Freescale Semiconductor
Application Note

Document Number: AN3979
Rev. 0, 01/2010

Adding UART Bluetooth Solution to i.MX Windows CE 5.0/6.0

by Multimedia Applications Division
Freescale Semiconductor, Inc.
Austin, TX

Bluetooth is a wireless communication technology, widely accepted by the developer community. Enhanced data throughput and audio have improved user’s experience with multimedia appliances.

This application note describes the main characteristics that should be taken into consideration when selecting a Bluetooth solution that best fits the requirements; the common requirements that the Bluetooth solution maker must provide to integrate its solution into Freescale’s platforms; and the way to integrate the Bluetooth solution in Freescale WinCE BSPs.

The Bluetooth solution is an external chip that can be configured and operated by other i.MX processors through a common interface and protocol. The Universal Asynchronous Receiver/Transmitter (UART) Bluetooth solution has some parts of the Bluetooth specifications embedded into the external chip and is capable of interfacing with other Bluetooth compliant devices.

Contents

1 Selecting a Bluetooth Solution

Consider the following parameters to select a Bluetooth solution that best fits the application needs:

- Throughput
- Profiles Supported
- Low Power Specifications
- Recognized Brand
- Support Model
- Communication Interfaces
- Protocols Supported such as H3, H4, H4ED, BCSP
- Drivers for the Operating System such as WinCE, Linux
- Norms

1.1 Throughput

Throughput is the measure of maximum speed with which the chip transmits data over a period of time. Normally the period of time is in seconds. This metric is taken into consideration by the Bluetooth communication when a large amount of data is transmitted using the Serial Port Profile (SPP).Compare the throughput specification of the Bluetooth solution with the throughput needed for the project requirements.

1.2 Profiles Supported

Check if the selected Bluetooth solution supports the Bluetooth profiles required for the application. At times, the Bluetooth solution supports all the profiles through the standard Bluetooth profiles that come by default with the Operating System or the Bluetooth solution maker provides the Bluetooth profiles for integrating into the Operating System or the Bluetooth solution supports some specific profiles embedded in the Bluetooth IC. Still it is necessary to confirm with the Bluetooth solution maker, if the Bluetooth solution supports the Bluetooth profiles required for the application and the way these Bluetooth profiles are supported by the Bluetooth solution.

1.3 Low Power Specifications

Applications with the Bluetooth capability are usually handheld, and have specific power consumption requirements. Verify if the selected Bluetooth solution is compliant with the low power requirements of the application in sleep mode, run mode, doze mode and so on. It is important to check with the Bluetooth solution maker for the procedure to turn off the Bluetooth solution and send to sleep mode.

1.4 Recognized Brand

Check if there are any devices available in the market pertaining to the brand of the Bluetooth solution selected. This would imply a matured Bluetooth solution and also increases the responsibility of the Bluetooth solution maker to have products with their Bluetooth solution.

1.5 Support Model

Determine the support model of the Bluetooth solution maker as it can impact the cost of the product and the time to market principally.

1.6 Communication Interfaces

Find the communication interfaces available in the Bluetooth solution chip. These interfaces must be compatible with the communication interfaces in the i.MX. Generally, the Bluetooth solution supports UART, USB and SPI.

1.7 Protocols Supported

Select the Bluetooth host transport protocol (H3, H4, H4ED and BCSP) to be used with the Bluetooth solution to communicate with the i.MX Processor. This selection depends on the type of application, environment in which the product operates, cost and so on.

The most common protocols used in the market are H4 and BCSP (CSR proprietary). The H4 protocol is defined by the Bluetooth standard and it cannot detect parity errors, data omissions and corruption. It cannot implement recovery strategy. BCSP was defined by CSR and it detects parity errors, data omissions, corruption and so on.

After defining the host transport protocol that best fits the project requirements, confirm that the Bluetooth solution supports it.

1.8 Drivers for the Operating System

The Bluetooth chip maker must provide Software drivers for WinCE 5.0 or 6.0 depending on the WinCE version to be used in the project and the instructions to install it into a WinCE image. It is possible that the Bluetooth solution maker uses the standard profiles and H4 host transport protocol provided by WinCE.

1.9 Norms

Verify if the Bluetooth solution selected is compliant with the international norms required for the end product. Also check if the end product depends on the target market (medical, automotive, home, industrial and so on).

Consider the following norms depending on the target market.

- FCC (http://www.fcc.gov/)
- NOM (http://www.economia-noms.gob.mx/)
- UL (http://www.ul.com/)
- CE (http://ec.europa.eu/enterprise/standards_policy/index_en.htm)
- Jedec (http://www.jedec.org/)
- PbFree (http://www.pb-free.org/)

2 Selecting UART Port in the i.MX Chip

Consider the following parameters to select the i.MX UART port that best fits the application needs:

- Port
- Set of Pins
- Serial Configuration
- Direct Memory Access (DMA) support

2.1 Port

Select a port that has not been used by other modules. Check if this port can be connected easily to the Bluetooth solution in schematic and routing.

2.2 Set of Pins

Some modules in the i.MX chip require a couple of signals to be connected (such as UART, SPI). These signals can be mapped to different pins in the chip through the registers of i.MX Processor. Each option of pins that can map the signals is called set of pins.

For UART and other WinCE serial drivers in Freescale BSPs, the driver is fixed to use one set of pins. Therefore, if the project uses the other set of pins, the serial driver must be reconfigured as mentioned in Section 5.1.2, “UART Port is Configured with Pins that do not Match with the Hardware.” Verify with the hardware designers team for the set of pins that can be used based on the routing, availability, and modules of the chip to be used.

2.3 Serial Configuration

For the configuration options and electrical specifications, see the Serial Module (UART) Section of i.MX chip Reference Manual. Check if these options and specifications are compatible with the Bluetooth solution chip. For example, in case of UART communication, check if i.MX UART module and the Bluetooth solution are compatible with speed, voltages, parity, flow control, stop bits and so on.

2.4 Direct Memory Access (DMA) Support

Each BSP has its own DMA configuration for use with the Serial Module (UART). Check if the default DMA configuration is accurate with the requirements, otherwise configure the DMA.

3 Integrating into the i.MX WinCE BSPs

The minimal information and/or files required for the Bluetooth solution chip working on the i.MX WinCE BSPs are as follows:

- The HCI Transport Layer driver—It is essential for successful communication with the Bluetooth solution. There are two possible scenarios for this driver which are as follows:
  - — The Bluetooth solution provides a custom HCI Transport Layer driver. The Bluetooth driver maker must specify if the custom HCI Transport Layer driver needs the Universal Loadable

Driver provided by WinCE through the catalog item “Bluetooth Stack with Universal Loadable Driver”.

— The Bluetooth solution specifies that its HCI Transport Layer works with the standard H4 protocol provided by the WinCE through the catalog item “Bluetooth Stack with Integrated UART Driver”.

Figure 1 shows options in HCI catalog.

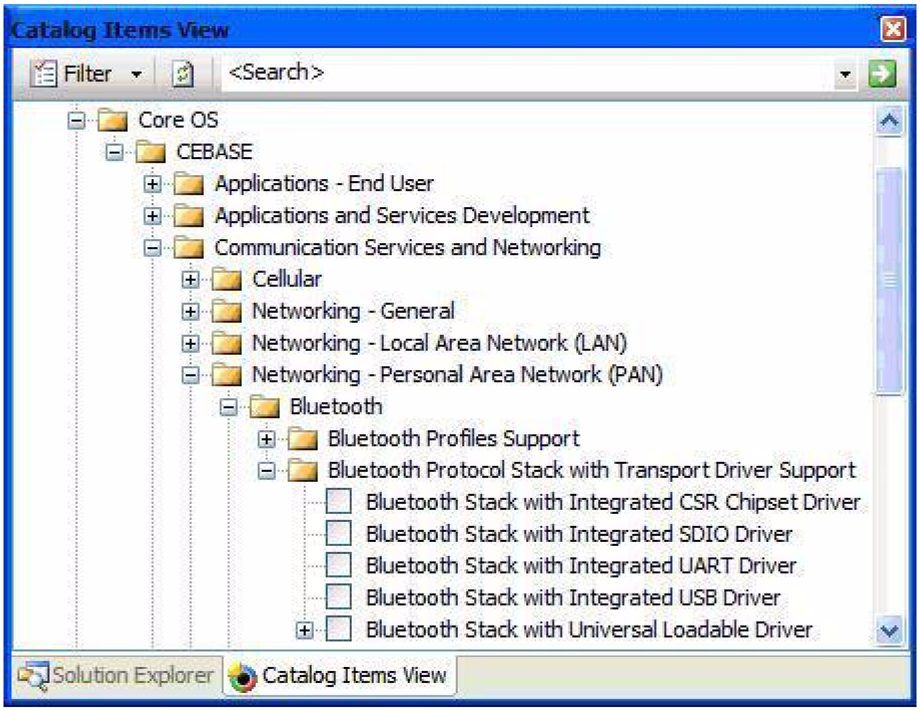

Catalog Items View
Filter
<Search>
Core OS
CEBASE
Applications - End User
Applications and Services Development
Communication Services and Networking
Cellular
Networking - General
Networking - Local Area Network (LAN)
Networking - Personal Area Network (PAN)
Bluetooth
Bluetooth Profiles Support
Bluetooth Protocol Stack with Transport Driver Support
Bluetooth Stack with Integrated CSR Chipset Driver
Bluetooth Stack with Integrated SDIO Driver
Bluetooth Stack with Integrated UART Driver
Bluetooth Stack with Integrated USB Driver
Bluetooth Stack with Universal Loadable Driver
Solution Explorer
Catalog Items View

Figure 1. HCI Catalog Items Options

- If required, use the Windows Registry configuration specifications to configure the Bluetooth host transport driver. These Registry configuration specifications are implemented in the registry file `platform.reg` located at `{WINCE DIR}\PLATFORM\{i.MX platform DIR}\FILES` through direct registry keys inputs or through a link to other registry file. Generally, the registry configurations are related to the UART communication settings such as speed, parity, bits, mode and so on.
- Include the following files for using the profiles— libraries (dll files), HCI driver, executable files, test files and so on. To include the files specified by the Bluetooth solution maker in the image, add lines of code with the filename and the startup allocation to the file `platform.bib` located at `{WINCE DIR}\PLATFORM\{i.MX platform DIR}\FILES`. There are some profiles included to select in WinCE such as PAN, LAP, DUN, HID, HSP, HFP and others.
- Instructions about the procedure to include the Bluetooth solution driver into Catalog Items View. This is optional.
- Dependence in the Catalog Items View. Generally they include OBEX items.

Refer to the document that comes with the software package of the Bluetooth driver maker such as Readme.txt, help files, quick start guides and so on. It gives a detailed description about the implementation of the Bluetooth solution driver into the i.MX BSPs.

4 Suggested Tests

Perform the following steps to test the communication through the Bluetooth solution chip.

4.1 Testing the Serial Communication—Transmit

- Requirements:
  — i.MX Processor Board or Evaluation Board (EVB) with the UART Transmit path available to connect an oscilloscope probe.
  — Oscilloscope
- Assumptions:
  — The board is powered ON and the user can download NK image. For more information, see *WinCE BSP User Guide*.
- Development:
  1. Develop a test program that enables the selected UART port and sends data in an infinite loop.
  2. Build the image, download it into the board and execute.
  3. Verify whether the Transmit path has signal or data in the path using an oscilloscope probe. In case of any issue, see Section 5, "Troubleshooting."
- Success criteria: A signal/data must be observed in the Transmit path.

4.2 Logical Testing of Serial Communication—Transmit/Receive

- Requirements:
  — i.MX Processor Board or EVB.
- Assumptions:
  — The board is powered ON and the user can download NK image For more information, see *WinCE BSP User Guide*.
- Development:
  1. Configure the UART driver in the port selected for loopback mode by connecting the UART Transmit path with the UART Receive path internally in the i.MX chip.
  2. Develop a test program that enables the selected UART port, sends data and waits for receiving the data sent by polling or interrupts with a time out. The send and the receive process could be in an infinite loop.
  3. Add a breakpoint or insert print debug instructions in the receive process for success and time out.
  4. Build the image, download it into the board and execute.
  5. Verify whether the system stops at the breakpoint instruction or the debug window shows specified print information for success. In case of any issue, see Section 5, "Troubleshooting."
- Success criteria: A signal/data must be observed in the Transmit/Receive path and the system must stop at the breakpoint instruction or the debug window must show specified print information for success.

4.3 Hardware Testing of Serial Communication—Transmit/Receive

- Requirements:
  - — i.MX Processor Board or EVB with the UART Transmit/Receive paths available to connect an oscilloscope probe.
  - — The Transmitter must be physically connected with the Receiver path of the same UART port selected.
  - — Oscilloscope
- Assumptions:
  - — The board is powered ON and the user can download NK image. For more information, see *WinCE BSP User Guide*.
- Development:
  1. Develop a test program that enables the selected UART port, sends data and waits for receiving the data sent by polling or interrupts with a time out. The send and the receive process could be in an infinite loop.
  2. Add a breakpoint or insert print debug instructions in the receive process for success and time out.
  3. Build the image, download it into the board and execute.
  4. Verify whether the Transmit and the Receive paths have the same signal/data using an oscilloscope.
  5. Verify whether the system stops at the breakpoint instruction or the debug window shows specified print information for success. In case of any issue, see Section 5, “Troubleshooting.”
- Success criteria: A signal/data must be observed in the Transmit/Receive path and the system must stop at the breakpoint instruction or the debug window must show the specified print information for success.

4.4 Testing the Bluetooth Solution Chip Communication

- Requirements:
  - — i.MX Processor Board or EVB and the Bluetooth solution connected through a UART port with the UART Transmit/Receive paths available to connect an oscilloscope probe.
  - — Oscilloscope
  - — The Bluetooth solution chip has some defined test commands in its specification.
- Assumptions:
  - — The board is powered ON and the user can download NK image. For more information, see *WinCE BSP User Guide*.
- Development:
  1. Develop a test program that enables the UART port selected, sends the test command defined in the Bluetooth solution specification and waits for a valid response into a time out window. The send and the receive process could be in an infinite loop.
  2. Add a breakpoint or insert print debug instructions in the receive process for success and time out.

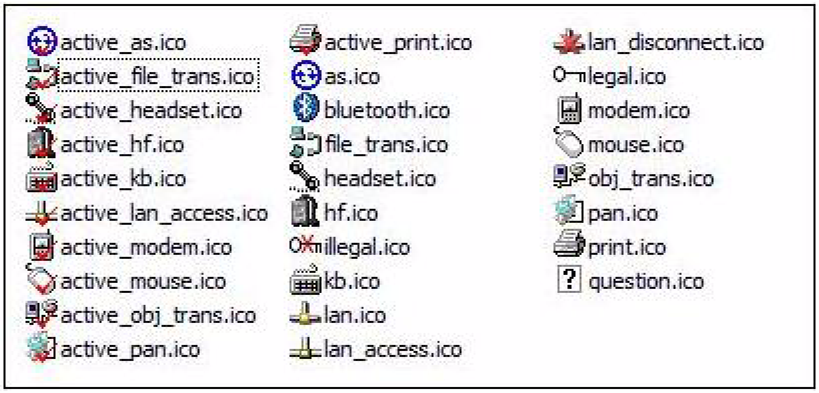
3. Build the image, download it into the board and execute.
4. Verify whether the Transmit and the Receive paths have signal/data using an oscilloscope.
5. Verify whether the system stops at the breakpoint instruction or the debug window shows specified print information for success. In case of any issue, see Section 5, “Troubleshooting.”

- Success criteria: A signal/data must be observed in the Transmit/Receive paths and the system must stop at the breakpoint instruction or the debug window must show the specified print information for success.

4.5 Testing the Bluetooth Solution

- Requirements:
  — i.MX Processor Board or EVB with the Bluetooth solution connected through a UART port.
  — A Bluetooth device that supports the Bluetooth profile.
- Assumptions:
  — The board is powered ON and has flashed the WinCE image. The Bluetooth driver and the profiles to be tested are included. For more information, see *WinCE BSP User Guide*.
- Development:
  1. Turn ON the board, open Control Panel and open Bluetooth Manager window.
  2. Scan for the Bluetooth devices and the profile needed. Each profile is represented by a unique image. Each icon filename describes the profile which is represented in `{WINCE DIR}\PUBLIC\COMMON\OAK\DRIVERS\NETUI`. The Bluetooth device documentation explains the procedure to search from the Bluetooth PAN.

Figure 2 shows icons in the Bluetooth Profile.



active_as.ico
active_file_trans.ico
active_headset.ico
active_hf.ico
active_kb.ico
active_lan_access.ico
active_modem.ico
active_mouse.ico
active_obj_trans.ico
active_pan.ico
active_print.ico
as.ico
bluetooth.ico
file_trans.ico
headset.ico
hf.ico
illegal.ico
kb.ico
lan.ico
lan_access.ico
lan_disconnect.ico
legal.ico
modem.ico
mouse.ico
obj_trans.ico
pan.ico
print.ico
question.ico

Figure 2. Bluetooth Profile Icons

3. Click device profile item and select trust. Depending on the Bluetooth device to pair, an authentication would be required. Typically the key is “0000”.
4. Activate the Bluetooth profile by clicking device profile item and select activate.

NOTE

Depending on the Bluetooth solution driver specifications, the above two steps (3 and 4) can vary.

5. Verify whether the Bluetooth profile works in accordance with the Bluetooth solution chip and the Bluetooth device documents.

- Success criteria: No issues must be observed.

5 Troubleshooting

This section explains the common errors and its solutions when a Bluetooth solution is integrated into the i.MX BSPs.

5.1 No Traffic in the UART Transmit Path

This error signifies that the WinCE test program created by the developer for the UART opens the UART port and sends data in an infinite loop.

There are three possible scenarios related to this problem:

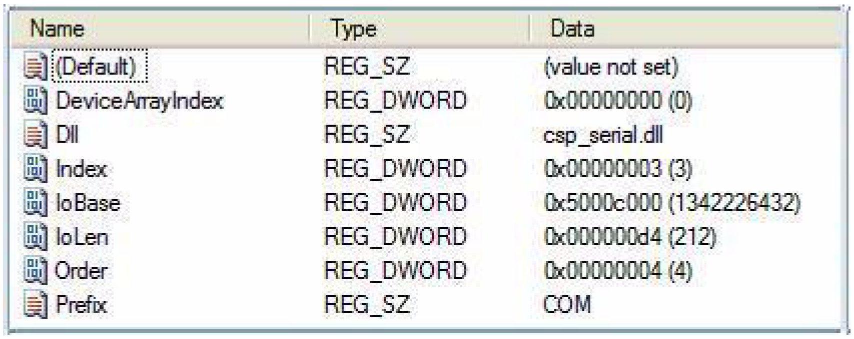
- WinCE COM port does not correspond to the physical UART port. The developer uses an incorrect WinCE COM port.
- UART port used is configured with pins that do not match with the hardware.
- Custom board swaps the Transmit/Receive UART path.

5.1.1 WinCE COM Port does not Correspond to the Physical UART Port

This problem arises when the BSP uses the COM port numbers that corresponds to the physical UART port numbers for other modules such as SPI, VSP and so on.

To verify the physical UART port associated with the WinCE COM port, check in the file `platform.reg` located at `{WINCE DIR}\PLATFORM\{i.MX platform DIR}\FILES` under `HKEY_LOCAL_MACHINE\Drivers\BuiltIn\COMn` registry key, its associated i.MX registry address under the property IoBase. This address can be found in the i.MX chip Reference Manual associated with a specific UART port of the chip. With this information, WinCE COM port that corresponds to the i.MX UART port can be found.

Figure 3 shows an example of contents in the Registry Key `HKEY_LOCAL_MACHINE\Drivers\BuiltIn\COMn`.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DeviceArrayIndex	REG_DWORD	0x00000000 (0)
Dll	REG_SZ	csp_serial.dll
Index	REG_DWORD	0x00000003 (3)
IoBase	REG_DWORD	0x5000c000 (1342226432)
IoLen	REG_DWORD	0x000000d4 (212)
Order	REG_DWORD	0x00000004 (4)
Prefix	REG_SZ	COM

Figure 3. An Example of Content in the Registry Key

Verify in the Windows Registry configuration whether the keys related to the Bluetooth solution driver has configured the correct COM port.

5.1.2 UART Port is Configured with Pins that do not Match with the Hardware

Some modules in the i.MX chips and some UART ports have two sets of pins for TX, RX, RTC, CTS, DTR, and DSR in the chip. The set of pins to be used by the WinCE UART driver are hard configured by the software. This implies that the WinCE UART driver must be updated if the other set of pins in the i.MX chip are used.

This problem arises when the hardware team selects a specific set of pins (for easy layout, easy map) for modules like UART that are not the default set of pins used by the driver.

In case of UART, this issue can be solved by updating the file `bspserial.c` located at `{WINCE DIR}\PLATFORM\{i.MX platform}\SRC\DRIVERS\SERIAL` and the function `BSPUartConfigureGPIO`. This function receives the base address as parameter for the UART to be configured. Go to the UART to be configured in the switch and configure the correct set of pins. For more information about the names of the pins and their alternative functions for UART, see *i.MX Reference Manual*.

5.1.3 Custom Board Swaps the Transmit/Receive UART Path

This problem arises when the transmit and the receive lines are swapped. This can be confirmed by using a WinCE test applications that opens the UART port and sends data in an infinite loop. Check whether the Transmit path does not show any signal using an oscilloscope. Check if the Receive path shows data.

The solution to the above problem involves board rework that includes swapping the Receive and the Transmit paths and updating the schematic to correct from origin for future releases.

5.2 There is Traffic in the Transmit/Receive Lines but the Bluetooth Solution is not Recognized by WinCE

This error implies that WinCE program is establishing communication with the Bluetooth solution chip through test communication messages, defined by the Bluetooth solution chip, or through a production application.

There are three possible scenarios related to this problem:

- WinCE COM port configuration does not correspond to the Bluetooth solution UART configuration.
- Selected protocol does not correspond to the protocol configured in the Bluetooth solution chip.
- The Bluetooth solution is not in the Operational Mode.

5.2.1 WinCE COM Port Configuration does not Correspond to the Bluetooth Solution UART Configuration

Verify whether the baud rate, data length, parity, stop bits and flow control of UART in WinCE and in the Bluetooth solution chip specifications are compatible.

5.2.2 Selected Protocol does not Correspond to the Protocol Configured in the Bluetooth Solution Chip

The Bluetooth specification defines host transport protocols H3 and H4. But, lack of recovery and errors detection mechanisms has forced some companies to develop their own proprietary Bluetooth host transport protocol specifications. For example, CSR has developed the host transport protocol BCSP.

To communicate correctly, the WinCE Bluetooth stack with the Bluetooth solution chip must use the same Bluetooth host transport protocol. The specifications of the Bluetooth solution chip must describe the Bluetooth host transport protocols supported and the ways by which it can be configured.

5.2.3 Bluetooth Solution is not in the Operational Mode

Verify whether the power levels of the Bluetooth solution are according to the Bluetooth solution specifications. Check the functionality of reset pin as well.

Normally in Operational Mode, reset pin must be active-high.

6 References

This application note has been written using the following references:

- Freescale i.MX platforms documentation http://www.freescale.com/imx
- Windows Embedded CE 6.0 Online Help.
- Bluetooth web page: http://www.bluetooth.org
- CSR web page: http://www.csr.com

7 Revision History

Table 1 provides a revision history for this application note.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
0	01/2010	Initial release.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.