a Trizeps and several interface options. This BSP can be used to build Windows-Embedded CE6 images for i-PAN7 LC and i-PAN7 LC-E.

The TC-Box is a small SODIMM200-baseboard, which got everything for a Thin-Client solution (Ethernet, USB, Audio, HDMI-connector, WLAN and Bluetooth-antennas,...)

See Keith&Koep Wiki at the company website: www.keith-koep.com\service for additional information, application notes and data-sheets.

Username: wince
Password: secret

Note the „Terms of Use“ which can be found at http://www.keith-koep.com/service/doku.php/service/disclaimer!!

2.0 Getting Started

2.1 Evaluation-Kit Contents

The kit contents varies, dependant on what kit you have bought.

Typically it consist of the hardware and a CD/DVD containing Board-Support-Packages, tools and documentation.

If bought as commercial evaluation-kit, it includes a power-supply and cables.

The iPAN7, TC-Box and (u)ConXS when bought as evaluation kit, contain a pre-flashed Windows Embedded CE6 R3 image.

If you bought the Trizeps seperatly, it will only contain the Keith&Koep bootloader.

2.2 Starting the board

Normally, when bought as evaluation-board, everything should be assembled and ready to use. All you must do, is connect the power-supply to the board and wait till the pre-installed Windows Embedded CE-image boots up. You may attach a display to the VGA-connector, when using the uConXS or HDMI/DVI-connector, when using a TC-Box.

If the SODIMM of the baseboard is not fitted with a Trizeps module when you receive your board, follow these instructions:

1. Slide the Trizeps into the socket taking account of the polarity mark. Do not touch the gold contacts. You can see that there is a polarization mark cut in the Trizeps ; this ensures that the module is adjusted correctly. Put the Trizeps module carefully at an angle of about 30 degrees into the socket.
2. Support the underside of the board and push the Trizeps down into the socket. It should click into its place.

2.3 Using the board

The Windows Embedded CE-Image already stored on the Trizeps-module should include everything you will need to get started with your own project. Windows Embedded CE is made up from several components, which may be selected to be included or not. This allows fitting images to several devices. If something is missing, you might want to build an image using the Keith & Koep Board-Support-Package and Platform-Builder.

With Windows Embedded CE you can do almost everything, that you can do with a „big“ computer. Allthough a full OS can be squizzed into devices with less than 32MB flash, many interfaces from the bigger Desktop-Windows can be used. Simple Applications written for Windows would also recompile for WinCE with only minor changes.

The network-driver is set up to get its IP-address through DHCP. If you have no DHCP-server connected, you can change network settings at:

Start->Settings->Network and Dial-up Connections.
Double-click PXAMFU1 and set your network settings accordingly.

2.3.1 Internet Explorer 6

Surf the Internet with Internet Explorer 6.

2.3.2 Media Player 9

Play media-files (audio and video) from local storage devices or using network-media-streams.

2.3.3 Microsoft Word Pad

Do simple Text editing.

2.3.4 Remote Desktop

Connect to your Home-PC or Server using a remote-desktop-connection.

2.3.5 Use the HTTP, FTP Server

To attach to these servers, obtain the IP-address of the ethernet-controller (‚double-tap the network-symbol in the task-bar) and open your browser with:

- ftp://wince:secret@<IP-Address>

  i.e. ftp://wince:secret@195.8.230.20

  View [HKEY_LOCAL_MACHINE\COMM\FTPD] registry keys for additional settings.

  The ftp-server is configured to use the „\temp“-directory on default.
- http://<IP-Address>/ADMIN

  i.e. http://195.8.230.20/ADMIN

  Use username: wince, password: secret.

  View [HKEY_LOCAL_MACHINE\COMM\HTTPD] registry keys for additional settings.

2.3.6 Connect to the Device with Telnet

On your PC run „telnet <IP-Address>“.

login (username): wince

password: secret

2.3.7 ActiveSync Tools

ActiveSync is a tool, that helps you connect the board to your development workstation. Besides the functionality you know from PDA’s or other devices, that use ActiveSync to synchronise phone-books or download programs, you may also use ActiveSync to debug your program or use some of the remote tools (i.e. Remote Registry Editor). If not already installed, you can get ActiveSync from Microsoft for download.

To use ActiveSync, connect the board through an USB-cable. ActiveSync will connect automatically. If not, open Microsoft ActiveSync and choose „File->Connection Settings...“ and validate that „Allow USB connections“ is selected.

Note: On Trizeps6, there is no dedicated USB-Slave port. USB-Host port 2 functions as OTG-port and can be configured to run as slave (see: http://www.keith-koep.com/service/doku.php/service/wince/bsp/driversxscale/usb). On (u)ConXS this is the lower USB-Host port. The ConXS also got a 5-pin-header to connect an OTG-connector.

2.3.8 RotateScreen

A little tool named RotateScreen exists, which is shown in the taskbar:

- Tap it once to rotate the screen.
- Tap it twice to open a control-panel, which lets you set the rotation-angles.

2.3.9 KuK-Tools, drvlib_app.dll and the persistant registry

drvlib_app.dll is a DLL which contains many useful functions to do device-specific things like reset, gpio-functions and i2c-communication.

The KuK-Tools-application is not included in this release of the BSP. Since the default registry-type is hive-based, the registry-store functionality is not needed. KuK-Tools will be implemented in the next BSP, when RAM-based registry store and erase will be included again.

2.3.10 Build your own application using Visual-Studio

The typical work-flow would be, to create a Windows-Embedded CE6 R3 project. Than build an image and after that create an SDK (Software-Development-Kit). With this SDK you may build applications, that fit to the components included in this image.

View Visual-Studio Help and the Keith&Koep Wiki pages for details.

To start application-programming quickly, pre-built SDK's can be found at: [http://www.keith-koep.com/service/doku.php/service/wince/sdk/sdk]
This is an example on how to install and use a SDK:

1. Install a Trizeps6 SDK.
2. Start Visual Studio 2005.
3. File->New->Project...
4. Select Visual C++ -> Smart Device -> MFC Smart Device Application.
5. Name it mfcsample and press ok.
6. Press Next. This will select the Platforms-Page.
7. Move the Trizeps6-SDK to the Selected-Tab.
8. Remove other SDK's from the Selected-Tab.
9. Press „Finish" to complete configuration. Press Next if you would like to do further modifications.
10. Connect the baseboard with USB to your workstation.
11. ActiveSync should connect. If not, make sure you got ActiveSync installed.
12. Press „F5"-key. This will build the project, deploy it to the Trizeps6-device and start debugging.

If USB is not easily accessible on your baseboard, use ethernet instead. (See FAQ: http://www.keith-koep.com/service/doku.php/service/faq#how_to_debug_an_application_running_on_windows_embedded_ce6_using_ethernet)

3.0 Using the bootloader

The bootloader is the „BIOS“ of a Trizeps-module. It decides what to boot and may assist in testing your hardware.

To enter the bootloader-command-interpreter:

1. Connect the board with a serial null-modem-cable to your pc.
2. Open Hyperterminal with 38,4kBaud, 8 data-bits, no parity, 1 stop bit, no flow control.
3. Press ESC and hold it pressed while turning on the board.
4. Take a look at the supported bootloader-functions: „?“.

TABLE 1. Often used Bootloader-Commands

Command	**Description**
?	print supported functions.
ereg	erase permanent registry (recommended, before loading a new WinCE-image).
eflashb0	erase everything from flash, except the bootloader.
epsm	erase user-partition (contains registry).
mount mmc	mount storage card. This must be called, before you can use the card.
cd, dir	navigate through the directory structure of a mounted storage card.
boot mmc <file>	load a file from a storage card.
fb	flash boot. Boots image stored in flash.
eboot	start download of an image from Platform-Builder.
menu	Configure additional boot-settings (i.e. frequency).

A complete list of all bootloader commands can be found at: [http://www.keith-koep.com/service/doku.php/service/software/bootloader/cmdreference]

3.1 Loading files

The bootloader can load different file-types. Files can be stored to flash, copied to RAM or executed if they are boot-scripts.

Examples for loadable files are:

- WinCE-Images:
  [CD::\wince600\Images]

[http://www.keith-koep.com/service/doku.php/service/wince/images/images]

- Scripts
- Bitmaps (usually used in conjunction with a script)

The bootloader can load files from various locations:

- Ethernet using Windows Embedded CE Eboot.
- USB using Windows Embedded CE Eboot.
- SD/MMC-Cards (formated as FAT12/16/32).

3.1.1 Loading files using Ethernet or USB (Eboot)

This procedure can only be used for Windows Embedded CE images (nk.bin or xip.bin). See chapter „Using the WinCE Board-Support-Package“ for details.

3.1.2 Loading files using storage-card-interface

To load a file using a SD/MMC-Card:

1. Copy the file to a SD or MMC -Card.
2. Insert it into the (u)ConXS-Board or i-PAN7.
3. Type „mount mmc“ to mount the card.
4. Type „boot mmc <filename>“.
5. Loading should begin.

or

1. Copy the file to a SD/MMC (note the filename should not be longer than 8 characters).
2. Create a file on the SD-card named autoboot.bat.
3. This file can have bootloader-commands like „boot mmc <filename>“. Where filename is the name of the file to boot.
4. Insert card and switch on power of the board.
5. While booting, the bootloader will check if a file named „autoboot.bat“ exists on a connected storage-media and will boot it.

see [http://www.keith-koep.com/service/doku.php/service/software/bootloader/appnote_autoboot]

4.0 Using the Windows Embedded CE Board-Support-Package (BSP)

The Windows Embedded CE-BSP helps you in creating your own custom CE-images. Prebuild CE-Images may be found at [CD::\wince600\Images] or [http://www.keith-koep.com/service/doku.php/service/wince/images/images] . Prior to installing the Board-Support-Package, you will need to install Visual-Studio-Professional 2005 and Windows Embedded CE6.0 R3 Platform Builder.

For more detailed instructions view the wiki-page at [http://www.keith-koep.com/service/doku.php/service/wince/appnotepub/tr6conxs_buildce6fromscratch].

4.1 Installing the Trizeps6 Board-Support-Package

The BSP contains platform-files (tr6conxs) and three projects (Minimal, CoreMax and MaxIE).
Copy the tr6conxs-directory to <WINCEROOT>\platform\tr6conxs.
Copy the three projects to <WINCEROOT>\OSDesigns.
[WINCEROOT = <Windows Embedded CE Installation path>\WinCE600]

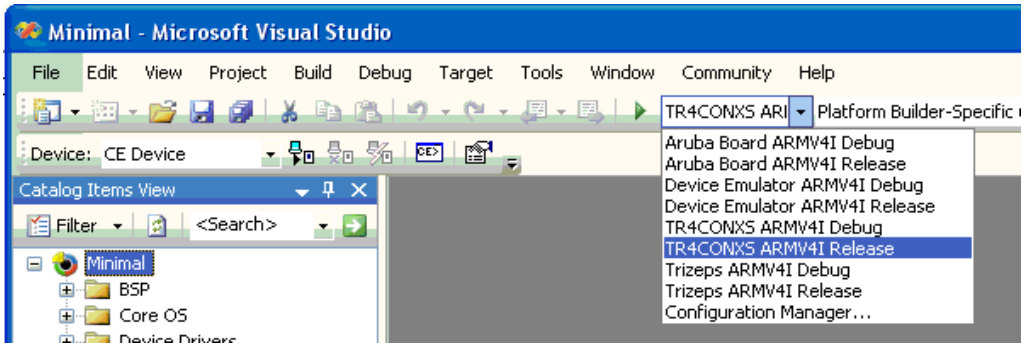
4.2 Creating an image with an existing project

To give you a good starting point, 3 projects are included with this BSP.

- Minimal
  This is a minimal configuration, including all components needed, to use all of the uConXS-Board interfaces (VGA, UART, SD/MMC, etc.).
  [Windows Embedded CE 6.0 Core Run-Time License]
- CoreMax
  This is a typical configuration. It's a good example of what features are still part of a Core Run-Time License.
  [Windows Embedded CE 6.0 Core Run-Time License]
- MaxIE
  This is a full-featured Windows Embedded CE 6.0 image, including Internet-Explorer and Media-Player. And yet it does not include all Windows Embedded CE-components.
  [Windows Embedded CE 6.0 Professional Run-Time License]

To open one of these projects double-click the Microsoft Visual Studio Solution (*.sln) file in the <WINCEROOT>\OSDesigns\<project-name> directory.

Select a solution configuration (TR6CONXS ARMVI Release or Debug):



To build a project, choose Build->Build <project-name> from the menu.

If no errors occur, the build process runs makeimg.exe. which will create a xip.bin file in your _FLATRELEASEDIR

This file can be downloaded to the device through the standard bootloader in many ways (see 2.3.1. Loading files).

4.3 Downloading the Windows Embedded CE Image

The simplest way to download your Windows Embedded CE Image to your board is through SD-card.

Copy xip.bin to the card.

Create an autoboot.bat-file containing:
boot nanddisk
boot mmc xip.bin
ereg
epsm
fb

(File must end with a blank line)
Insert the card and power-up the board.
boot nanddisk: runs an initial script, which typical shows a boot bitmap.
boot mmc xip.bin will store the CE-image to flash.
ereg and epsm erase the registry and user-store.
fb will start the stored image.

Important:
If you don't erase the registry with ereg, your new CE-Image might hang during boot.

The „boot nanddisk" command in the autoboot.bat file might be important to let the display-driver know the correct display-settings. „boot nanddisk" actually calls „boot nanddisk autoboot.bat" -- that is the autoboot.bat file stored in flash, which is run on every boot and typically contains the display settings.
If this command is not used or if you don't have or want a boot-bitmap, the correct display must be set in the Platform-Builder project. (Catalog->Third Party->BSP->TR6CONXS: ARMV4I->Device Drivers->Display->Trizeps Framebuffer Device->Type)

4.4 Downloading the Windows Embedded CE Image with Eboot

Eboot is the „Microsoft Bootloader" for Windows Embedded CE -devices and offers a good integration in the Visual-Studio IDE. This Eboot-implementation is integrated in the Keith&Koep bootloader.

See [http://www.keith-koep.com/service/doku.php/service/wince/appnotepub/tr6conxs_usingeboot] for the latest information on eboot-usage.

If you want to use an Eboot-interface for downloading images to the (u)ConXS or i-PAN7 board, you have two options:

- Ethernet
- USB (using RNDIS)

To use the autoboot-mechanism prepare a SD-card with an autoboot.bat with following content:
boot nanddisk
eboot kitl=0xd eth

„*kitl=0xd eth*“ are command-line parameters used by eboot and the Windows-Embedded CE image.
The **kitl** parameter specifies if and how to use KITL (Kernel Independant Transport Layer). If **kitl** is not given on the command-line, no kitl will be used. The value assigned to kitl is made up out of these flags

TABLE 2. Kitl-flags

kitl-flags	**description**
0x01	Kitl enable
0x04	DHCP enable
0x08	VMINI enable

0xd will enable Kitl, DHCP and VMINI. VMINI is a virtual ethernet adapter that sits on top of kitl and allows TCP/IP connectivity while the ethernet-adapter is used for kitl-debugging.

If you don’t work on a network containing a DHCP-server, you may want to enter a static IP-address. To use a static IP-address pass „ipAddress=ipnumber“ to the command-line.
i.e. eboot kitl eth ipAddress=0x01fe8c0
Note that the IP-address is coded hexadecimal: 192.168.254.1 (c0.a8.fe.01)
When using USB-RNDIS, the IP will default to the static address 192.168.254.1.

eth selects ethernet as transport.
rndis selects USB-RNDIS as transport.

After the SD-card is prepared, put it into the SD-card slot of the board and power it on. Now eboot will start. Don’t forget to plug the ethernet or USB cable.

In Visual-Studio open Target->Connectivity Options... and select Ethernet as download.
Press the settings button next to the download-listbox and wait till an active target device shows up. Select this one and press OK, Apply and Close.
After building the image select Target-> Attach Device.
Donwload should start.

4.5 Using Debug-Images

There are two options to connect to a debug-image, which you may build when using configuration „TR6CONXS ARMV4I Debug“. (These example assume you build an image with „Type: nk.nb0 (RAM)“-configuration.)

1. Using Trizeps6-Ethernetcontroller.
Copy nk.nb0 to a SD-card and create an autoboot.bat-file with this content:
boot mmc nk.nb0 kitl eth

2.) Using USB-Slave as RNDIS-device.
Copy nk.nb0 to a SD-card and create an autoboot.bat-file with this content:
boot mmc nk.nb0 kitl rndis

Start the device with the created SD-Card.
Open Target->Connectivity Options... and select Ethernet as transport (download set to none, debugger set to KdStub). You must select Ethernet, even if you want to use USB as kitl-transport!
Press the settings button next to the transport-lisbox and wait till an active target device shows up. Select this one and press OK, Apply and Close.
Select Target-> Attach Device.
Debugger should connect.

When using Eboot set Download to Ethernet in the Connectivity Options dialog and select: „Use device name from bootloader“ in the transport settings dialog.
The command-line parameters described in the previous chapter will work here too.

If you are using the USB-RNDIS option for the first time, you may have to specify a valid usb8023.inf-file and usb8023.sys:
C:\WINCE600\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\RNDIS-MINI\HOST\usb8023.inf
C:\WINDOWS\system32\drivers\usb8023.sys
After the network-adapter appears, assign a static IP-address of 192.168.254.2.

4.6 Creating an image from scratch

1. File-> New Project...
2. Choose Project type: „Platform Builder for CE 6.0“
3. Select „OS-Design“ and press OK.
4. When Windows Embedded CE6.0 OS Design Wizard opens, press Next.
5. Choose TR6CONXS: ARMV4I as Board Support Package.
6. Follow the instruction of the design wizard..
7. Select which type of image you want to build:

Type: nk.nb0 (RAM): Build an image, which is not stored to flash. This is quite usefull if you want to directly download the image through Platform-Builder to RAM and start debugging. It can also be used to start images from SD-card. This configuration uses a RAM-based registry and is comparable to the mode used on earlier Trizeps-products.

Type: xip.bin (ROM, BinFS): Build an image which is stored on a seperate flash-partition. It uses a hive-based registry and the flashdisk is mounted as root-filesystem. That results in all changes to be persistant (like you experience on desktop operating-systems). Note: When using this configuration, you should set:

Project-> ProjectName Properties..->Configuration Properties->General->Target file name for debugger: to *xip.bin*. Be sure to change this for both build-configurations: TR6CONXS ARMV4I Debug and TR6CONXS ARMV4I Release.

Note:

If a component added is excluded from build (red-cross instead a green check mark), you get the reason by pressing the right-mouse-button on that icon and choosing „Reason for Exclusion of Item“.

Sometimes drivers need additional components from the Microsoft Platform-Builder to function as expected!

- To use a SD/MMC storage card, you have to include the FAT-Filesystem component.

SD/MMC: add „SD Memory“ SYSGEN_SD_MEMORY

USB-Slave: add at least one of the USB-Function clients (RNDIS, Serial(ActivSync), Mass Storage) (i.e. Serial: SYSGEN_USBFN_SERIAL).