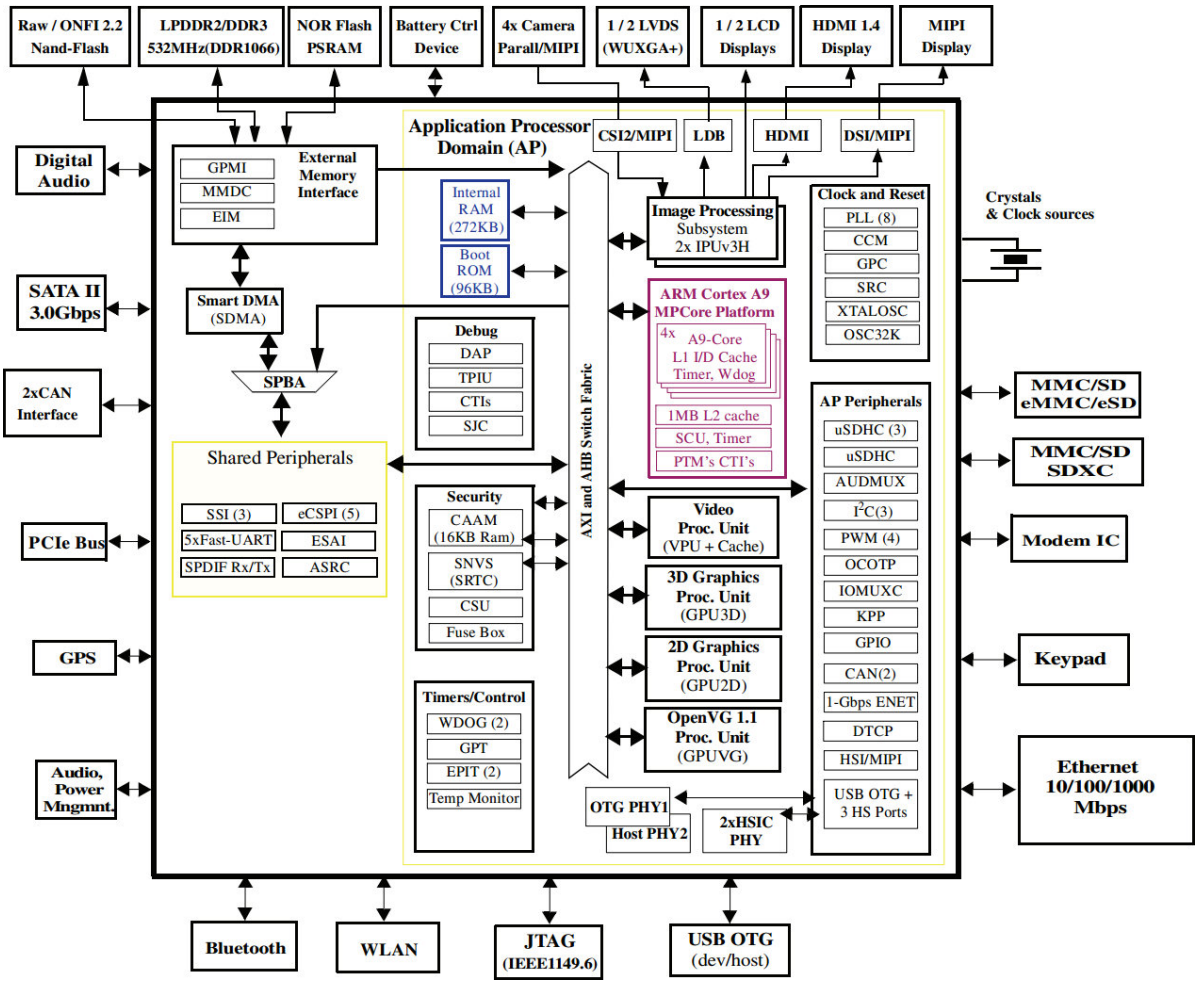
Embedded Technologies Experts

GURUCE I.MX6 GETTING STARTED GUIDE

GETTING STARTED WITH THE GURUCE I.MX6 WEC 7 & 2013 BSP AND EVALUATION KERNELS



CONTENTS

INTRODUCTION

This document describes how to get started with the GuruCE i.MX6 Board Support Package (BSP) and evaluation images for some of the supported standard hardware platforms. The BSP, developed by GuruCE, targets any board containing a NXP i.MX6 series processor and supports, out of the box, the following off-the-shelf hardware platforms:

- ✓ Element14 RIoTboard
- ✓ Element14 SabreLite
- ✓ Boundary Devices BD-SL
- ✓ Boundary Devices Nitrogen6X
- ✓ Boundary Devices Nitrogen6_VM
- ✓ Device Solutions Opal6-S
- ✓ Device Solutions Opal6-DL
- ✓ Device Solutions Opal6-Q
- ✓ Device Solutions Opal6-QP
- ✓ Congatec Conga-QMX6
- ✓ Digi ConnectCore6-Q
- ✓ Digi ConnectCore6-DL
- ✓ NXP Sabre SDP-DL
- ✓ NXP Sabre SDP-DQ
- ✓ NXP Sabre SDB-QP
- ✓ NXP MCIMX6ULL EVK
- ✓ Technologic Systems TS4900-S
- ✓ Technologic Systems TS4900-Q
- ✓ Toradex Colibri-DL
- ✓ Variscite VAR-SOM-SOLO
- ✓ Variscite VAR-SOM-DQ
- ✓ Emtrion DIMM-MX6DL
- ✓ Advantech DMS-BA16
- ✓ iWave iW-RainboW-G15M-Q7

Support for more boards will be added to the BSP over time and GuruCE can of course add support for your custom board on request. Please contact support@guruce.com for more information.

Evaluation kernel and bootloader images for selected boards are available on the GuruCE website at https://guruce.com/imx6/latest. Instructions for working with the evaluation images can be found in this document.

The GuruCE i.MX6 BSP fully supports the Microsoft Windows Embedded Compact[1] 7 and Microsoft Windows Embedded Compact 2013 operating systems. Evaluation images for both OS versions are available online.

[1] In this document we sometimes use “Windows CE” and sometimes “Windows Embedded Compact”. Windows CE was the name used for Windows Embedded Compact before it became Windows Embedded Compact. Many articles, blog posts, documentation and source code still refer to this OS as “Windows CE”, and so do we. For all intents and purposes; Windows CE == Windows Embedded Compact.

INSTALLING THE GURUCE EVALUATION IMAGES

PREREQUISITES

To be able to flash the GuruCE i.MX6 evaluation kernel images to the board you need the following tools:

- NXP's Manufacturing Tool (MfgTool2)
- CEWriter
- CELoader
- Kernel and bootloader images (nk.bin or nklz4.bin, eboot.bin and eboot.nb0)
- Serial console terminal (e.g. TeraTerm) connected to the serial debug port of the development board
- 7-zip to unpack the packages

Everything, apart from 7-zip and a serial console terminal, is supplied by GuruCE in the evaluation image 7-zip package.

In this document we walk you through the steps required to setup your chosen board so you can evaluate the GuruCE kernel and bootloader evaluation images.

Please note that our license does not allow the use of our evaluation images and accompanying tools for anything other than evaluating our i.MX6 BSP (in other words: the evaluation images and tools cannot be used in, or for preparation of, production devices).

Please also note that GuruCE does not manufacture any board or module, so **please refer to the board manufacturer documentation** for the following hardware specific settings:

- Serial debug port
- Boot switches (see also APPENDIX A)
- Power supply
- Reset button
- etc.

FLASHING THE GURUCE I.MX6 EVALUATION IMAGES

Many development kits are delivered containing pre-flashed bootloader and kernel images. To evaluate the GuruCE i.MX6 evaluation images you will, in some cases, need to replace the pre-flashed bootloader and kernel image with the GuruCE i.MX6 evaluation images. Some boards allow 'dual boot' so you don't have to erase the existing bootloader and kernel images already flashed on the board.

IMPORTANT NOTE: If you already have our eboot bootloader flashed and running on your device, please follow the much easier and quicker steps in the chapter "Updating the kernel, bootloader or boot splash image".

If your device is booting or can boot from an SD card or from a SATA disk, it is much easier to use our CEWriter tool, see chapter "Using the GuruCE CEWriter" below.

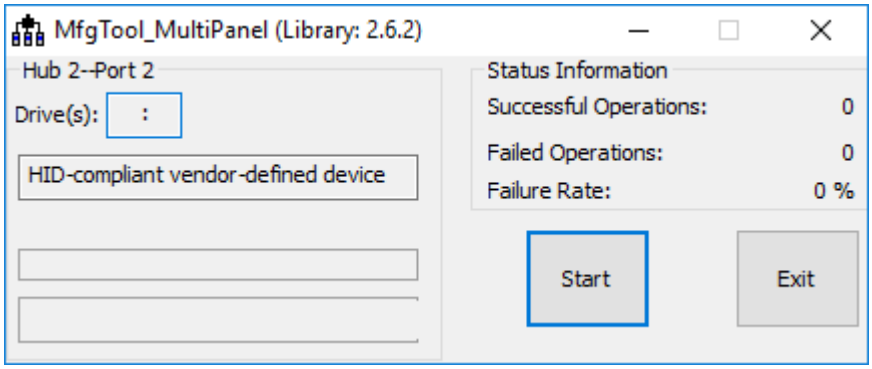
If you can boot the device from SD but want to boot from eMMC, it is easier to first boot from SD and then transfer the firmware files from SD to eMMC using the bootloader menu option "[M] SD/MMC menu -> [C] Copy Firmware". Once that is done you can change the boot switches to boot from eMMC. It's much quicker than following the long version below!

The steps below are only required once, on a new or bricked device that is booting from SPI NOR Flash or soldered down eMMC (that you are not able to boot from SD at all)!

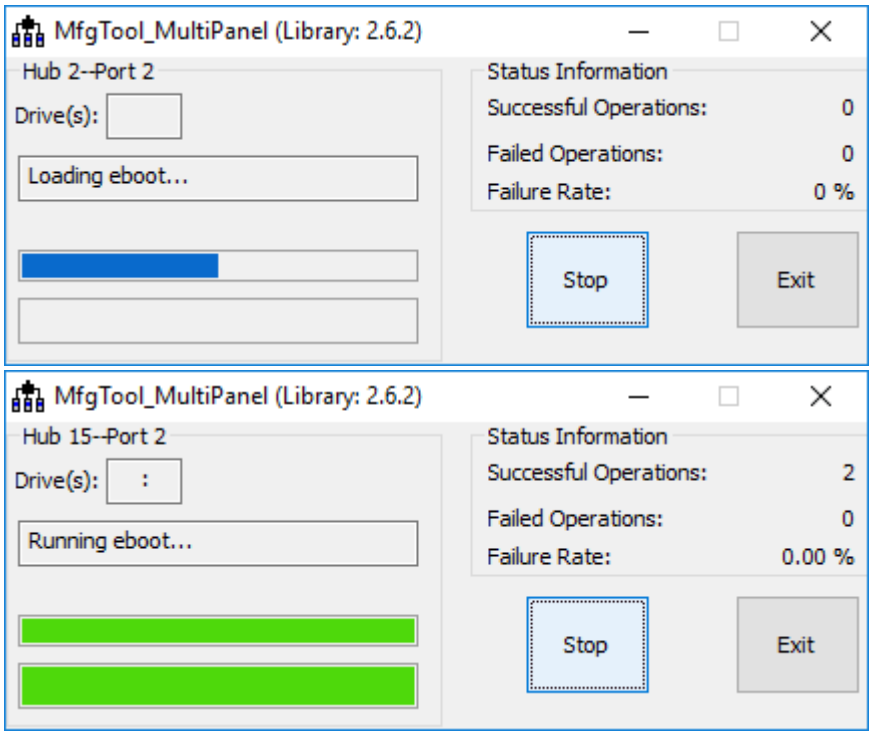
Follow these instructions to get started with your evaluation of the GuruCE i.MX6 evaluation images:

1. Extract the contents of the GuruCE i.MX6 evaluation images 7-zip file (make sure to download the correct package for your board) to a new folder. NOTE: Downloading kernel images created for other hardware than what you intend to run the kernel on will not work properly or at all.
2. Connect a serial cable between the board's serial debug port and your development PC.
3. Open a serial console window (e.g. TeraTerm) and set the serial connection to 8N1, 115k2, no flow control.
4. If you are booting from SD, µSD, eMMC or SATA, you may be able to use our CEWriter tool. This will greatly reduce the number of steps you need to follow, so please check the instructions in APPENDIX A to see if this is possible on your board.
   If you used CEWriter to flash the bootloader and kernel images: insert the µSD or SD card in (or attach the SATA drive to) the board, apply power to the board, press space in the terminal window to break into the bootloader menu and **jump to step 37 below**.
5. Put the board in USB Boot mode.
   See APPENDIX A for detailed instructions for your board. Double-check the board-manufacturer's instructions in case something doesn't work as expected or your hardware is not listed in Appendix A.
6. Connect a USB cable to the client (USB OTG) port of the board and your development PC.
7. Connect an Ethernet cable to the board's LAN port and make sure your development PC is on the same network subnet.
8. Apply power to the board.
   Some boards do not correctly enter USB boot mode when a card is inserted into the SD slot. If your device does not enter USB boot mode: remove any µSD, SD or SDIO cards from the device and power-cycle.
9. Start MfgTool2.exe.

10. A *HID-compliant vendor defined* device should be detected in MfgTool:



11. If you want to flash the bootloader or kernel to SD or µSD card, make sure a card is inserted in the correct device slot **before pressing Start** in the dialog above.
12. Press the Start button. This will load the GuruCE bootloader into RAM and run from it from there (note that nothing is flashed at this time):



13. Exit the NXP Manufacturing Tool by pressing ‘Stop’ and then ‘Exit’.

14. The serial console window should now show the GuruCE i.MX6 bootloader menu (output will vary slightly with different boards):

```
GuruCE Windows Embedded Compact 2013 Bootloader 3.6 for the NXP SDB-QP
Built on [month] [day] [year] at [time] (rXXXX)

Reset reason: power-up or reset button.

Manufacturing Tool bootloader; redirecting to USDHC3
Initializing USDHC3   : OK!

-----------------------------------------------------------------
 GuruCE Bootloader - Main menu
-----------------------------------------------------------------
 [1] Load kernel from..... : Visual Studio
 [2] Number of active cores..... : 4
 [3] Boot menu keypress wait..... : 3
 [W] Hardware watchdog..... : Enabled
 [Q] L2 cache..... : Enabled
 [C] Clean registry & databases.. : Disabled
 [K] KITL..... : Disabled
 [P] KITL passive mode..... : Disabled
 [O] KITL/Download device..... : USB Serial
 [U] Windows CE debug UART..... : UART1
 [G] Display menu..... : Disabled
 [E] Ethernet menu
 [M] SD/MMC menu
 [A] SATA menu
 [H] HAB menu
 [R] Reset menu
 [I] Show i.MX6 info
 [B] Bootloader shell
 [F] Reset to factory defaults
 [D] Download image over..... : USB Serial
 [L] Launch stored kernel from... : SD3
 [S] Save settings

 Selection:
```

15. Enter the Ethernet menu and select ENET1 by pressing [E][1]:

```
Selection: e

-----------------------------------------------------------------
 GuruCE Bootloader - Ethernet menu
-----------------------------------------------------------------
 [1] Configure ENET1
 [2] Configure USB RNDIS
 [X] Return to Main menu

 Selection: 1

-----------------------------------------------------------------
 GuruCE Bootloader - Configure ENET1
-----------------------------------------------------------------
 [1] IP address..... : 192.168.1.201
 [2] Subnet mask..... : 255.255.255.0
 [3] Gateway..... : 192.168.1.1
 [4] MAC address..... : 00:04:9f:04:10:2e (from MAC0 fuses)
 [5] DHCP..... : Disabled
 [C] Copy settings to CE..... : Enabled
 [X] Return to Main menu

 Selection:
```

16. If the MAC address shows all zeroes, configure a MAC address by pressing [4]. In the output above for the NXP SDB-QP board, you can see the MAC address is taken from the fuses, so no need to set a MAC address in that case.
17. If you don't have a DHCP server on the network you'll need to configure a static IP through options 1, 2 and 3 in the Ethernet->ENET1 menu. Make sure to select a free IP in the same subnet as your

development PC. If you have a DHCP server on your network you can enable it through option [5] DHCP.

18. Press [X] until you are back in the Main menu.
19. If booting from SD/MMC, make sure the correct SD slot is activated: press [M], followed by [A] to select the active SD slot.
20. It is best to erase the entire boot medium before flashing our evaluation bootloader and kernel images to make sure no MBR or partition data exists on the medium. To do this, press [E] in the SD/MMC menu followed by 'y' to erase the entire SD/MMC or, if you are booting from SATA or NAND, enter the [A] SATA menu or [N] NAND menu to do the same.
21. Press [X] until you are back in the Main menu.
22. Make sure the Download device is set to ENET1 by pressing "[O] KITL/Download device" until ENET1 is shown.
23. Make sure everything else is setup as you want and press [S] to save the settings.
24. Now open a command prompt on your development machine, browse to the folder you extracted the GuruCE i.MX6 evaluation images and type "**CELOADER CEX\eboot.bin**" (where 'X' is either '7' or '8') followed by ENTER:

```
C:\SDB-QP_r2350>celoader ce7\eboot.bin
**************************************************************************
*                                                                        *
*                           GuruCE CELoader                              *
*       Loading binary images over TFTP to Embedded Compact Device       *
*    syntax: celoader [binfile-name] [target-id] -c(wait for cerdisp)    *
*                                                                        *
*                   Special thanks to Thierry Joubert                    *
*                     https://celoader.codeplex.com/                     *
*                                                                        *
**************************************************************************

CELOADER 1.3.1 - Waiting for BOOTME...
```

25. In the serial console press [D] to download the bootloader image.
    The bootloader menu should show output something very similar to the below (in the listing below DHCP was disabled):

```
Selection: d
Using ENET1 MAC address 00:04:9f:04:10:2e
Found AR8031 PHY (0x004DD074) at address 1
Waiting for Ethernet link...
100 Mbps full duplex link established!
+EbootSendBootmeAndWaitForTftp
Sent BOOTME to 255.255.255.255
Packet has the following data:
  boot.bin[NULL]octet[NULL]
TFTP packet could have 1 name/value pairs
Locked Down Link 1
Src IP 192.168.1.201 Port Dest IP 192.168.1.100 Port Default TFTP block size set to: 512
bytes
There were no options detected in the TFTP
EthDown::TFTPD_OPEN::boot.bin
-EbootSendBootmeAndWaitForTftp
Using device name: "SDB-QP4142"

BL_IMAGE_TYPE_BIN

Downloading EBOOT SD/eMMC image...
rom_offset=0x280c0400.
ImageStart = 0xA8100400, ImageLength = 427908, LaunchAddr = 0xA8102C00

Completed file(s):
--------------------------------------------------------------------------------
[0]: Address=0xA8100400  Length=0x00068784  Name="" Target=FLASH

WARNING:  Flash update requested.
Do you want to continue? [y/n]:
```

26. Press ‘y’ and the bootloader image will be flashed.
27. When flashing is completed you can power off the board and set the board’s boot switches to boot from the medium you just flashed the bootloader to. Refer to APPENDIX A for detailed instructions for your board, or the boards hardware manual if not listed in APPENDIX A.
28. Disconnect the USB cable from the USB client port of the board.
29. Apply power to the board again and immediately press the space bar in the serial console window to break into the bootloader menu.
30. Make sure the bootloader is still setup correctly for download so you can download the kernel image.
31. In the command prompt window on your development PC, you now run CELOADER to upload the kernel image to the device by typing “**CELOADER CEX\nklz4.bin**” (where ‘X’ is either ‘7’ or ‘8’) followed by ENTER:

```
C:\SDP-DL_r2031>celoader ce7\NKlz4.bin
**************************************************************************
*                                                                        *
*                           GuruCE CELoader                              *
*       Loading binary images over TFTP to Embedded Compact Device       *
*    syntax: celoader [binfile-name] [target-id] -c(wait for cerdisp)    *
*                                                                        *
*                   Special thanks to Thierry Joubert                    *
*                     https://celoader.codeplex.com/                     *
*                                                                        *
**************************************************************************

CELOADER 1.3.1 - Waiting for BOOTME...
```

32. In the serial console press [D] to download the kernel image.
33. Once the kernel is downloaded, the bootloader will ask if you want to flash the kernel to the device. Press ‘y’ to proceed.
34. Press any key to reset the device when the flash update is done.
35. Immediately press space to break into the bootloader menu again.
36. Optionally you can flash a boot splash image that will be displayed immediately after power is applied to the board on any of the configured displays. To do this, repeat steps 30 to 35 but exchange the CELoader command with “CELoader logo.bmp”. The bitmap has to be 32bpp and smaller than the display resolution.

37. Press [1] until the correct boot device is shown:

```
Selection: 1

-------------------------------------------------------------
 GuruCE Bootloader - Main menu                 (UNSAVED CHANGES)
-------------------------------------------------------------
 [1] Load kernel from..... : SD/MMC
 [2] Number of active cores..... : 4
 [3] Boot menu keypress wait..... : 3
 [W] Hardware watchdog..... : Enabled
 [Q] L2 cache..... : Enabled
 [C] Clean registry & databases.. : Disabled
 [K] KITL..... : Disabled
 [P] KITL passive mode..... : Disabled
 [O] KITL/Download device..... : ENET1
 [U] Windows CE debug UART..... : UART1
 [G] Display menu..... : Disabled
 [E] Ethernet menu
 [M] SD/MMC menu
 [A] SATA menu
 [H] HAB menu
 [R] Reset menu
 [I] Show i.MX6 info
 [B] Bootloader shell
 [F] Reset to factory defaults
 [D] Download image over..... : ENET1
 [L] Launch stored kernel from... : SD3
 [S] Save settings

 Selection:
```

38. Make sure all settings are correct (don't forget to select the correct display configuration in menu [G]), and press [S] to save the settings.
39. Press [R][R] to reset the board.
40. The kernel should be automatically loaded and your debug output should look similar to the output below (of course RAM size may differ on your board):

```
GuruCE Windows Embedded Compact 7 Bootloader 3.6 for the NXP SDB-QP
Built on [month] [day] [year] at [time] (rXXXX)

Reset reason: power-up or reset button.

Booted from USDHC3!
Initializing USDHC3   : OK!
Reading splash image  : 100%

Press [ENTER] to launch image stored in SD/MMC or [SPACE] to cancel.

Initiating image launch in 0 seconds...
Launching image...
Reading LZ4 compressed kernel: 100%
Decompressing LZ4 compressed kernel image...
Kernel read from SD successfully!
Authenticating kernel image... OK!

Windows CE Kernel for ARM (Thumb Enabled)
GuruCE Windows Embedded Compact 7 for the NXP SDB-QP
Built on May  6 2019 at 11:32:04 (r2350)

 512 MB base memory         (PA: 0x10000000 - 0x30000000)
 384 MB extension memory    (PA: 0x30000000 - 0x48000000)
 128 MB shared video memory (PA: 0x48000000 - 0x50000000)
==========================================================
   1 GB total memory


Welcome to the Windows CE Command Line Interface (CLI)

Pocket CMD v 7.00
\>
```

UPDATING THE KERNEL, BOOTLOADER OR BOOT SPLASH IMAGE

There is no need to follow the lengthy procedure of the previous chapter if you already have the bootloader flashed, configured and running on your device. In that case, simply follow this procedure:

1. Apply power to the board and immediately press the space bar in the serial console window to break into the bootloader menu.
2. Enter the Ethernet menu and select ENET1 by pressing [E][1]:

```
Selection: e

-------------------------------------------------------------
 GuruCE Bootloader - Ethernet menu
-------------------------------------------------------------
 [1] Configure ENET1
 [2] Configure USB RNDIS
 [X] Return to Main menu

 Selection: 1

-------------------------------------------------------------
 GuruCE Bootloader - Configure ENET1
-------------------------------------------------------------
 [1] IP address.................. : 192.168.1.201
 [2] Subnet mask................. : 255.255.255.0
 [3] Gateway..................... : 192.168.1.1
 [4] MAC address................. : 00:04:9f:04:10:2e (from MAC0 fuses)
 [5] DHCP........................ : Disabled
 [C] Copy settings to CE......... : Enabled
 [X] Return to Main menu

 Selection:
```

3. If the MAC address shows all zeroes, configure a MAC address by pressing [4]. In the output above for the NXP SDB-QP board, you can see the MAC address is taken from the fuses, so no need to set a MAC address in that case.
4. If you don't have a DHCP server on the network you'll need to configure a static IP through options 1, 2 and 3 in the Ethernet->ENET1 menu. Make sure to select a free IP in the same subnet as your development PC. If you have a DHCP server on your network you can enable it through option [5] DHCP.
5. Press [X] until you are back in the Main menu.
6. Make sure the Download device is set to ENET1 by pressing "[O] KITL/Download device" until ENET1 is shown.
7. Make sure everything else is setup as you want and press [S] to save the settings.
8. Open a command prompt window on your development PC and browse to the CELoader folder. Run CELOADER to upload the kernel or bootloader image to the device by typing "**CELOADER CEX\nklz4.bin**" or "**CELOADER CEX\eboot.bin**" (where 'X' is either '7' or '8'), or "**CELOADER logo.bmp**" followed by ENTER:

```
C:\SDP-DL_r2031>celoader ce7\NKlz4.bin
**************************************************************************
*                                                                        *
*                          GuruCE CELoader                               *
*       Loading binary images over TFTP to Embedded Compact Device       *
*    syntax: celoader [binfile-name] [target-id] -c(wait for cerdisp)    *
*                                                                        *
*                   Special thanks to Thierry Joubert                    *
*                     https://celoader.codeplex.com/                     *
*                                                                        *
**************************************************************************

CELOADER 1.3.1 - Waiting for BOOTME...
```

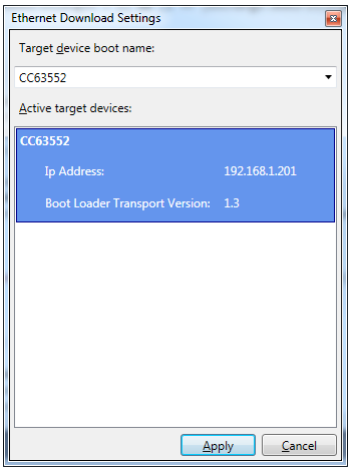
9. In the serial console bootloader menu press [D] to download the kernel image.

10. Once the bootloader, boot splash image or kernel is downloaded, the bootloader will ask if you want to flash it to the device.

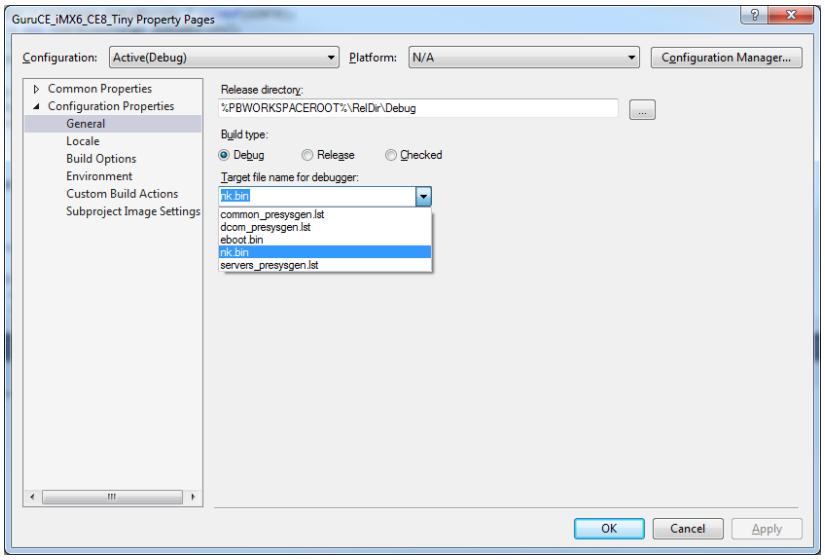
```
WARNING:  Flash update requested.
Do you want to continue (y/n)?
```

    Press 'y' to proceed.
11. Press any key to reset the device when the flash update is done.

Of course, you can also download eboot.bin, nk.bin or nklz4.bin and even logo.bmp directly from within Visual Studio (so you don't need CELoader). In that case simply replace step 2 above with clicking menu "Target -> Attach Device" in Visual Studio.



You can select whether you want to download eboot.bin, nk.bin, nklz4.bin or logo.bmp in the OSDesign properties window under "General".

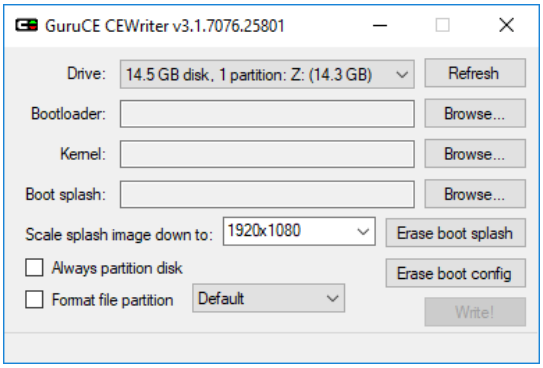


Using Visual Studio has another benefit over CELoader; you can use USB Serial as the download medium. This is our preferred and recommended medium for download and the KITL connection, as it needs no additional setup and is super-fast!

To use USB Serial for download and KITL, make sure you have a USB cable connected between the USB client (OTG) port on the device and your PC. Select "UsbSer" as the download device in Visual Studio and in the GuruCE bootloader menu on the device type [O] until it shows "USB Serial". Enable KITL if you want to debug the kernel over KITL, and press [D] to download the kernel, bootloader or splash image from Visual Studio.

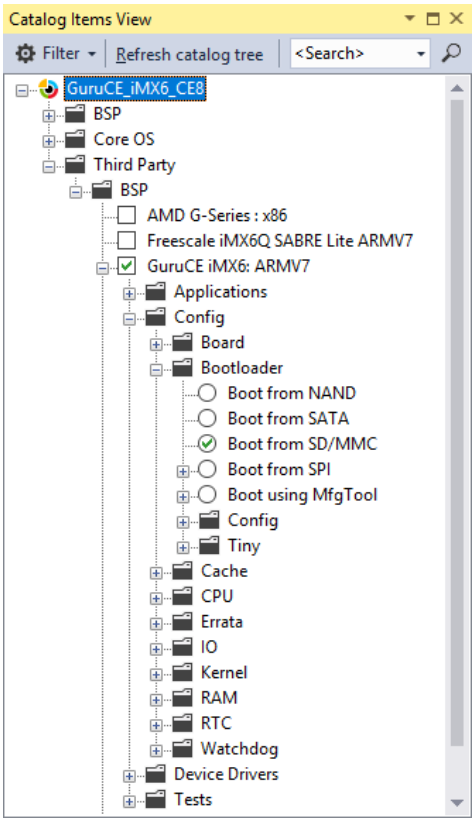
For USB Serial to work for download and kernel debugging over KITL, you need to stop the "Windows Mobile-2003-based device connectivity" (WcesComm) and "Windows Mobile-based device connectivity" (RapiMgr) services on your PC. In the root of the BSP folder (\WINCEX00\PLATFORM\GuruCE_iMX6) you will find a batch file "enable_usb_serial_kitl.cmd" that will do just that. Double click it to stop those services.

USING THE GURUCE CEWRITER



The GuruCE CEWriter tool allows you to quickly and easily flash the bootloader, kernel and any boot splash image to an attached SD card or SATA disk.

CEWriter reserves 136 MB at the very beginning of the SD card or SATA drive: 1 MB for the bootloader, 128 KB for the bootloader settings, 6 MB for an optional boot splash image and 128 MB for the kernel image. The bootloader, bootloader configuration, boot splash image and kernel are all stored in this reserved space. CEWriter will check if there is a 136 (or larger) reserved area at the beginning of the disk and, if not, create a partition starting at an offset of 136 MB spanning the remaining space on the disk.



If you get a popup saying the image you selected is “intended to be uploaded to RAM/SPI/NAND”, it means you are trying to flash a kernel that was built without IMGFLASH set in the build options or that was destined to either SPI or NAND. CEWriter will still allow the image to be written to SD or SATA, but it is better you correctly configure your OS Design to build the right images. Always build a correctly configured Shipbuild configuration (this has WINCESHIP=1 and IMGFLASH=1) or manually set IMGFLASH=1 in the configuration’s build options to create a “flashable” image.

You can specify the maximum resolution for the boot splash image. If the image is larger than the resolution specified, CEWriter will automatically scale the image down (keeping the aspect ratio) to match the maximum resolution. You can select jpg, png or bmp files as boot splash image. CEWriter automatically converts the image to be a single plane, 32bpp bitmap image.

When using CEWriter make sure you select the correct drive from the “Drive” list. This list only shows removable and external drives to reduce the chances of total disaster (imagine selecting one of your installed hard disks!). This means that if you want to flash to a SATA drive you will have to connect the SATA drive through a USB-to-SATA adapter (so it is classified as an ‘external’ drive).

The “Erase boot splash” button erases an existing boot splash image from the SD card or SATA drive.

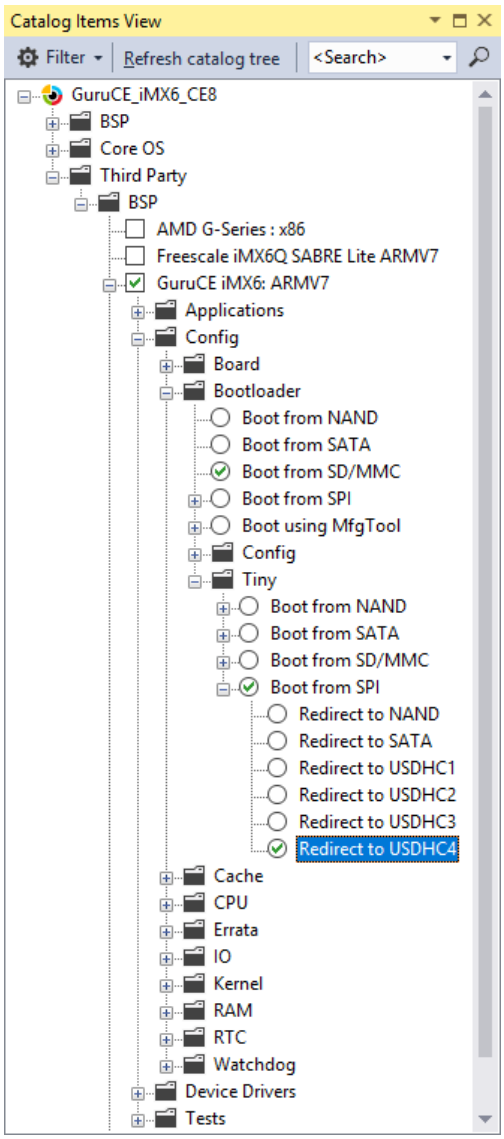
The “Erase boot config” button erases the boot configuration, so the next time you boot using this SD card or SATA drive, the bootloader will revert to factory defaults.

You can check “Always partition disk” to repartition the SD card or SATA drive, even if the reserved space at the beginning of the disk is sufficient to store the bootloader, boot splash image and kernel. Normally you would leave this checkbox unchecked for automatic detection of required partitioning.

CEWriter can be instructed to format the file partition with the FAT32 or exFAT filesystem so that the partition can be used in Windows CE (and on your PC) to store files. Note that if you don’t format the partition, CE will format it at first boot. If you have setup the storage device in the BSP catalog as T(ex)FAT, then you should not instruct CEWriter to format but instead let CE format the partition (because ‘big’ Windows has no capability to format T(ex)FAT).

REDIRECTING SPI FLASH BOOT TO SD, MMC, SATA OR NAND

Some boards (like SabreLite, Nitrogen6X/VM, Conga-QMX6, TS-4900 and Colibri) are hardwired to only allow boot from SPI NOR Flash. This makes updating the bootloader a bit more difficult since you can't just use CEWriter like you can for SD or SATA. At the expense of a slightly longer boot time, you can opt to use a 2-stage boot: If you flash our TinyBoot bootloader to the SPI NOR Flash it will boot from SPI NOR Flash and immediately reset the i.MX6 instructing it to boot from another medium (like SD, eMMC, SATA or NAND). Our BSP Catalog allows easy selection of which medium to redirect to.

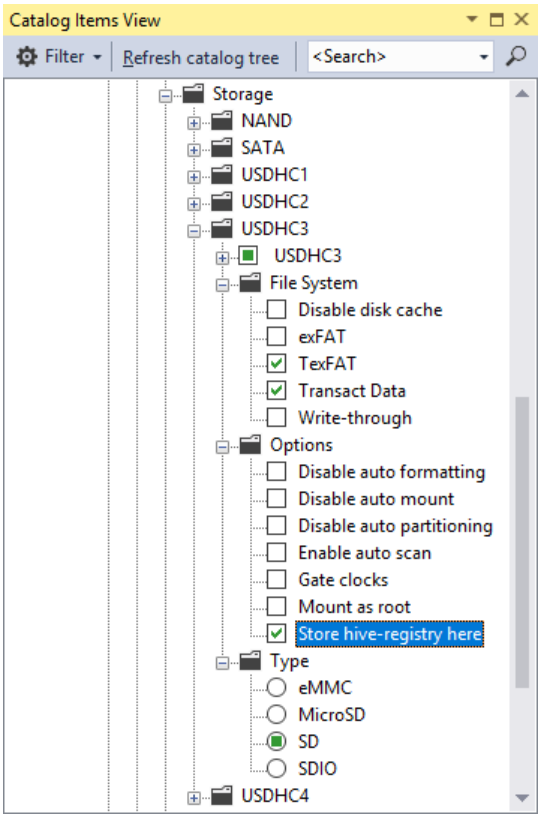


EVALUATION KERNEL COMPONENTS

All evaluation kernels include our Command Line Interface component that redirects Pocket CMD (cmd.exe) over the serial debug port. The kernels also include the network components and an FTP and Telnet server with anonymous access, so you can also connect to the device over the network. To find the IP address of the device, type 'ipconfig' in the debug terminal. The images also include the .NET Compact Framework (v3.5 on WEC7 and v3.9 on WEC2013) and start up the Connection Manager automatically so you can develop and debug native or managed applications using Visual Studio; see below for detailed instructions. Audio, USB HID (Mouse and Keyboard) and USB Storage are also included, as well as the appropriate USDHC port drivers (SD, MicroSD and/or eMMC), UART, SATA, GPIO, GPT, PWM, FlexCAN, I2C, ECSPI, TempMon, PCIe, FTDI Virtual COM USB, USB Function, OpenGL-ES, OpenVG, OpenCL (on DQ/DQP), GPU tutorials, Silverlight, Touch and CE Remote Display (CERDISP). If you have the USB client port connected to your PC, the device will present itself as a USB Serial device (Mobile Devices -> Microsoft USB Sync) so you can connect to the device with the Windows Mobile Device Center application on your PC. For more information see "Applications and utilities" **Error! Reference source not found.**.

PERSISTENT REGISTRY

Our evaluation kernels do not persist the registry and do not mount any medium as persisted root. However, configuring hive-based persistent registry or which medium to mount as root is super easy using our BSP. It's simply a matter of selecting the "Store hive-registry here" or "Mount as root" checkbox in the catalog under the medium you want to store the registry hives on or that you want to mount as root. No difficult registry setup required, all required components will be included automatically and the registry is automatically configured to enable the functionality you require.



FILE SYSTEM OPTIONS

The Storage catalog items (NAND, SATA, USDHCn) allow for easy selection of file system type and various options, so you can, for instance, set USDHC3 to be formatted with TexFAT and all data transacted, without the need for any complex and error-prone registry configuration.

BOOTSPLASH IMAGE

The GuruCE i.MX6 BSP fully supports bootsplash images in the system reserved area (where eboot and NK are stored). If a bootsplash image can't be found in the system partition, the bootloader will try to mount the first FAT12/16/32 partition (exFAT is NOT supported) on the kernel boot medium (SD, MMC or SATA) and will try to find and load "logo.bmp" from the root folder.

The bootloader will center and display the bootsplash image on all active displays with a background color determined by the color of the bottom right pixel of the boot splash image. The bootsplash image bitmap needs to be a single plane, 24 bpp bitmap (if you are using CEWriter then any image you select will be automatically converted to the correct format).

TESTING OPENGL-ES, OPENVG & OPENCL

The BSP fully supports 2D, 3D and vector hardware acceleration using the latest VeriSilicon drivers for the GC2000 (OpenGL-ES 1.1, 2.0 & 3.0, OpenCL), GC355 (OpenVG 1.1) & GC320 (2D) GPUs on i.MX6 Dual and Quad, and the GC880 (OpenGL-ES 1.1, 2.0 & 3.0) and GC320 (2D) on the Solo and DualLite. All of the VeriSilicon supplied OpenGL-ES, OpenVG and OpenCL source code examples are included in the BSP and can be included in the kernel image by selecting "GPU Example Code" in the catalog. Note that OpenGL/VG/CL is not available on the UL/ULL.

OPENGL-ES 1.1

- **Rotating Three Color Triangle: es11_tut1**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut1 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Rotating Six Color Cube: es11_tut2**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut2 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Rotating Multi-Textured Cube: es11_tut3**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut3 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Lighting and Fog: es11_tut4**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut4 –w 1920 –h 1080
  Press SPACE to pause, ENTER to add green fog and ESC to exit.
- **Blending and Bit-mapped Fonts: es11_tut5**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut5 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Particles Using Point Sprites: es11_tut6**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut6 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Vertex Buffer Objects: es11_tut7**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es11_tut7 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Overlay: es11_overlay**
  Press ESC to exit.
- **DDB pixmap: es11_pixmapddb**
  Use the mouse to close the window.
- **DIB pixmap: es11_pixmapdib**
  Use the mouse to close the window.
- **Native DDB pixmap: es11_npixmapddb**
  Use the mouse to close the window.
- **Native DIB pixmap: es11_npixmapdib**
  Use the mouse to close the window.

OPENGL-ES 2.0

- **Rotating Three Color Triangle: es20_tut1**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es20_tut1 –w 1920 –h 1080
  Press SPACE to pause, ENTER to change between ORTHO and PERSPECTIVE projection and ESC to exit.
- **Rotating Six Color Cube: es20_tut2**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es20_tut2 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Rotating Reflecting Ball: es20_tut3**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es20_tut3 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.
- **Rotating Refracting Ball: es20_tut4**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: es20_tut4 –w 1920 –h 1080
  Press SPACE to pause and ESC to exit.

OPENVG

NOTE: OpenVG acceleration is handled by the GC355 on the i.MX6 Dual and Quad, and by the GC880 on the i.MX6 Solo and DualLite.

- **Spinning & Scaling Tiger: vg_tiger**
  Without command line options this runs in a window. To run full-screen specify the screen resolution as follows: vg_tiger –w 1920 –h 1080
  Press ESC to exit.

OPENCL

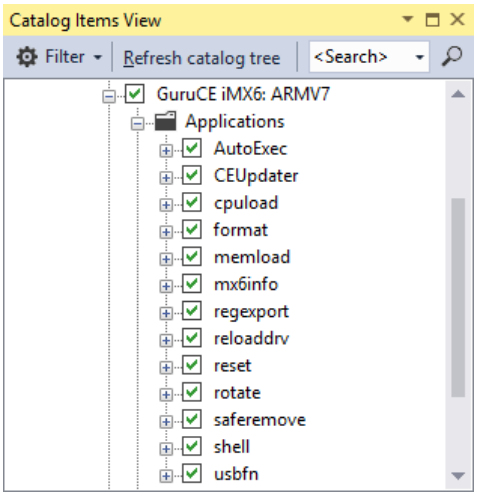
NOTE: OpenCL is not supported on the i.MX6 Solo and DualLite.

- **Show OpenCL info: cl_info**
  No command line options.
- **FFT calculation: cl_fft**
  Specify command line option “fft_length” >= 16 and <= 65536.
- **Threadwalker: cl_threadwalker**
  Specify command line option “test_case” >= 0 and <= 5. Default = 3.
- **Math: cl_math**
  Specify command line option “test_case” >= 0 and <= 25. Default = run all tests.
- **Load/Store: cl_loadstore**
  Specify command line option “test_case” >= 0 and <= 25. Default = run all tests.

APPLICATIONS AND UTILITIES

We've included many handy applications in our BSP, OS Designs and evaluation kernels.

All of these applications come with full source code and can be easily included in your kernel image simply by checking a box in the BSP catalog.



AUTOEXEC

This application allows you to run any type of application at boot. You can configure the AutoExec component through the registry. Configuration options include:

- Wait4Mount
  Wait for a medium to be mounted and fully available before starting your application.
- Wait4Net
  Wait for a valid IP and working network stack before starting your application.
- Wait4WMGR
  Wait until all the window manager is ready and available (needed for running .NET Compact Framework applications) before running your application.
- Kill
  Kill one or more running processes before running your application.
- Run
  Run one or more programs at boot
- Fallback
  This key has values for Kill and Run as well. If running any application from the Run key failed, fall back to killing or running applications under this key.

CETOUCHVIEW

You can use this application to test multi-touch.

Example: cetouchview (then use the application dialogs to setup behavior)

CEUPDATER

This application and SDK library can be used to update kernel, bootloader and splash image to an SD, eMMC or SATA disk.

Example: ceupdater (then follow on-screen instructions)

CPULOAD

Show the load distributed over the number of cores.

Example: cpuload

FORMAT

Format (and partition if needed) any store using the specified filesystem and options.

Examples:

Show usage	: format -?
Show information on all disks in the system	: format -i
Quick format "USBDisk" with default FAT32 filesystem	: format USBDisk
Full format "eMMC" with TexFAT filesystem	: format eMMC4 -texfat -full
Securely wipe & quick format "USDHC2" with the exFAT filesystem	: format USDHC2 -wipe -exfat
Delete “Part00” on USDHC1	: format -s:”USDHC1” -p:”Part00” -del
Delete the partition mounted as "SDC"	: format SDC -del

MEMLOAD

Show memory load information. Pass parameter “-a” to include virtual memory information.

Example: memload -a

MX6INFO

Show iMX6 hardware information (like processor type, speed grade, chip revision, amount of RAM, etc).

Example: mx6info

REGEXPORT

This application can be used to inspect the registry from the command line.

Examples:

Show usage and examples	: regexport -?
Export entire HKEY_LOCAL_MACHINE to console	: regexport
Export HKLM\Drivers\BuiltIn to console	: regexport Drivers\BuiltIn
Export HKLM\Drivers\BuiltIn to console	: regexport HKLM\Drivers\BuiltIn
Export HKCU\ControlPanel to console	: regexport HKCU\ControlPanel
Export HKCR\CLSID to file \Release\hkcr.txt	: regexport HKCR\CLSID -f\Release\hkcr.txt
Export HKU to file \users.txt and to console	: regexport HKU -f\users.txt -c

RELOADDRV

A tool that allows you to unload or (re)load a driver at runtime. See https://guruce.com/blogpost/load-unload-or-reload-a-driver-at-runtime for more information.

Examples:

Unload GPT1	: reloaddrv -u GPT1:
List all loaded device drivers	: reloaddrv -l
Unload (if loaded) and (re)load WAV1:	: reloaddrv WAV1:

RESET

A simple application that allows you to software reset the device from the command prompt (over CLI, telnet or on the device itself).

Example: reset

ROTATE

Rotates the screen. Parameter can be 0, 90, 180 or 270, eg “rotate 90” for 90 degree counterclockwise rotation of the display. Note that screen rotation is not supported in multi-monitor configurations.

Examples:

Rotate 90 degrees : rotate 90
Rotate 180 degrees : rotate 180
Rotate 270 degrees : rotate 270
Restore normal display : rotate 0

SAFEREMOVE

A simple application that allows you to safe remove a removable storage (like a USB key or SD card). The application simply unmounts the store from CE.

Example: saferemove USBDisk

SHELL

The Windows CE Target Control Shel (CESH). Type “shell –c ?” for a list of available commands.

Examples:

Show a list of running processes : shell –c gi proc
Kill the process with ID 7 : shell –c kp 7
Show a list of loaded device drivers : shell –c dev

USBFN

usbfn [<c|d> [q|m|r|s]]
USB Function: c = Current, d = Default
Operand:
- q Query
- m Set Mass Storage Device
- r Set RNDIS (Network Adapter) Device
- s Set Serial (USB Sync) Device

If no operand is specified, Current|Default USB Function will be disabled
If no arguments, Current USB Function will be set to Default

Example: usbfn c m
The example above will expose an inserted USB Flash Disk as mass storage device to the connected PC.

CLI

This is our Command Line Interface component. It redirects the command line to a configurable serial port. The evaluation kernels always use the debug port, so that you can easily access the device's command prompt over serial. Very handy for debug and test! Note that you should never select the debug UART port for CLI for kernels that output debug messages (so any kernel that was built without WINCESHIP set), as the debug messages will collide with CLI's use of the UART causing many problems!

IPCONFIG, PING, ROUTE, NETSTAT, TRACERT, NET, IPV6, NDISCONFIG

Network utility tools, working roughly the same as on 'big' Windows.

FTP

The evaluation kernels all run an FTP server. You can login using username "anonymous" and any characters as password. If you don't know the IP address of the device, type "ipconfig" on the CLI prompt or configure a static IP in the bootloader (make sure that bootloader option "[4] DHCP" is set to Disabled and option "[5] Copy network settings to CE" is Enabled). Type 'bye' to exit.

Example: ftp 192.168.1.201

TELNET

The evaluation kernels all run a Telnet server. If you don't know the IP address of the device, type "ipconfig" on the CLI prompt or configure a static IP in the bootloader (make sure that bootloader option "[4] DHCP" is set to Disabled and option "[5] Copy network settings to CE" is Enabled). Type 'exit' to exit.

Example: telnet 192.168.1.201

PWMAPP

Control PWM ports.

Examples:

pwmapp 1	Toggle state of PWM1: (run with same params as last run)
pwmapp 2 5000	Enable PWM2: for 5 seconds
pwmapp 3 20000 10	Set PWM3: output to 20 kHz at 10% duty
pwmapp 4 4000 50 10000	Set PWM4: output to 4 kHz at 50% duty for 10 seconds
pwmapp 5 15000 cycle	Set PWM5: output to 15 kHz and cycle the duty
pwmapp 6 8000000 cycle 5000	Set PWM6: output to 8 MHz and cycle the duty for 5 seconds

NETIOCE

NETIO - Network Throughput Benchmark, Version 1.32
(C) 1997-2012 Kai Uwe Rommel

Usage: netio [options] [<server>]

-s	run server side of benchmark (otherwise run client)
-b <size>[k]	use this block size (otherwise run with 1,2,4,8,16 and 32k)
-B -K -M -G	force number formatting to Bytes, K, M or G Bytes
-t	use TCP protocol for benchmark
-u	use UDP protocol for benchmark
-h <addr/name>	bind TCP and UDP sockets to this local host address/name defaults to all (server) or unspecified (client)
-p <port>	bind TCP and UDP servers to this port (default is 18767)
<server>	If the client side of the benchmark is running, a server name or address is required.

The server side can run either TCP (-t) or UDP (-u) protocol or both (default, if neither -t or -u is specified). The client runs one of these protocols only (must specify -t or -u).

Example: netioce -s
The example above will start the NETIO server on the device. Download NETIO for your PC here.

CERDISP/CERHOST

CE Remote Display (CERDISP) is started automatically on the device at boot (if included) and will start broadcasting to any listening CE Remote Host (CERHOST) applications. Start cerhost.exe or cerhostm.exe on your PC and Connect to any listed device to see and control the CE desktop from your PC. Cerhostm.exe is the managed version of cerhost and offers a bit more functionality than the native version. Performance of the managed cerhost is very similar to that of the native cerhost, and of course both versions come with full source code included.

APPLICATION DEVELOPMENT

When the kernel image is running on the device, you can start developing your application. The first thing you need to do is install the SDK for the correct OS version (either Windows Embedded Compact 7[2] or Windows Embedded Compact 2013[3]).

The following sections describe the steps to install the SDK and how to create a simple "HelloWorld" application.

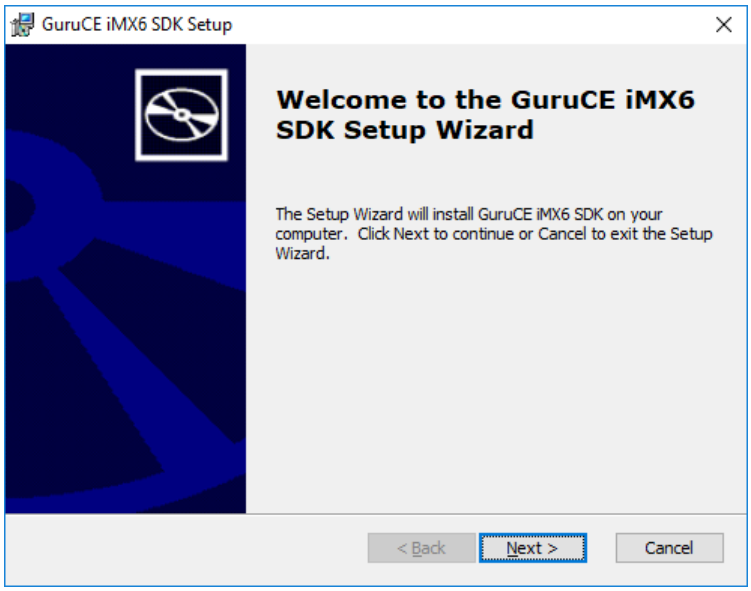
SDK INSTALLATION

Any developer wishing to write applications for your device will need to install the SDK generated for that specific device. With the SDK, the developer can create applications for the device without the need for the Windows Embedded Compact Platform Builder plugin. For standalone application development you need (in this order):

1. Visual Studio 2008 (if you target WEC7) or Visual Studio 2012/2013/2015 (if you target WEC2013), and make sure to get all the latest updates for Visual Studio.
2. Only if targeting WEC2013; Application Builder for Visual Studio 2012/2013/2015.
3. The SDK generated for the device and for the correct OS version. In our case this would be either GuruCE_iMX6_SDK_CE7.msi or GuruCE_iMX6_SDK_CE8.msi.

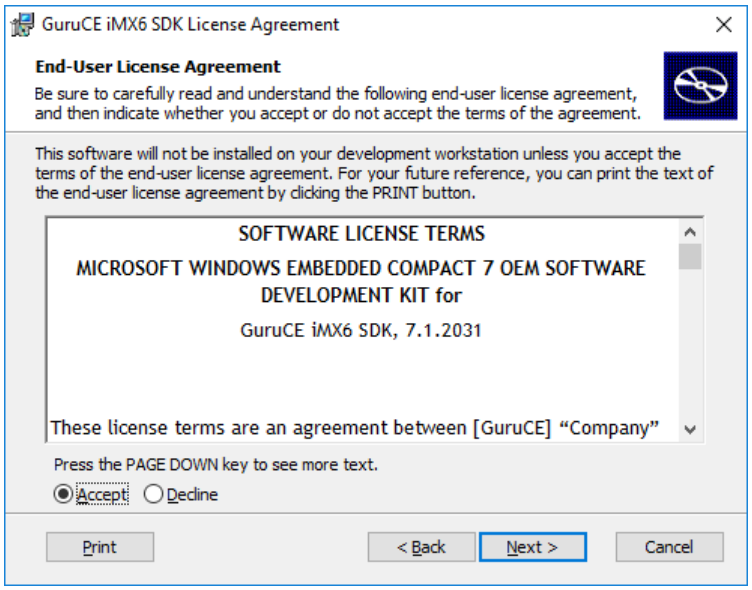
In this example we will guide you through the steps needed to install the SDK that will allow you to create applications targeting Windows Embedded Compact 7. The procedure for WEC2013 is similar.

1. Close any instance of Visual Studio.
2. Double click the GuruCE_iMX6_SDK_CE7.msi. The installation setup welcome dialog will appear:

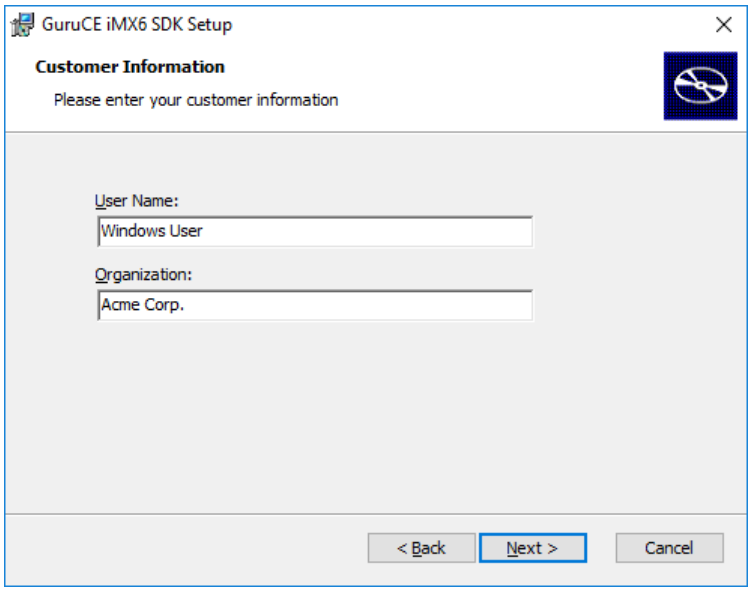


3. Click "Next".

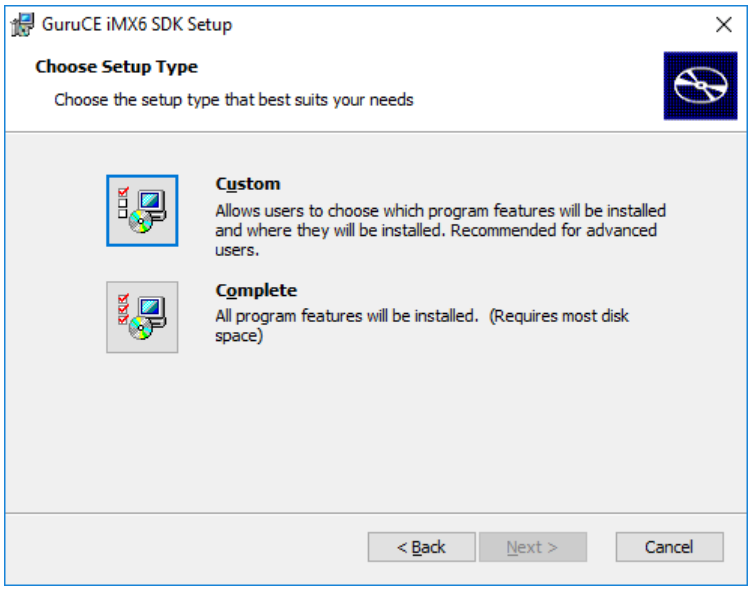
[2] In this document we use CE7 and/or WEC7 to indicate Windows Embedded Compact 7
[3] In this document we use CE8, WEC8 and/or WEC2013 to indicate Windows Embedded Compact 2013



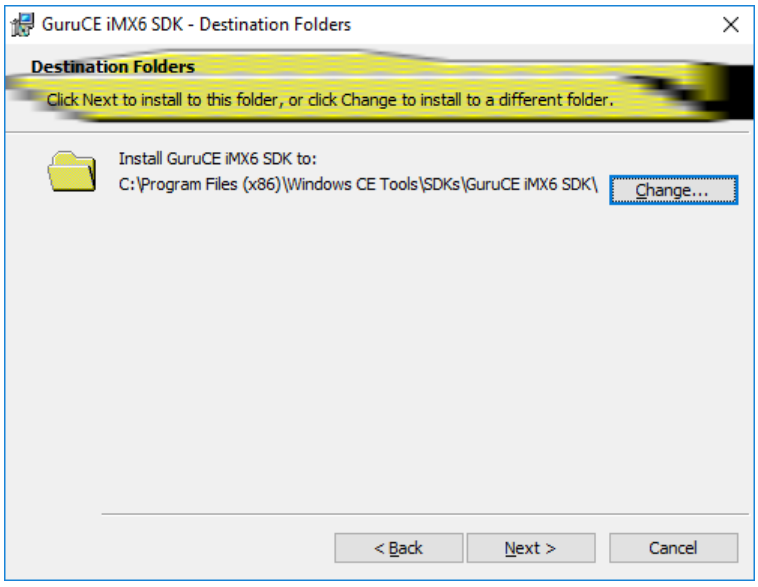
4. To continue installation you must accept the license terms, and click "Next".



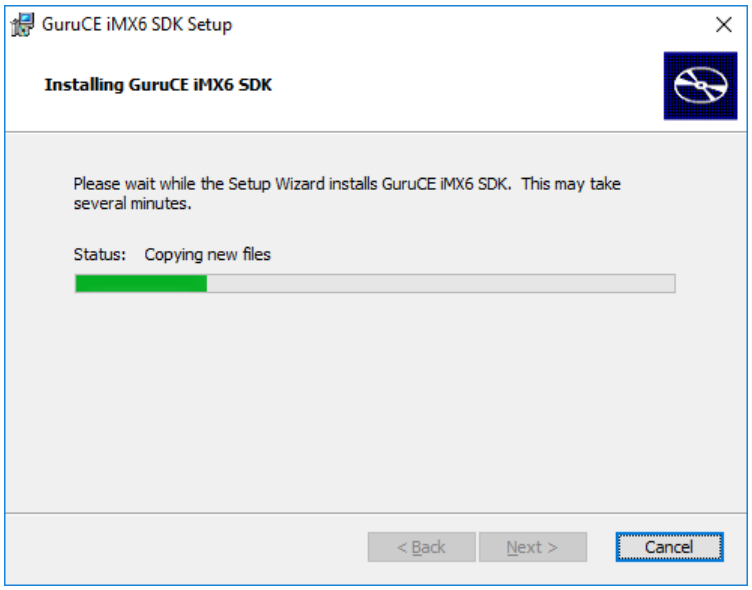
5. Enter the user name and organization (optional) and click "Next".



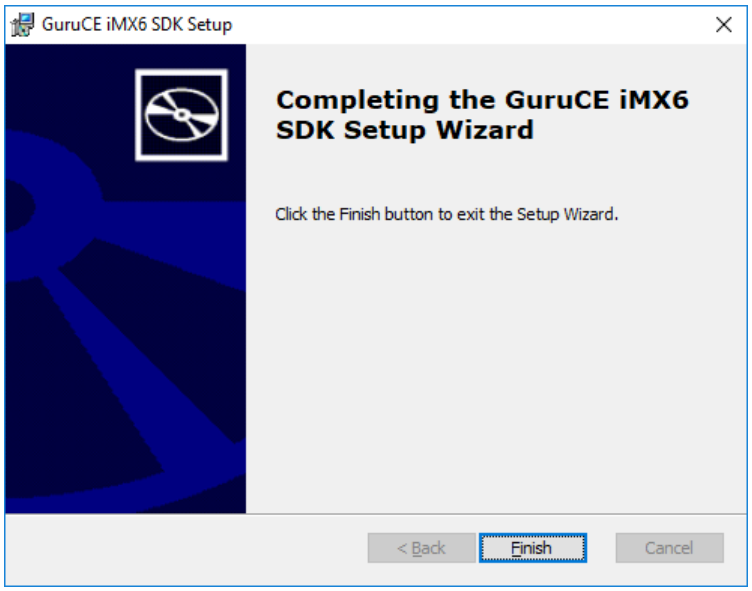
6. Click Complete and configure where you want to install the SDK to. Usually the default location is fine:



7. Select Next, then Install to start the installation.



8. Click “Finish” to complete the installation:

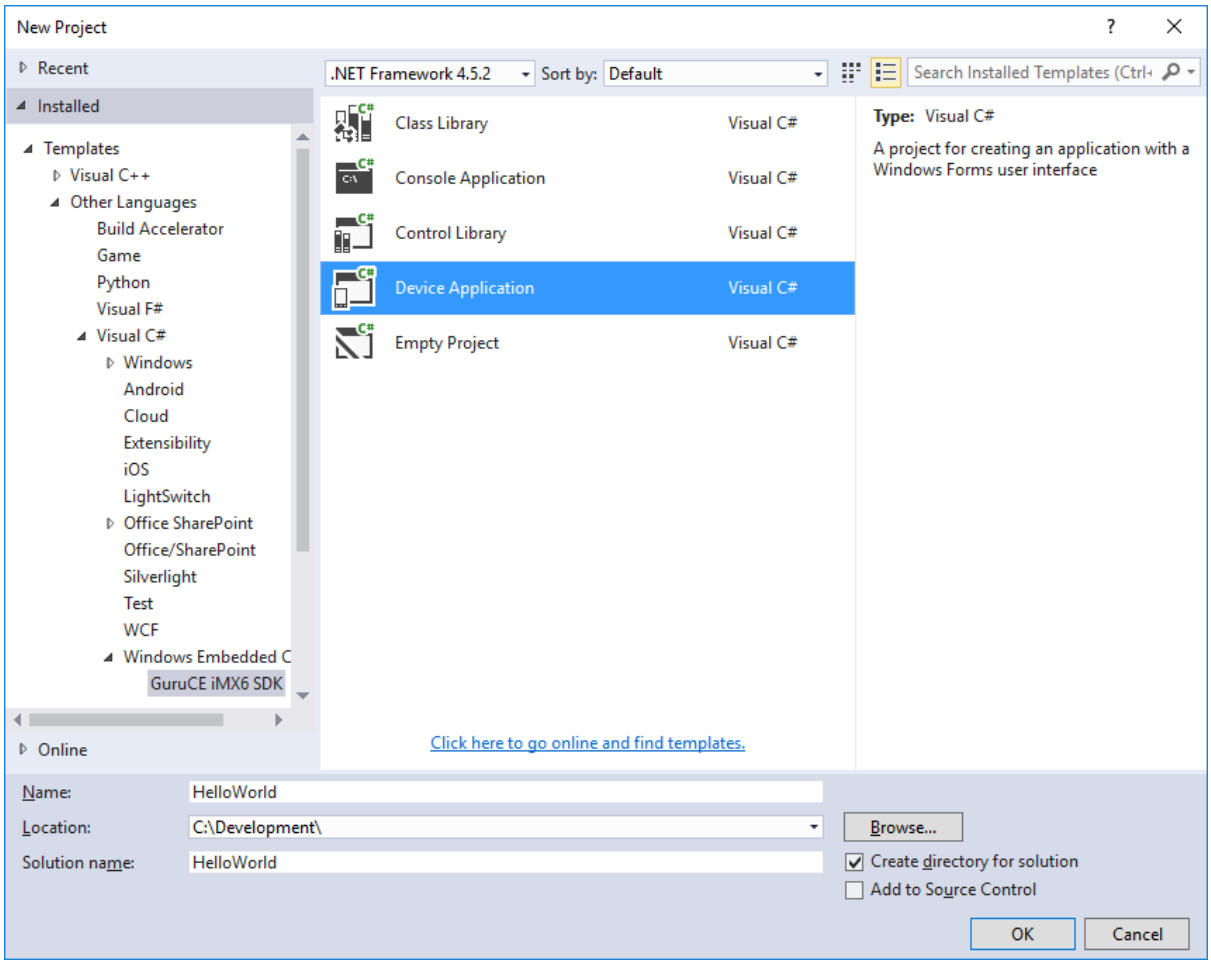


CREATE AND START A MANAGED C# "HELLOWORLD" APPLICATION

Once the installation of the SDK has been completed you can start developing your application.

As an example, let's create an application for Windows Embedded Compact 2013 using Visual Studio 2013 (the same steps apply for Windows Embedded Compact 7 using Visual Studio 2012).

1. Open Visual Studio.
2. In the "File" menu, choose "New | Project" (CTRL+SHIFT+N) and the "New Project" dialog will pop-up.
3. In the left panel, in the "Templates" section for "Visual C# | Windows Embedded Compact", select the "GuruCE i.MX6 SDK".

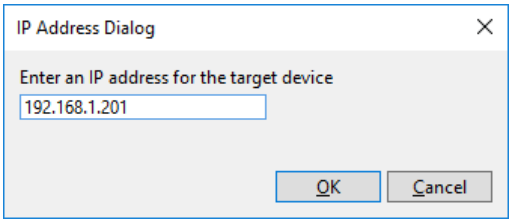


4. Select the "Device Application" template in the right panel and enter "HelloWorld" as the name of the application.
5. Click "OK".
6. Before you can build and deploy the application you need to start the Connection Manager on the target itself to establish the debug connection.
   NOTE: if you are using any of our evaluation images that you downloaded from our website you can jump to step 10 below (the Connection Manager components are automatically started at boot).
7. Use the serial terminal Command Line Interface or open a telnet session to your device (open a command prompt and enter "telnet [device IP address]"; to be able to do this you must know the IP address of the device).
8. In the CLI terminal or the telnet session enter "start conmanclient3" and press "enter". For Embedded Compact 7 the command is "start conmanclient**2**".

```
Welcome to the Windows CE Command Line Interface (CLI)

Pocket CMD v 8.00
\> start conmanclient3
```

9. Switch back to Visual Studio.
10. In Visual Studio, press “F5” to debug our application. The following dialog will pop up:



11. Enter the IP address of the device and press “OK”.
12. The Debug connection will be established and the application is deployed.
13. You should see an empty form on your device’s display:

14. You can now set breakpoints, step debug your code, watch variables, etc.

OS DEVELOPMENT

The GuruCE i.MX6 BSP can be used to create a custom Windows Embedded Compact 7 or Windows Embedded Compact 2013 kernel that can run on your board containing an NXP i.MX6 UltraLight, ULL, Solo, DualLite, Dual, DualPlus, Quad or QuadPlus processor. The BSP is the software layer between the hardware and the operating system. It contains all the low-level initialization code, drivers and configuration settings to allow the OS to communicate with the underlying hardware in a transparent manner.

PREREQUISITES

To be able to create custom images you need to install the Windows Embedded Compact development environment for either WEC7 or WEC2013. For both OS versions we have listed the components you need to install and the order you need to install them in:

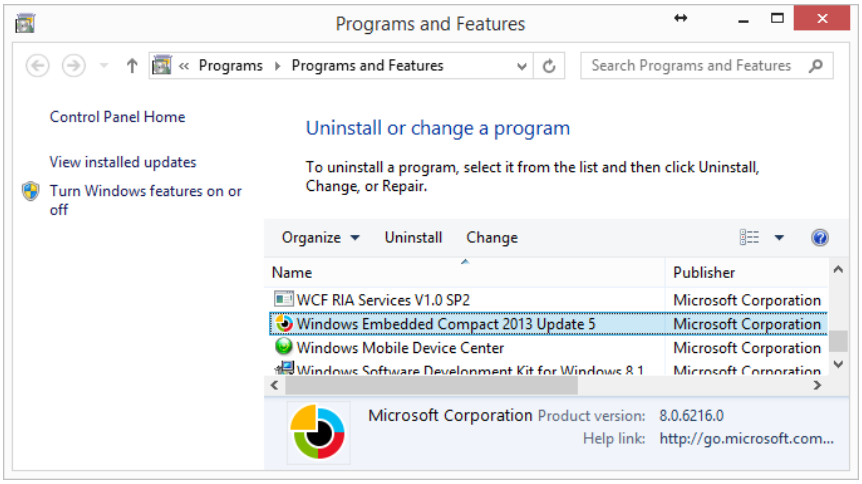
Windows Embedded Compact 7

1. Visual Studio 2008 Professional
2. Visual Studio 2008 Service Pack 1 (SP1)
3. Expression Blend 3 (Optional)
4. Windows Embedded Silverlight Tools (Optional)
5. Windows Embedded Compact 7
6. Visual Studio 2008 update for Windows Embedded Compact 7
7. Updates for WEC7*

Windows Embedded Compact 2013

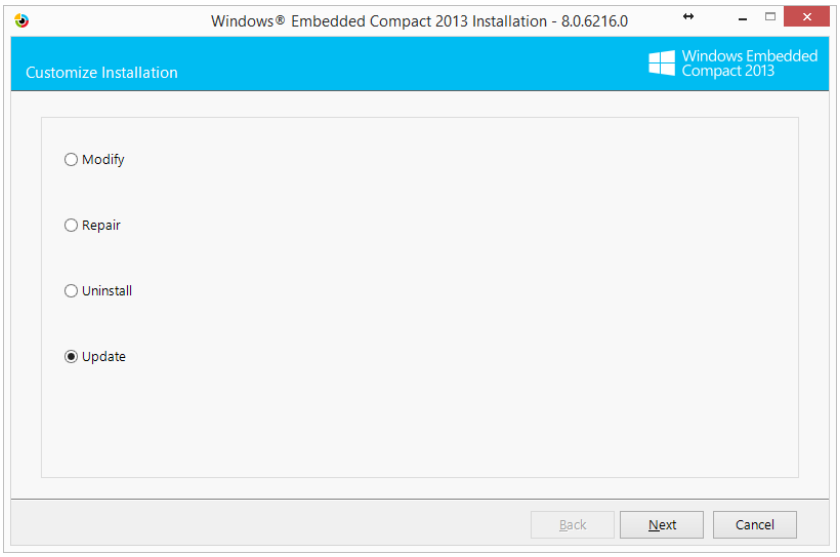
1. Visual Studio 2012/2013/2015
2. Visual Studio 2012/2013/2015 updates
3. Application Builder for Visual Studio 2012/2013/2015
4. Windows Embedded Compact 2013
5. Updates for WEC2013*

* The last step “Updates for WECx” is handled via WEDU[4]. For both environments (WEC7 and WEC8) you will need to go to the “Programs and Features” on your development machine and locate “Windows Embedded Compact”. If you want to install WEC2013 for use inside Visual Studio 2015 you need to start the installation at the September 2015 update (Wave2 update). The September 2015 Wave2 update for WEC2013 is a full installation, so none of the older updates are required.



[4] The WEDU update process is notoriously flaky. If you encounter problems the best way to resolve this is register on the Microsoft Device Partner site and request “Indirect Embedded/IoT OEM" permissions so you can download ISOs of all WEC updates.

In the “Programs and Features” dialog select option “Change” and the Windows Embedded Compact installation dialog will show up. Select “Update” and complete the process to update your environment to the latest.



Select “Next” and follow the update procedure. You will need the latest ISO file of the Windows Embedded Compact installation DVD for updating shared source. This is a selectable option during the installation wizard of Windows Embedded Compact 2013. The ISO is needed because the update procedure does not allow you to download the shared source code when there are updates available. If the update fails because it cannot find the shared source, contact your distributor or check the Microsoft Device Partner (see footnote 4 above) or MSDN subscriptions resource locations to retrieve the latest version of the ISO. If you did not select shared source when you initially installed WEC you should not encounter any issues with shared source availability.

BSP AND OS DESIGNS INSTALLATION

After you install the WEC development environment you can install the GuruCE BSP and the example OS Designs. The GuruCE package contains the BSP and multiple example OSDesigns as a plain “files and folders” structure, so you just have to copy the files and folders into the right place. We have chosen this simple format in favor of an automated installer to give you absolute control over how & where you want to install our BSP.

GURUCE I.MX6 BSP PACKAGE OUTLINE

The GuruCE i.MX6 BSP package delivered by GuruCE contains the following folder structure:

└──WINCEX00.....	CE7 or CE8 root folder.
├──osdesigns.....	OS Designs root folder.
│ ├──GuruCE_iMX6_CE7.....	Full featured WEC7 sample
│ │ └──GuruCE_iMX6_CE7	OS Design.
│ │ ├──ConMGR.....	VS Connection Manager.
│ │ └──SDKs.....	SDK configuration for this
│ │ └──GuruCE_iMX6_SDK_CE7	OS Design.
│ ├──GuruCE_iMX6_CE7_Tiny.....	Tiny WEC7 OS Design for debugging
│ │ └──GuruCE_iMX6_CE7_Tiny.....	isolated issues.
│ ├──GuruCE_iMX6_CE8.....	Full featured WEC2013 sample
│ │ └──GuruCE_iMX6_CE8	OS Design.
│ │ ├──ConMGR.....	VS Connection Manager.
│ │ ├──SDKFiles.....	Extra files to include in the SDK.
│ │ └──SDKs.....	SDK configuration for this
│ │ └──GuruCE_iMX6_SDK_CE8	OS Design.
│ └──GuruCE_iMX6_CE8_Tiny.....	Tiny WEC2013 OS Design for
│ └──GuruCE_iMX6_CE8_Tiny	debugging isolated issues.
└──PLATFORM.....	Platforms root folder.
└──GuruCE_iMX6.....	GuruCE i.MX6 BSP root folder.
├──CATALOG.....	Platform Builder Catalog xml.
│ └──1033.....	Platform Builder Catalog Strings.
├──CESYSGEN.....	CESYSGEN makefile.
├──FILES.....	Some tools, splash logo and
│	platform registry, bib, dat and
│	other configuration files.
├──HABv4.....	High Assurance Boot tools.
└──SRC.....	Source code root folder.
├──APPS.....	Applications & utilities.
├──BOARDS.....	Board specific files.
├──BOOTLOADER.....	Bootloader source.
├──COMMON.....	Bootloader & kernel common code.
├──DRIVERS.....	Driver source code.
├──INC.....	Include files.
├──KITL.....	KITL source code.
├──MS_CLONED.....	Cloned MS component source code.
├──OAL.....	OAL sources code.
└──TEST.....	TEST applications.

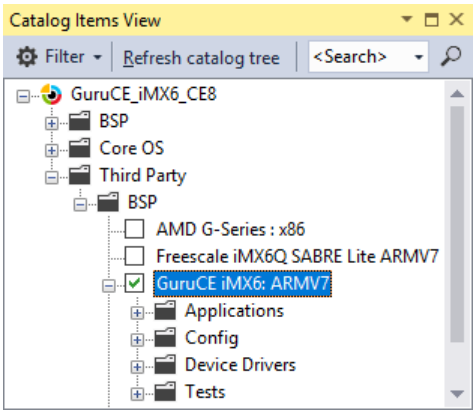
INSTALLING THE GURUCE I.MX6 BSP PACKAGE

You can unpack the entire zip file over your existing WINCEX00 (where 'X' is either '7' or '8') installation folder or choose to manually copy the GuruCE_iMX6 folder into the WINCEX00\PLATFORM folder and the required OS Design files into your OS Design storage location.

GURUCE I.MX6 BSP CATALOG

The GuruCE i.MX6 BSP includes catalog items that allow you to easily configure the BSP to work optimally on your board. The catalog selections take care of most of the dependent main catalog selections as well. The actual catalog XML can be found in WINCEX00\PLATFORM\GuruCE_iMX6\CATALOG.

The GuruCE i.MX6 BSP catalog consists of three main categories:



1. **Applications**
   The items in this category allow you to include handy applications and utilities in your kernel image.
2. **Config**
   The items in this category allow you to configure core BSP settings (like board type, boot source, CPU type, HAB secure boot, silent bootloader, etc.).
3. **Device Drivers**
   The items in this category allow you to select specific device drivers for inclusion in the kernel image.
4. **Tests**
   The items in this category allow you to include test files.

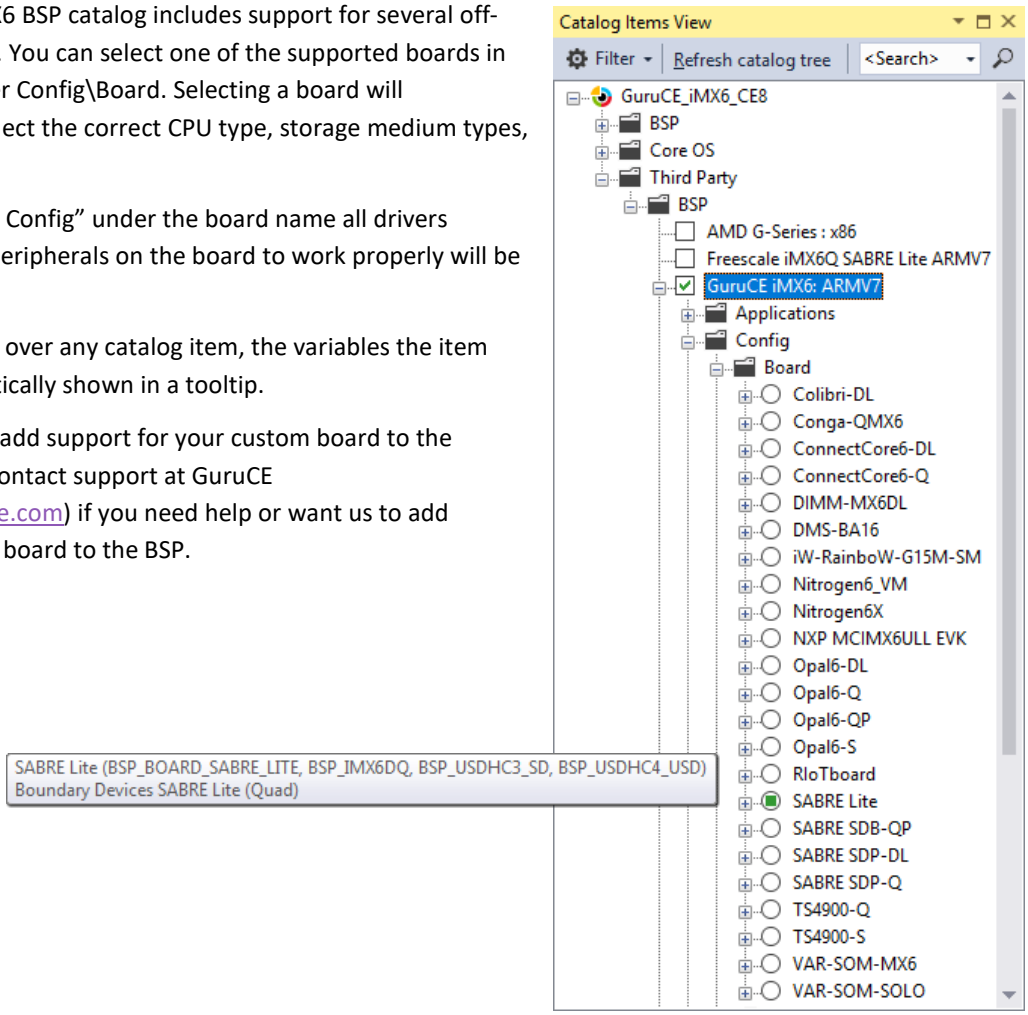
OFF-THE-SHELF BOARD SUPPORT

The GuruCE i.MX6 BSP catalog includes support for several off-the-shelf boards. You can select one of the supported boards in the catalog under Config\Board. Selecting a board will automatically select the correct CPU type, storage medium types, etc.

If you check “Set Config” under the board name all drivers required for all peripherals on the board to work properly will be selected.

When you hover over any catalog item, the variables the item sets are automatically shown in a tooltip.

It is very easy to add support for your custom board to the catalog. Please contact support at GuruCE (support@guruce.com) if you need help or want us to add support for your board to the BSP.



BOOTLOADER

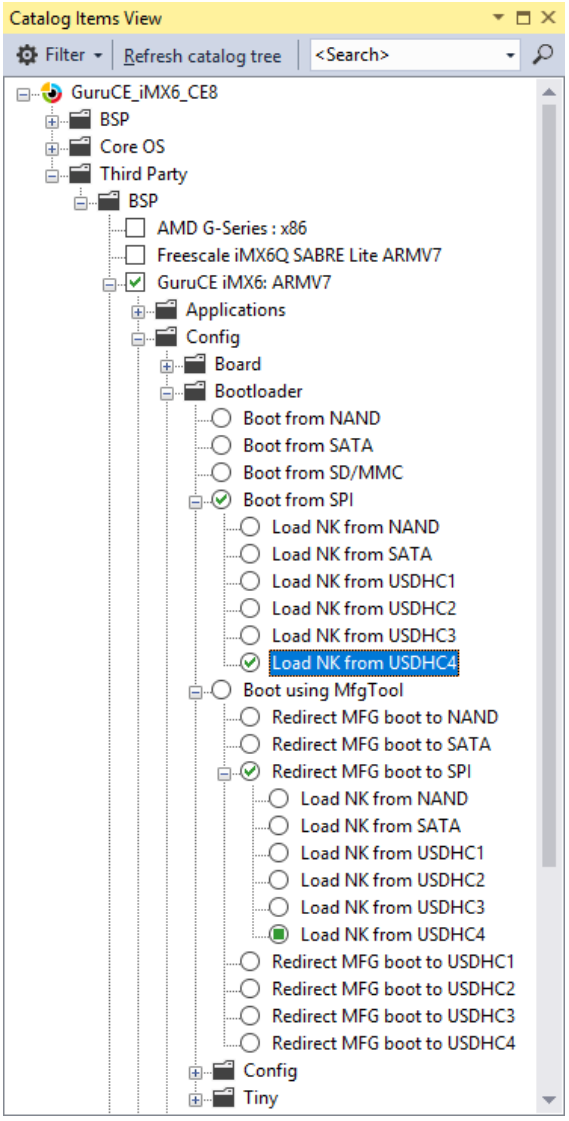
The GuruCE bootloader can be configured to boot from many different boot sources. The bootloader code supports booting from SATA, SD, MMC, NAND and SPI Flash.

Not all boards allow booting from all boot sources. The SABRE Lite design, for instance, is hardwired to boot from SPI Flash only. For a way around this and boot from the medium you want, see “Redirecting SPI Flash boot to SD, MMC, SATA or NAND” above.

If you are booting from SPI Flash there is usually not enough space in the SPI Flash to store the entire CE kernel. Therefore, the catalog has options to redirect the SPI boot to a secondary storage location like SATA, NAND or USDHC.

Most of the components show more information through tooltips if you hover your mouse over it.

You can also create a special bootloader image that can be used together with the NXP Manufacturing Tool. This tool can be used to unbrick a bricked device. Say, for instance, that you flashed a bootloader containing a bug to your board and the bootloader hangs very early in the boot process. You are now unable to re-flash the bootloader since a proper working bootloader is required to flash a new bootloader. In this case, you need to put the board in USB boot mode, build a special bootloader for MfgTool, and use MfgTool to load that bootloader in memory so you can use that bootloader to flash a new bootloader to the board again (detailed instructions in “Flashing the GuruCE i.MX6 evaluation images” above and APPENDIX A below).



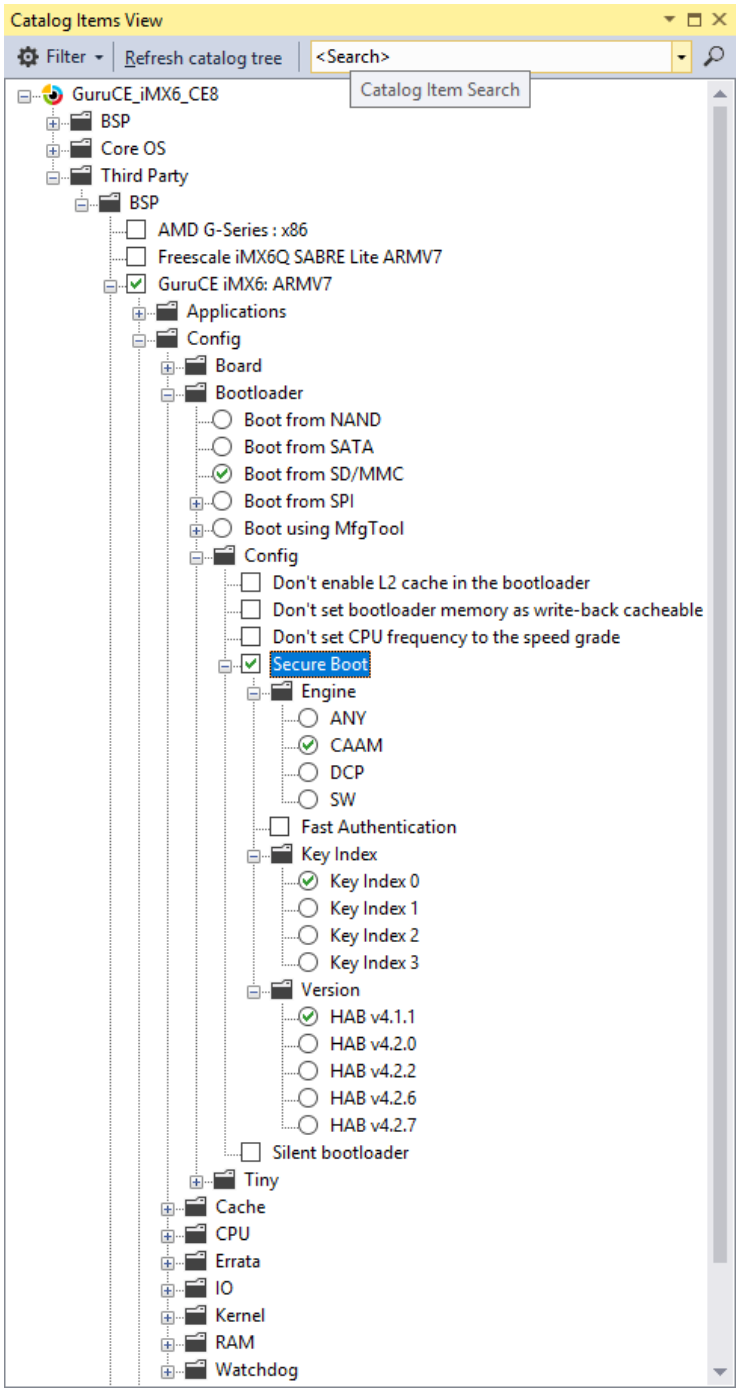
HIGH ASSURANCE BOOT

Executing trusted and authentic code on an applications processor starts with securely booting the device. The i.MX family of applications processors provide this capability with the High Assurance Boot (HAB) component of the on-chip ROM. The ROM is responsible for loading the initial bootloader image from the boot medium. HAB enables the ROM to authenticate the bootloader image by using digital signatures.

Once the bootloader is authenticated it can be executed. HAB provides a mechanism to establish a root of trust for the remaining software components and establishes a secure state on the i.MX IC's secure state machine in hardware. This allows the bootloader to use HAB to authenticate the kernel image and assure only correctly signed kernel images can be executed.

Enabling High Assurance Boot in the GuruCE iMX6 BSP is as easy as following the simple steps in the readme.txt in \WINCEX00\PLATFORM\GuruCE_iMX6\HABv4\ to generate the public and private keys and certificates, and checking the desired HAB functionality boxes in the catalog.

The entire signing process is completely integrated into the build system so, when you build the OS Design, the bootloader and kernel images will automatically be signed and prepared for HAB boot.

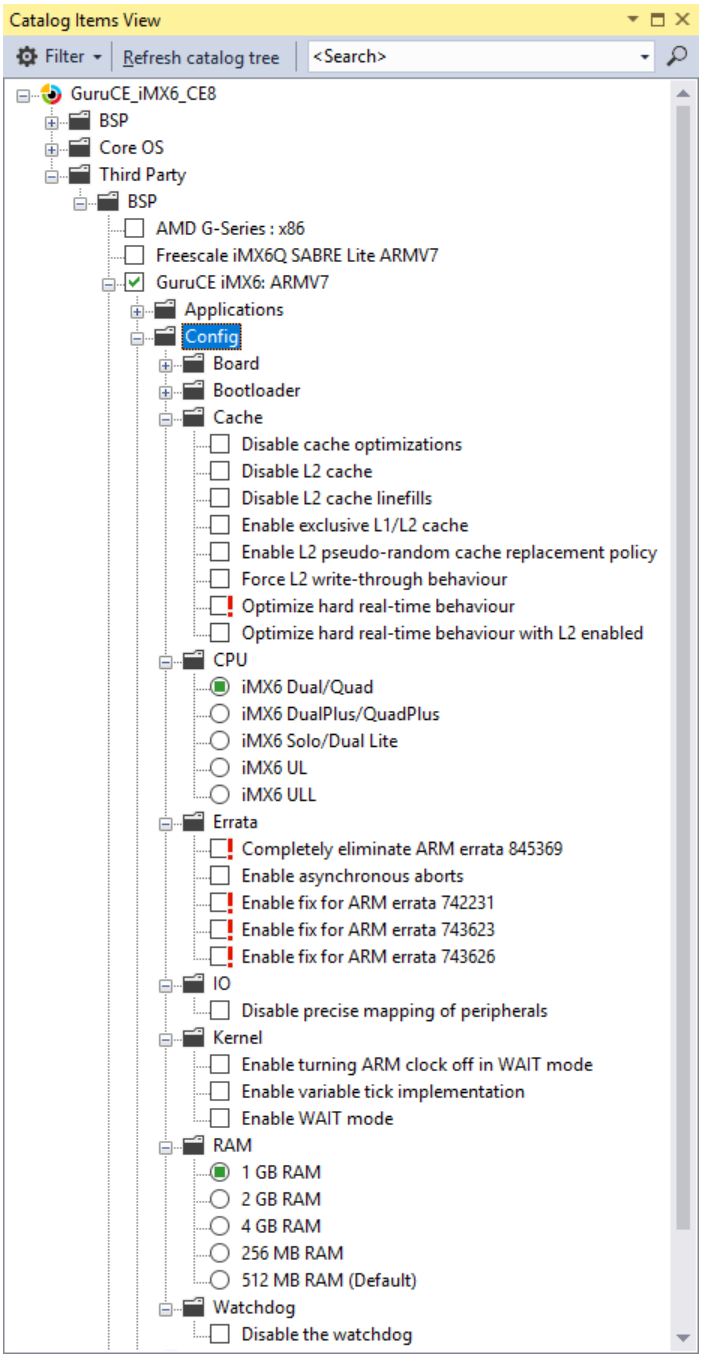


SILENT BOOTLOADER

If you don't want any output over the serial port at boot and want the fastest possible startup time, silent boot is for you. When you select this option, all bootloader output is disabled and the kernel is loaded without any delay. You can still break into the bootloader menu by (repeatedly) sending a password over the serial port while powering up your device. The BSP comes with a utility (Silent Boot Breaker, SBB.exe) demonstrating this, of course with full source included.

OTHER CONFIGURATION OPTIONS

There are several other options in the "Config" category of the GuruCE i.MX6 BSP catalog that allow you fine control over caching, CPU type, optional errata, IO mapping, RAM size, Kernel options and the hardware watchdog.



The bootloader is using L1 and L2 cache to speed up the load-time of the kernel and is setting the CPU frequency to the speed grade of the processor. Some board designs don't allow for a higher frequency than the out-of-reset frequency of 792 MHz (due to power limitations or non-optimal placement of decoupling capacitors for instance). The options under "Bootloader->Config" allow you to disable some of the advanced optimization features in the bootloader, in case you encounter issues on your board.

If you are writing a new device driver and you suspect you may have a problem related to the Cortex-A9 caching mechanisms, you can use the items under "Cache" to disable cache optimizations or even the entire L2 cache. You can also enable exclusive L1/L2 cache and change from a round-robin cache replacement policy to a psuedo-random cache replacement policy for L2 cache. The bootloader can also be used to dynamically disable L2 cache, or set L2 cache to write-through mode (through the bootloader menu, option [Q]).

Performance benefits of the cache options largely depend on the end-user application code. We have created the Config section of the catalog in such a way that, in most cases, the best performance is with all items unchecked.

The items under "Errata" allow you to enable the implementations of some optional errata. We have implemented all required errata and they are always enabled (not configurable in the catalog). The errata listed in the catalog should never occur (due to other errata being implemented) or only occur in very extreme edge cases where the performance degradation of the errata implementation outweighs the risk of the errata ever occurring. We recommend leaving all checkboxes unchecked, unless you suspect you are dealing with an issue that could be related to the specific ARM errata number in the catalog. If you think you found an issue related to any of these specific errata, please let us know.

If you leave “Disable precise mapping of peripherals” under “IO” unchecked, the BSP and bootloader map device IO exactly. This means that if you access any non-existing (reserved) memory in the i.MX6, an exception is generated. This increases device security at the expense of needing more page table entries and incurring a small performance hit because of that. If you check this option, the BSP will map the entire region from 0x0000_0000 to 0x1000_0000 as device memory in one big page table entry, slightly increasing performance at the expense of device security. We recommend keeping this option unchecked.

The options under “RAM” allow you to select the RAM size on the board. Some modules, like the Congatec QMX6 modules, support different revisions that have different sizes of RAM. Use these catalog items to select the correct RAM size used on your board. If the board has a fixed configuration, the correct RAM size will be automatically selected together with the selection of the board.

The options under “Kernel” allow for power reductions. At this moment these options are experimental and not deep tested to work in all situations. We recommend keeping these options unchecked and contact us for more information if you need to reduce power consumption on your device.

The last option allows you to overwrite the bootloader setting for the watchdog (option [9] in the bootloader menu) and disable the hardware watchdog in a Shipbuild configuration. The watchdog is normally only enabled in a Shipbuild configuration (WINCESHIP==1) because having the watchdog enabled while debugging code would cause the watchdog to reset the board shortly after you hit a breakpoint in your code.

We moved the RTC items to the “Device Drivers” section:

The options under “Device Drivers\RTC” allow you to configure how the kernel will use the external RTC (if one is present). If you select “Don’t use external RTC” the BSP will not sync the internal RTC with the external RTC and the internal SOC RTC will be initialized to 1 January 2016, 0:00. The option “Don’t use internal SOC RTC” causes the kernel to only use the external RTC. This is not recommended because it will generate some unnecessary I2C traffic. The best (and default) option is “Use external RTC to sync internal SOC RTC”. This causes the kernel to sync the time of the internal SOC RTC with the external RTC at boot, and sync back to the external RTC any time the device’s system time is updated. Some board designs don’t connect a 32.768 kHz crystal to the i.MX6 RTC XTAL pins. In this case the internal RTC uses the internal ring oscillator. The frequency of this ring oscillator can vary between i.MX6 processors from 14 to 66 kHz. If this is the case on your board (and this is bad hardware design so you should change it) you will have to enable the “Calibrate SOC RTC frequency at boot” checkbox. This will add 1 second to the boot time of your kernel, used to accurately measure and determine the frequency of the internal SOC oscillator. If you don’t check this box when no crystal is connected to the RTC XTAL pins of the i.MX6, the kernel assumes the internal ring oscillator runs at 32 kHz. In most cases this will not be true, so time will run either too fast or too slow (depending on the actual frequency of the internal ring-oscillator).

LZ4 COMPRESSED KERNELS

You can shorten the boot time by LZ4 compressing the kernel. All OS Design configurations that have IMGFLASH set to 1 automatically execute PackBin.exe in the _FLATRELEASEDIR so NKlz4.bin is created. You can manually compress NK.BIN by running PackBin from the command line in the folder with NK.bin (or provide the full path to NK.bin on the command line). PackBin creates an LZ4 compressed kernel; NKlz4.bin. It’s a normal bin file with a single record that can be flashed to the device using Visual Studio, CELoader or CEWriter. When loading the kernel, the bootloader will recognize this is a compressed binary, decompress it and run as normal. Using LZ4 compression can shorten the boot time by up to 40%!

ADDING SUPPORT FOR YOUR CUSTOM BOARD TO THE GURUCE I.MX6 BSP

Follow these steps to add support for your custom board to the BSP (or ask us to add support for your board to our BSP):

1. Create a detailed IO mux file for your board and use comments to name any GPIO including whether this GPIO is active high low. Example: “TOUCH_INTn”. This indicates clearly this GPIO is an active low input. Use the IOMUX tool or the i.MX Pins Tool version 3 (version 4 and later is not supported if you want us to generate the board header file out of your pins tool config file).
2. Send us the IOMUX of Pins Tool (v3) file so we can generate the board header file for you (using an internal tool).
3. Download the DDR FSL stress test and DDR calculation excel sheet from the NXP community website, then run the DDR3 calibration and stress tests (you should run this at various temperatures and take the worst settings).
4. From the FSL stress test results, adjust the calibration items and use the generated script to create your \PLATFORM\GuruCE_iMX6\SRC\BOARDS\[boardname].s file. Take any of the other .s files as a template.
5. Create any custom initialization (like setting GPIOs or configuring a PMIC) in your \PLATFORM\GuruCE_iMX6\SRC\BOARDS\[boardname].c file (optional).
6. Create any custom registry settings (like default touch calibration) in your \PLATFORM\GuruCE_iMX6\SRC\BOARDS\[boardname].reg file (optional).
7. Add your board to the catalog (in \PLATFORM\GuruCE_iMX6\CATALOG and subfolder 1033).
8. Add your BOARDID_ to the BOARDID enumeration in \PLATFORM\GuruCE_iMX6\SRC\BOARDS\BoardIDs.h.
9. Add your BSP_BOARD_ define to the #if list and include the correct board.h file in \PLATFORM\GuruCE_iMX6\SRC\BOARDS\BoardIDs.h
10. Add your BSP_BOARD_ define to the IF :DEF: list to include the correct [boardname].s file in \PLATFORM\GuruCE_iMX6\SRC\BOARDS\BoardInit.s
11. Add your BSP_BOARD_ define to the #if defined list to include the correct [boardname].c file in \PLATFORM\GuruCE_iMX6\SRC\BOARDS\BoardInit2.c
12. Add your BSP_BOARD_ define to the #if defined list to include the correct [boardname].reg file at the very end of \PLATFORM\GuruCE_iMX6\FILES\platform.reg
13. Add the compiler and assembler defines for your board in \PLATFORM\GuruCE_iMX6\sources.cmn

If you search the GuruCE_iMX6 BSP folder for "BSP_BOARD_" you will find all locations required to add your board. As much as we could possibly get into the BOARDS folder is there, but some things (the catalog and sources.cmn file) are in different locations. There is NO NEED to change any driver code to setup IO muxing specific for your board!

BOOTLOADER

MENU

```
------------------------------------------------------------------
 GuruCE Bootloader - Main menu
------------------------------------------------------------------
 [1] Load kernel from..... : SD/MMC
 [2] Number of active cores..... : 4
 [3] Boot menu keypress wait..... : 3
 [W] Hardware watchdog..... : Enabled
 [Q] L2 cache..... : Enabled
 [C] Clean registry & databases.. : Disabled
 [K] KITL..... : Disabled
 [P] KITL passive mode..... : Disabled
 [O] KITL/Download device..... : USB Serial
 [U] Windows CE debug UART..... : UART1
 [G] Display menu..... : Disabled
 [E] Ethernet menu
 [M] SD/MMC menu
 [A] SATA menu
 [H] HAB menu
 [R] Reset menu
 [I] Show i.MX6 info
 [B] Bootloader shell
 [F] Reset to factory defaults
 [D] Download image over..... : USB Serial
 [L] Launch stored kernel from... : SD3
 [S] Save settings

 Selection:
```

[1] LOAD KERNEL FROM

This option can be used to select the storage medium to load the CE kernel image from. You can choose from the following options:

- NK from SD/MMC
- NK from SATA
- NK from NAND
- Visual Studio

Set the boot device to “Visual Studio” to download the kernel image over USB Serial, Ethernet or USB RNDIS from Visual Studio or CELoader. If you want to debug the kernel in Visual Studio (with the Platform Builder plugin) make sure you also enable KITL (see “[K] KITL Enable Mode” below).

[2] NUMBER OF ACTIVE CORES

The number of cores inside the i.MX6 processor differ per type; UL, ULL & Solo: 1 core, DualLite, Dual & DualPlus: 2 cores, Quad & QuadPlus: 4 cores. The bootloader automatically sets this menu option to the maximum number of cores detected, but you can change this to limit the number of active cores. For debugging purposes, it is often very nice to enable just 1 core to simplify the debugging process, also because Microsoft never designed nor tested the kernel debugger on a multi-core platform and as a result you will be treated with random exceptions in KDSTUB if you debug with multiple cores enabled. We strongly recommend setting the number of active cores to 1 when using the kernel debugger.

[3] BOOT MENU KEYPRESS WAIT

Option [3] in the menu allows you to configure the number of seconds the bootloader will wait before it jumps to its configured action. The default boot delay is set to 3 seconds. The user can cancel the default action and jump into the bootloader menu by pressing the space bar within the delay specified here. To reduce boot time set the delay to 0 seconds (or build a Silent bootloader). When 0 seconds is set as delay, the bootloader will

immediately load the kernel without any delay. You can still break into the bootloader by pressing and holding the spacebar in your terminal window while powering or resetting the device.

[W] HARDWARE WATCHDOG

This option allows you to disable or enable the hardware watchdog. The watchdog is normally only enabled in the Shipbuild build configuration. It is never enabled in a Debug kernel (even though the menu will still show it is enabled) because then the board would reset shortly after you hit a breakpoint. The "Disable the watchdog" item in the catalog (see "Other configuration options" above) overrides the setting in the bootloader. Disabling the watchdog from the bootloader can be very handy when running CETK tests (some tests run a tight loop at a high priority causing the watchdog to reset the board) or when you encounter board resets and you need to find the offending process.

[Q] L2 CACHE

This option allows you to disable L2 cache completely, set L2 cache to write-through mode or enable L2 cache in normal mode from the bootloader. For best overall performance leave this option enabled. If hard real-time control is required, set to write-through for slightly better real-time behavior or disable L2 cache completely for best real-time behavior.

Not available on UL/ULL.

[C] CLEAN REGISTRY & DATABASES

This option is very useful when the OS running on the board uses a hive-based persistent registry. With a hive-based persisted registry, the registry settings made in Windows CE are persisted (saved) on a storage medium between power cycles or resets. If, for some reason, the registry on the device gets corrupted it may prevent the kernel from booting up. Using this option in the boot menu will force the OS to discard the stored registry and databases and start clean.
The normal way of using this option is to set it to 'Enabled' and immediately choose option [L] to load the existing Windows kernel in flash (so without saving the settings to flash). If you don't want Windows CE to persist the registry & databases in between power cycles and resets you can of course enable this option and save this setting to flash (by choosing the [S] option). That way Windows CE will always start up with a clean registry and databases.

[K] KITL

The Kernel Independent Transport Layer (KITL) is a communications protocol between Visual Studio on your development PC and a board running Windows CE. This option allows you to enable or disable KITL. By default this option is disabled because the image automatically starts from persistent storage without KITL being required. If you want to debug your kernel from Visual Studio (with the Platform Builder plugin) you need to set this option to "Enabled" and download the kernel from Visual Studio (see option [6] above).

[P] KITL PASSIVE MODE

KITL can be started and used in two different modes: Passive and Active mode. Active KITL is best suited for development (the debugger can maintain a constant connection) and passive KITL is better suited in a real-world scenario where the debugger is not constantly needed. Passive KITL allows you to connect a debugger after the device has crashed, allowing you to investigate the cause of the crash. For a more detailed description about these two modes please refer to MSDN: https://msdn.microsoft.com/en-us/library/jj556189.aspx.

[O] KITL/DOWNLOAD DEVICE

This option allows you to select the communication device used to establish the KITL connection. You can choose either ENET (the i.MX6 internal Ethernet module), USB Serial or USB RNDIS. Selecting USB RNDIS or

USB Serial frees up the ENET module so you can debug the Ethernet driver in CE if required. If you want to use USB RNDIS you need to install a driver to allow downloading and debugging over USB. Full setup instructions are described in "Configure KITL over USB RNDIS" below. We recommend to always use USB Serial for KITL as this is the only option not requiring any configuration on the PC or in the bootloader, and it is by far the fastest communication channel (>8 MB/s).

[U] WINDOWS CE DEBUG UART

This option allows you to select the UART that will be used by the kernel to output serial debug messages. You can select any of the 5 UARTs in the i.MX6, or set this to 'Disabled' to disable serial debug messages altogether. Note that your CE kernel should not load the UART driver for the UART port selected as debug UART in the bootloader because this will cause issues with applications using that same UART. The evaluation kernels load the driver for the UART used for debug so it can display the Command Line Interface. To prevent garbage output at boot when using the evaluation kernel images you should disable the Windows CE Debug UART option in the bootloader.

[G] DISPLAY MENU

This option allows you to select the display configuration used by the kernel. You can configure multiple displays; each selected display will immediately initialize and show the boot splash if there is one.

```
-------------------------------------------------------------
 GuruCE Bootloader - Display menu
-------------------------------------------------------------
 [0] No display
 [P] Primary display..... : Disabled
 [S] Secondary display..... : Disabled
 [D] DirectDraw surface..... : Disabled
 [C] Clone mode..... : Disabled
 [X] Return to Main menu
```

The menu options are built up from the various display modes defined in \SRC\DRIVERS\IPUV3\DISPLAY\MDD. If you add a definition for a particular display to HDMIDisplayModes.cpp, LVDSDisplayModes.cpp or LCDDisplayModes.cpp and you rebuild the bootloader, the menu will automatically be adjusted to include your new display mode.

Use the "[P]rimary display" option to select the primary display, and the "[S]econdary display" option to select an optional secondary display. There are some bus bandwidth limitations on the i.MX6 Solo and DualLite processors, so not all combinations of displays will work on the i.MX6 Solo or DualLite. The i.MX6 Dual, Dual+ Quad and Quad+ processors have enough bandwidth to support any combination of displays and resolutions supported.

Due to limitations in the Microsoft DirectDraw implementation, only one display can use DirectDraw. Use option [D]irectDraw surface to select which display will use the DirectDraw surface.

Option [C]lone mode allows you to enable or disable the display clone mode. When enabled, the primary display contents are copied (and scaled if needed) to the secondary display.

The GuruCE BSP currently supports the following HDMI resolutions:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Primary Display Mode
-----------------------------------------------------------------
 [0] HDMI 800x600
 [1] HDMI 1024x600
 [2] HDMI 1024x768
 [3] HDMI 1280x800
 [4] HDMI 1280x1024
 [5] HDMI 1360x768
 [6] HDMI 1680x1050
 [7] HDMI 1920x1080
 [8] HDMI Auto Detect
 [X] Return to Display Port menu
```

Option [8] HDMI Auto Detect will make the kernel read the EDID information from the monitor itself and setup the display according to the resolution information received from the monitor. It will also enable hot-plugging HDMI monitors, even if they are using different resolutions. You could for instance attach a 1360x768 HDMI monitor (the CE desktop is shown in 1360x768), unplug the monitor and attach a 1920x1080 HDMI monitor (the CE desktop will dynamically change its resolution to 1920x1080). Due to limitations in the CE display driver and multi-monitor architecture, auto detection of HDMI monitor modes is not possible in multi-display scenarios.

The following LVDS displays are supported (on both LVDS0 and LVDS1):

```
-----------------------------------------------------------------
 GuruCE Bootloader - Primary Display Mode
-----------------------------------------------------------------
 [0] LVDS0 800x480 18bpp
 [1] LVDS0 800x480 24bpp
 [2] LVDS0 1024x600 18bpp
 [3] LVDS0 1024x600 24bpp
 [4] LVDS0 1024x768 18bpp
 [5] LVDS0 1024x768 24bpp
 [6] LVDS0 1280x800 18bpp
 [7] LVDS0 1280x800 24bpp
 [X] Return to Display Port menu
```

The following LCD displays are supported on DISP0:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Primary Display Mode
-----------------------------------------------------------------
 [0] KOE TX16D20VM5BAA 6.2" 640x240
 [1] DataVision 7" 800x480
 [2] Samsung 7" LMS700K 800x480
 [3] Okaya 7" 800x480
 [4] Tianma TM035KBH02-09 3.5" 320x240
 [5] ET043080DH6-GP 4.3" 480x272
 [6] Emtrion TX14D17 5.7" 640x480
 [X] Return to Display Port menu
```

And the following LCD displays are supported on DISP1:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Primary Display Mode
-----------------------------------------------------------------
 [0] NEC NL4827HC19-05A 4.3" 480x272
 [X] Return to Display Port menu
```

[E] ETHERNET MENU

This option allows you to configure all available Ethernet interfaces on the device.

```
-----------------------------------------------------------------
 GuruCE Bootloader - Ethernet menu
-----------------------------------------------------------------
 [1] Configure ENET1
 [2] Configure USB RNDIS
 [X] Return to Main menu

 Selection: 1

-----------------------------------------------------------------
 GuruCE Bootloader - Configure ENET1
-----------------------------------------------------------------
 [1] IP address.................. : 192.168.1.201
 [2] Subnet mask................. : 255.255.255.0
 [3] Gateway..................... : 192.168.1.1
 [4] MAC address................. : 00:04:9f:04:10:2e (from MAC0 fuses)
 [5] DHCP........................ : Disabled
 [C] Copy settings to CE......... : Enabled
 [X] Return to Main menu
```

[1..5] NETWORK SETTINGS

Options [1] to [5] allow you to configure the network. You can set the IP address, subnet mask, gateway, MAC address and whether or not to use DHCP. Note that these settings are not only used by the bootloader but can also be reflected in Windows CE through option [C], see below.

[C] COPY SETTINGS TO CE

When this option is enabled, Windows CE will use the network settings from options 1-5 to configure the network. Disable this if you want to control the network setup in Windows CE through registry settings.

[M] SD/MMC MENU

This option takes you to a sub-menu that provides configuration utilities for SD & MMC storage cards.

```
-----------------------------------------------------------------
 GuruCE Bootloader - SD/MMC menu
-----------------------------------------------------------------
 [A] Select active USDHC port
 [E] Erase SD/MMC
 [F] Create file-system partition
 [C] Copy firmware
 [X] Return to Main menu
```

With option [A] you can select the current active SD or MMC card. All other options always act on the currently selected SD or MMC card.
Option [E] will erase the entire SD or MMC card. Use with care!
Option [F] will create a file-partition at an offset of 136 MB. The first 136 MB of the card is reserved for the kernel, bootloader and splash logo. The remainder of the card is partitioned and can be formatted for file storage in CE or on your PC. You normally never need this option because the bootloader does this automatically when needed.
Option [C] allows you to copy the contents of the system reserved space (the 136 MB where the bootloader, kernel and splash logo are stored) to another medium. This is very handy when you want to update the system files on eMMC by using an SD card prepared with CEWriter.

[A] SATA UTILITIES

This option takes you to a sub-menu that provides configuration utilities for a SATA hard disk (only available on Quad/QuadPlus).

```
-----------------------------------------------------------------
 GuruCE Bootloader - SATA menu
-----------------------------------------------------------------
 [E] Erase SATA disk
 [F] Create file-system partition
 [X] Return to Main menu
```

Option [E] will erase the entire SATA disk. Use with care!

Option [F] will create a file-partition at an offset of 136 MB. The first 136 MB of the disk is reserved for the kernel, bootloader and splash logo. The remainder of the disk is partitioned and can be formatted for file storage in CE or on your PC. You normally never need this option because the bootloader does this automatically when needed.

[H] HAB MENU

The HAB menu allows you to check for HAB events, write the public key pairs and secure (close) the device.

```
-----------------------------------------------------------------
 GuruCE Bootloader - HAB menu
-----------------------------------------------------------------
 [I] Show HAB info
 [C] Check HAB Events
 [W] Write SRK fuses
 [S] Secure (close) device
 [X] Return to Main menu
```

Before enabling HAB, follow the steps in \WINCEX00\PLATFORM\GuruCE_iMX6\HABv4\readme.txt.

Option [I] shows the HAB status. If the status is anything else than 0xF0 (HAB_SUCCESS) you can use option "[C] Check HAB Events" for details about the failure.

If the HAB status is showing HAB_SUCCESS, you can use options "[W] Write SRK" fuses and option "[S] Secure (close) device" to write the public key to the device and completely secure the device so that it will be impossible to run unauthorized code on the device.

As long as the #define HAB_ALLOW_FUSING in \WINCEX00\PLATFORM\GuruCE_iMX6\SRC\BOOTLOADER\HAB\hab.cpp is set to FALSE, you can try options [W] and [S] without any risk; the functions are not allowed to actually burn any fuses, so you can try before you die.

WARNING: Once the #define HAB_ALLOW_FUSING is set to TRUE you must make absolutely sure [C] Check HAB Events returns HAB_SUCCESS without any additional HAB events and the SRK fuses are burned successfully and correct before securing the device.
Failure to do so will permanently brick your device!

[R] RESET MENU

This menu has many options to reset the device. To quickly reset the device press [R][R]. Make sure to save any changed settings before choosing this option. The bootloader reset menu allows you to dynamically change the medium the i.MX6 loads the bootloader from:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Reset menu
-----------------------------------------------------------------
 [R] Reset
 [0] Reset to fuses
 [1] Reset to USB (MfgTool)
 [2] Reset to NOR Flash
 [3] Reset to OneNAND
 [4] Reset to SATA SSD/HD
 [5] Reset to ECSPI1
 [6] Reset to ECSPI2
 [7] Reset to ECSPI3
 [8] Reset to ECSPI4
 [9] Reset to ECSPI5
 [A] Reset to I2C1
 [B] Reset to I2C2
 [C] Reset to I2C3
 [D] Reset to USDHC1
 [E] Reset to USDHC2
 [F] Reset to USDHC3
 [G] Reset to USDHC4
 [H] Reset to eMMC1
 [I] Reset to eMMC2
 [J] Reset to eMMC3
 [K] Reset to eMMC4
 [L] Reset to RAW NAND
 [M] Reset to Toggle NAND
 [X] Return to Main menu
```

To reset and boot from another boot medium, select the boot medium from the menu. The board will reset and try to load the bootloader from the selected medium. Once you have selected an alternate medium to reset to, pressing [R][R] will reset to this medium as long as power is applied. To revert to the medium as set by the fuses either press [R][0] or hard reset/power-cycle the device. If the i.MX6 fails to boot from the selected medium, USB boot mode is entered. If this happens simply power-cycle the device or press the reset button to load the bootloader from the medium as set by the fuses.

[I] SHOW I.MX6 INFO

This menu option shows i.MX6 Hardware information:

```
------------------
System Information
------------------
Processor                       = iMX6 QuadPlus (4 cores)
Speed grade                     = 996 MHz (core @ 996 MHz)
Temperature grade               = Automotive -40 to 125C
Chip silicon revision           = 1.0
RAM                             = 1 GB at 0x10000000
Fuse MAC0                       = 00:04:9F:04:10:2E
USB VID/PID                     = 0000/0000
GP1                             = 0x00003100
GP2                             = 0x00000000
64 bit UID                      = 0B1309D4E3167ABB
256 bit SRK                     = 0000000000000000000000000000000000000000000000000000000000000000
SRC Reset Status Register       = 0x00010010 (software initiated reset)
```

[B] BOOTLOADER SHELL

The bootloader shell provides a way to peek and poke memory and I/O, burn fuses, erase SPI Flash, display image definitions, inspect detailed clock settings and set or even overclock the CPU frequency, etc.

WARNING: The bootloader shell is for advanced users only! Incorrect use of the options here can damage your board and/or **permanently** brick your device!

```
-----------  GuruCE Boot Shell  ----------
Type ? for help
> ?
?                                          Help
i                                          Show iMX6 Info

d                                          Dump Image Defines
g                                          Dump Clock Gates
c                                          Dump Clock Frequencies
t                                          Dump Clock Tree

a freqMHz                                  Set ARM CPU frequency (81..1296 MHz)
  Example: a 996                           Set ARM CPU frequency to 996 MHz
o                                          Configure Clock Output 1/2

v RegAddress [# of object]                 View register value
s RegAddress RegValue                      Set register value
b RegAddress BitOffset(0-31) Value(0 or 1) Set register bit

e startSector numSectors                   Erase SPI Flash sectors
  Example: e 0x7F 0x1                      Erase SPI Flash sector 127
           e 0x0 0x100                     Erase SPI Flash sectors 0 to 256 (not included)

f Command (r/w) FuseAddress(0-0x2F) FuseValue   Fuse Read/Write/Dump fuses
  Example: f r 0x26                        Read fuse at 0x26 (Bank 4, word 6)
           f w 0x27 0x1                    Write fuse at 0x27 (Bank 4, word 7) with 0x1
           f d                             Dumps all OCOTP fuses from 0x0-0x2F

m MemAddress Length                        Memory test (1 GB total)
  Example: m 0x10200000 0x3FDFFFFF         Test all device memory (excl. bootloader memory)

l [wt]                                     L1/L2 Cache Test. Specify wt to enable write-through
  Example: l                               Perform cache tests
  Example: l wt                            Perform cache tests with write-through enabled

x [or] q                                   Exit Shell

All the input parameters are separated by space key!
>
```

[F] RESET TO FACTORY DEFAULTS

Select this option to reset the bootloader menu configuration settings to the factory default configuration settings.

[D] DOWNLOAD IMAGE OVER

This option will initialize the selected download device and start the image download.

[L] LAUNCH STORED KERNEL FROM

Pressing [L] will try to load and launch the image stored on SD, MMC, NAND or SATA depending on the boot device setting (see option [1] above).

[S] SAVE SETTINGS

This option writes the menu settings to persistent storage. The storage location depends on the boot source that was selected in the catalog at the time of building the bootloader, and in case of SD/MMC on the active SD/MMC device (see option [M] above).

CONFIGURE KITL OVER USB RNDIS

The Remote Network Driver Interface Specification (RNDIS) is a Microsoft proprietary protocol used mostly on top of USB. It provides a virtual Ethernet link to most versions of the Windows operating system. Using USB RNDIS to download and debug a CE kernel frees up the ENET module so you can debug the Ethernet driver itself, or debug the kernel on a system without an Ethernet port. Note that we recommend to use USB Serial for KITL as this is the only option not requiring any configuration on the PC or in the bootloader, and it is by far the fastest communication channel (>8 MB/s).

Before you can use USB RNDIS you need to install a driver on the desktop to support this virtual network adapter.

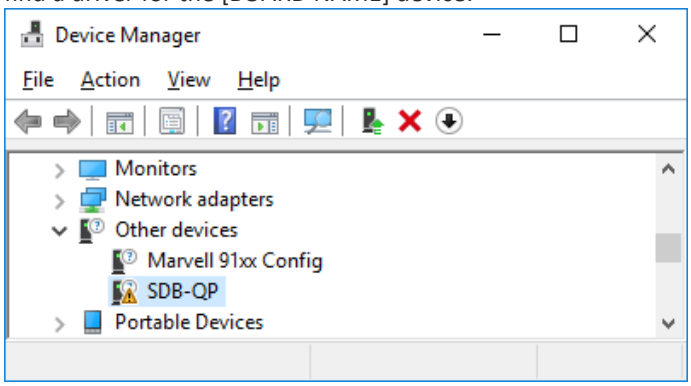
USB RNDIS DRIVER INSTALLATION

Make sure that USB cable is connected between the client port of the board and the development machine.

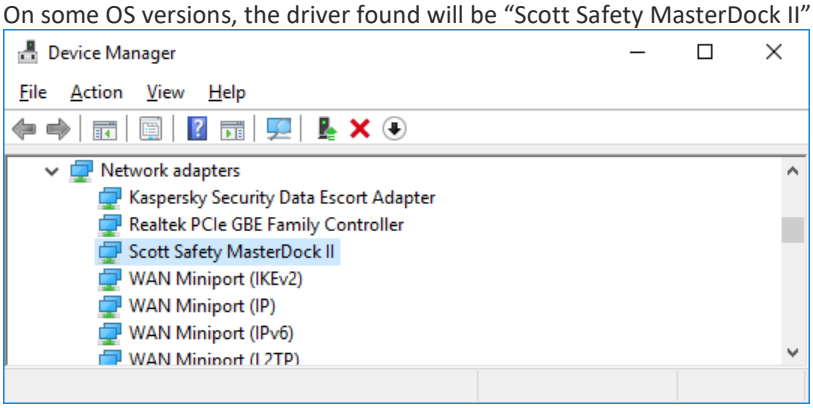
1. Set the KITL communication device (option [O]) in the bootloader menu to USB RNDIS:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Main menu
-----------------------------------------------------------------
 [1] Load kernel from..... : SD/MMC
 [2] Number of active cores..... : 4
 [3] Boot menu keypress wait..... : 3
 [W] Hardware watchdog..... : Enabled
 [Q] L2 cache..... : Enabled
 [C] Clean registry & databases.. : Disabled
 [K] KITL..... : Disabled
 [P] KITL passive mode..... : Disabled
 [O] KITL/Download device..... : USB RNDIS
 [U] Windows CE debug UART..... : UART1
 [G] Display menu..... : Disabled
 [E] Ethernet menu
 [M] SD/MMC menu
 [A] SATA menu
 [H] HAB menu
 [R] Reset menu
 [I] Show i.MX6 info
 [B] Bootloader shell
 [F] Reset to factory defaults
 [D] Download image over..... : USB RNDIS
 [L] Launch stored kernel from... : SD3
 [S] Save settings
```

2. Press [S] to save the new settings.
3. Press [D] to start download over USB RNDIS.
4. At this moment USB insertion should be detected on the desktop machine and Windows will try to find a driver for the [BOARD NAME] device:

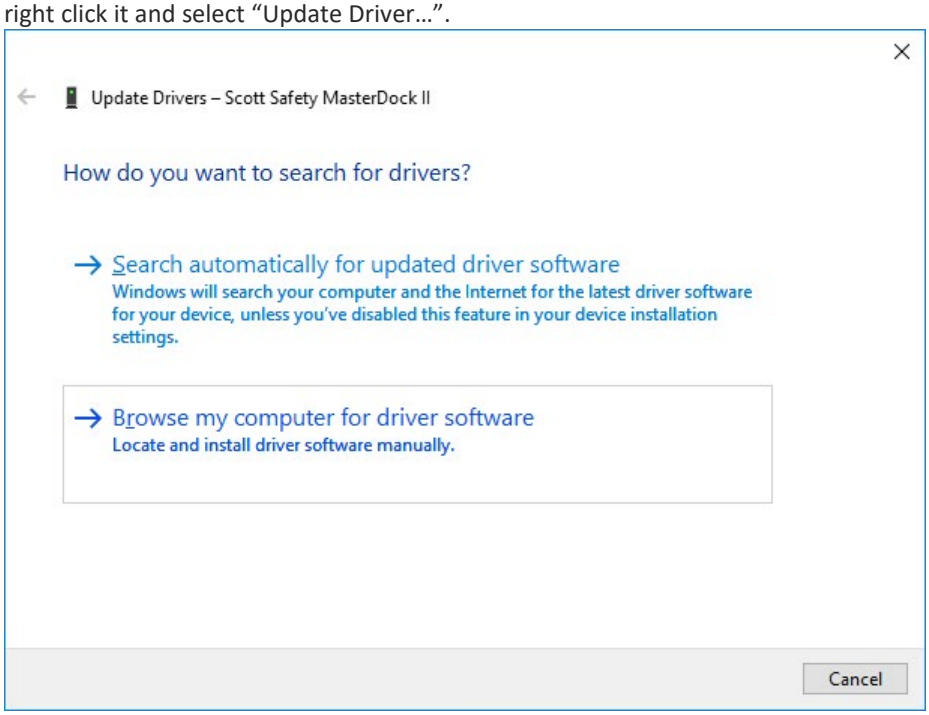


5. On some OS versions, the driver found will be “Scott Safety MasterDock II”:



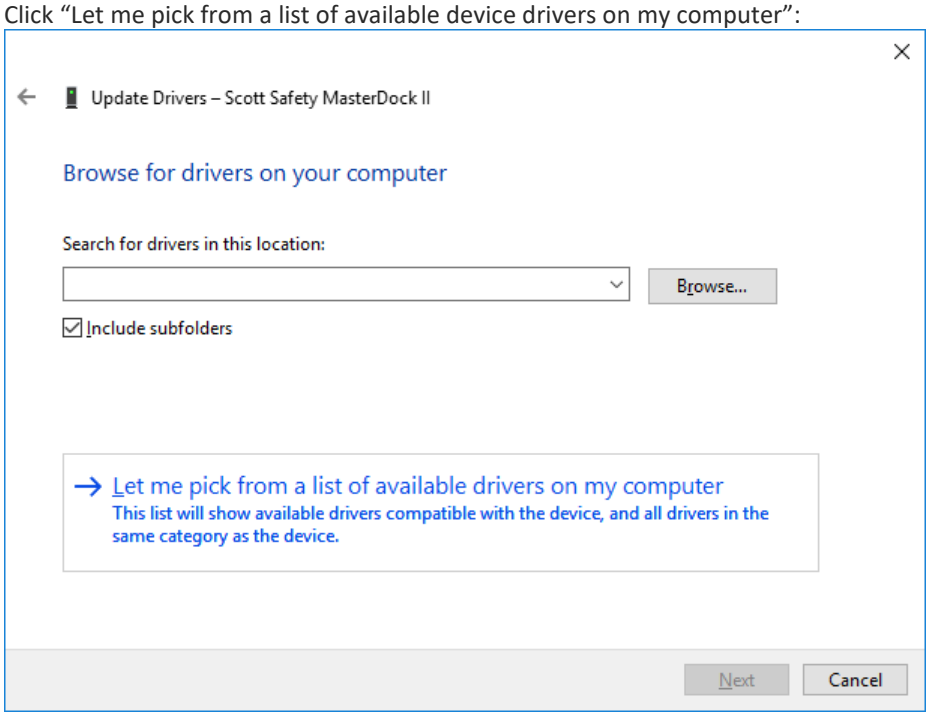
This will work just fine, but you can change it to “Remote NDIS Compatible Device” by following the steps below. If the correct driver is not found, follow the steps below:

6. Open the device manager on your desktop machine by pressing the Windows start button + R and in the run dialog type “devmgmt.msc” followed by enter. This will open the device manager.
7. Find your device under “Other devices” (or under “Network adapters -> Scott Safety MasterDock II”), right click it and select “Update Driver…”.

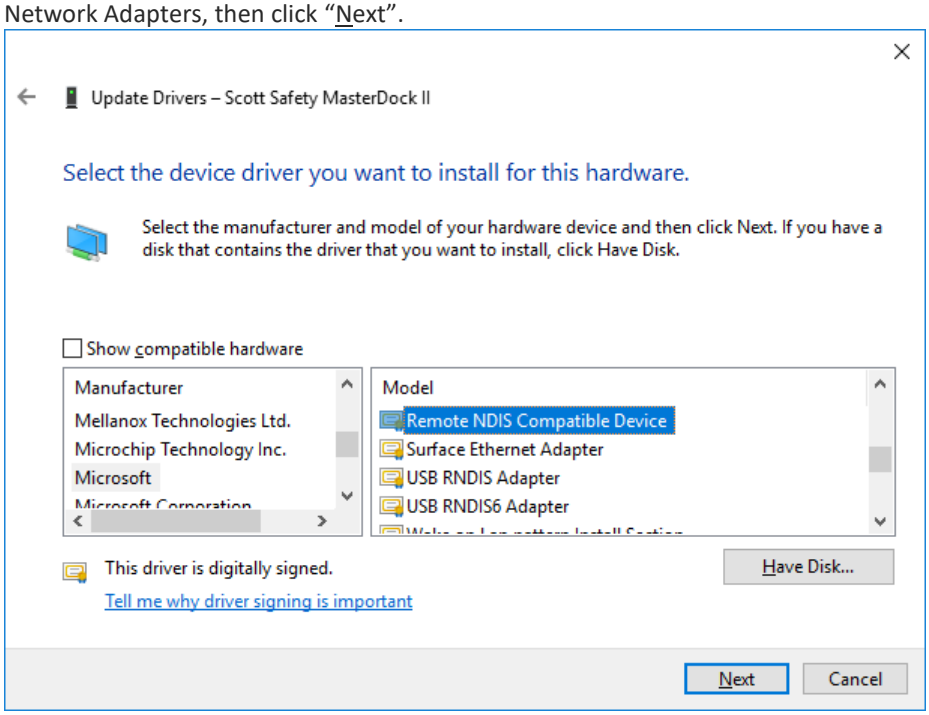


8. Click “Browse my computer for driver software” in the “Update Drivers” dialog:

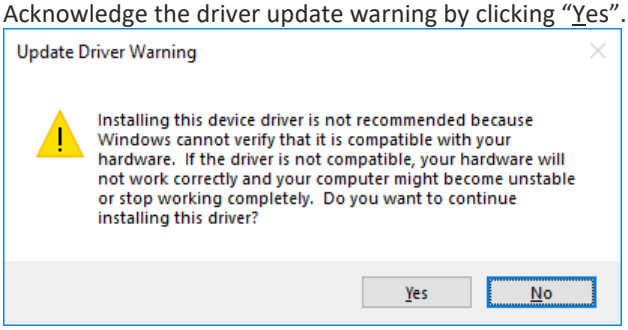
9. Click “Let me pick from a list of available device drivers on my computer”:



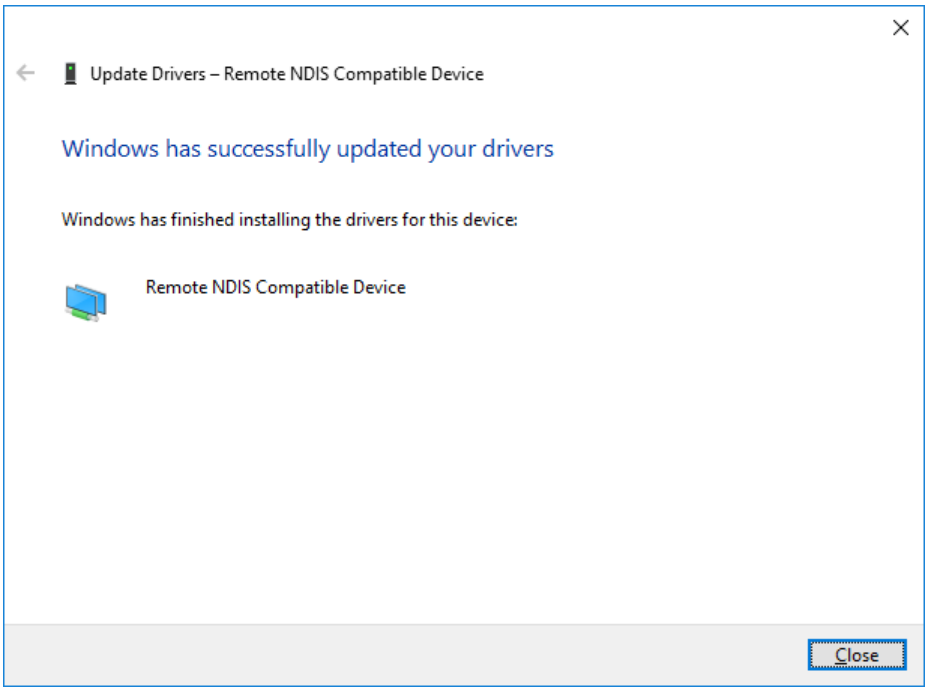
10. If you get a dialog with a checked checkbox “Show compatible devices”, uncheck it. If you get a dialog with “Common hardware types”, choose Network adapters.
11. Select “Microsoft” in the manufacturers list and find “Remote NDIS Compatible Device” in the list of Network Adapters, then click “Next”.



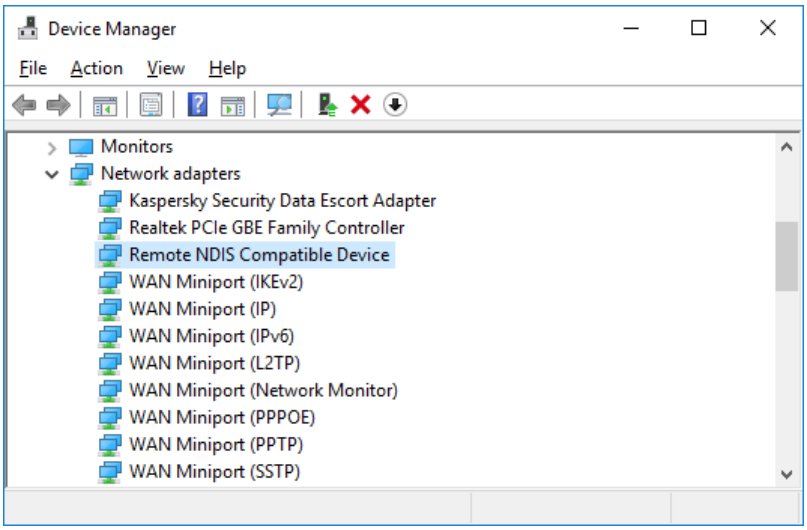
12. Acknowledge the driver update warning by clicking “Yes”.



13. The driver should install without any issue. If it does not, restart the process from step 1 above.
14. If the driver installs successfully the final dialog will show:



15. Click Close and the device manager should now show the Remote NDIS Compatible device under Network Adapters:



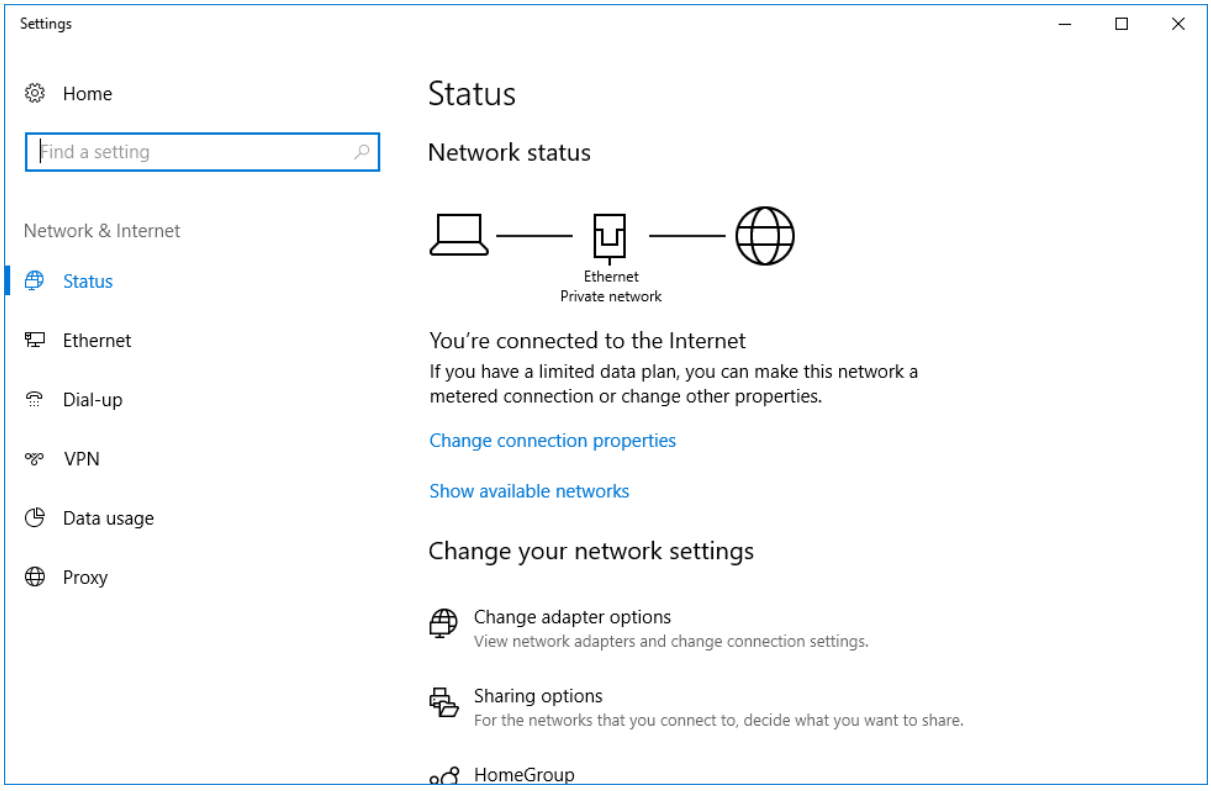
CONFIGURE USB RNDIS NETWORK SETTINGS

Now that the driver is installed you need to configure a static IP for the device and the development machine:

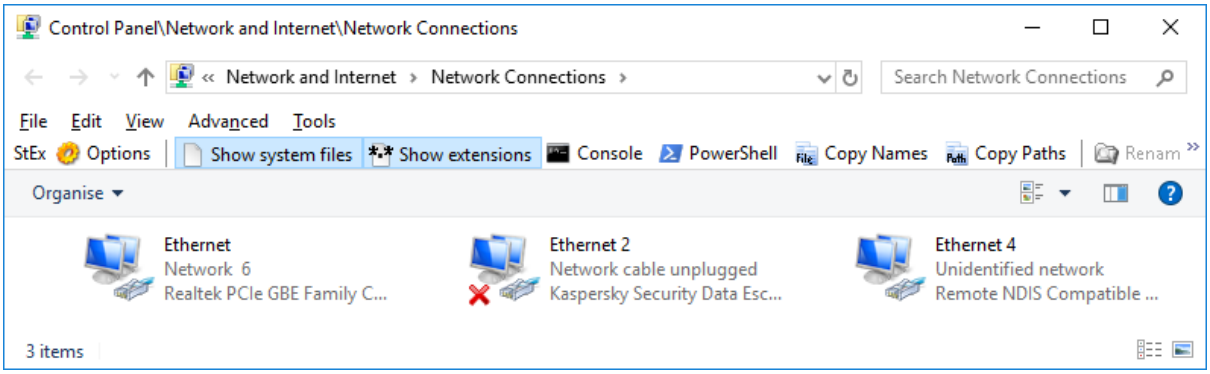
1. Reset the board and press the space bar to enter the bootloader menu.
2. Make sure USB RNDIS (in the [E] Ethernet menu) is configured for a static IP inside your desktop PC's network subnet. For instance, if your desktop PC has an IP address of 192.168.1.105 you could configure the bootloader to use the following settings:

```
------------------------------------------------------------
 GuruCE Bootloader - Configure USB RNDIS
------------------------------------------------------------
 [1] IP address.................. : 192.168.1.203
 [2] Subnet mask................. : 255.255.255.0
 [3] Gateway..................... : 192.168.1.200
 [4] MAC address................. : 00:3c:51:01:02:03
 [5] DHCP........................ : Disabled
 [C] Copy settings to CE......... : Enabled
 [X] Return to Main menu
```

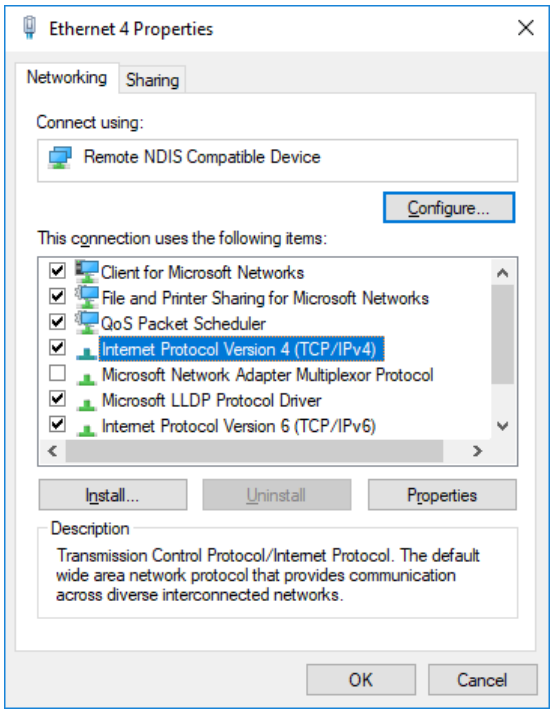
   The gateway address is not needed for USB RNDIS, but if you want Windows CE to use a gateway you could set it in the bootloader and make sure option [C] is set to "Enabled".
3. Press [X] until you are back in the Main menu.
4. Press [S] to save the settings.
5. Press [D] to start the download.
6. Open the Network Status dialog on your desktop machine by right-clicking the network icon in the taskbar and clicking "Open Network Internet Settings":



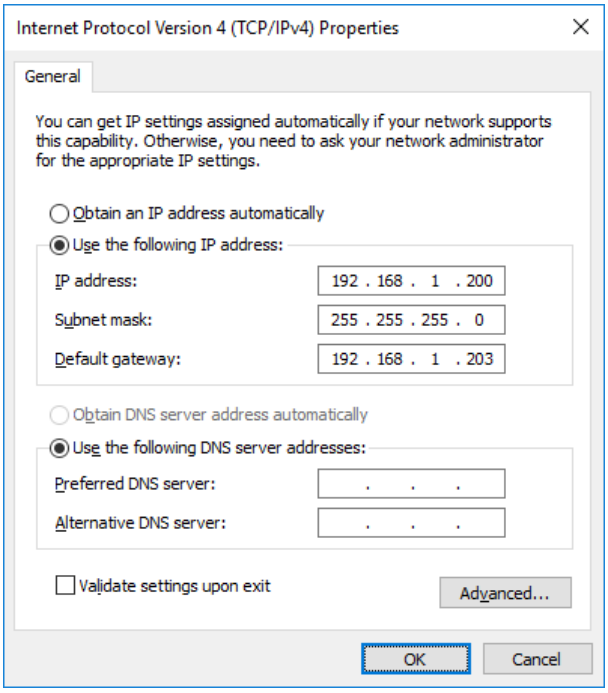
7. Click “Change adapter options”. The following Dialog should then open:



8. Right click the “Remote NDIS Compatible” Ethernet adapter and select “Properties”.
9. Select “Internet protocol version 4 (TCP/IPv4)” and click “Properties”.



10. Configure a static IP address in the same range as the device and set the gateway to the device's IP address:



11. Click "OK". You should now be able to see the BOOTMEs sent by the device in Visual Studio. If the download does not start from Visual Studio but VS does detect the device, make sure you have used an IP address in the same subnet as your development PC and make sure you have set a valid MAC address.
12. For complete reference, the important bootloader settings (in red) for downloading and debugging a Windows Embedded Compact image with KITL via USB RNDIS are:

```
-----------------------------------------------------------------
 GuruCE Bootloader - Main menu
-----------------------------------------------------------------
 [1] Load kernel from.......... : Visual Studio
 [2] Number of active cores..... : 1
 [3] Boot menu keypress wait.... : 3
 [W] Hardware watchdog.......... : Enabled
 [Q] L2 cache................... : Enabled
 [C] Clean registry & databases.. : Disabled
 [K] KITL....................... : Enabled
 [P] KITL passive mode.......... : Disabled
 [O] KITL/Download device....... : USB RNDIS
 [U] Windows CE debug UART...... : UART1
 [G] Display menu............... : Disabled
 [E] Ethernet menu
 [M] SD/MMC menu
 [A] SATA menu
 [H] HAB menu
 [R] Reset menu
 [I] Show i.MX6 info
 [B] Bootloader shell
 [F] Reset to factory defaults
 [D] Download image over........ : USB RNDIS
 [L] Launch stored kernel from... : SD3
 [S] Save settings
```

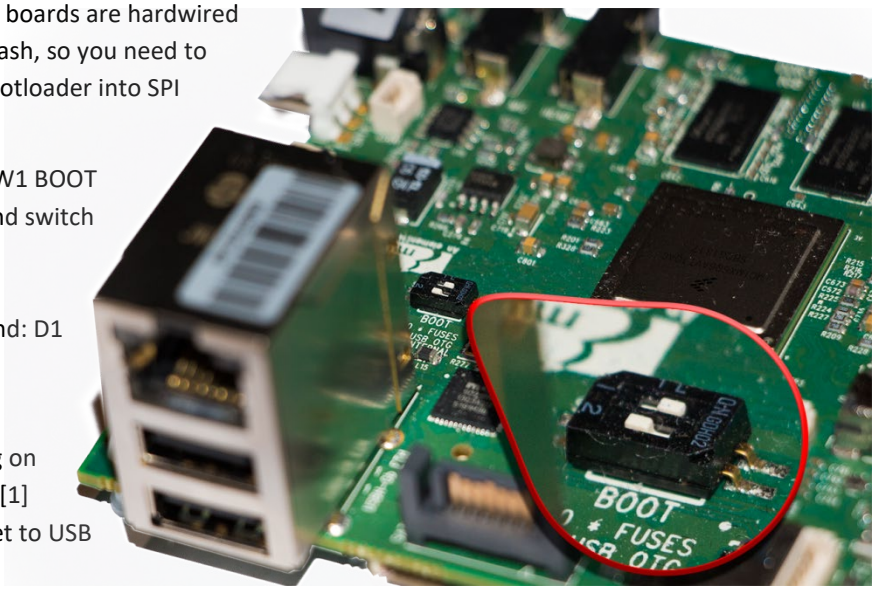
APPENDIX A – BOARD SPECIFIC SETTINGS

BOUNDARY DEVICES SABRE-LITE, NITROGEN6X & NITROGEN6_VM

The SABRE-Lite and Nitrogen6X boards are hardwired to always boot from SPI NOR Flash, so you need to use the MfgTool to flash the bootloader into SPI NOR Flash.

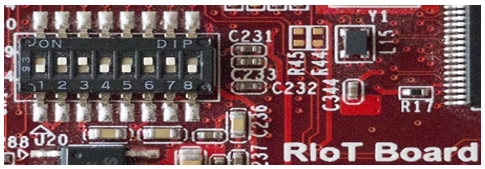
On the SABRE-Lite board, set SW1 BOOT switch D1 to the ON position and switch D2 to the OFF position. On the Nitrogen6X and Nitrogen6_VM boards it is the other way around: D1 OFF, D2 ON.

If you already have the GuruCE bootloader flashed and running on the device, you can also use [R][1] in the bootloader menu to Reset to USB (MfgTool).



ELEMENT14 RIOTBOARD

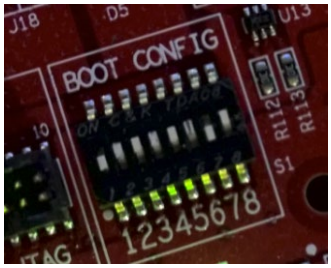
The RIoTboard can boot from a variety of mediums. You can use the CEWriter tool to write the bootloader (and kernel) to an SD or µSD card if you are booting from that. If you are booting from eMMC you will have to use USB boot mode together with the MfgTool to flash the bootloader into eMMC.



Set the SW1 dip switches according to this table:

Boot mode	D1	D2	D3	D4	D5	D6	D7	D8
USB boot mode	0	1	x	x	x	x	x	x
SD (J6, bottom)	1	0	1	0	0	1	0	1
uSD (J7, top)	1	0	1	0	0	1	1	0
eMMC	1	0	1	1	0	1	1	1

DEVICE SOLUTIONS OPAL6-DL & OPAL6-Q



The Opal6 can boot from a variety of mediums. You can use the CEWriter tool to write the bootloader (and kernel) to an SD card if you are booting from that. If you are booting from eMMC you will have to use USB boot mode together with the MfgTool to flash the bootloader into eMMC.

Set the boot config dip switches according to this table:

Boot from:	1	2	3	4	5	6	7	8
eMMC	N/A	ON	ON	ON	ON	ON	ON	OFF
SD Card	N/A	OFF	ON	OFF	ON	OFF	ON	OFF
USB	N/A	X	X	X	X	X	OFF	ON

CONGATEC CONGA-QMX6

The Congatec QMX6 on the QKIT-ARM board is hardwired to always boot from SPI NOR Flash, so you need to use the MfgTool to flash the bootloader into SPI NOR Flash.

To get the board in USB mode, put a jumper on pin 1-2 of CN31, make sure a jumper is on pin 2-3 of CN55 to select USB OTG and connect CN20 to the host computer.

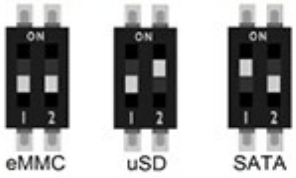
DIGI CONNECTCORE6

The Digi ConnectCore6 can boot from a variety of mediums. You can use the CEWriter tool to write the bootloader (and kernel) to a SATA harddisk or a µSD card if you are booting from that. If you are booting from eMMC you will have to use USB boot mode together with the MfgTool to flash the bootloader into eMMC.

To get the board in USB mode, set switch 1 to off, switch 2 to on and make sure there is no µSD card in the µSD slot on the bottom of the board when you apply power to the board.

SBC boot source switches

The ConnectCore 6 SBC provides two switches to configure the boot source.



Pos1	Pos2	Comments
Off	Off	Boot from eMMC
Off	On	Boot from microSD
On	Off	Boot from SATA
On	On	Reserved

NXP SDP-DL, SDP-Q & SDB-QP

The NXP SDP/SDB boards can boot from a variety of mediums. You can use the CEWriter tool to write the bootloader (and kernel) to a SATA harddisk or an SD card if you are booting from that. If you are booting from eMMC you will have to first boot from SD and then use the "Copy firmware" option in the bootloader to copy the firmware from SD to eMMC.

Set the SW6 dip switches according to this table:

Boot mode	D8 (CFG1_7)	D7 (CFG1_6)	D6 (CFG1_5)	D5 (CFG1_6)	D4 (CFG2_6)	D3 (CFG2_5)	D2 (CFG2_4)	D1 (CFG2_3)
USB boot mode	0	0	0	0	0	0	0	0
eMMC4	0	1	1	0	00 – 1-bit 01 – 4-bit 11 – 8-bit		1	1
SD3	0	1	0	X	X0 – 1-bit X1 – 4-bit		1	0
SATA	0	0	1	0	X	X	X	0

An alternative way to get the NXP SDP/SDB board into USB boot mode is to set SW6 to boot from SD and take out the card before powering the board.

TECHNOLOGIC SYSTEMS TS4900-S & TS4900-Q

The Technologic Systems TS4900 is hardwired to always boot from SPI NOR Flash and all devices come with uboot flashed in SPI NOR Flash. If you are evaluating it is best to leave uboot in SPI NOR Flash and configure it in such a way that it will boot the GuruCE bootloader from eMMC; this keeps your Linux options open. There is one catch; the TS4900 needs to run the April 2016 release of uboot (or later) to be able to do this.

There is no switch or jumper on the TS4900 hardware to easily get the board into USB boot mode, so we'll have to do some more steps to accomplish this task.

If the GuruCE bootloader is already on the device it is really easy: break into the bootloader menu and type 'R1'. This will reset to USB mode.

If u-boot is running on the device you'll have to press CTRL-C to break into uboot and type 'bmode usb' followed by ENTER. For this command to work you need the Technologic Solutions April 2016 uboot release. If 'bmode usb' does not work, you will first need to update uboot to at least the April 2016 version: contact Technologic Solutions for instructions.

If the 'bmode usb' command does work, you'll see the following:

```
Press Ctrl+C to abort autoboot in 1 second(s)
U-Boot > bmode usb
resetting ...
```

Power cycle the device, press CTRL-C to break into uboot and enter the following commands to configure uboot to start our bootloader from eMMC:

```
setenv bootcmd "bmode emmc"
saveenv
bmode usb
```

Now you can start MfgTool and continue the instructions in chapter “Flashing the GuruCE i.MX6 evaluation images” above.

If you want to restore the environment variables (including the ‘bootcmd’ variable we changed) to the factory default, execute the following on the uboot command line:

```
env run clearenv
```

If you are not just evaluating but already developing a product based on WEC, it is of course best to remove uboot and replace it with the GuruCE bootloader to speed up the boot and display the boot splash image immediately at power on. To do this, you can follow the same procedure as above, but make sure you build a bootloader destined for SPI Flash so that uboot will be erased and replaced with the GuruCE bootloader.

TORADEX COLIBRI

The Toradex Colibri board comes flashed with uboot and boots from eMMC. The Toradex uboot supports the ‘bmode’ command, so we can configure uboot in such a way that it boots our bootloader from SD card without the need to replace uboot. This is great for evaluating CE while keeping your Linux options open. Of course once you’ve made the decision to use the GuruCE BSP, it is better to remove uboot and boot the GuruCE Bootloader in one go, to shorten the boot time.

To get the Toradex Colibri uboot to load & run the GuruCE bootloader from an SD card prepared with CEWriter; break into the uboot bootloader, then type the following commands:

```
setenv bootdelay 1
setenv bootcmd “bmode esdhc1”
saveenv
reset
```

OTHER BOARDS

For all other boards; check the manufactures hardware manual for instructions how to get the board into USB recovery mode.

Note: Most development kits enter USB Recovery Mode when you take out the boot medium the board is set to boot from. For instance, if a board normally boots from SD3 and you take the SD card out of the slot, then power the board, it will usually enter USB Recovery Mode. If the board boots from soldered down eMMC or SPI NOR Flash, erasing the boot medium may get the board to enter USB Recovery Mode after power cycling. Use this last option as a last resort! Not all boards enter USB Recovery Mode when you erase the boot medium. If it doesn’t, you have now bricked your device until you find another way to enter USB Recovery Mode.

ABOUT US

GURUCE

GuruCE offers deep technical knowledge of the Windows Embedded CE / Embedded Compact and Windows 10 IoT Core operating systems. The consultants of GuruCE are among the best in Windows BSP & driver development, training and consulting. GuruCE works with many real experts around the world, so help is never far away.

GuruCE can help you and your company get to market faster by taking care of all the Windows low-level issues so that your experts can focus on what they do best, or we can teach you how to do it yourself through training by one of our consultants. We can help you with general system design (both hardware & software), application design & development, real-time embedded design issues and driver development.

BLOG

For general tips & tricks on ThreadX/Windows CE/Embedded Compact/Win10 IoT Core and other related issues please have a look at our blog at https://guruce.com/blog.

SUPPORT OPTIONS

GuruCE offers yearly support contracts that include:

- Direct access to dedicated, highly skilled, support engineers
- 50 support hours
- Free BSP updates

Further details are available online at https://guruce.com/support.

Feel free to contact us online at https://guruce.com/contact or direct using the contact details below:

GuruCE Limited
Sales
Email : sales@guruce.com

Phone : +64 (0)7 929 5807
Mobile : +64 (0)21 104 6208

240 Ohiwa Harbour Road
3198, Opotiki
Bay of Plenty
New Zealand

GuruCE Limited
Technical Support
Email : support@guruce.com