



Coinel Technology Solutions LLP<sup>®</sup>

---

# **USB BOOT LOADER Tutorial on LPC2148 BASED BOARD**

# CoiNel Technology Solutions LLP

---

Designed by

CoiNel Technology Solutions LLP

No-816, 2<sup>nd</sup> Floor, 4<sup>th</sup> B Cross, 9<sup>th</sup> A Main,

RPC Layout, Vijaynagar,

Bangalore-560040

State: Karnataka

Country: India

[www.coineltech.com](http://www.coineltech.com)

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where CoiNel is aware of trademark claim, the product name appears in initial capital letters, in all capital or in accordance with the vendor's capitalization preference. Users should contact appropriate companies for more complete information on trademark and trademark registrations. All trademarks and registered trademarks in this manual are the property of their respective holders.

No part of this document may be reproduced or distributed in any form or by any means, or stored in the database or retrieval system, without the prior written permission from CoiNel Technology Solutions LLP; with the exception that the listings may be entered, stored and executed in a computer system, but they may not be reproduced.

The content in this document are presented for instruction value. The details have been carefully tested, but are not guaranteed for any particular purpose. CoiNel Technology Solutions does not offer any warranties and does not guarantee the accuracy, adequacy, or completeness of any information herein and is not responsible for any errors or omissions. CoiNel Technology Solutions LLP assumes no liability for damages resulting from use of such information in this document or for any infringement of intellectual property rights of third parties that would result from use of this information.

CoiNel Technology Solutions LLP assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

For any enquires, kindly contact [info@coineltech.com](mailto:info@coineltech.com)

## CONTENTS:

I. What is a Boot loader?

II. Why would you use the LPC2148 USB Boot loader?

III. How do you use the Boot loader?

IV. Loading the Bin file onto the Bootloader Enabled Board

The USB Bootloader for the LPC2148 is a cool trick that CoiNel created for some of our ARM7 based boards. Since the ARM architecture allows for large flash space, loading code onto the LPC2148 ARM7 over the serial port was very slow. The bootloader brings various techniques together to create an easy to use and very fast development system to load the program without any need for expensive hardware or software.

In this tutorial we will cover what a Bootloader is, why should we use it, and how to use it.

## I. What is a Boot loader?

A Boot loader is a small piece of code that runs before the operating system starts running. In our case the bootloader is the code that runs before the device firmware starts up. Typically a bootloader is used because the system memory is too small to contain the entire program, and so the bootloader uses a set of routines to call the program from a different part of memory.

The CoiNel LPC2148 ARM7 USB Boot loader performs three steps:

1. First, the Boot loader checks to see if a USB cable has been plugged in. If the LPC2148 is connected as a USB device then it initiates a USB Mass Storage system. This will cause the target board to appear on any computer platform as a removable flash drive. The user can then seamlessly transfer files to the flash drive.
2. The next thing the Boot loader does is look for a firmware file. This file contains the desired operating firmware (in a binary file format) for the LPC2148. If the Boot loader finds this file system then it programs the contents of this file to the flash memory of the LPC2148. In this way, the Boot loader acts as a “programmer” and we can upgrade the firmware on the LPC2148 simply by loading a new file.
3. After performing for first two checks, the boot loader calls the main firmware.

## II. Why would you use the LPC2148 USB Bootloader?

The bootloader provides several distinct advantages. First, if you have a LPC2148 Board that has been preloaded with the bootloader then you don't even need a programmer to load code onto the board! Another reason that LPC2148 USB Boot loader is great, because it is very fast at loading code. When using a serial programmer to load code onto an ARM, it can take several minutes to get your code up. This can be a major setback when you're writing very large programs, and you start debugging. Using the LPC2148 USB bootloader allows you to load code in seconds.

## III. How do you use the Boot loader?

First you have to program the Boot loader code onto the LPC2148 Board. The Boot loader code can be downloaded at the download section of LPC2148 Based product page. You can check the product page at [www.coineltech.com/shop](http://www.coineltech.com/shop). You may either use FLASH MAGIC software for programming (i.e., serial programming), or H-JTAG (parallel port programming) to load this code. Once the Boot loader program is on the LPC2148, using it is very simple!

**Note: Make sure you erase the old program before loading the bootloader code.**

Let us assume that you have written some neat code to use on LPC2148 Board. Plug in a USB cable, and when the USB Mass Storage Device opens on your desktop, just drag and drop the binary file onto the device then press reset; the bootloader will look for the binary file and if the file is found the bootloader will program the new code onto the controller.

That seems easy enough, but how do I get this binary file? In order to get the binary file, first we have to compile the code. In order to compile the code you need have Kiel uVision4! Below steps would show you, how to create a binary file.

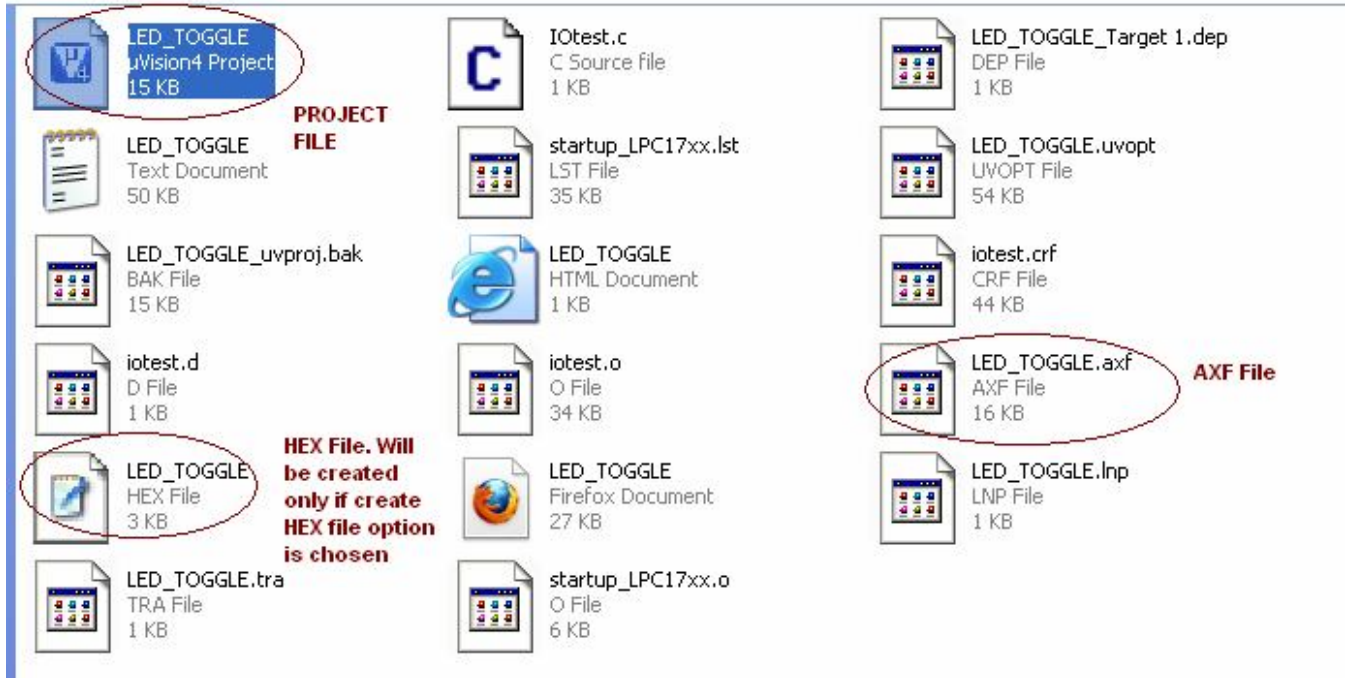
**Note: You should be familiar with Keil environment & programming. For more details on Keil programming, check tutorials on [how to use Keil](#).**

# CoiNel Technology Solutions LLP

## Compiling and Loading Code onto a USB Boot loader Enabled Board:

The axf file would be created once you compile the code. The axf file will be created in the folder where you have saved the project.

Note: You can change the location of .axf file storage if required whose details are not explained in this document.

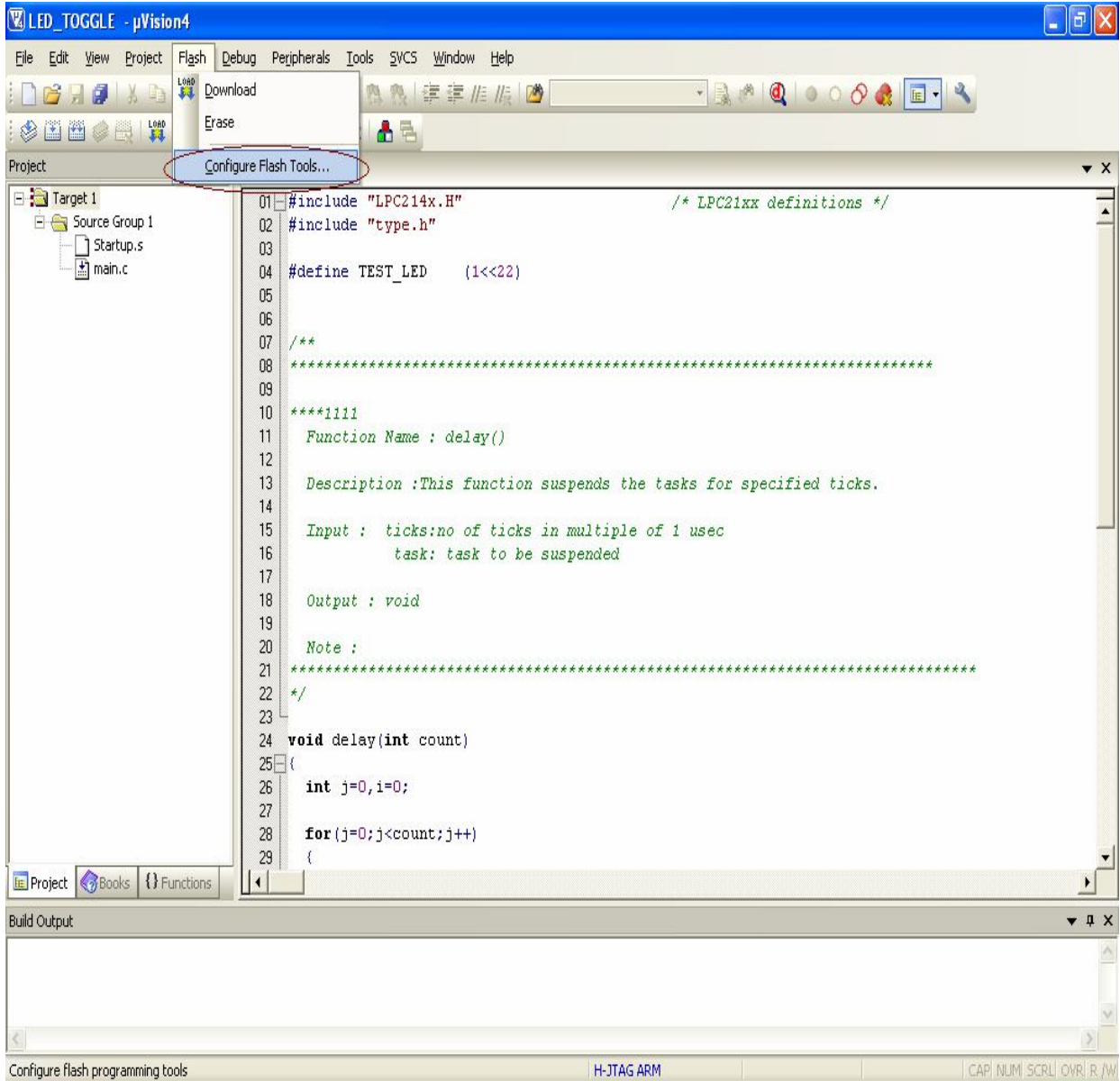


Now that you have created .axf file, how can it be converted into a binary file so that it can be loaded onto USB bootloader enabled LPC2148 board. The following documentation gives you the detail.

## ➤ How to create binary file using Keil uVision4:

To create binary file follow the settings as shown in the images below.

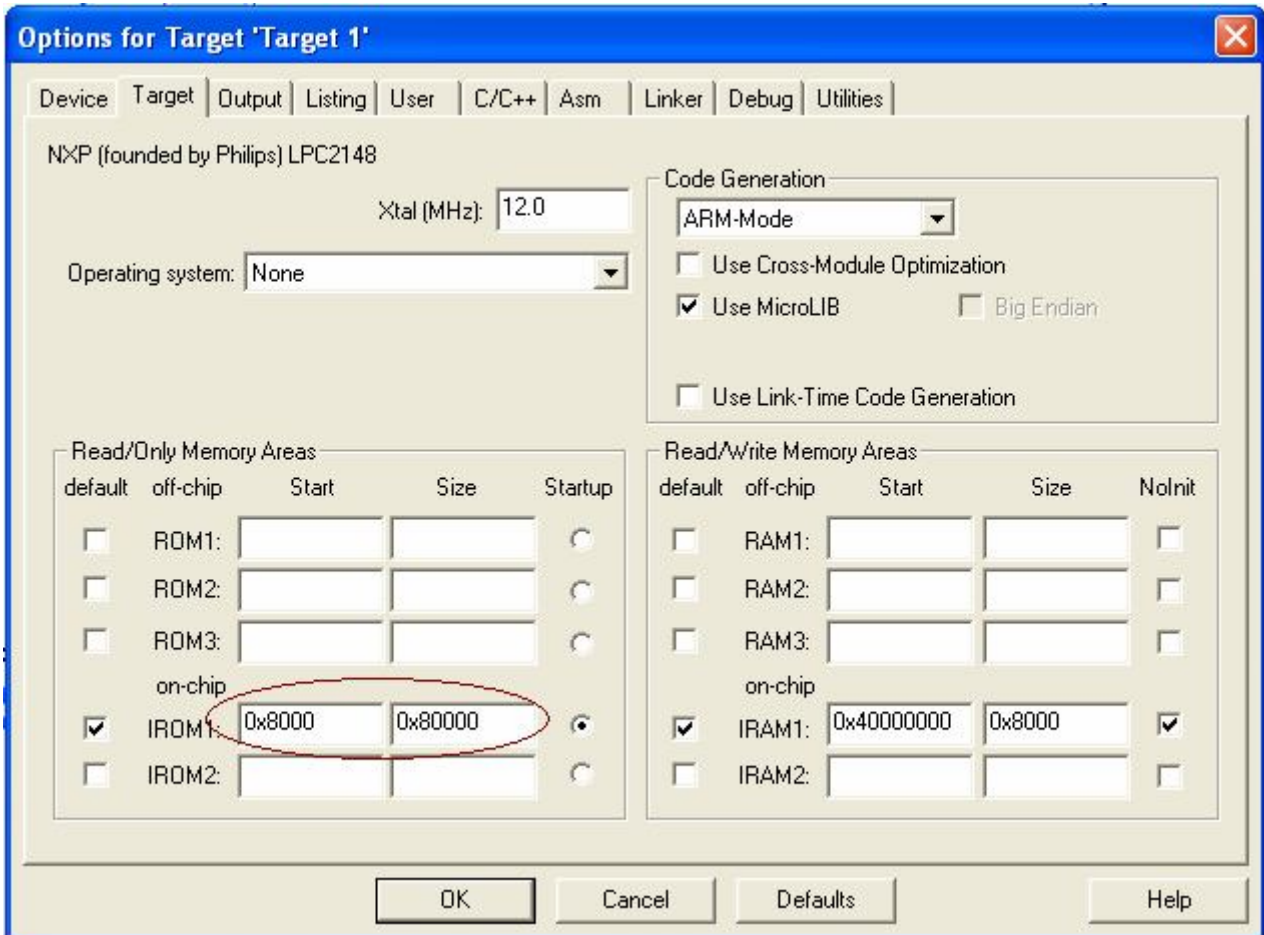
### 1. Click on Flash Menu → Go to Configure Flash Tools



# CoiNel Technology Solutions LLP

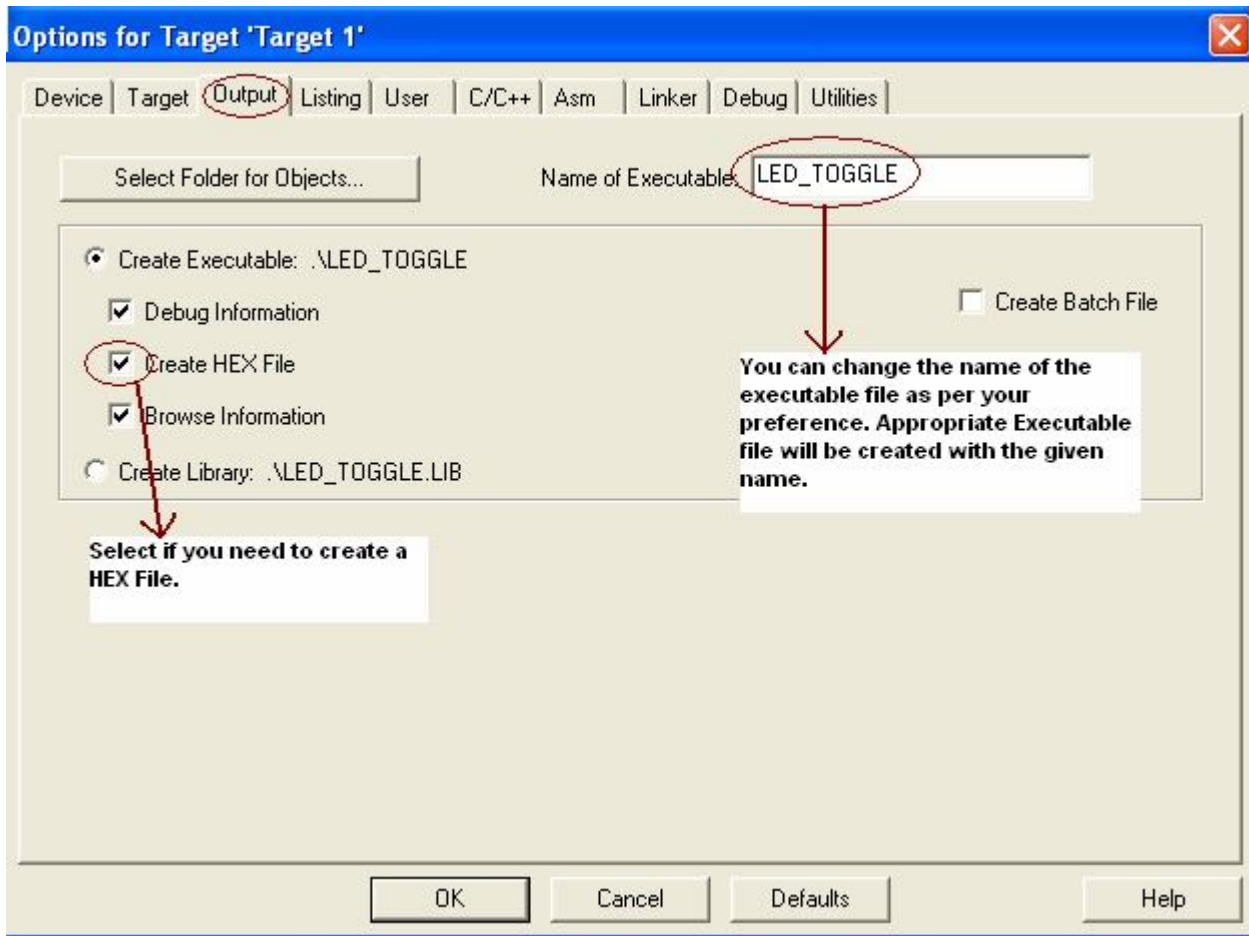
2. Click on **Target Tab** and type in **IROM1** as shown in below image.

Change the start address of on-chip IROM1 memory to 0x8000 to 0x80000 as shown in fig. Below .This is the area where user program is placed and has to be run when the controller is reset.

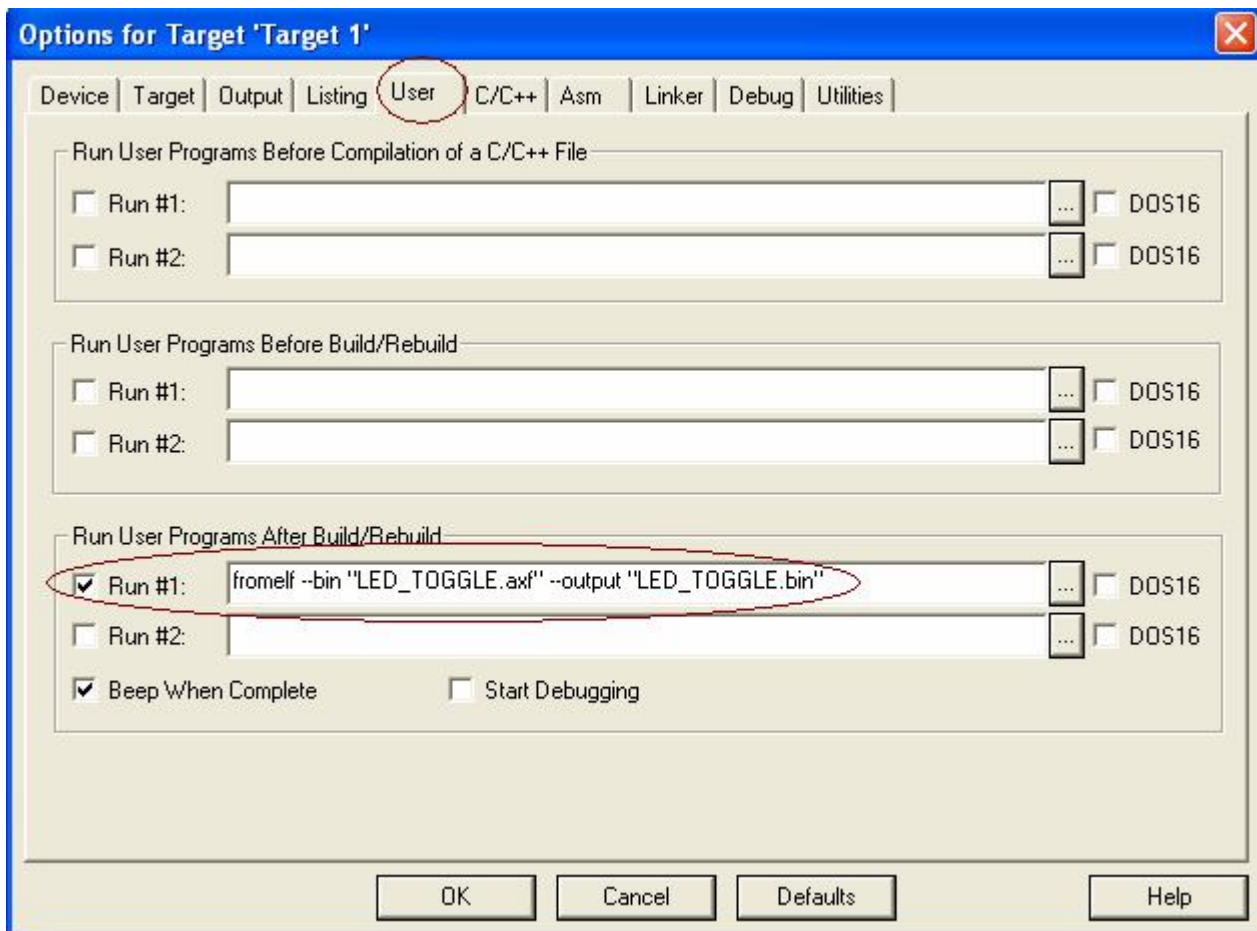




3. Click on **Output Tab** and make settings as required.



4. Click on **User Tab** and type in box next to **RUN #1** as shown in below image.

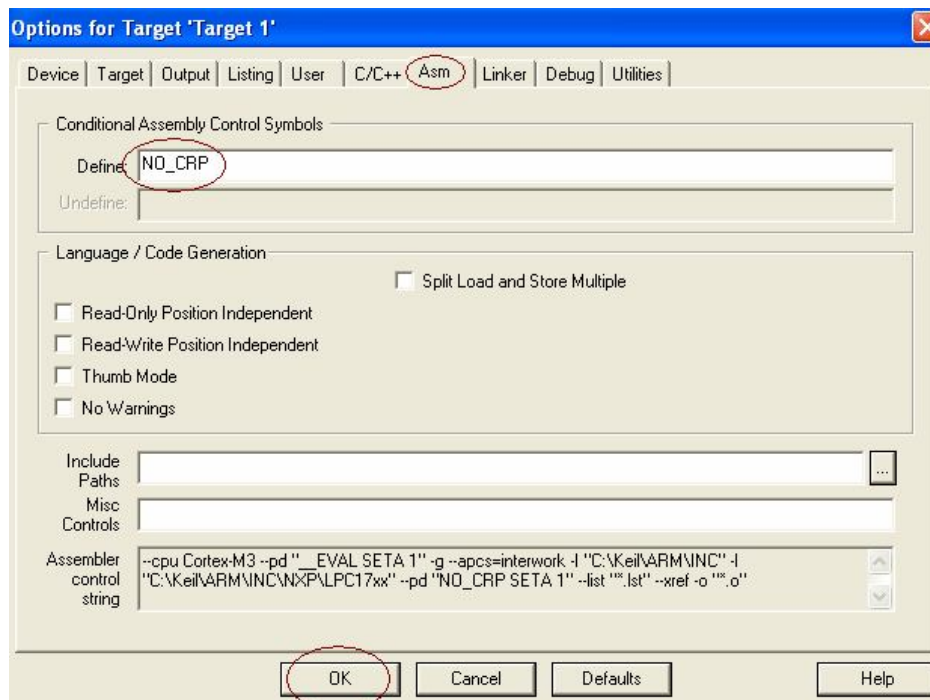


You can also copy and paste the following

```
fromelf --bin "LED_TOGGLE.axf" --output "LED_TOGGLE.bin"
```

Above image shows an example for LED\_TOGGLE. If you are doing this for other code, replace LED\_TOGGLE with name that you have given to your code/project.

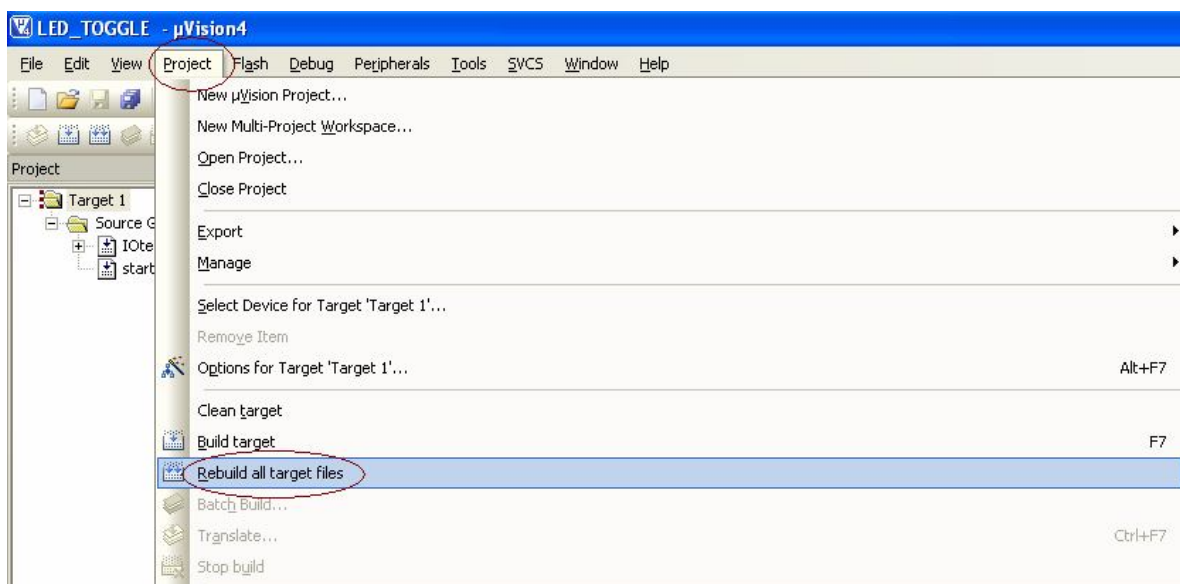
4. Click on **Asm Tab** and make settings as below and then click **OK** button.



After all the settings made, press ok.

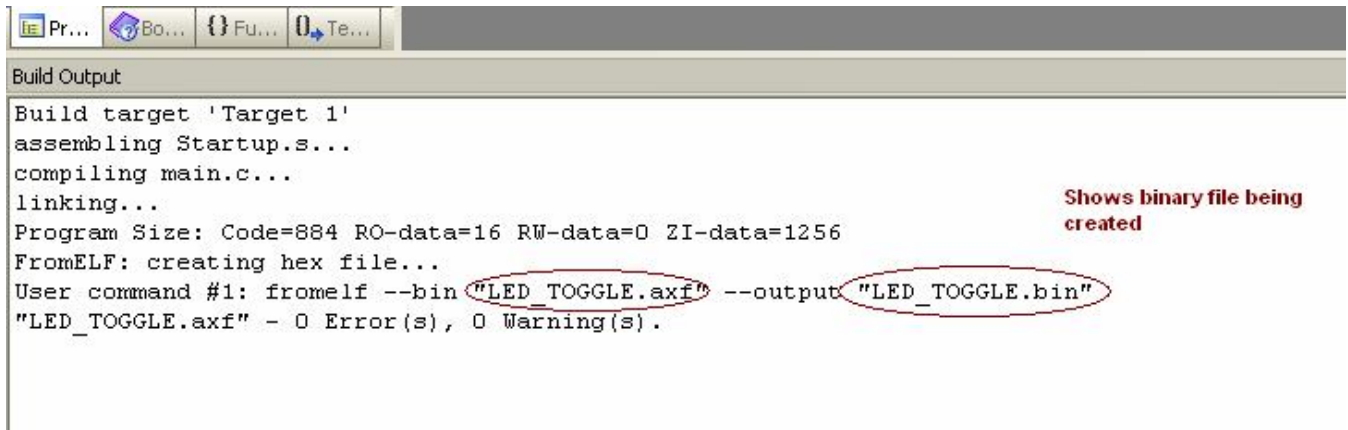
You need to rebuild the code for compilation. When the code is compiled, binary files are created for your program that can be loaded on your LPC 2148 Board.

Rebuilding the code



# CoiNel Technology Solutions LLP

After compilation, you will see the following in build output window.

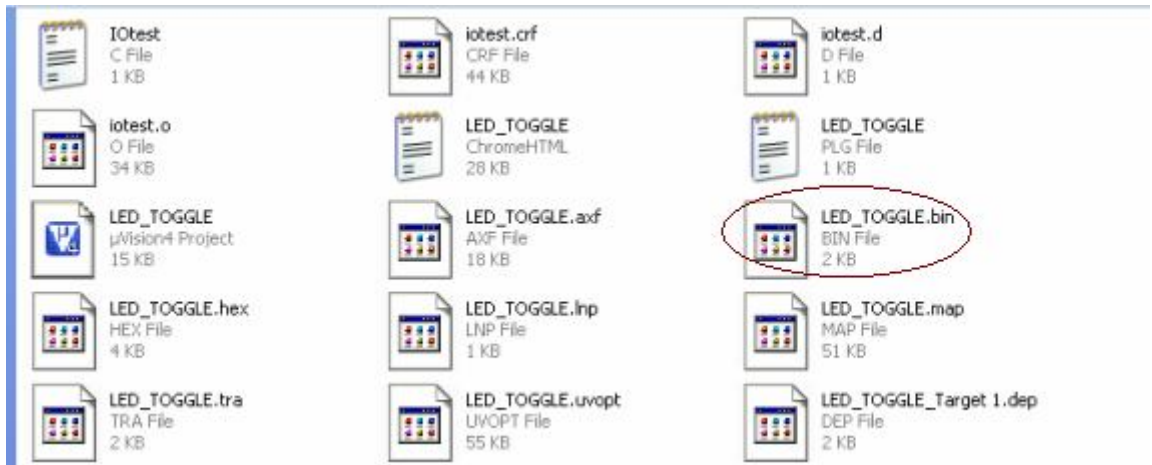


```
Build target 'Target 1'
assembling Startup.s...
compiling main.c...
linking...
Program Size: Code=884 RO-data=16 RW-data=0 ZI-data=1256
FromELF: creating hex file...
User command #1: fromelf --bin "LED_TOGGLE.axf" --output "LED_TOGGLE.bin"
"LED_TOGGLE.axf" - 0 Error(s), 0 Warning(s).
```

The bin File will be created in the same folder where project file is created.

Note: You can change the location of the bin file by changing the output location in Flash → Configure Flash Tools → Output → Select Folder for objects... The details of which are not explained in this document.

BIN File is created as shown



This bin file need to be loaded on the bootloader enabled board.

## IV. Loading the Bin file onto the Bootloader Enabled Board:

After following the steps above, we would be ready with the following

**LPC2148 Board** (with the USB Boot loader firmware).

**USB cable.**

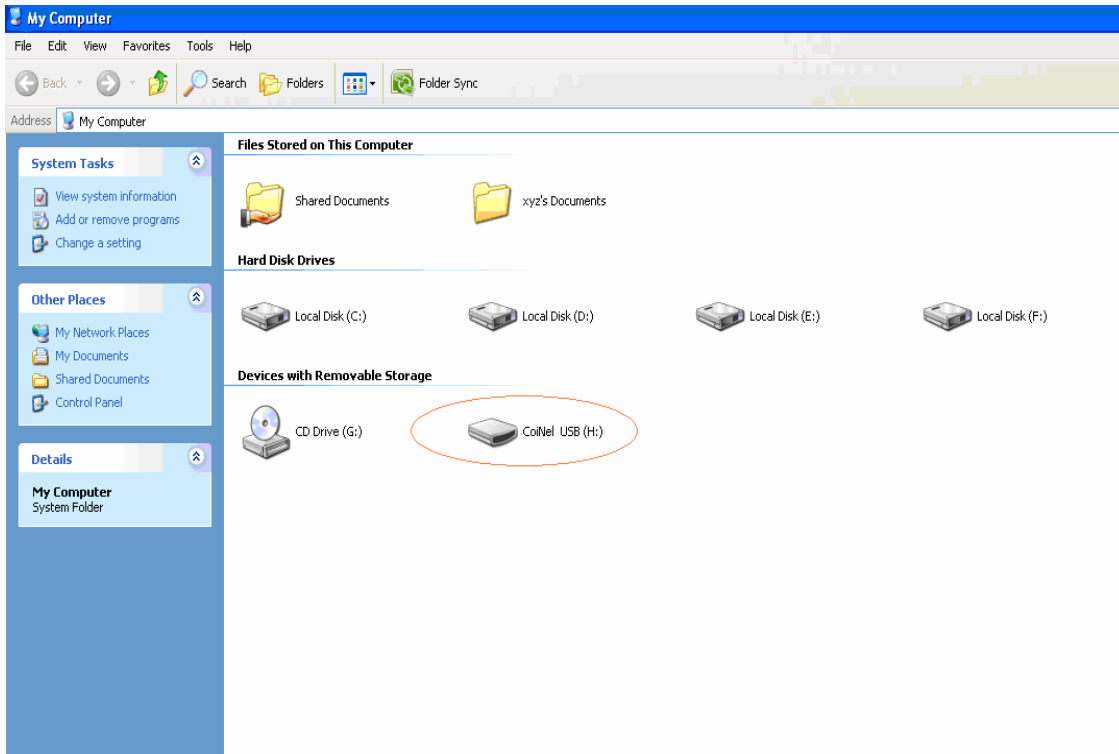
**Required .bin file for the project.**

Now we will see how we can load this bin file on the controller.

### Procedure:

- Plug in the USB cable to your computer and to the Board. Once the USB cable is plugged in, Board is powered up; a Removable Disk Drive will be available. This drive represents the memory space on the card.

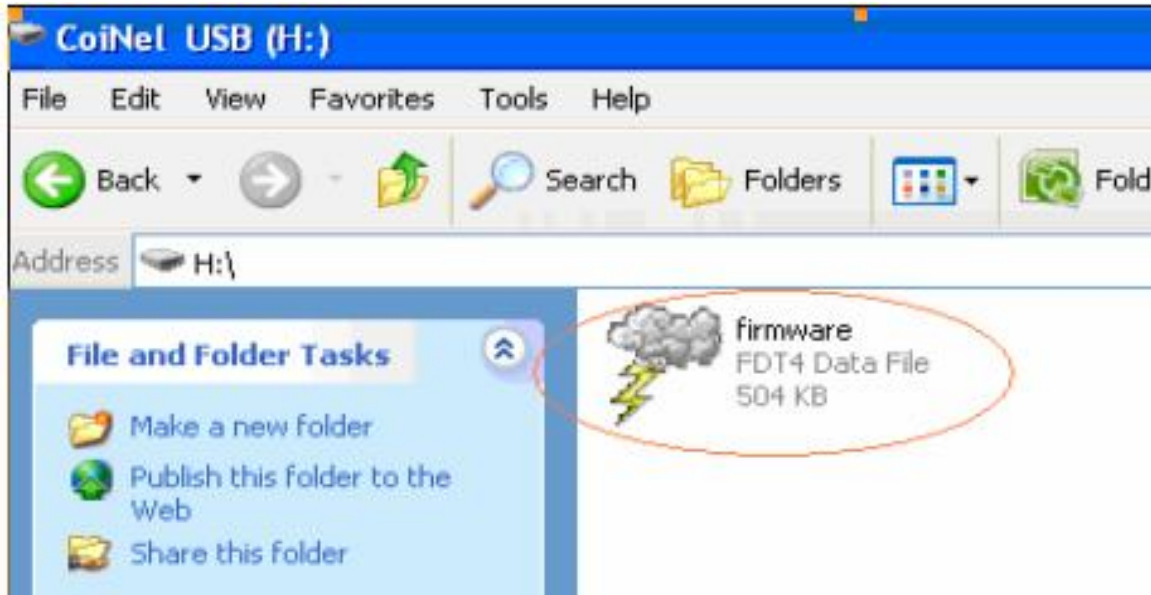
In the below image, you can notice, a drive named CoiNel USB.



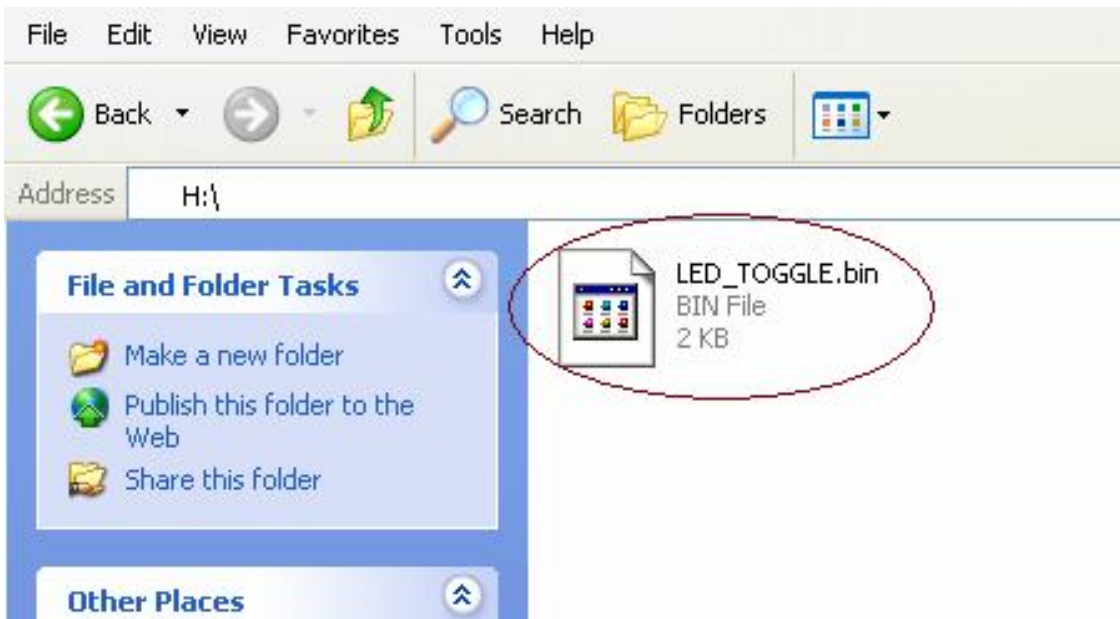
# CoiNel Technology Solutions LLP

---

- When you enter into this CoiNel USB drive, you could notice a default firmware file already present.



You need to delete this firmware file. *Copy and paste* the newly created project binary file into this drive. Below figure shows a newly created project binary file is pasted.



Once the file has been copied to this new location, **Press reset on your board, and your new code will be running.**

**Note:**

In case, you want to dump a new program into LPC2148 Board, you need to connect port P1.20 to ground pin of the board and then press reset switch.

Doing this will again have the board detected as a removable disc (ie the board will exit from application mode and will switch back to bootloader mode). Follow the above procedure and copy/load the new bin file.

**Note:** Once programmed, make sure that the P1.20 pin is removed / not grounded.

## **CLARIFICATIONS AND SUPPORT**

CoiNel is at your service. If you have any clarifications or suggestions on this document or need any technical support and consultation, we are available in forms of telephone and E-mail.

TEL: +91-80-23154423

Technical Support E-mail: [support@coineltech.com](mailto:support@coineltech.com)

For any questions or concerns submit them to [info@coineltech.com](mailto:info@coineltech.com)