Opal Developers Guide for Windows Embedded Compact 7

Table of Contents

This document includes the following sections:

1. **Getting Started**
   This section will take you through the basics of setting up the Opal Development Kit, installing the required software, creating and downloading an application.

2. **Configuring the Bootloader**
   The bootloader runs when Opal first powers up. This section details how to connect to the bootloader and use it to configure boot, network and display settings.

3. **Application Development**
   This section includes details of the supported drivers and how to access them from your application.

1 Getting Started

This section describes the basics of how to get up and running with the Opal Development Kit. This includes:

1. How to install the development environment so you can successfully develop applications for Windows Embedded Compact 7 running on the Opal platform
2. How to flash your Windows Embedded Compact 7 image to Opal
3. How to create, deploy and debug a simple managed application

The Opal Development Kit comes with a pre-installed Windows Embedded Compact image that will automatically run when you power up the board. You can power the board through the provided power adapter.

It is important to check your device is running the latest version of the Windows Embedded Compact 7 image. GuruCE will release new images containing new features and bug fixes when needed. The latest version can always be found at this link: http://guruce.com/opal/release/latest

For instructions on how to upgrade the firmware in your device please read "Upgrading the firmware" below.

1.1 Software Requirements

You need the following software installed before you can start developing applications on Opal.

- **Visual Studio 2008 with Service Pack 1**
  Download the evaluation version from
  http://msdn.microsoft.com/en-us/vstudio/default.aspx
- **Windows Embedded Compact 7 Opal SDK**
  Download from http://guruce.com/opal/release/latest

For a successful installation you need to follow this exact sequence:

1. Install Visual Studio 2008
2. Install Visual Studio 2008 Service Pack 1
3. Install the Opal SDK[1]
   a. Double click on the SDK installer file (OpalSDKrXXX.msi where XXX is the release version number).
   b. Simply follow the installation wizard and within a few clicks your SDK is installed.

1.2 Upgrading the firmware

The Opal Development Kit boots from internal NAND flash by default.

It is also possible to boot Opal from SD card. Refer to the Opal Technical Reference Manual for Boot Switch settings to enable SD card boot.

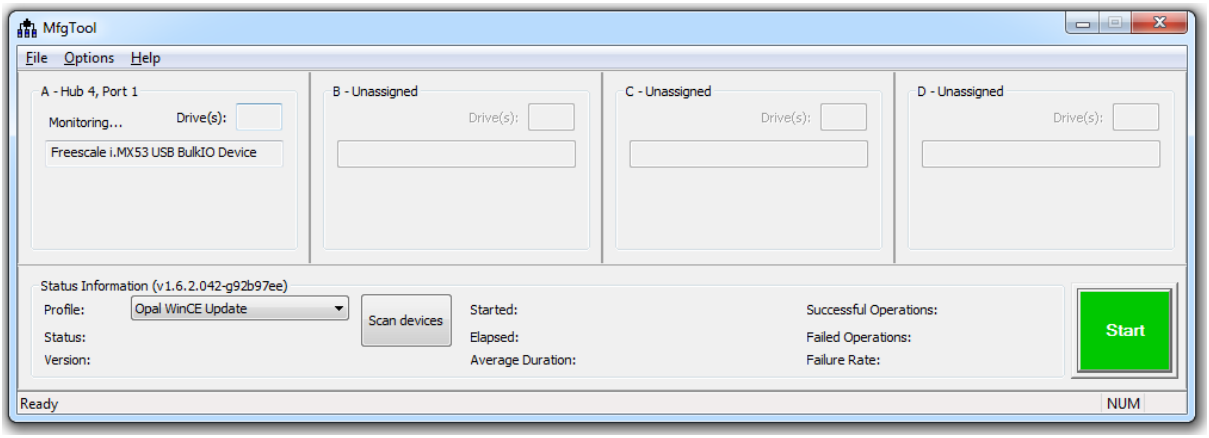
[1] The SDK is built for use with Visual Studio 2008. VS2010 does not support Smart Device Development.

1.2.1 Installing firmware on the NAND flash

Updating the NAND flash is done using the Freescale Manufacturing Tool.

Follow this procedure to update the firmware:

1. Download and unzip the firmware package from the GuruCE web site
2. Run MfgTool.exe
3. On the Opal Development kit, set BOOT CONFIG switch 8 ON
4. Connect the micro USB cable to the port on the left hand side of the board (between the Ethernet connector and GPS antenna). Make sure you have this connected directly to your computer or through a powered hub.
5. Power on the Opal Development Kit.
6. If necessary, install the device driver from the "Drivers" folder when prompted.
7. When the Opal Development Kit is connected and ready to program, you will see the following:



8. Press Start and wait for programming to complete.
9. When complete set BOOT CONFIG switch 8 to OFF and reset (or power cycle) the board.

1.2.2 Upgrading firmware on SD card:

The firmware can be updated on a PC with a card reader and the cfimager.exe tool. cfimager.exe is included with the firmware supplied.

Upgrading or installing new firmware onto an SD card

The following command line is used to update the bootloader on an SD card:

```
cfimager –a –f eboot.nb0 –d X –imx53
```

where X is the drive letter of your card reader. No ":" is required.

To update the main firmware use:

```
cfimager –f nk.nb0 –d X –imx53
```

The card is automatically formatted and you will be able to add files to the card in Windows and access them on Opal.

Before the image will run you will need to setup the MAC address. Follow the instructions in section 2.1.2 below to set the MAC address. Remember to press "S" to save the new MAC address.

1.3 Creating a Simple C# Application

This example will show you how to create a simple managed C# application targeting the Opal using Visual Studio 2008.

1.3.1 Creating your application

1) Open Visual Studio and select **New | Project** from the **File** menu.
2) Navigate to the **Visual C#** language and select the **Smart Device** project type.

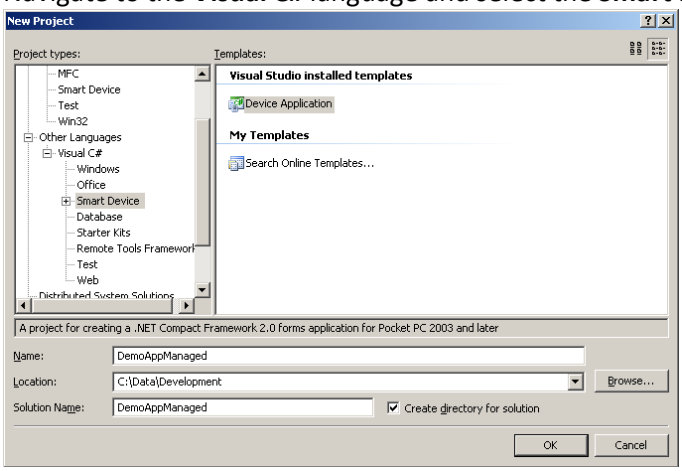
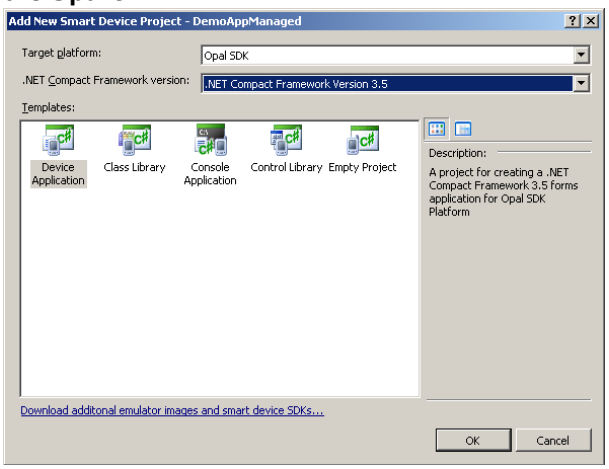


Figure 1: New Project

3) Name the solution **DemoAppManaged** and select **OK**. Visual Studio will generate the necessary files.
4) Make sure **Opal** is selected as your target. If it is not selected, right click on the **DemoAppManaged** project and select **Change Target Platform**. In the pop-up dialog select the **Opal SDK**.



5) Figure 2 shows the generated files.

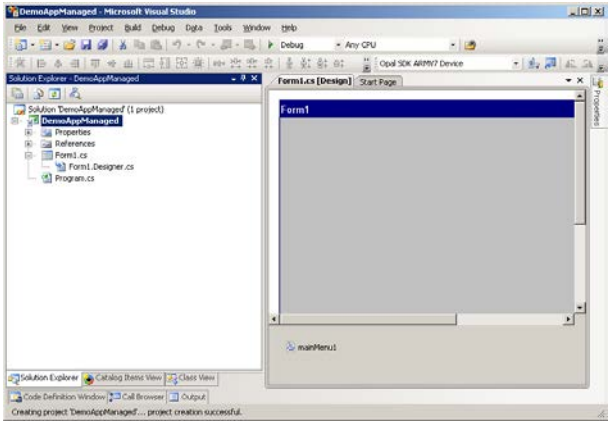


Figure 2: DemoAppManaged

1.3.2 Setting up the connection

1) Power up the Opal
2) When the system has booted you'll need to get the IP address of the Opal. There are several ways to do this:
   - Double click the network icon in the lower right corner and check the detailed view for the IP address, or
   - On the target's command line (start | Run | cmd.exe) run ipconfig.exe, - or-
   - Configure a static IP in the boot loader menu (see section 2.1.2 below), -or-
   - Check the list of assigned IP addresses in your DHCP server, -or-
   - Check the ARP table on your router
3) Click on menu **Tools** in Visual Studio. Select **Options** and browse to **Device Tools** -> **Devices**.
4) In the **Options** dialog select the **Properties** for the **Opal SDK ARMV4I Device** (see Figure 3)

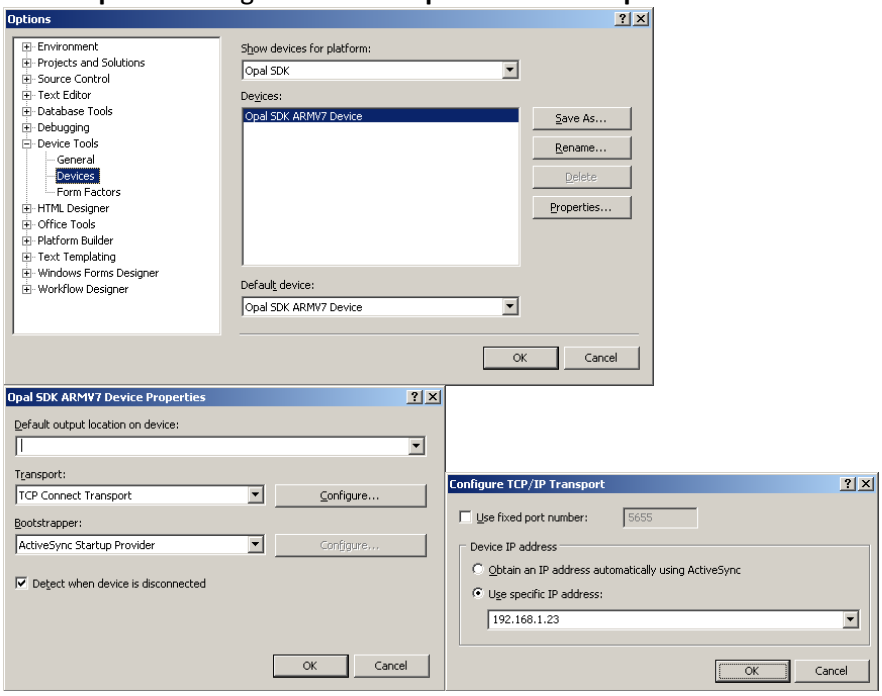


Figure 3: Configuring Connectivity Options

5) In the **Opal SDK ARMV4I Device Properties** window select **Configure**
6) In the **Configure TCP/IP Transport** dialog select **Use specific IP address** and enter the IP address of the Opal retrieved in step 2) above

7) Select **OK** 3 times to get back to the solution
8) In Visual Studio click **Connect to Device** on the **Tools** menu:
9) You are now connected to the Opal device.

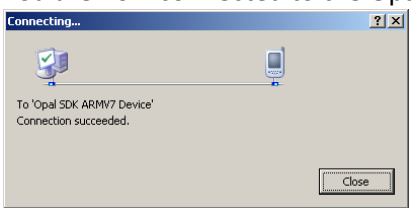


Figure 4: Connection succeeded

1.3.3 Debugging your application

To start debugging your application press F5 or navigate to the **Debug** menu and select **Start debugging**. The application will start running on the Opal. You can set breakpoints and view variables in the same way as you would in a normal desktop application.

2 Configuring the Bootloader

The bootloader is the very first piece of software that runs when Opal boots. Its main responsibility is to start the Windows Embedded Compact image and it can be configured to do so in several different ways. This section describes how to enter the bootloader menu and what the various options are. This includes:

1. Connecting to the bootloader
2. Bootloader menu
3. Configuring the network
4. Configuring the display options
5. Configuring boot and debug settings

2.1.1 Connecting to the bootloader

The physical bootloader connection is a USB mini-B connector which is located between the audio and power jack on the Opal Development Kit. Use the supplied cable to connect to a PC. The Opal development kit includes an FTDI USB-to-serial chip for which device drivers are already included in Windows 7.

This USB debug port is bus-powered and fully functional even before you power up the Opal Development Kit. This means you can configure your favourite terminal program (like TeraTerm Pro, RealTerm, etc.) before you power up the board. Serial parameters are 115200 baud, no parity, 8 data bits and 1 stop bit (115k2, 8N1).

> *If you have a RevD board, you will need to power the Opal Development Kit, press the power button and only then attach the USB cable before you can successfully connect your terminal program. For Rev 1.0 and above hardware you just need to ensure your cable is connected before you connect your terminal – it doesn't matter if Opal is powered or not.*

Power on the board (or press RESET) and you should see the bootloader outputting its version number and some debug information, followed by a prompt with a count-down. Press <SPACE BAR> to launch the booloader menu.

2.1.2 Configuring the Network

Network settings for Opal are setup in the bootloader. From here you can:

- Set the MAC Address
- Enable DHCP, or configure a static IP address and mask
- Select the network device

These items are part of the bootloader menu. Select the appropriate menu option and enter the new information. For example, press 3 to enter a new MAC address.

When you have finished entering your new settings, make sure to press "S" to save the new settings to the flash if you want to persist them. You can press "L" to launch the OS image, or reboot the board.

2.2 Bootloader Menu

The Opal bootloader menu includes the following commands:

```
-------------------------------------------------------------------------------
 GuruCE iMX Menu
-------------------------------------------------------------------------------
 [0] IP Address : 0.0.0.0
 [1] Set IP Mask : 0.0.0.0
 [2] Set Gateway : 0.0.0.0
 [3] Set MAC Address : 0-50-c2-3f-98-21
 [4] DHCP : Enabled
 [5] Copy network settings to CE : Enabled
 [6] Select Boot Device : NK from NAND
 [7] Boot Delay : 3
 [8] Set display mode (Current: "Disabled")
 [9] Reset to Factory Default Configuration
 [C] Clean registry & databases : Disabled
 [F] Format OS NAND Region
 [N] Format All NAND Regions
 [M] MMC and SD Utilities : ESDHC[1] is activated
 [A] ATA Utilities
 [B] Bootloader Shell
 [W] KITL Work Mode : Interrupt
 [K] KITL Enable Mode : Disabled
 [P] KITL Passive Mode : Disabled
 [E] Select Ether Device : FEC
 [D] Download Image Now
 [L] Launch Existing Flash Resident Image Now
 [R] Reset
 [S] Save Settings

 Selection:
```

2.2.1 [0..4] Network settings

Options 0-4 allow you to configure the network. You can set IP address, IP Mask, Gateway, MAC address and whether or not to use DHCP. Note that these settings are not only used to establish the debug connection (see section 1.3.2 above) but can also be reflected in Windows CE through option 5, see below.

2.2.2 [5] Copy network settings to CE

When this option is enabled, Windows CE will use the network settings from options 0-4 to configure the network. Disable this setting if you want to control the network setup from Windows CE.

2.2.3 [6] Select Boot Device

This option can be used to select the device to load the kernel image from. You can choose from the following boot devices:

- NK from NAND (default)
- NK from SD/MMC
- NK from ATA
- Disabled

Set the boot device to “Disabled” to automatically load the kernel image over Ethernet from Platform Builder. If you want to debug the kernel make sure you also enable KITL (see paragraph 2.2.14 below).

2.2.4 [7] Boot Delay

Option 5 in the menu lets you configure the number of seconds the bootloader will wait before it jumps to its configured action. The default boot delay is set to 3 second. The user can cancel the default action and jump into the bootloader menu by pressing the space within the delay specified here. If boot time is important set the delay to 1 second.

2.2.5 [8] Set Display Mode

```
--------------------------------------------------------------------------------
 GuruCE iMX Menu
--------------------------------------------------------------------------------
 [0] Disabled
 [1] DVI
 [2] LCD
 [3] VGA
 [4] LVDS0
 [5] LVDS1
 [6] Dual LVDS
 [7] TV
 [R] Return to Main Menu

 Selection:
```

Use this menu item to select the display you wish to use when Windows Embedded Compact boots. Each menu item shown above has its own sub-menu with display options including display type and resolution where appropriate. The selected mode will show in the main menu.

2.2.6 [9] Reset to Factory Default Configuration

This will restore the original boot loader settings as they were when the device was shipped.

2.2.7 [C] Clean registry & databases

The default Opal OS Design uses a hive based persistent registry. This means registry settings made in Windows Embedded Compact are persisted (saved) between power cycles or resets. If, for some reason, the registry on the device gets corrupted it may prevent Windows Embedded Compact from booting up. Using this option in the boot menu will force Windows to start with a clean registry and clean databases.
The normal way of using this option is to enable this option and immediately choose option 'L' to load the existing Windows kernel in flash (so without saving the setting to flash). If you don't want Windows to persist any settings in between power cycles or resets, you can of course enable this option and saving this setting to flash (by choosing the 'S' option). That way Windows will always start up with clean registry and databases.

2.2.8 [F] Format OS NAND Region

This option will erase the Windows Embedded Compact image from flash.

2.2.9 [N] Format All NAND Regions

This option will erase the entire flash, including the bootloader and its settings. After this command you'll have to re-flash EBOOT and NK using the manufacturing tool to get the device working again, and you'll have to re-enter the MAC address and re-setup the boot settings. Note that the device will not boot without a valid MAC address set!

2.2.10 [M] MMC and SD Utilities

This will option will take you to a sub-menu which provides configuration utilities for SD/MMC storage cards.

2.2.11 [A] ATA Utilities

This option takes you to a sub-menu which provides configuration options for a SATA HDD.

2.2.12 [B] Bootloader Shell

The bootloader shell provides a way to peek and poke memory and memory mapped I/O.
WARNING: This option is for advanced users only. Incorrect use of the options here can damage your board!

```
-----------  GuruCE iMX Boot Shell  ----------
type ? for help
command -- ?

----------------------------- Help ------------------------------
?                                                 Help
e                                                 Exit Shell
d RegAddress                                      Show Reg
s RegAddress RegValue                             Set Reg
b RegAddress BitOffset(0-31) Value(0 or 1)        Set Bit
------------------------------ End -------------------------------

All the input parameters are separated by space key!
command --
```

2.2.13 [W] KITL Work Mode

The Kernel Independent Transport Layer (KITL) is a communications protocol between Platform Builder on your development PC and the Opal i.MX53 CPU Module. KITL can work in either interrupt or polling mode. Interrupt mode is recommended because it's more efficient and hence performs better. Polling is slower and does not allow near-instantaneous user initiated breaks but it can be useful in early system debugging before peripheral interrupts are enabled.

2.2.14 [K] KITL Enable Mode

Enable or disable KITL. By default this option is disabled because the image automatically starts from NAND without KITL being required. If you want to debug your kernel with Platform Builder you need to set this option to "Enabled".

2.2.15 [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:
http://msdn.microsoft.com/en-us/library/ms898144.aspx

2.2.16 [E] Select Ether Device

This option allows you to select the Ethernet transport. For the fastest option choose FEC (the Ethernet module). Other options are USB Serial and USB RNDIS.

2.2.17 [D] Download Image Now

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

2.2.18 [L] Launch Existing Flash Resident Image Now

Selecting 'L' will try to load and launch the image stored on NAND, SD or ATA depending on the boot device setting (option 6).

2.2.19 [R] Reset

This option will reset the Opal

2.2.20 [S] Save Settings

The “Save Settings” option writes the menu settings to NAND flash.

3 Application Development

This section includes detailed information about the features supported on Opal and how to access them from your application.

The Opal BSP is derived from the Freescale i.MX53 BSP. This document only describes the additions & modifications to the Freescale BSP. More detailed information can be found in the Freescale documentation which is available at this location: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=RDIMX53SABRETAB&fpsp=1&tab=Documentation_Tab#.

Note: We have managed wrappers available for most of the SDK headers. Please contact us directly for pricing information.

3.1 Supported Features

The BSP and OS Design support the standard Windows Embedded Compact 7 features like KITL Debugging and downloading, power management and support for all available Platform Builder Remote Tools. Additionally the Opal BSP supports the following features:

Feature	Standard API	SDK	Comments
Memory and Storage			
512Mbytes or 1GByte DDR3 Memory	•		Kernel autodetects available RAM
Opal NAND Flash	•		Supported on Rev D and above (letters are pre-release boards, numbers are final release revisions).
SD Card	•		SDHC compatible
microSD Card	•		Supported on Rev D and above
SATA			Not fully tested yet
Hive Registry	•		Store on NAND, SD or SATA
Display and User Interface			
VGA	•		Various resolutions
LVDS	•		10” Freescale capacitive multi-touch screen supported
LCD (parallel interface)	•		7” TFT with resistive (single) touch supported
Audio			
Freescale SGTL5000 Audio Codec	•		
Communications			
10/100 Ethernet	•		
USB Host	•		
USB Function	•		
Serial Ports	•		
FlexCAN		•	Standard FlexCAN driver included, High Performance driver available, contact us for details.
WiFi		•	Fully supported
Bluetooth			Not working yet (possible driver or

			hardware problem)
I2C		•	
SPI		•	
I/O			
GPIO		•	
Digital I/O		•	Control via the GPIO driver
Analog Inputs		•	
PWM			Not yet supported in this release
Other Features			
GPS	•		Support through location framework
GPT (General Purpose Timers)		•	
Freescale MMA 8540Q Accelerometer	•		Ensure R4 removed from Rev C dev kit. Not fully tested yet.

Table 1 – Opal BSP features

3.1.1 SDK Libraries

The Opal BSP is derived from the Freescale iMX53 BSP. We have improved and added several SDK libraries. The additions and changes are described below. Please refer to the Freescale i.MX53 SDK documentation and use the below information for additional, overriding, documentation.

The following lists the SDK headers and libraries included in the standard kernel image:

- ADC
  Analog to Digital Converter: pmic_adc.h, pmicSdk_da9052.lib
- SPI
  SPI Bus: ecspisdk.h, ecspisdk.lib
- FlexCAN
  CAN Bus: flexcansdk.h, flexcansdk.lib
- GPIO
  General Purpose IO: gpiosdk.h, gpiosdk.lib
- GPT
  General Purpose Timers: gptsdk.h, gptsdk.lib
- I2C
  I2C Bus: i2csdk.h, i2csdk.lib
- OWIRE
  One-Wire (1-Wire): owiresdk.h, owiresdk.lib
- WIFI
  Summit WIFI SDC40NBT: sdc_sdk.h, sdk.lib

We have managed wrappers available for most SDK libraries. Please contact us for pricing information.

3.1.2 Opal GPIO Driver

The Opal GPIO driver makes configuring, reading and writing GPIO very easy. The i.MX53 uses a very complex multiplexing scheme to set various functions per pin and to set pin features (like drive strength, voltage, pull-ups, etc). Some pins can be configured for up to 7 different functions and require you to set various PAD configurations plus the GPIO direction and data registers. The GPIO

driver takes away the complexity of it all and offers you a simple interface through which you can set all features per pin in one single call.

The GPIO driver needs to be configured to set the “safe” set of GPIO’s available. Free GPIO depends on the board design. For instance; if you are using SPI module 1 you won’t be able to use GPIO 5.22, 5.23, 5.24, 5.25, 5.9, 4.11 and 4.12. If you are not using SPI module 1 then these GPIOs can be freely used by the GPIO driver.

Configure the free GPIOs by setting the port masks in the registry (HKLM\Drivers\BuiltIn\GPIO).

3.1.3 FlexCAN Driver

The standard BSP and image binaries include the standard FlexCAN driver and SDK header for the 2 FlexCAN ports of the iMX53. GuruCE also offers a high performance FlexCAN driver supporting many more configuration options and making use of the FlexCAN receive FIFO for uninterrupted high speed reception of messages on the CAN bus. It also supports complex filtering schemes so you can block reception of unwanted messages. The High Performance FlexCAN driver supports all features offered by the iMX53 FlexCAN module. Please contact GuruCE for pricing information.

FlexCAN Driver Comparison Chart

	Standard FlexCAN Driver	High Performance FlexCAN Driver
Configure bitrate in registry	√	√
Set/Get bitrate (runtime)	√	√
Low-power mode	√	√
Error and Activity LEDs (code configurable)	√	√
Manual fine-tuning of CAN timing parameters and clock frequency	×	√
Configure IST priority in registry	√	√
Send a single CAN message	√	√
Receive a single CAN message	√	√
Send/receive multiple CAN messages in one call	√	√
High speed reception & buffering of an unlimited amount of messages	×	√
Configure “Receive Own” in registry	×	√
Enable/Disable “Receive Own” (runtime)	×	√
Configure Rx FIFO (Enable/Receive All/FIFO Size) in registry	×	√
Enable/Disable Rx FIFO (runtime)	×	√
Set/Get Rx FIFO ID filtering (runtime)	×	√
Configure “Listen Only” mode in registry	×	√
Enable/Disable “Listen Only” mode (runtime)	×	√
Configure oversampling in registry	×	√
Enable/Disable oversampling (runtime)	×	√
Configure loopback mode in registry	×	√
Enable/Disable loopback mode (runtime)	×	√
Configure Timer Synchronization in registry	×	√
Enable/Disable Timer Synchronization (runtime)	×	√

4 About us

4.1 GuruCE

GuruCE offers deep technical knowledge of the Windows Embedded CE (Windows Embedded Compact Edition) operating system. The consultants of GuruCE are among the best in Windows CE BSP & driver development, training and consulting.

GuruCE can help you and your company get to market faster by taking care of all the Windows CE 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.

4.2 Blog

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

4.3 Support options

GuruCE support contracts include:

- Free Source BSP
- Unlimited questions and answers
- Development hours

Details on the support options available are online at: http://guruce.com/support

Please contact us directly for more support options and more detailed information on how we can help you:

GuruCE APAC/NZ

Contact : Michel Verhagen
Email : michel@guruce.com
Phone : +64 (0)7 929 5807
Mobile : +64 (0)21 104 6208

240 Ohiwa Harbour Road
RD2
3198, Opotiki
Bay of Plenty
New Zealand

GuruCE EMEA/NL

Contact : Erwin Zwart
Email : erwin@guruce.com
Phone : +31 (0)725 719 386
Mobile : +31 (0)629 512 116

Overbroek 6
Heerhugowaard, 1702 HG
The Netherlands