

Building EWL Compiler Libraries with a Custom UART Baud Rate

1. Introduction

This application note describes the steps required to change the universal asynchronous receiver/transmitter (UART) console baud rate for the Embedded Warrior Library (EWL) compiler libraries.

This document tells how to:

- Rebuild the EWL libraries with a new UART console baud rate
- Rebuild a UART project with the custom baud rate
- Customize a stationary project to work with the custom baud rate

Contents

1. Introduction	1
2. Preliminary background	2
3. Rebuilding EWL libraries	2
4. Rebuilding UART project	3
5. Using new UART library.....	5

2. Preliminary background

By default, Power Architecture projects contain UART library built for baud rate of 115200 bps. If you need to use a different baud rate, you need to rebuild the UART and EWL libraries for the new baud rate value.

3. Rebuilding EWL libraries

To define a new UART console baud rate and rebuild the EWL libraries, perform the following steps:

1. Open `uart_console_config.h` from `<CWInstallDir>\PA\PA_Support\ewl\EWL_C\include\pa`, and replace the default macro with the new baud rate value, as shown in the figure below.

```

uart_console_config.h
/* EWL
 * Copyright © 1995-2009 Freescale Corporation. All rights reserved.
 *
 * $Date: 2011/06/01 10:06:59 $
 * $Revision: 1.2 $
 */

/*
 * uart_console_config.h
 */

#ifndef __pa_uart_console_config__
#define __pa_uart_console_config__

#define UART_CONSOLE_BAUD_RATE kBaud768000

#endif
    
```

Figure 1. Defining new UART console baud rate

2. Rebuild the EWL libraries using the steps given in Section 22.3.3, “How to Rebuild the EWL Libraries,” of `<CWInstallDir>\PA\Help\PDF\Power Arch Build Tools Reference.pdf`.

4. Rebuilding UART project

Before building the UART project, perform these steps:

1. Open `UART.h` from `<CWInstallDir>\PA\PA_Support\Serial\Common`.
2. Add the new baud rate value to the `UARTBaudRate` enumeration.

```

/* EWL
 * Copyright © 1995-2009 Freescale Corporation. All rights reserved.
 *
 * $Date: 2009/05/14 16:55:58 $
 * $Revision: 1.1 $
 */

#ifndef UART_H
#define UART_H

#if !_EWL_CONSOLE_SUPPORT
#error _EWL_CONSOLE_SUPPORT must not be defined to 1.
#endif

typedef int UARTError;

enum {
    kUARTNoError = 0,
    kUARTUnknownBaudRate,
    kUARTConfigurationError,
    kUARTBufferOverflow,           /* specified buffer was too small */
    kUARTNoData                   /* no data available from polling */
};

typedef enum {
    kBaudHWSet = -1,               /* use HW settings such as DIP switches */
    kBaud300 = 300,               /* valid baud rates */
    kBaud600 = 600,
    kBaud1200 = 1200,
    kBaud1800 = 1800,
    kBaud2000 = 2000,
    kBaud2400 = 2400,
    kBaud3600 = 3600,
    kBaud4800 = 4800,
    kBaud7200 = 7200,
    kBaud9600 = 9600,
    kBaud19200 = 19200,
    kBaud38400 = 38400,
    kBaud57600 = 57600,
    kBaud115200 = 115200,
    kBaud230400 = 230400,
    kBaud76800 = 76800
} UARTBaudRate;

UARTError InitializeUART(UARTBaudRate baudRate);
UARTError TerminateUART(void);
    
```

Figure 2. Adding new baud rate value to `UARTBaudRate` enumeration

Rebuilding UART project

To rebuild the UART project, perform these steps:

1. Start CodeWarrior for Power Architecture.
2. Choose **File > Import** from the menu bar. The **Import Projects** page of the **Import** wizard appears.
3. Browse and select a UART project, specific to the board being used, from `<CWInstallDir>\PA\PA_Support\Serial`, as shown in the figure below.

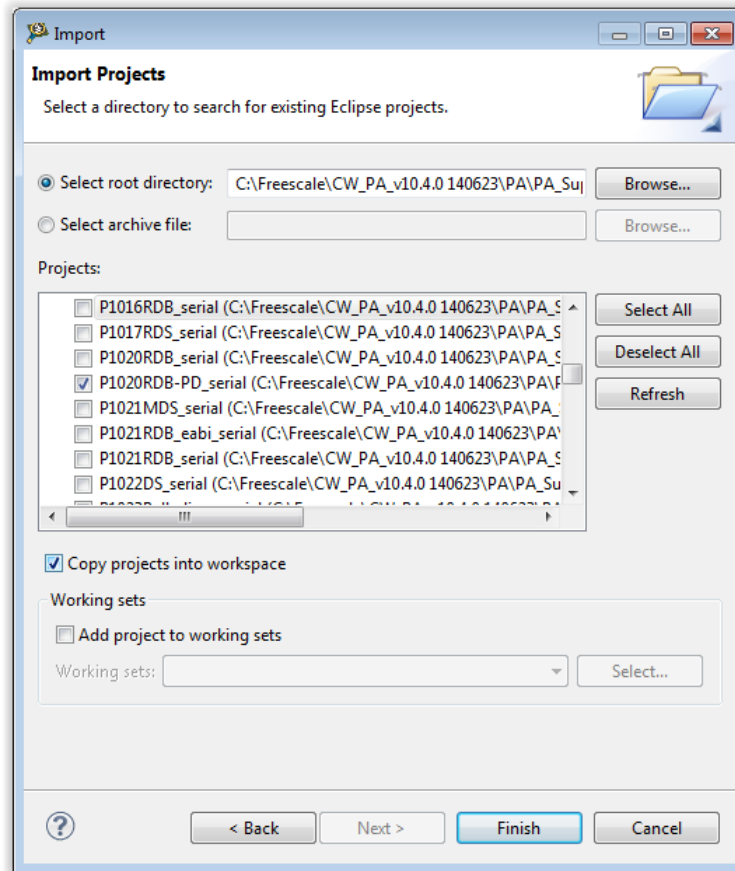


Figure 3. Importing UART project

4. Select the **Copy projects into workspace** checkbox.
5. Click **Finish** to end the **Import** wizard.
6. Build the project using the **Project > Build Project** option.

5. Using new UART library

To use the new UART console baud rate in a project, perform these steps:

1. Start CodeWarrior for Power Architecture.
2. Create a new project using the **File > New > CodeWarrior Bareboard Project Wizard** option.
3. Choose **Project > Properties** from the menu bar. The **Properties for <project>** window appears.
4. Expand the **C/C++ Build** property and select **Settings > PowerPC Linker > Input**.
5. Replace the library files in the **Library Files** pane with the ones built in sections 3 and 4, as shown in the figure below.

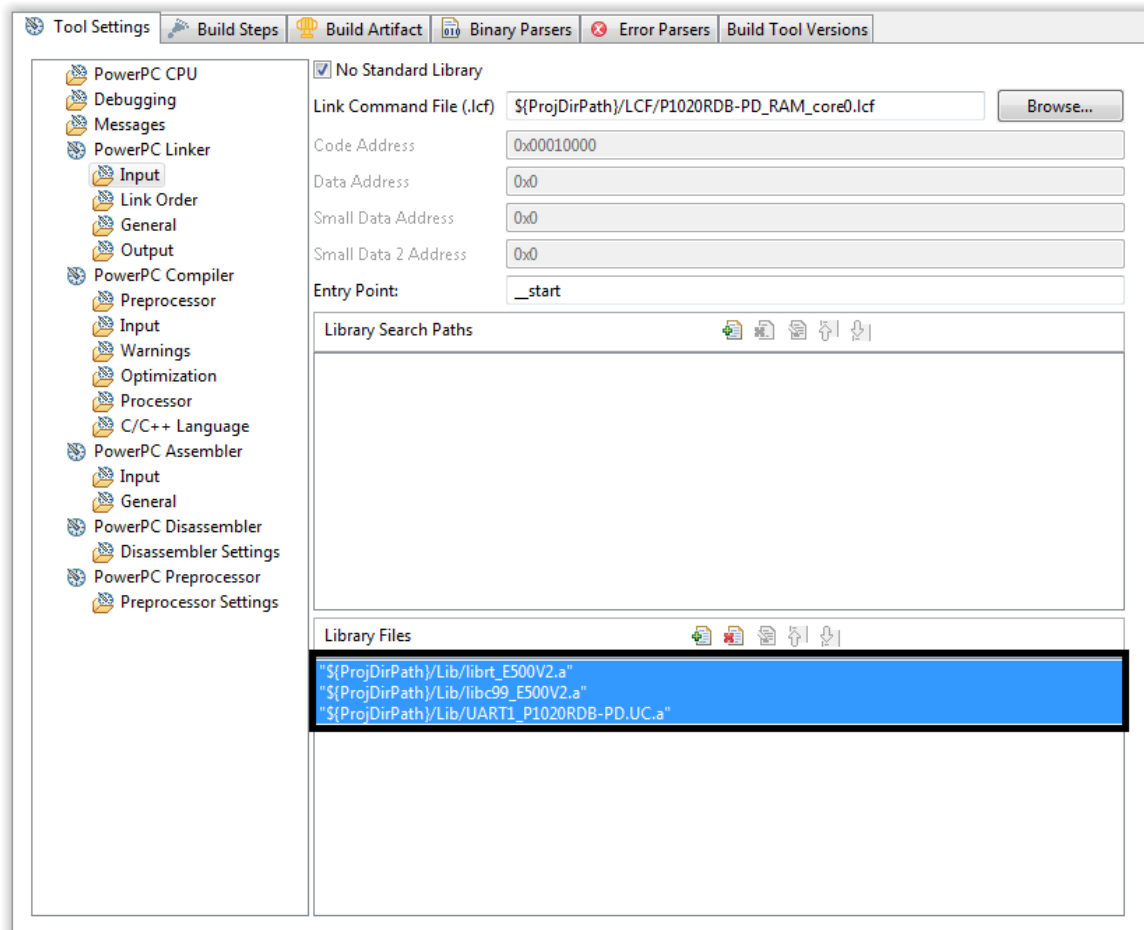


Figure 4. Modifying project settings

6. Click **Apply** and then **OK** in the **Properties for <project>** window.
7. Build the project with the new libraries using the **Project > Build Project** option.

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.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 reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 that may be provided in Freescale 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, CodeWarrior, and QorIQ are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Layerscape is trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, Cortex and TrustZone are trademarks or registered trademarks of ARM Ltd or its subsidiaries in the EU and/or elsewhere. All rights reserved.

© 2014 Freescale Semiconductor, Inc.