

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Programing of embedded systems

1. Introduction

Lead Partner: Warsaw University of Technology

Authors: Daniel Krol

University of Applied Sciences in Tarnow

Programing of embedded systems

1. Introduction

Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

© Copyright 2021 - 2023 the ENGINE Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Funding Disclaimer

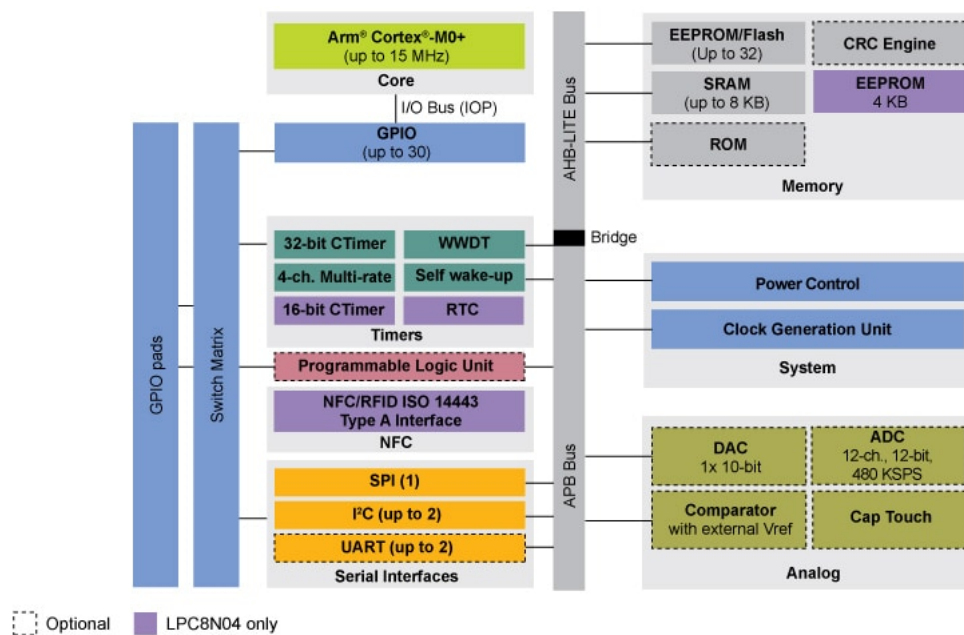
This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Programming of embedded systems

1. Introduction

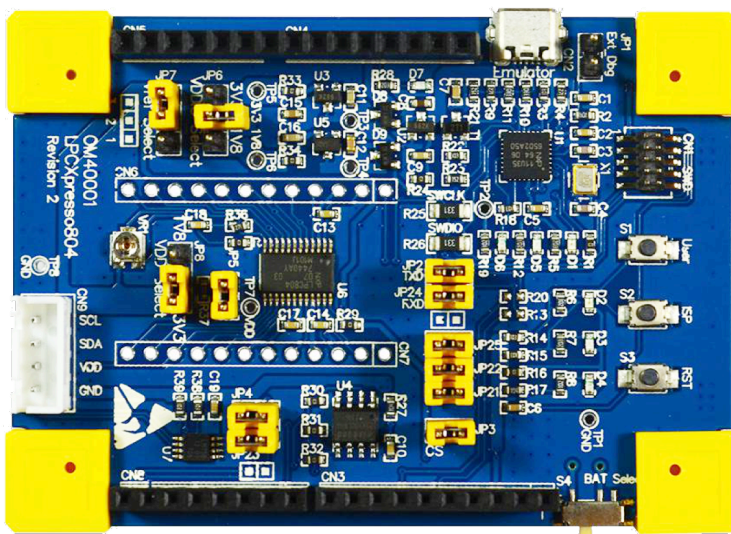
I. LPC804 features

1. The LPC804 are an Arm Cortex-M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 15 MHz. The LPC804 supports 32 KB of flash memory and 4 KB of SRAM. The peripheral complement of the LPC804 includes a CRC engine, two I2C-bus interfaces, up to two USARTs, one SPI interface, Capacitive Touch Interface (Cap Touch), one multi-rate timer, self-wake-up timer, one general purpose 32-bit counter/timer, one 12-bit ADC, one 10-bit DAC, one analog comparator, function-configurable I/O ports through a switch matrix, an input pattern match engine, Programmable Logic Unit (PLU), and up to 30 general-purpose I/O pins [1].

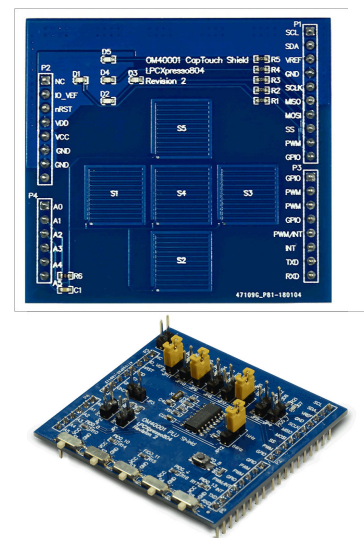


Source: NXP

2. Development board and Shields:



Source: NXP



Programming of embedded systems

1. Introduction

II. MCUXpresso Integrated Development Environment (IDE)

1. Go to the website: <https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE>, register and download the MCUXpresso IDE installer. The installer is available for Windows, MacOS and Linux:

Product Download

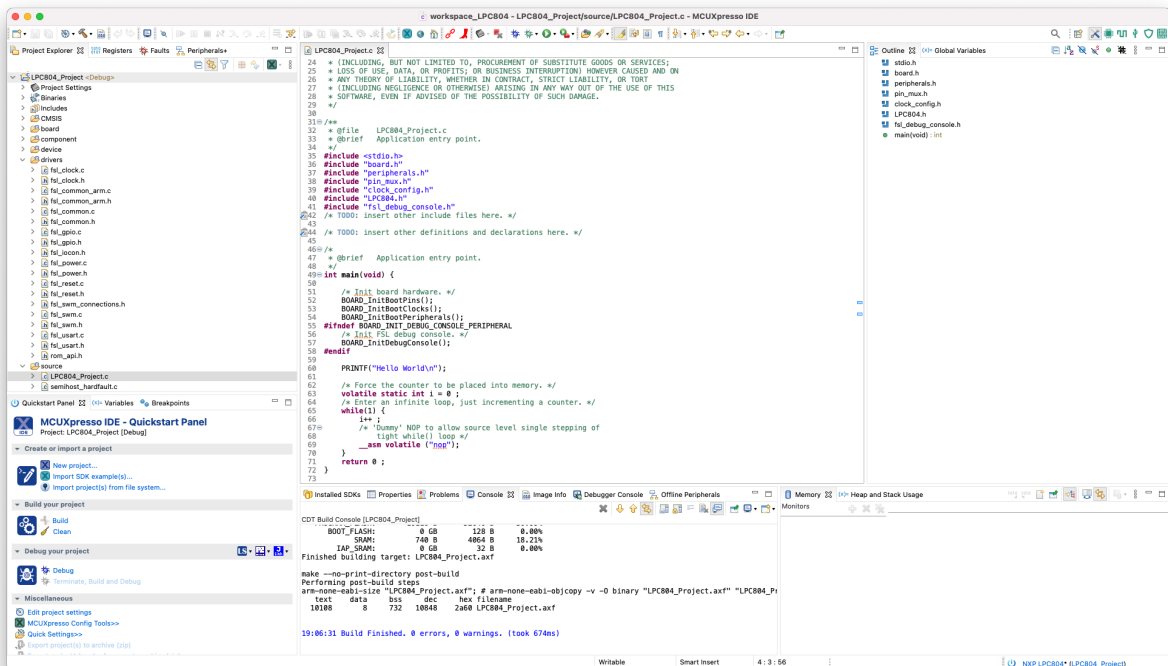
MCUXpresso IDE

Files License Keys Notes Download Help

Show All Files 3 Files

+	File Description	File Size	File Name
+	MCUXpressoIDE 11.4.1 - Linux	928.6 MB	mcuxpressoide-11.4.1_6260.x86_64.deb.bin
+	MCUXpressoIDE 11.4.1 - Mac	885.1 MB	MCUXpressoIDE_11.4.1_6260.pkg
+	MCUXpressoIDE 11.4.1 - Windows	833.3 MB	MCUXpressoIDE_11.4.1_6260.exe

2. Install MCUXpresso environment on your computer:



Programing of embedded systems

1. Introduction

3. Then go to the website: <https://mcuxpresso.nxp.com/en/welcome>

The screenshot shows the MCUXpresso SDK Builder homepage. At the top, there are navigation icons for IDE, CFG, SEC, and SDK, along with a SUPPORT link. The main heading is "MCUXpresso SDK Builder". Below this, a paragraph states: "The MCUXpresso SDK brings open source drivers, middleware, and reference example applications to speed your software development. Customize and download an SDK specific to your processor or evaluation board selections." There are two buttons: "Select Development Board" and "Access My SDK Dashboard". A light blue box contains a message: "NXP strongly recommends you update to the latest version of MCUXpresso SDK that contains essential security updates for Bluetooth Low Energy software stack implementations." Below this is a navigation bar with "OVERVIEW", "SOFTWARE AND TOOLS", and "DEVELOPER RESOURCES". The "SOFTWARE AND TOOLS" section is active, showing four tool categories: MCUXpresso SDK, MCUXpresso IDE, MCUXpresso Config Tools, and MCUXpresso Secure Provisioning Tool. Each category has a brief description and a "Learn More" link.

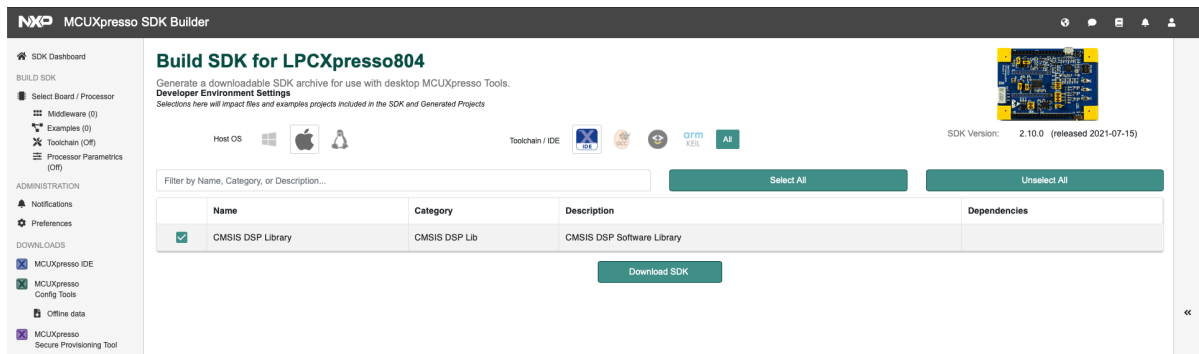
4. Click *Select Development Board* and then select the prototype board with the LPC804 microcontroller, and then click *Build MCUXpresso SDK* (actual version):

The screenshot shows the "Select Development Board" interface. On the left is a sidebar with navigation options like "SDK Dashboard", "BUILD SDK", "Select Board / Processor", "Administration", "Downloads", and "MCUXpresso IDE". The main area has a search bar for hardware and a list of boards. The "LPCXpresso804 (LPC804)" board is selected and highlighted in blue. To the right, the "Selection Details" for the LPCXpresso804 are shown, including a product image, a "Build MCUXpresso SDK v2.10.0" button, and a list of "Matched Hardware Platforms". The platform list shows "Found 633 HW solutions that match your criteria." with filters for Boards (131), Kits (77), and Processors (399). Below this, filtering criteria are listed, such as "Required Middleware" and "Required Example Projects", with a note that "Middleware filtering not applied" and "Example Project filtering not applied".

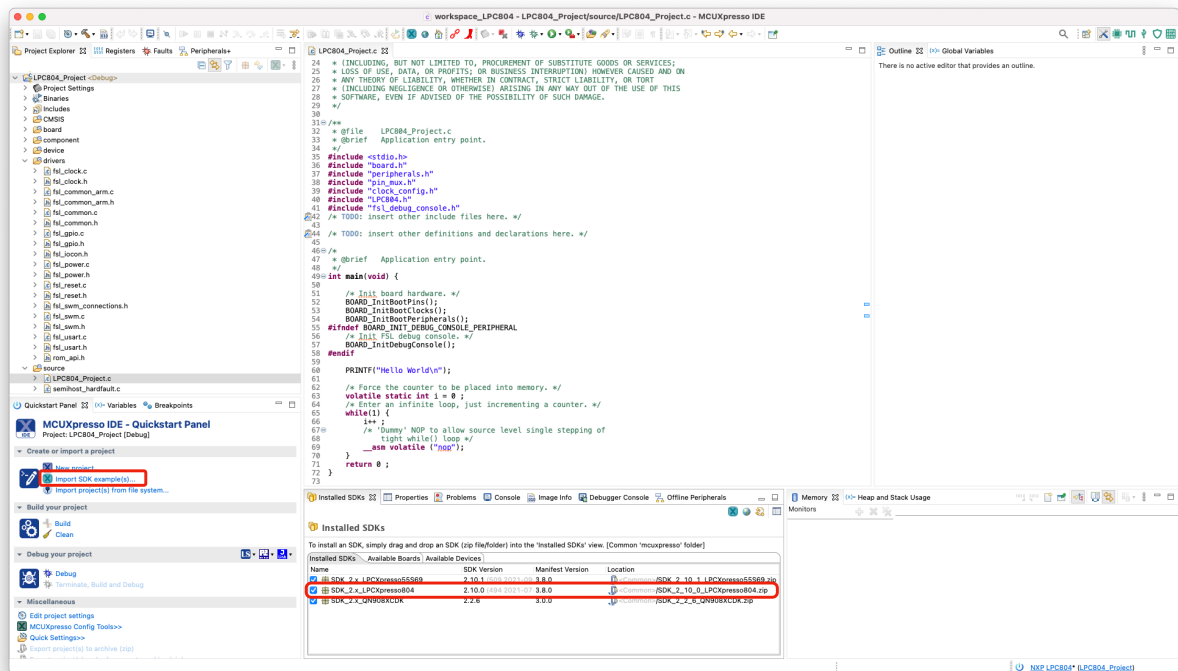
Programming of embedded systems

1. Introduction

5. Select the operating system you are using and select *Toolchain / IDE: MCUXpresso*. Select *All* available components:



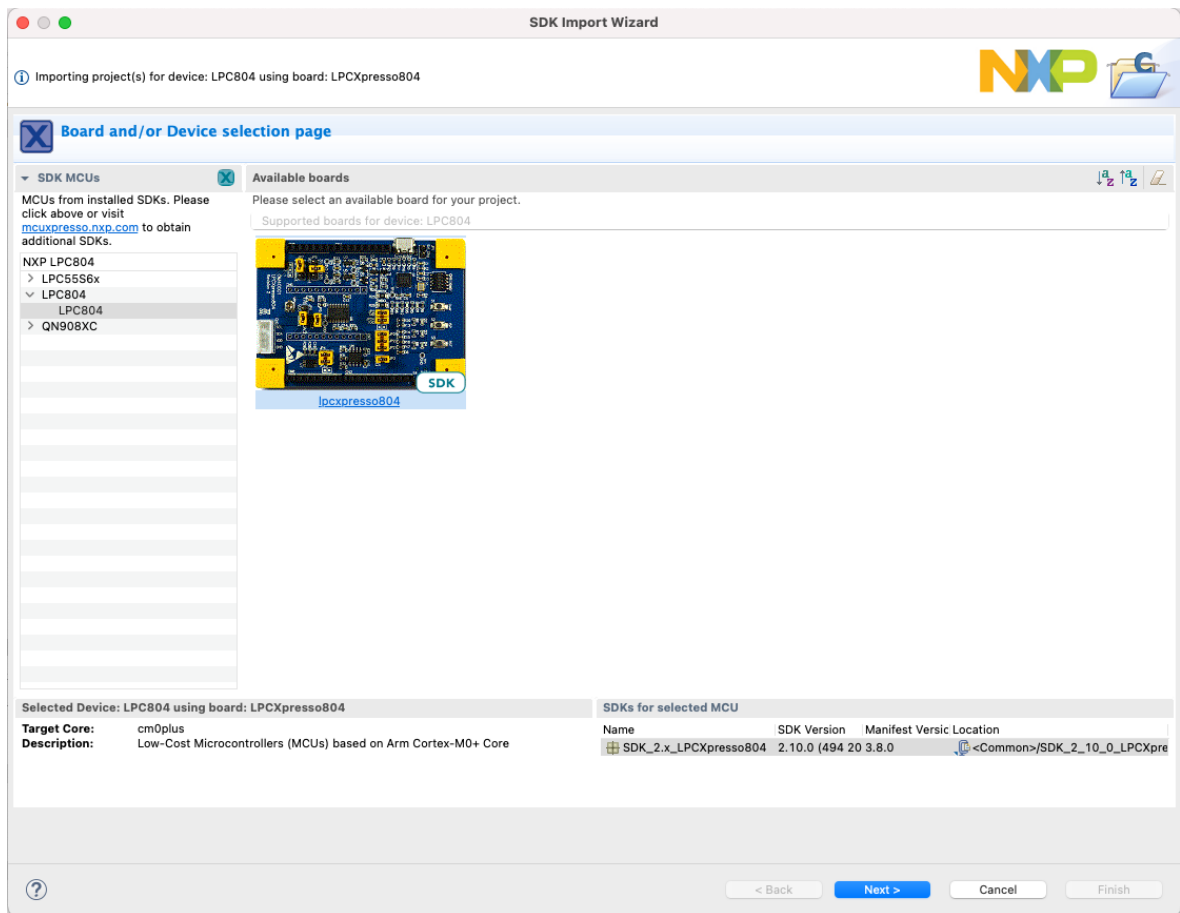
6. After a while, the ready SDK will appear in your Dashboard. Click *Download SDK* and in the next window select *Download SDK Archive*:
7. Open *MCUXpresso IDE* and move (Drag and Drop) Archive SDK to *Installed SDKs* tab:



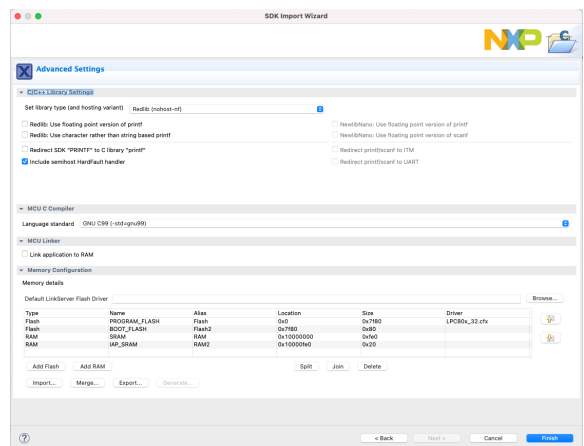
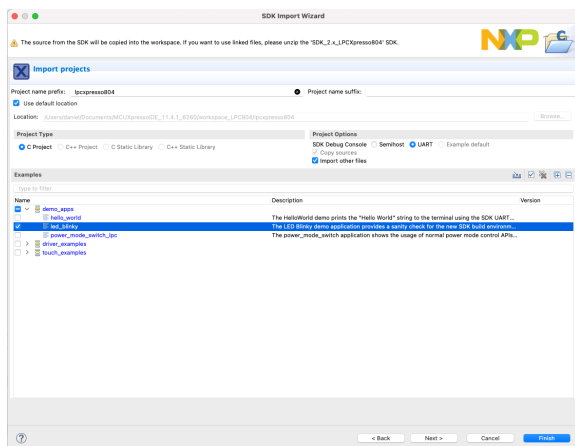
Programing of embedded systems

1. Introduction

8. Go to *Import SDK examples* (picture above) and select *lpcpresso804*:



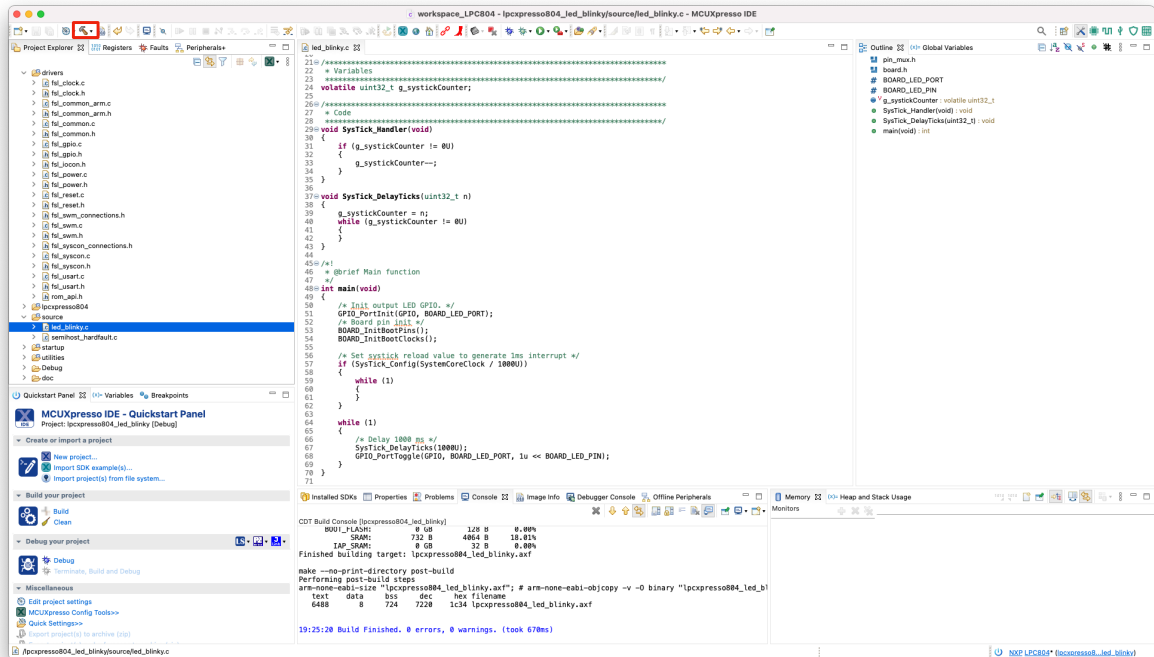
9. Select *demo_apps* and then *led_blinky*. Click Finish:



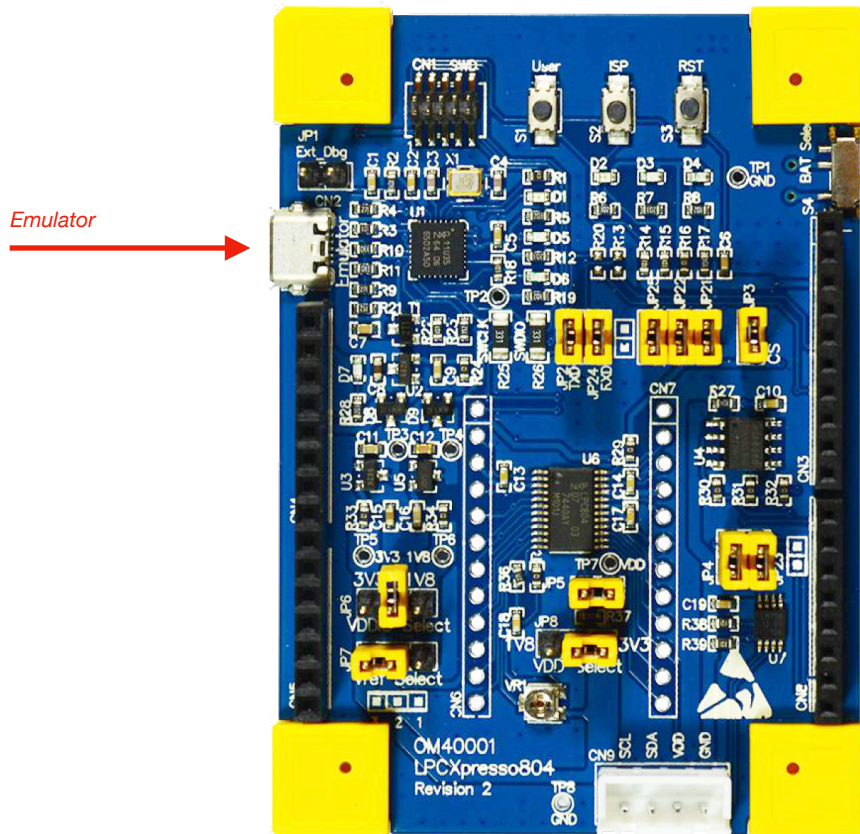
Programming of embedded systems

1. Introduction

10. Build the project by clicking *Build*:



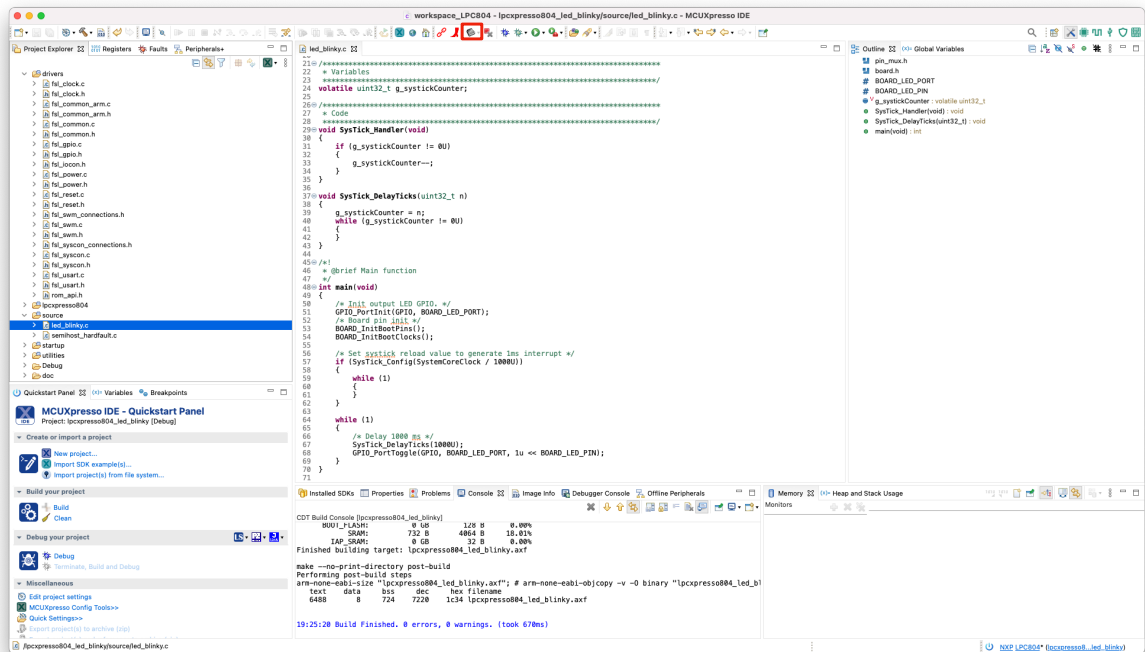
11. Connect the *LPCXpresso804* board with the USB interface labeled *Emulator* to the computer:



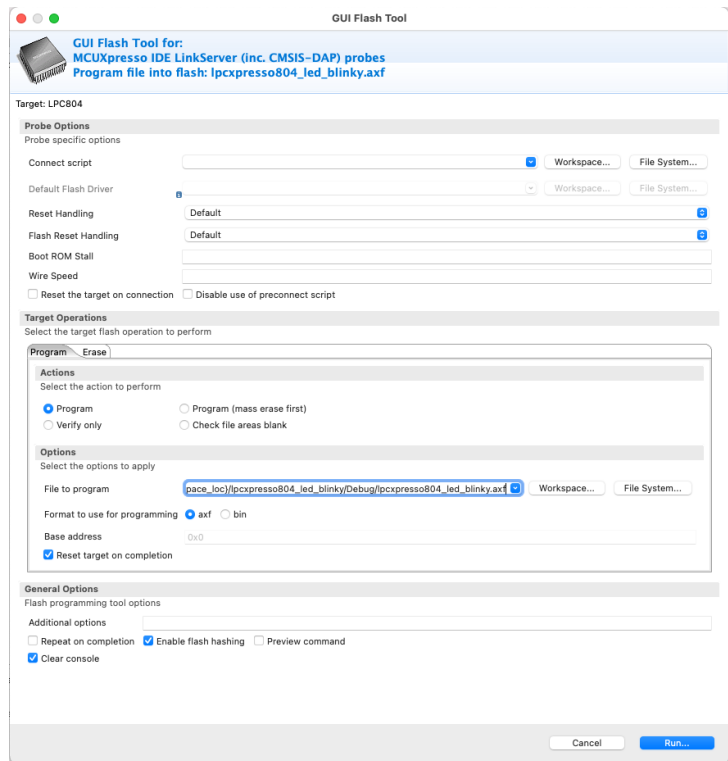
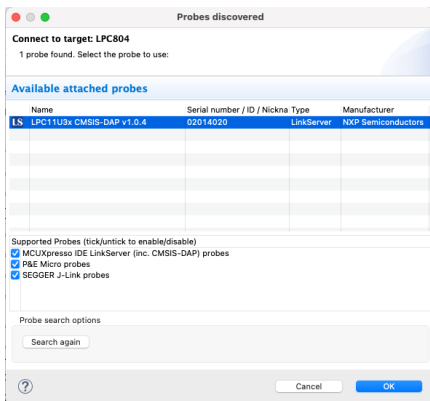
Programming of embedded systems

1. Introduction

12. Program the microcontroller by pressing *GUI Flash Tool*:



13. Leaving the default settings in the following programmer windows and click Run:

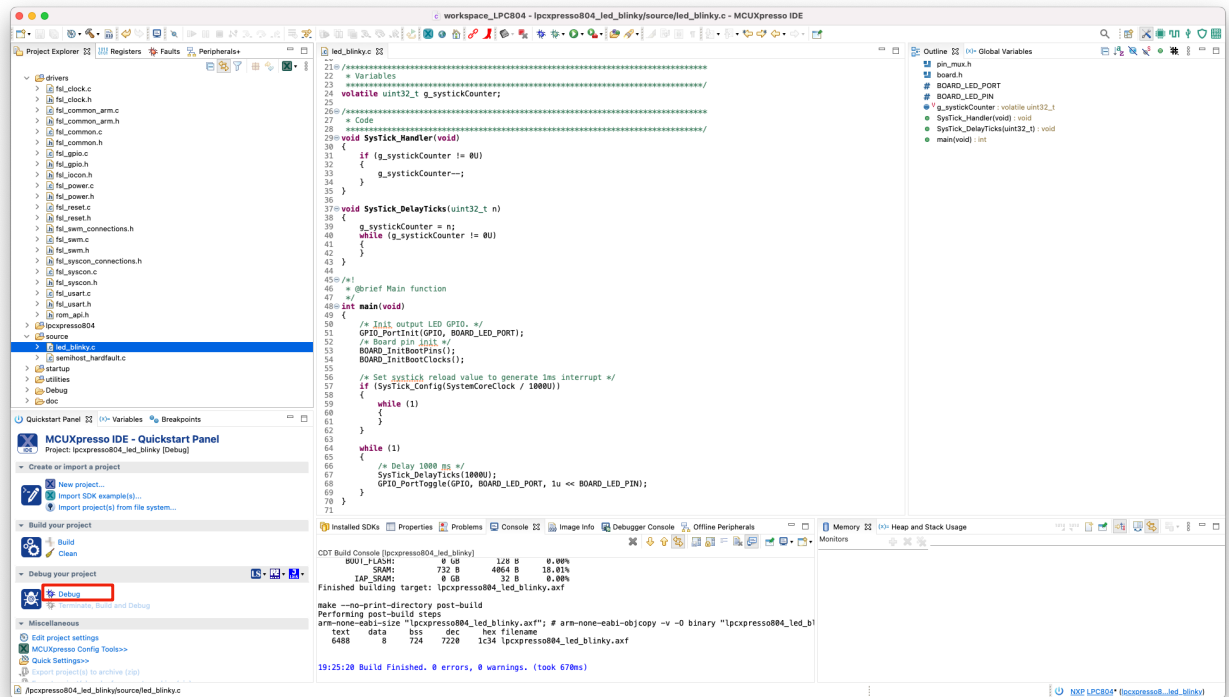


14. The LED on the evaluation board should flash.

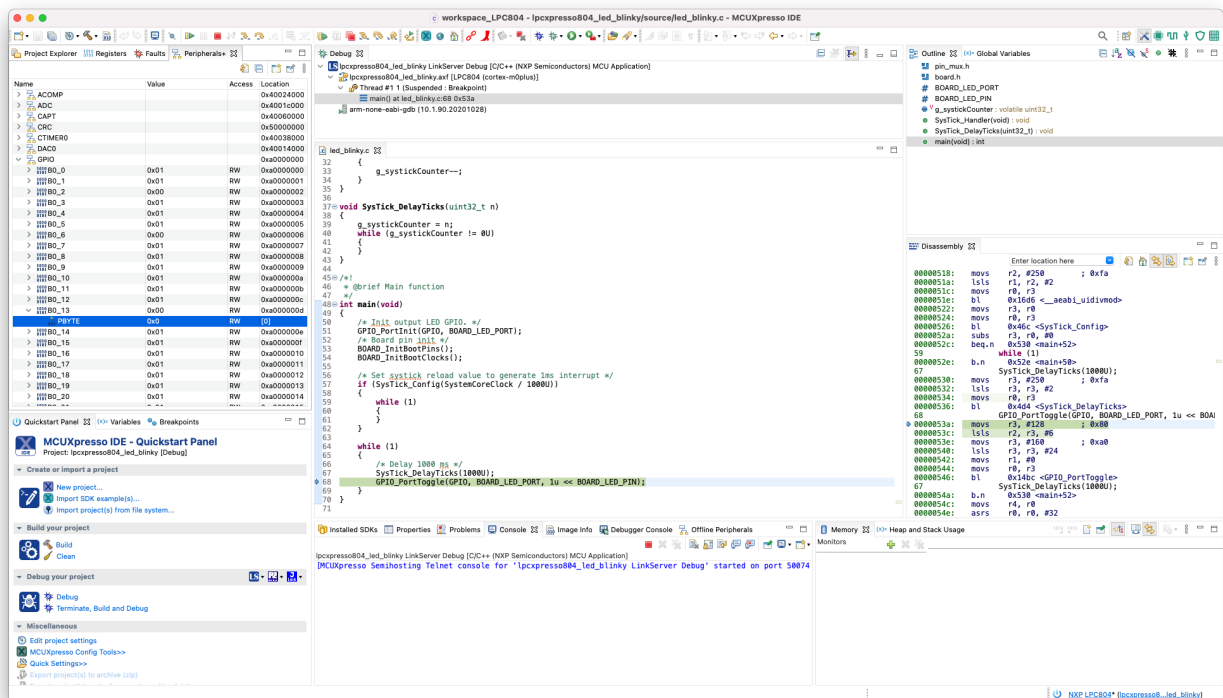
Programming of embedded systems

1. Introduction

15. Start *Debug* mode:



16. By pressing Step Over (F6) you can execute the program in steps. Pressing *Instruction Stepping Mode* opens the *Disassembly* window where you can see the generated assembly instructions. Go to the *Peripherals* tab and find the GPIO and then the B0_13 registry:



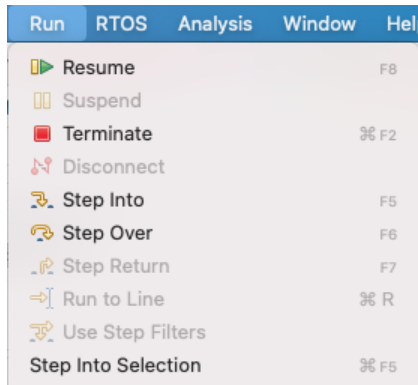
Programing of embedded systems

1. Introduction

17. Check how the value of the B0_13 register changes when the LED is on and off.
18. You can change value of B0_13 register directly in *Peripherals* tab, by write 0 or 1.

III. Exercises

1. Check the operation of other *Debugger* functions in the *Run* menu:



2. Check out the other examples provided with the SDK.

References

1. *LPC804 Data Sheet*, <https://www.nxp.com/>
2. *LPC804 User manual*, <https://www.nxp.com/>
3. *User Manual for LPCXpresso804 Board*, <https://www.nxp.com/>