Abstract
This document describes how to build and use the Windows Embedded RIL library for u-blox cellular modules on Windows Embedded CE 6.0 / Windows Embedded Compact 7 / Windows Mobile 6.5 OS operating systems.
This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.

u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice.

u-blox reserves all rights to this document and the information contained herein. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2015, u-blox AG.
Contents

1 Introduction .................................................................................................................. 5

2 Production delivery ...................................................................................................... 6
   2.1 Software release ................................................................................................. 6
   2.2 Package contents for Windows Embedded ......................................................... 6
   2.3 Package contents for Windows Mobile ................................................................. 6

3 Host PC configuration for Windows Embedded CE 6.0 ............................................. 7

4 Host PC configuration for Windows Embedded Compact 7 ...................................... 9

5 Host PC configuration for Windows Mobile 6.5 ......................................................... 10

6 Build source code ....................................................................................................... 11
   6.1 Windows CE 6 / Windows EC 7 ......................................................................... 11
      6.1.1 Environment Variable required ................................................................. 12
   6.2 Windows Mobile 6.5 ........................................................................................ 14

7 Project configuration ................................................................................................... 15
   7.1 Windows CE 6.0 / Windows EC 7 ................................................................. 15
      7.1.1 Catalog .................................................................................................... 15
      7.1.2 <platform>.bat ....................................................................................... 19
      7.1.3 sources.cmn ........................................................................................... 20
      7.1.4 platform.bib ......................................................................................... 20
      7.1.5 platform.reg ........................................................................................ 21
      7.1.6 710mux.reg ............................................................................................ 21
      7.1.7 ril_leon.reg ........................................................................................... 22
      7.1.8 ril_usb.reg ............................................................................................. 23
   7.2 Windows Mobile 6.5 ......................................................................................... 24
      7.2.1 <platform>.bat ....................................................................................... 24
      7.2.2 platform.bib ........................................................................................... 25
      7.2.3 platform.reg ........................................................................................ 25
      7.2.4 710mux.reg ............................................................................................ 26
      7.2.5 rilgsm.reg .............................................................................................. 27

8 Setting up the data connection .................................................................................. 28
   8.1 Windows Embedded CE 6.0 ............................................................................. 28

9 Debug RIL .................................................................................................................... 32
   9.1 Host PC configuration ....................................................................................... 32
A  Compatibility matrix ........................................................................................................................................34
  A.1  Windows software deliveries compatibility matrix ......................................................................................34
  A.2  Supported interfaces compatibility matrix ....................................................................................................34

Related documents .............................................................................................................................................35

Revision history ....................................................................................................................................................35

Contact ...............................................................................................................................................................36
1 Introduction

⚠️ u-blox assumes no responsibility for inappropriate use of RIL by customers.

✍️ The document applies to the following software deliveries:
- Windows Embedded CE 6.0
- Windows Embedded Compact 7
- Windows Mobile 6.5

✍️ The software has been developed and tested on either of the following platform:
- Windows Emulator (DEVICEEMULATOR/VIRTUALPC)
- BeagleBoard-xM Platform Rev C (SBC)

See Appendix A.1 and A.2 for the list of Windows software deliveries and interfaces supported by u-blox cellular modules.

✍️ In the following sections <name> indicates a parameter that can:
- be customized
- be set according to system configuration
- provide software version

The following sections describe the production packages delivered by u-blox.
An overview of the system setup is provided as well as the procedure to perform a log.
The following symbols are used to highlight important information within this document:

🔍 An index finger points out key information pertaining to integration and performance.

⚠️ A warning symbol indicates actions that could negatively impact performance or damage the device.
2 Production delivery

2.1 Software release

The delivery consists of the RIL library source code.

Connecting the u-blox cellular module to the platform can be done via UART serial port or USB port depending on the series module.

RIL driver does not offer standard interface for power off module, so it should be necessary to modify power off manager and send command AT+CPWROFF using an available AT command port.

2.2 Package contents for Windows Embedded

The RIL production delivery is provided in the RIL_we<version>.zip compressed file with the content structured as follows. Some components may not be present in some releases due to non-applicability:

RIL_we<version>.zip
  ril_we<version>
    SRC\DRIVERS\710MUX
    SRC\DRIVERS\RILGSM
    SRC\INC

  Source overlay for Windows platform
  Multiplexing driver files
  RIL core directory
  Include directory

2.3 Package contents for Windows Mobile

The RIL production delivery is provided in the RIL_wm<version>.zip compressed file with the content structured as follows:

RIL_wm<version>.zip
  ril_wm<version>
    SRC\DRIVERS\710MUX
    SRC\DRIVERS\RILGSM
    SRC\DRIVERS\SERIAL
    SRC\INC

  Source overlay for Windows platform
  Multiplexing driver files
  RIL core directory
  Serial drivers
  Include directory
3 Host PC configuration for Windows Embedded CE 6.0

The host development environment for Window Embedded CE 6.0 is based on the Windows operating system. Verify the platform requirements for the Windows distribution according to the tutorial, manual, etc. related to the specific platform’s distribution. The required software for development is:

- Visual Studio 2005
- Windows Embedded CE 6.0

An evaluation version of the software on Windows Embedded CE 6.0 can be downloaded from the internet [1]. To create the environment for Windows Embedded CE 6.0, download and install the following software packages in this order:

1. Visual Studio 2005
3. Windows Embedded CE 6.0
4. Windows Embedded CE 6.0 Platform Builder Service Pack 1
5. Windows Embedded CE 6.0 R2
6. Windows Embedded CE 6.0 R3
8. Windows Embedded CE 6.0 Monthly Updates (QFEs)

Follow these steps:

Step 1:
- Install Visual Studio 2005
- Install SP1 for Visual Studio 2005

Step 2:
- Install Windows Embedded CE 6.0 (integrated with Visual Studio 2005 using the Platform Builder for CE 6.0 development toolkit)
  - For the source code reference, other processor architectures can be selected and the shared source license can be accepted

Step 3:
- Install Windows Embedded CE 6.0 Platform Builder SP1, R2 & R3
- Install Windows Embedded CE 6.0 Cumulative Product Update Rollup Package
  - Download and run the .msi file corresponding to the CPU architecture installed in Step 2
- Install Windows Embedded CE 6.0 monthly updates in order of release (January, February,…etc)
  - Download and run the .msi files corresponding to the CPU architecture installed in Step 2

Step 4:
- The first time VS 2005 is launched, set the default environment settings. Be sure to select “Platform Builder Development Settings”

On Windows 7, it is possible to have an error during the monthly update installation, as shown in Figure 1. To avoid this error, set “Run in compatibility mode for previous Windows system”, as in Figure 2, in “Compatibility Tab” of File properties.
A directory is created in which Windows Embedded CE 6.0 source code is installed. This directory is called `<winemb>` in the following sections.
4 Host PC configuration for Windows Embedded Compact 7

The host development environment for Window Embedded Compact 7 is based on the Windows operating system.
Verify the platform requirements for the Windows distribution according to the tutorial, manual, etc. related to the specific platform's distribution.
The required software for development is:

- Visual Studio 2008
- Windows Embedded Compact 7

An evaluation version of the software on Windows Embedded Compact 7 can be downloaded from the internet [2].
To create an environment for Windows Embedded Compact 7, download and install the following software packages in this order:
1. .NET Framework 3.5
2. Visual Studio 2008
3. Visual Studio 2008 Service Pack 1
4. Windows Embedded Compact 7
5. Windows Embedded Developer Update

Follow these steps:
Step 1:
- Install .NET Framework 3.5
- Install Visual Studio 2008
- Install Visual Studio 2008 Service Pack 1

Step 2:
- Install Windows Embedded Compact 7
  o For the source code reference, other processor architectures can be selected and the shared source license can be accepted

Step 3:
- Update Visual Studio 2008 and Windows Embedded Compact 7 using Windows Update from “Control Panel” menu

Now a directory is created in which Windows Embedded Compact 7 source code is installed. This directory will be called <winemb> in the following sections.
5 Host PC configuration for Windows Mobile 6.5

The host development environment for Window Mobile 6.5 is based on the Windows operating system.

Verify the platform requirements for the Windows distribution according to the tutorial, manual, etc. related to the specific platform’s distribution.

The required software for development is:

- Visual Studio 2005
- Platform Builder for Windows Mobile 5.0

To create an environment for Windows Mobile 6.5, download and install the following software packages in this order:

1. Visual Studio 2005
2. Platform Builder for Windows Mobile 5.0
   - Choose Tools Only option
   - a. At the end of the installation, it is prompted to install BetaDEVS: press Cancel, not required
3. Windows Mobile 6.5 AKU (currently using AKU6530)
4. Visual Studio 2005 Service Pack 1
5. ActiveSync 4.5
6. .NET Compact Framework 2.0 SP2 – usually comes with the SDK, see the next step
7. Windows Mobile 6 Professional SDK Refresh
8. Platform Builder QFE2 update
   - a. This comes with one dll file (OsAxsHCe50.dll): copy this on the PC into C:\Program Files\Platform Builder for Windows Mobile\5.00\CEPB\BIN
   - b. No need to install QFE1
9. Reboot the PC

A directory is created in which Windows Mobile 6.5 source code is installed. This directory is called <winemb> in the following sections.
6 Build source code

6.1 Windows CE 6 / Windows EC7

- The package installation described in previous sections created a directory `<winemb>` with Windows Embedded
- Board Support Packages (BSP) installation will create `<platform>` directory with specific drivers for device
- Create a directory (e.g. `<ril_dir>`) and uncompress the RIL file

Decrypting the compressed RIL file requires the password provided by u-blox.

```bash
mkdir <ril_dir>
cd <ril_dir>
cp <path_of_ril>
\RIL_we_<version>.zip .
unzip -P <ril_password> RIL_we_<version>.zip
```

- Copy RIL's source code in the Window Embedded distribution
  ```bash
cp ril_we_<version>\SRC\DRIVERS <winemb>\PLATFORM\<platform>\SRC\DRIVERS
cp ril_we_<version>\SRC\INC\* <winemb>\PLATFORM\<platform>\SRC\INC\n```
- Modify file `<winemb>\PLATFORM\<platform>\SRC\DRIVERS\dirs` adding the following lines
  ```
  710MUX\ (Not required with USB configuration)
  RILGSM\n  ```
- See the section 7 to modify the platform configuration
- Open Project > Properties and select in “Configuration” combo-box “Release Build” option as in Figure 3

![Figure 3: Project properties](image-url)
Windows Embedded RIL - Application Note

- Enable the following building options in Project > Properties > Build options menu as in Figure 3.
  Enable eboot space in memory (IMGEBOOT=1)
  Enable KITL (no IMGNOKITL=1)

6.1.1 Environment Variable required

The following step applies only to TOBY-L2 series modules.

- Environment Variable required: enables the following building options in Project > Properties > Environment menu as in Figure 4. This environment variable includes the USB serial driver which is used by TOBY-L2.

![Figure 4: Project properties](image-url)
- Build the Windows Embedded system using “Build” menu (Build > Build Solution) as in Figure 5

![Build Solution Menu]

Figure 5: Build solution

- Insert a microSD card (minimum 2 GB) into the PC
- Use the command provided in the platform’s distribution for an SD card creation for the platform

⚠️ **This step can delete the hard disk drive if the SD card device name entered after the script name is incorrect.**

- Insert the SD card into the board’s SD slot
- Connect the board to the u-blox cellular module using an interface as described in section A.2
- Power on both devices
6.2 Windows Mobile 6.5

- The package installation, described in previous sections, created a directory <winemb> with Windows Mobile
- Board Support Packages (BSP) installation will create <platform> directory with specific drivers for device
- Create a directory (e.g. <ril_dir>) and uncompress the RIL file

Decrypting the compressed RIL file requires the password provided by u-blox.

```
mkdir <ril_dir>
cd <ril_dir>
cp <path_of_ril>\RIL_wm_sc_<version>.zip .
unzip -P <ril_password> RIL_wm_sc_<version>.zip
```

- Copy the RIL source code to the Windows Embedded distribution
  ```
  cp ril_wm_<version>\SRC\DRIVERS <winemb>\PLATFORM\<platform>\SRC\DRIVERS
  cp ril_wm_<version>\SRC\INC\* <winemb>\PLATFORM\<platform>\SRC\INC
  ```

- Modify the file in <winemb>\PLATFORM\<platform>\SRC\DRIVERS\dirs adding the following line:
  ```
  710MUX
  ```
- See section 7 for how to modify the platform configuration
- Open the command window and then navigate to <winmob>\BuildScripts\<platform>
- Run the BuildAll.bat command
- Insert a microSD card (minimum 2 GB) into the PC
- Use the command provided in the platform’s distribution for an SD card creation for the platform

This step can delete the hard disk drive if the SD card device name entered after the script name is incorrect.

- Insert the SD card into the board’s SD slot
- Connect the board to the u-blox cellular module using an interface as described in section A.2
- Power on both devices
7 Project configuration

7.1 Windows CE 6.0 / Windows EC 7

Windows Embedded RIL requires modifications in the following components of Board Support Packages (BSP):

- Catalog
- `<platform>.bat`
- `sources.cmn`
- `platform.bib`
- `platform.reg`
- `710mux.reg`
- `ril_leon.reg`
- `ril_usb.reg`

7.1.1 Catalog

Click on “Catalog Items View” tab (or View > Other Windows > Catalog Items View from the menu) and open “Catalog” as in Figure 6.

![Figure 6: Select Catalog Items View tab](image)
Under Core OS > <OSVersion> > Communication Services and Networking > Cellular > CELLCORE, verify that “All Modules” and “RIL Proxy Log” are included, as shown in Figure 7.

**Figure 7: Set CellCore and RIL Proxy Log**
Under Core OS > <OSVersion> > Shell and User Interface > Shell > Command Shell, verify that “Command Processor” and “Console Window” are included, as in Figure 8.

Figure 8: Set Command Processor and Console Window
Following steps are needed only for Windows CE 6.0.

Under Core OS > CEBASE > Communication Services and Networking > Networking – General > Connection Manager verify that “Connection Manager” and “Network Utilities” are included as shown in Figure 9.

The connection manager is part of Windows Embedded Compact 7.

Figure 9: Set Connection Manager and Network Utilities for Windows Embedded CE 6.0
Under “Device Drivers”, verify that “Windows Embedded CE Test Kit” is included as shown in Figure 10.

![Figure 10: Set Windows Embedded CE Test Kit for Windows Embedded CE 6.0](image)

### 7.1.2 `<platform>.bat`

Add to `<winemb>`\`PLATFORM\`<platform>\`<platform>\`.bat the following lines:

```batch
REM -- ------------------------------ RIL-specific variables
REM -- ------------------------------
set IMGLEONRIL=1
set IMGFAKERIL=

REM -- Logging/Debugging
REM
REM Enable RIL Proxy logging
set SYSGEN_RILPROXY_LOG=1

REM Enable AT Cmd logging in RETAIL mode (send/response)
set RIL_RETAIL_OUTPUT=1
```
REM Enable MUX debugging
set GSM0710_LOG_MSG=1
set GSM0710_ERR_MSG=1
REM
REM --End Logging/Debugging

REM Enable EONS (Enhanced Operator Name) support
set RIL_EONS=1

REM Disable location updates (enabled by default)
set RIL_NOLOCATIONUPDATES=

REM Enable cell broadcast
set RIL_ENABLE_CELL_BROADCAST=1

REM Enable Call Progress Notifications
set IMG_ENABLE_CALLPROG=1

REM The following must be set if we are using the GSM0710 MUX.
set IMGDUALPORTRIL=1
REM End of MUX settings.

7.1.3 sources.cmn
Verify that the following paths are inserted into <winemb>\PLATFORM\<platform>\sources.cmn:

_COMMONPUBROOT=${_PROJECTROOT}\cesysgen
_OEMINCPATH=${_COMMONPUBROOT}\sdk\inc;
_OEMINCPATH=${_OEMINCPATH};${_COMMONPUBROOT}\oak\inc;
_OEMINCPATH=${_OEMINCPATH};${_COMMONPUBROOT}\ddk\inc;

7.1.4 platform.bib
Add to <winemb>\PLATFORM\<platform>\FILES\platform.bib the following lines:

IF IMGLEONRIL
   rilgsm.dll   ${_FLATRELEASEDIR}\ril_leon.dll   NK SHK
   710mux.dll   ${_FLATRELEASEDIR}\710mux.dll   NK SHK
   usbcdc.dll   ${_FLATRELEASEDIR}\usbcdc.dll   NK SHK ENDIF

710mux.dll is required only with MUX configuration.

Usbcdc.dll is required with SARA-U series only. To use the driver, place the usbcdc.dll in <winemb>\PLATFORM\<platform>\target\ARMV4I\retail in CE6 or release directory in EC7.
7.1.5  platform.reg
The following files are added to include the registry settings in the BSP image.
Add to `<winemb>\PLATFORM\<platform>\FILES\platform.reg` the following lines:

```plaintext
; Enable RIL
IF IMGLEONRIL
#include "$(TARGETPLATROOT)\src\drivers\rilgsm\dll\ril_leon.reg"
#include "$(TARGETPLATROOT)\SRC\DRIVERS\710MUX\710mux.reg"
#include "$(TARGETPLATROOT)\SRC\DRIVERS\710MUX\ril_usb.reg"ENDIF ; IMGLEONRIL
```

ril_leon.reg and 710mux.reg are required with MUX configuration only.

ril_usb.reg is required with USB configuration only.

For SARA-U series module, add the following lines:

```plaintext
; Enable RIL
IF IMGLEONRIL
#include "$(TARGETPLATROOT)\src\drivers\rilgsm\dll\ublox_sara.reg"
ENDIF ; IMGLEONRIL
```

For TOBY-L2 series module, add the following lines:

```plaintext
; Enable RIL
IF IMGLEONRIL
#include "$(TARGETPLATROOT)\src\drivers\rilgsm\dll\ublox_toby.reg"ENDIF ; IMGLEONRIL
```

Also copy the `ubxdummy.dll` in "SD CARD" root to prevent pop up message requesting the driver for Flash loader device in Windows CE6/EC7.

7.1.6  710mux.reg
This file configures the 710mux driver. It is located in the `\SRC\DRIVERS\710MUX\` directory.

710mux.reg must be initialized after the port driver (e.g. serial port driver, SPI driver) and before the RIL driver.

Use `Order` parameter to set the booting sequence.

This driver creates three virtual ports: two ports are used by RIL and the other one could be used to send the commands directly to the module.

Use `Prefix` and `Index` parameters to set the virtual port name.

```plaintext
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x>]
 "Prefix"="<NAME>"
 "Dll"="710mux.dll"
 "DeviceArrayIndex"=dword:1
 "Index"=dword:<x>
 "Order"=dword:<ord>

[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x+2>]
```

"Prefix"="<NAME>"
"Dll"="710mux.dll"
"DeviceArrayIndex"=dword:1
"Index"=dword:<x+2>
"Order"=dword:<ord+1>

[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x+3>]
 "Prefix"="<NAME>"
 "Dll"="710mux.dll"
 "DeviceArrayIndex"=dword:1
 "Index"=dword:<x+3>
 "Order"=dword:<ord+2>

The standard configuration is:
 <x>=1
 <NAME>=VCA
 <ord>=2

Use the ComPort parameter for the 710mux driver port configuration as described below:

[HKEY_LOCAL_MACHINE\Software\Microsoft\GSM07_10]
 "ComPort"="<COM>:" - port exposed by the Mux

The standard configuration is <COM>=COM1.

7.1.7  ril_leon.reg

This file configures the RIL driver and is used with MUX configuration only. It is located in the \SRC\DRIVERS\RILGSM\DLL\ directory.

Initialize the RIL driver after the port driver i.e. the UART.

The Order parameter may be used to set the booting sequence if required

[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\RIL]
 "Order"=dword:<y> ;

When the 710mux driver is used, configure <y> > <x>. The standard configuration is <y>=4.

ComPort and DataPort parameters configure communication and data ports used by RIL.

[HKEY_LOCAL_MACHINE\Software\Microsoft\RIL]
 "ComPort"="<COMP>:" - AT command port exposed by Mux for use by RRIL
 "DataPort"="<COMD>:" - Data port used by upper layers for PPP

The standard configuration is
 <COMP>=VCA1
 <COMD>=VCA3

where VCA<n> is a virtual port created by the 710mux driver.
7.1.8  ril_usb.reg

This file also configures the RIL driver and is used with USB configuration only. It is located in the \SRC\DRIVERS\RILGSM\DLL\ directory.

The Order parameter may be used to set the booting sequence if required

    IF IMGLEONRIL
    [HKEY_LOCAL_MACHINE\Drivers\BuiltIn\RIL]
    ; Registry key to load the RIL driver at boot time
    "Dll"="rilgsm.dll"
    "Prefix"="RIL"
    ;"Order"=dword:50
    "Index"=dword:1
    "IClass"="{A32942B7-920C-486b-B0E6-92A702A99B35}"
    
    [HKEY_LOCAL_MACHINE\Software\Microsoft\RIL]
    "ComInitString"=""

    "ComPort"="COM1:" - AT command port
    "VirtualData1Port"="COM3:" - Data1 Port

    ; This is the port that is handed to the upper layers for PPP
    "DataPort"="COM3:"  

The above configuration of COM1 and COM3 assigned to RIL is tested on Windows Embedded Compact 7. For Windows CE6, COM1 and COM4 were assigned to the RIL and tested.

In any case, if the RIL does not start properly due to unavailability of COM port, an AT terminal available for Windows Embedded may be used to see the list of COM ports enumerated and check which ones are available and responding to AT commands. The available ones may then be assigned in ril_usb.reg.

In both Windows CE6/ EC7, the u-blox module has been assigned COM2.
7.2 Windows Mobile 6.5

Windows Mobile RIL requires modifications in the following components of Board Support Packages (BSP):

- `<platform>.bat`
- `platform.bib`
- `platform.reg`
- `710mux.reg`
- `ril_leon.reg`

7.2.1 `<platform>.bat`

Add to `<winmob>\PLATFORM`\<platform>\<platform>.bat` the following lines:

```cmd
@REM ************************************************************
@REM ********** u-blox RIL Integration Related Variables **********
@REM ************************************************************

set BSP_TWO_SERIAL_PORTS=
set BSP_NOSIR=1
set IMG_UBLOX_RIL=1
set IMGFAKERIL=
REM Turn on Call Progress Notifications
set IMG_ENABLE_CALLPROG=1

REM Enable if our modem supports separate AT command for number of PIN retries remaining.
set RIL_SUPPORT_ATCMD_PINCOUNT=1

REM The following setting sets the 2 [HKEY_LOCAL_MACHINE\Comm\Cellular\RIL](Packets=1 and Contexts=3) values
REM in PUBLIC\CELLCORE\OAK\FILES. Requires rebuilt 710MUX and RILGSM and makeimg!
set IMGNDISGPRS=1

REM Turn on RIL Proxy logging
REM set SYSGEN_RILPROXY_LOG=1

REM The following must be set if we are using the GSM0710 MUX.
set RIL_USE_GSM0710=1
set IMGDUALPORTRIL=1
set IMGRILCOM1=
REM End of MUX settings.

REM MUX Debug Flags
set GSM0710_LOG_MSG=
set GSM0710_ERR_MSG=1
```
REM Disable using network card in emulator.
set BSP_NORNDIS=1
set BSP_NOSHAREETH=1
set BSP_NOUSB=1

REM Enable AT Cmd logging in retail mode
set RIL_RETAIL_OUTPUT=1

@REM ********************
@REM ****************************
@REM ************* u-blox RIL Integration Ends *******************
@REM ****************************

7.2.2  platform.bib
Add to <winmob>\PLATFORM\<platform>\FILES\platform.bib the following lines:

IF IMG_UBLOX_RIL
 ; **************************** u-blox RIL Start ****************************
 ; **************************** u-blox RIL Start ****************************
 #include "$_TARGETPLATROOT)\SRC\DRIVERS\RILGSM\DLL\rilgsm.bib"

IF RIL_USE_GSM0710
    #include "$_TARGETPLATROOT)\SRC\DRIVERS\710MUX\710mux.bib"
ENDIF ; RIL_USE_GSM0710

 ; **************************** u-blox RIL End ****************************
 ; **************************** u-blox RIL End ****************************
ENDIF ;IMG_UBLOX_RIL

710mux.dll is needed only if a multiplexer driver must be used. Check the configuration variables to enable only the u-blox RIL configuration.

7.2.3  platform.reg
Add to <winmob>\PLATFORM\<platform>\FILES\platform.reg the following lines:

IF IMG_UBLOX_RIL
 ; **************************** u-blox RIL Start ****************************
 ; **************************** u-blox RIL Start ****************************
 #include "$_TARGETPLATROOT)\SRC\DRIVERS\RILGSM\DLL\rilgsm.reg"
IF RIL_USE_GSM0710
    #include "$(TARGETPLATFORM)\SRC\DRIVERS\710MUX\710mux.reg"
ENDIF; RIL_USE_GSM0710

; ***************************************************************
; ********************** u-blox RIL End **************************
; ***************************************************************
ENDIF; IMG_UBLOX_RIL

710mux.dll is needed only if a multiplexer driver must be used.
Check the configuration variables to enable only the u-blox RIL configuration.

### 7.2.4 710mux.reg

This file configures the 710mux driver. It is located in the \SRC\DRIVERS\710MUX\ directory.

710mux.reg must be initialized after the port driver (e.g. serial port driver, SPI driver) and before the RIL driver. Use the Order parameter to set the booting sequence.

This driver creates three virtual ports: two ports are used by RIL and the other one could be used to send commands directly to the module.

Use the Prefix and Index parameters to set the virtual port name.

```plaintext
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x>]
"Prefix"="<NAME>"
"Dll"="710mux.dll"
"DeviceArrayIndex"=dword:1
"Index"=dword:<x>
"Order"=dword:<ord>

[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x+2>]
"Prefix"="<NAME>"
"Dll"="710mux.dll"
"DeviceArrayIndex"=dword:1
"Index"=dword:<x+2>
"Order"=dword:<ord+1>

[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\GSM0710_<x+3>]
"Prefix"="<NAME>"
"Dll"="710mux.dll"
"DeviceArrayIndex"=dword:1
"Index"=dword:<x+3>
"Order"=dword:<ord+2>
```

The standard configuration is:

```plaintext
<x>=1
<NAME>=VCA
<ord>=2
```
Use the ComPort parameter for the 710mux driver port configuration as described below.

\[
\begin{align*}
\text{[HKEY_LOCAL_MACHINE\Software\Microsoft\GSM07_10]} & \\
\text{\quad "ComPort"} & = "<\text{COM}>:" \quad \text{port exposed by the Mux}
\end{align*}
\]

The standard configuration is \text{<COM>=COM1}.

### 7.2.5 \text{rilgsm.reg}

This file configures the RIL driver. It is located in the \text{\SRC\DRIVERS\RILGSM\DLL} directory. The RIL driver must be initialized after the port driver (e.g. USB driver or multiplexer driver). Use the \text{Order} parameter to set the booting sequence as described below.

\[
\begin{align*}
\text{[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\RIL]} & \\
\text{\quad "Order"} & = \text{dword:<y> ;}
\end{align*}
\]

When the 710mux driver is used, configure \text{<y> \text{<x>}}. The standard configuration is \text{<y>=4}.

The ComPort and DataPort parameters configure the communication and the data ports used by RIL.

\[
\begin{align*}
\text{[HKEY_LOCAL_MACHINE\Software\Microsoft\RIL]} & \\
\text{\quad "ComPort"} & = "<\text{COMP}>:" \quad \text{AT command port exposed by Mux for use by RRIL} \\
\text{\quad "VirtualDataPort"} & = "<\text{COMD}>:" \quad \text{Data port that MUX exposes} \\
\text{\quad "DataPort"} & = "<\text{COMD}>:" \quad \text{Data port used by upper layers for PPP}
\end{align*}
\]

The standard configuration is

\text{<COMP>=VCA1} \\
\text{<COMD>=VCA3}

where \text{VCA<n>} is a virtual port created by 710mux driver.
8 Setting up the data connection

For a correct system initialization, properly configure the APN on the platform.

The dialup registry settings configure the Remote Access Service (RAS) connections.

The RAS registry settings could be added to the hardware platform in the Platform.reg file (in `<winemb>`\Platform\<platform>\Files).

The [HKEY_CURRENT_USER\Comm\RasBook\<connection>] registry key contains information about the RAS phone book entries for the specified connection.

It is necessary to define two hexadecimal parameters:

- DevCfg: indicates the serial device configuration
- Entry: contains the RAS phone book entry for the connection

After the project is built, the data connection setting will be permanent. See the Windows Embedded CE 6.0 RAS book [4] and Windows Embedded Compact 7 RAS book [5] for further details.

In Windows Embedded CE 6.0 it is possible to set the APN on runtime but the connection setting is not permanent: see the section 8.1 for more details.

In Windows Embedded Compact 7 it is not possible to set APN on runtime.

TOBY-L2 series does not support data connection through this process.

8.1 Windows Embedded CE 6.0

The provision application could be used to set network connection. It is necessary to enable Kernel Independent Transport Layer (KITL), as described in the section 6 and the build provision application as described in this section.

Open a build window by clicking Build > Open Release Directory in Build Window as in Figure 11. A command shell is opened as in Figure 12.

![Figure 11: Open Release Directory in Build Window](image-url)
Go to `<winemb>\PUBLIC\CELLCORE\SDK\SAMPLES\PROVISION`, using the following command:

```
cd <winemb>\PUBLIC\CELLCORE\SDK\SAMPLES\PROVISION
```

Type `build`, and obtain a build of `provision.exe` that will be copied to `Release` directory. Figure 13 shows the build result.

Copy the `provision.xml` file from the same directory, with the command:

```
copy provision.xml $(FLATRELEASEDIR)
```
The provision.xml file contains the network's parameters. Modify it with the network operator settings.

Open the provision.xml file copied in the Release directory. Change and save the APN name and other parameters (i.e. GPRSInfoAccessPointName, etc).

Click Target > Target Control (or Alt + 1) menu as in Figure 14.

---

**Figure 13: Build provision**

![Image of build output with details]

Build output showing necessary steps to build the provisioning file.
Figure 14: Target control

Figure 15: Start provision

Type “s provision”, as in Figure 15. The Output window of Visual Studio should display these messages:

```
Opening file provision.exe from desktop
Opening file Provision.xml from desktop
ProcessXML: Process Succeeds!
```

In the device window, click Start > Settings > Network and Dial-up Connections. Double-click the connection created in the section 8. Click “Connect”. The “Connected” message should appear in the connection status box if the data connection is successful.

Click “Hide” button to hide the connection status box. Close the “Control Panel” window.
9 Debug RIL

9.1 Host PC configuration

The Kernel Independent Transport Layer (KITL) can be used for debugging purposes, after the creation of a build image and enabling these options:

- Enable KITL (no IMGNOKITL=1)
- Enable profiling (IMGPROFILER=1)

Go to Project > Properties and set the previous parameters.

Open the Windows Embedded project and configure the target connection by selecting “Target > Connectivity Options...” from the menu, as in Figure 16.

![Figure 16: Start connectivity options](image)

Set the “Connectivity options”, as in Figure 17. Set “Transport” according to the PS’s device connection.

![Figure 17: Set connectivity options](image)
Apply the settings and close “Target Device Connectivity Options” window.
On the Target menu, click Attach Device.
Boot Windows Embedded device and after the completion of the image download, it is possible to debug it using the KITL service. The debug messages will be displayed in the “Output” window of Visual Studio.
A Compatibility matrix

A.1 Windows software deliveries compatibility matrix

<table>
<thead>
<tr>
<th>Cellular module</th>
<th>Windows CE 6.0</th>
<th>Windows EC 7</th>
<th>Windows Mobile 6.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEON-G100 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>LEON-G200 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>LISA-U1 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>LISA-U2 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>SARA-G350 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>SARA-U series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>TOBY-L2 series</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>

Table 1: Windows software deliveries compatibility matrix

A.2 Supported interfaces compatibility matrix

<table>
<thead>
<tr>
<th>Cellular module</th>
<th>SPI</th>
<th>UART</th>
<th>USB</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEON-G series</td>
<td>Not available</td>
<td>Supported</td>
<td>Not available</td>
</tr>
<tr>
<td>LISA-U series</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>SARA-G350 series</td>
<td>Not available</td>
<td>Supported</td>
<td>Not available</td>
</tr>
<tr>
<td>SARA-U series</td>
<td>Not Available</td>
<td>Not Available</td>
<td>Supported</td>
</tr>
<tr>
<td>TOBY-L2 series</td>
<td>Not Available</td>
<td>Not Available</td>
<td>Supported</td>
</tr>
</tbody>
</table>

Table 2: Supported interfaces compatibility matrix
Related documents

[1] Windows Embedded CE 6 download page:

[2] Windows Embedded Compact 7 download page:

[3] Windows Embedded Compact 7 Development update page:


For regular updates to u-blox documentation and to receive product change notifications, register on our homepage.

Revision history

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Name</th>
<th>Status / Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>20-Dec-2011</td>
<td>fpic</td>
<td>Initial release (Last revision with old doc number, WLS-CS-12001)</td>
</tr>
<tr>
<td>1</td>
<td>07-Jun-2013</td>
<td>fpic</td>
<td>Inserted Windows Mobile 6.5 delivery Added compatibility matrix Extended to include SARA-G3 series</td>
</tr>
<tr>
<td>R03</td>
<td>22-Dec-2014</td>
<td>bqam</td>
<td>Extended to include SARA-U and TOBY-L2 series Inserted support for Windows EC7 Added notes in project configuration files</td>
</tr>
</tbody>
</table>
Contact

For complete contact information visit us at www.u-blox.com

u-blox Offices

North, Central and South America
u-blox America, Inc.
Phone: +1 703 483 3180
E-mail: info_us@u-blox.com
Regional Office West Coast:
Phone: +1 408 573 3640
E-mail: info_us@u-blox.com
Technical Support:
Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters
Europe, Middle East, Africa
u-blox AG
Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific
u-blox Singapore Pte. Ltd.
Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com
Regional Office Australia:
Phone: +61 2 8448 2016
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com
Regional Office China (Beijing):
Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com
Regional Office China (Shenzhen):
Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com
Regional Office India:
Phone: +91 959 1302 450
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com
Regional Office Japan:
Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com
Regional Office Korea:
Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com
Regional Office Taiwan:
Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com