AN13790

Testing CoreMark and Active Power Consumption on LPC86x

Rev. 0 — 8 May 2023 Application note

Document Information

Information	Content
Keywords	AN13790, LPC86x, CoreMark, Power consumption
Abstract	This application note describes how to test CoreMark and active power consumption on LPC86x.

1 Introduction

CoreMark, developed by EEMBC, is an industry-standard benchmark that measures the performance of central processing units (CPU) and embedded microcontrollers (MCU). Running CoreMark produces a single-number score allowing users to make quick comparisons between processors.

The LPC86x is an Arm Cortex-M0+ based, low-cost 32-bit MCU family for embedded applications. These devices include:

- Up to 64 kB of flash memory and 8 kB of SRAM
- Running at frequencies of up to 60 MHz
- Strengthened Code Read Protection (SCRP)
- A CRC engine
- One I^2C bus interface, one I^3C MIPI bus, up to three USARTs, up to two SPI interfaces
- One Multi-Rate Timer (MRT), Self-Wake-up Timer (WKT), Windowed Watchdog Timer (WWDT), two FlexTimers with DMA support
- One analog comparator
- One 12-bit ADC with up to 12 input channels with multiple internal and external trigger inputs and with sample rates of up to 1.9 Msamples/s
- Up to 54 general-purpose I/O pins

This application note describes how to test CoreMark and active power consumption on LPC86x. It also describes the steps to set up the test project, including code building and project settings for different software development tools (Keil MDK, IAR EWARM, and MCUXpresso). Furthermore, the document describes how to measure CoreMark and active power consumption on the board LPCXpresso860-MAX. At the end, the application note explains how to get the test result and draw some conclusions.

2 Setting up test project

This chapter describes the steps to set up the test project.

2.1 Code building for CoreMark

The software package associated with this application note contains SDK 2.11.1 based project framework that allows developers to drop in the CoreMark library sources and quickly get up and running with benchmarking the LPC86x. To get started, go to: https://www.eembc.org/coremark. Click the download link as shown in Figure 1, and it can be downloaded on GitHub.

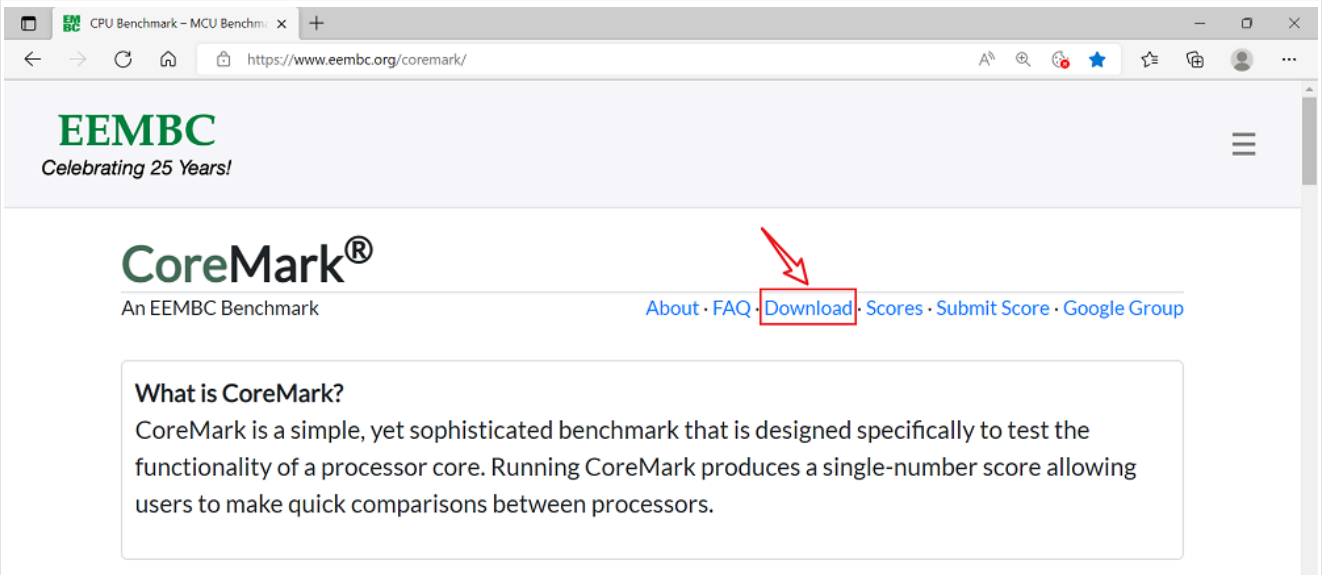

Figure 1. EEMBC CoreMark download link

2.1.1 Port CoreMark main-files into project

To port the CoreMark main files into the project, follow the steps below:

1. Copy the following mainfiles from the CoreMark package downloaded from EEMBC:
   - `core_list_join.c`
   - `core_main.c`
   - `core_matrix.c`
   - `core_state.c`
   - `core_util.c`
   - `coremark.h`

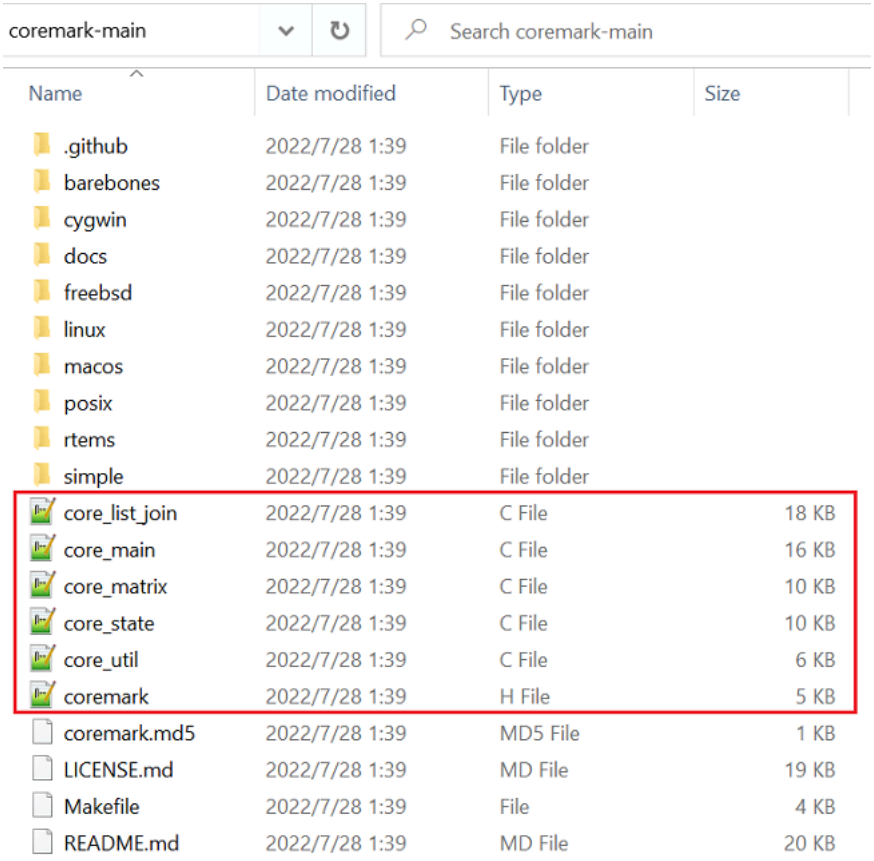

Figure 2. CoreMark main-files

2. Perform the following steps:
   a. For Keil MDK, place these files in the `coremark_mdk\source` project directory.
   b. For IAR Embedded Workbench, place these files in the `coremark_iar\source` project directory.
   c. For MCUXpresso, place these files in the `coremark_MCUXpresso\coremark\source` project directory.
3. Furthermore, all of these files must be added to the project. For Keil MDK, right-click the source folder. Select **Add Existing Files to Group 'xxx'**, and then add the files.

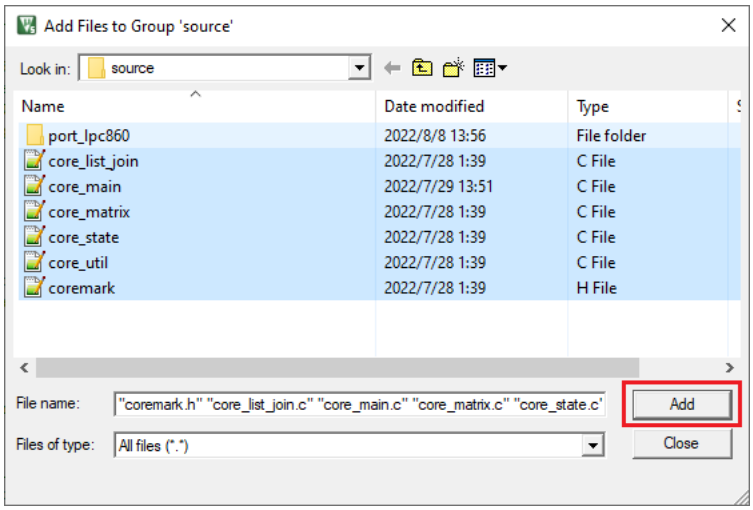

Figure 3. Adding Files in Keil MDK

4. For IAR Embedded Workbench, right-click the source folder. Select **Add** > **Add Files**, and add the files.

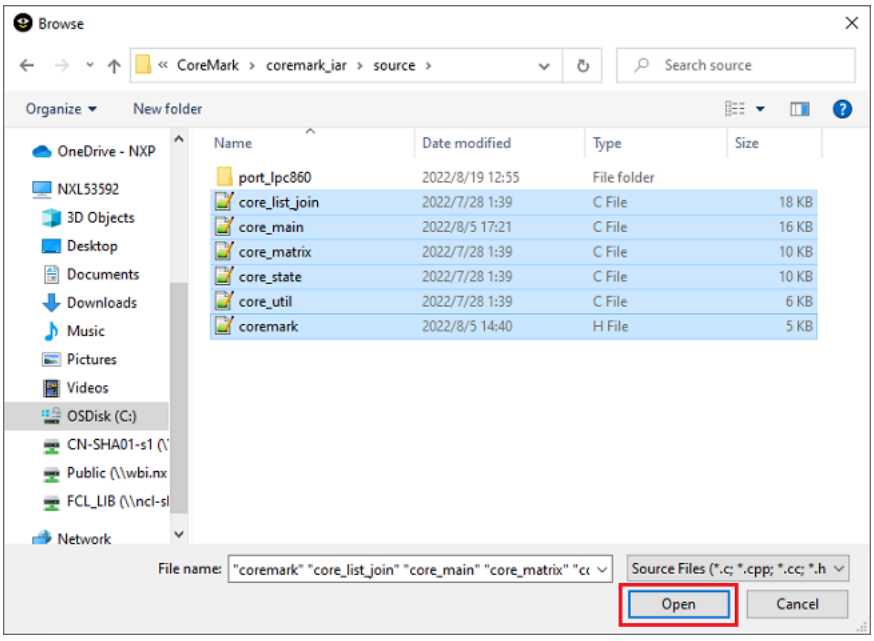

Figure 4. Adding Files in IAR Embedded Workbench

5. For MCUXpresso, copy the files in the "source" folder, then select **File** > **Refresh**.The files get added in project automatically.

2.1.2 Rebuild CoreMark port-files

Find the port-files `core_portme.c` and `core_portme.h` from the CoreMark package downloaded from EEMBC, in the `coremark_main\simple` directory, as shown in Figure 5.

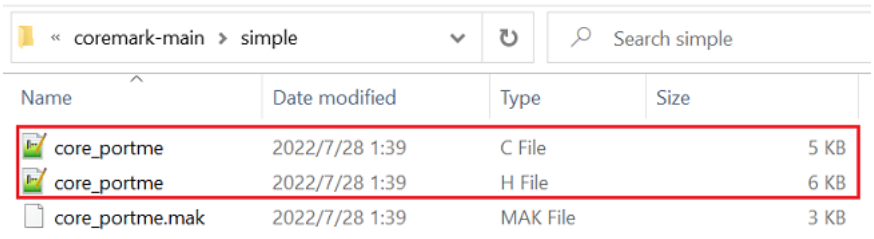

Figure 5. CoreMark port-files

To meet LPC86x's application requirements, the following files should be carefully rebuilt:

- For Keil MDK, the final files can be found in the `coremark_mdk\source\port_lpc860` project directory.
- For IAR Embedded Workbench, the final files can be found in the `coremark_iar\source\port_lpc860` project directory.
- For MCUXpresso, the final files can be found in the `coremark_MCUXpresso\coremark\source\port_lpc860` project directory.

2.2 Code building for active power measurement

Power measurements in active mode should be performed under the following conditions:

- Configure all pins as GPIO with pullup resistor disabled in the IOCON block
- Configure GPIO pins as outputs using the GPIO DIR register
- Write 1 to the GPIO CLR register to drive the outputs LOW

Follow the above mentioned rules, build the code, and integrate it into file `pin_mux.c`.

For Keil MDK, this file can be found in the `coremark_mdk\board` project directory.

For IAR Embedded Workbench, this file can be found in the `coremark_iar\board` project directory.

For MCUXpresso, the final files can be found in the `coremark_MCUXpresso\coremark\board` project directory.

2.3 IDE options setting

2.3.1 Execute code from internal flash

Projects can execute the code from the flash or SRAM memory regions. Here, the code executes from the flash memory region.

To execute the code from the flash memory region for Keil MDK:

Select **Project** > **Options for Target 'xxx'** > **Linker** tab. From the **Scatter File** list, select `LPC865_flash.scf`. The file is at the `coremark_mdk\LPC865_flash.scf` project directory.

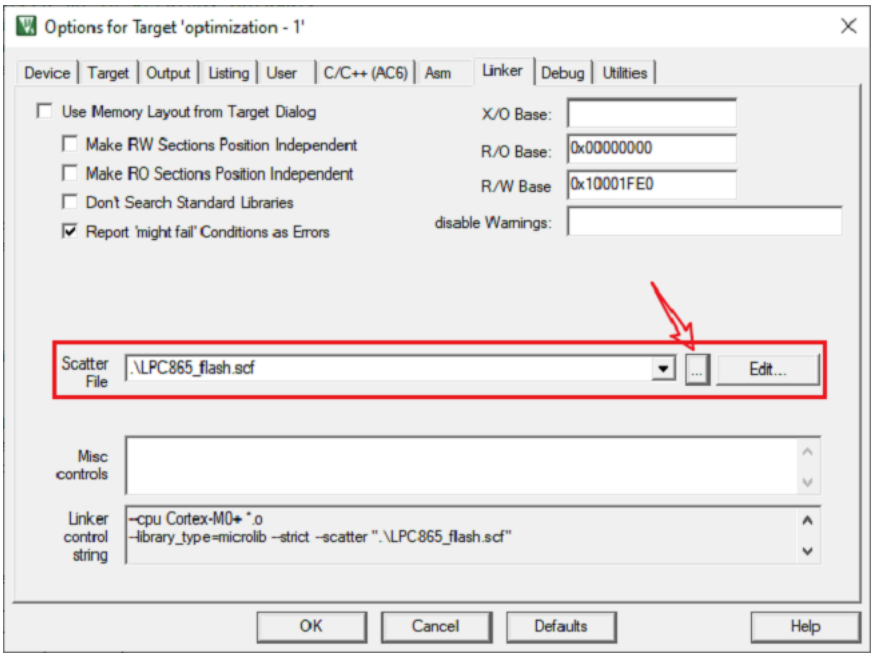

Figure 6. Keil MDK setting for executing code in internal flash

To execute code from flash memory region for IAR EWARM:

Select **Project** > **Options** > **Linker** > **Config** tab. In the **Linker configuration file** section, select `LPC865_flash.icf`. The file is at the `coremark_iar\LPC865_flash.scf` project directory.

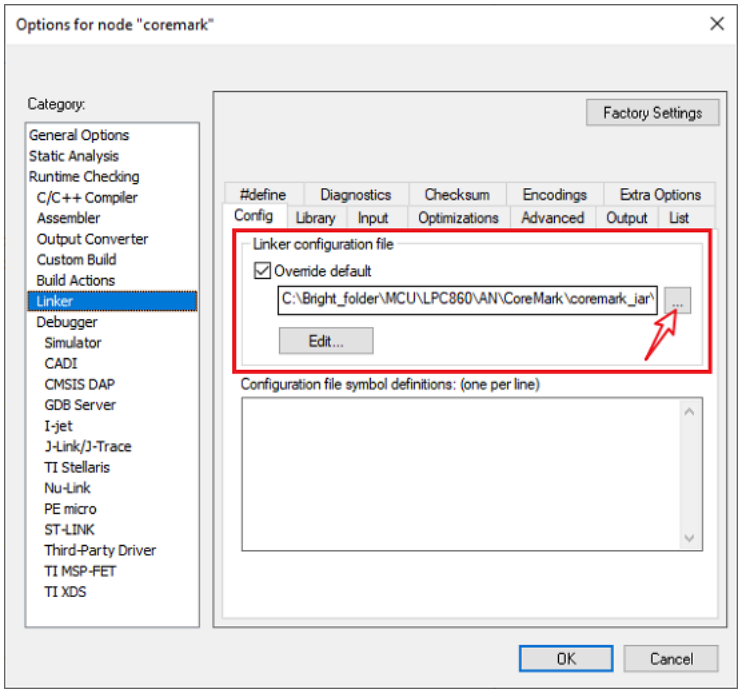

Figure 7. IAR EWARM setting for executing code in internal flash

To execute code from flash memory region for MCUXpresso:

Select **Project** > **Properties** > **C/C++ Build** > **Setting** and select the **Manage linker script** checkbox.

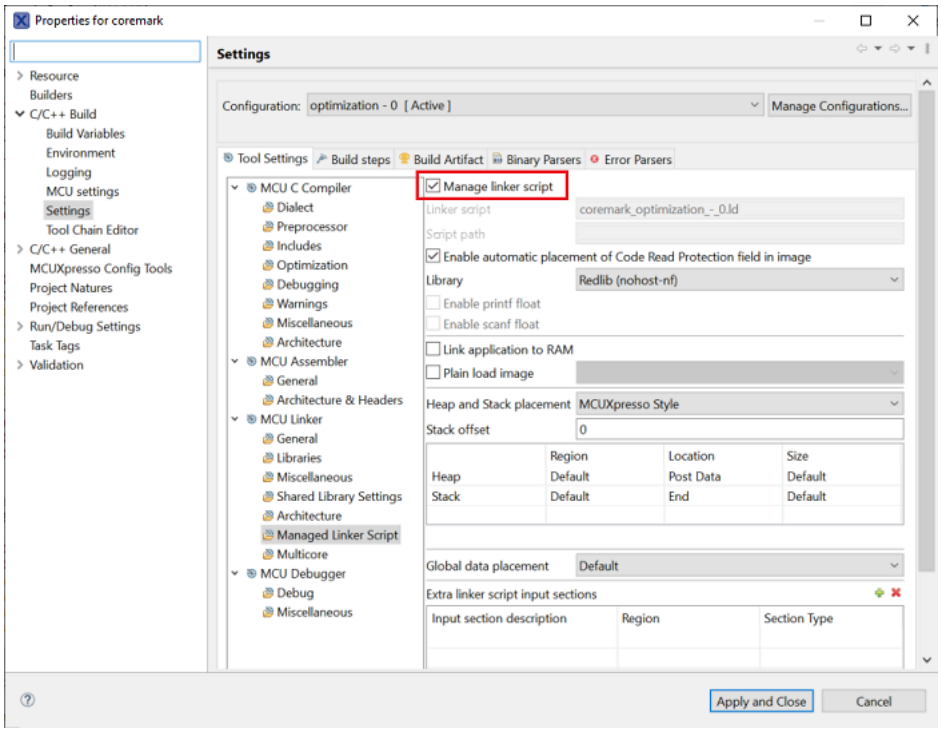

Figure 8. MCUXpresso setting for executing code in internal flash

2.3.2 Setup compiler include paths

To let the compiler find the header files, the path should be added in the project.

For Keil MDK:

Select **Project** > **Options for Target 'xxx'** > **C/C++ (AC6) tab** > **Include Paths**. Add the path that contains the header files.

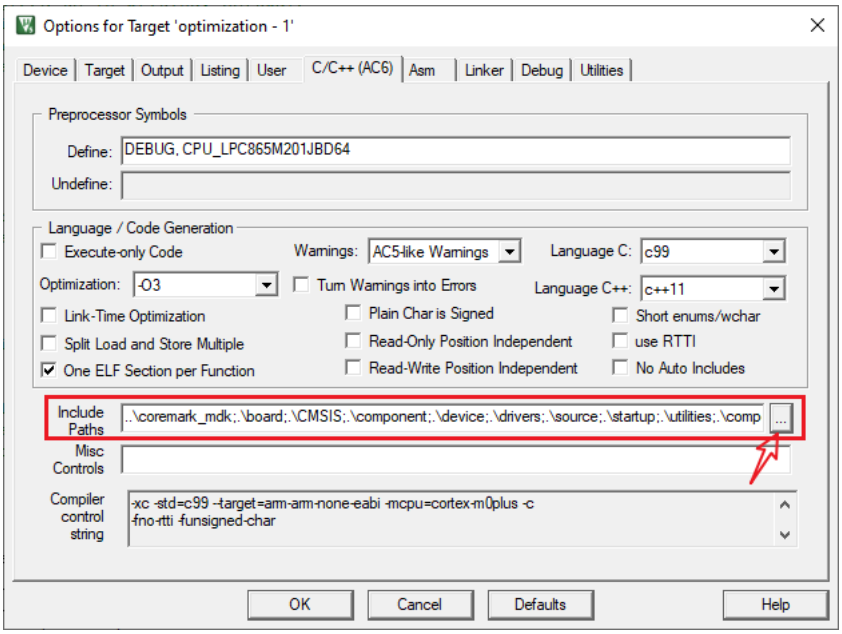

Figure 9. Keil MDK compiler include paths

For IAR EWARM:

Select **Project** > **Options** > **C/C++ Compiler** > **Preprocessor tab** > **Additional include directories: (one per line)**. Add the paths that contain the header files.

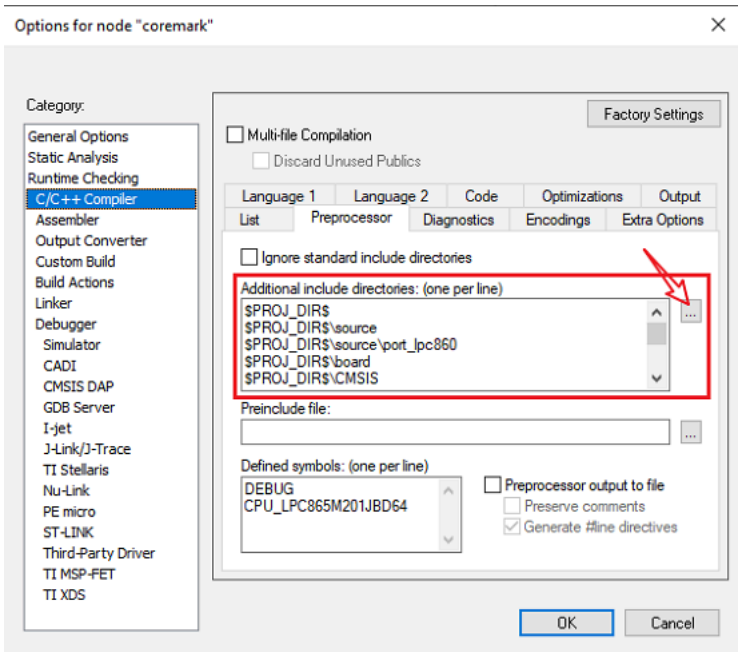

Figure 10. IAR EWARM compiler include paths

For MCUXpresso:

Select **Project** > **Properties** > **C/C++ Build** > **Setting** > **Includes tab** > **Include paths (-I)**. Add the paths that contain the header files.

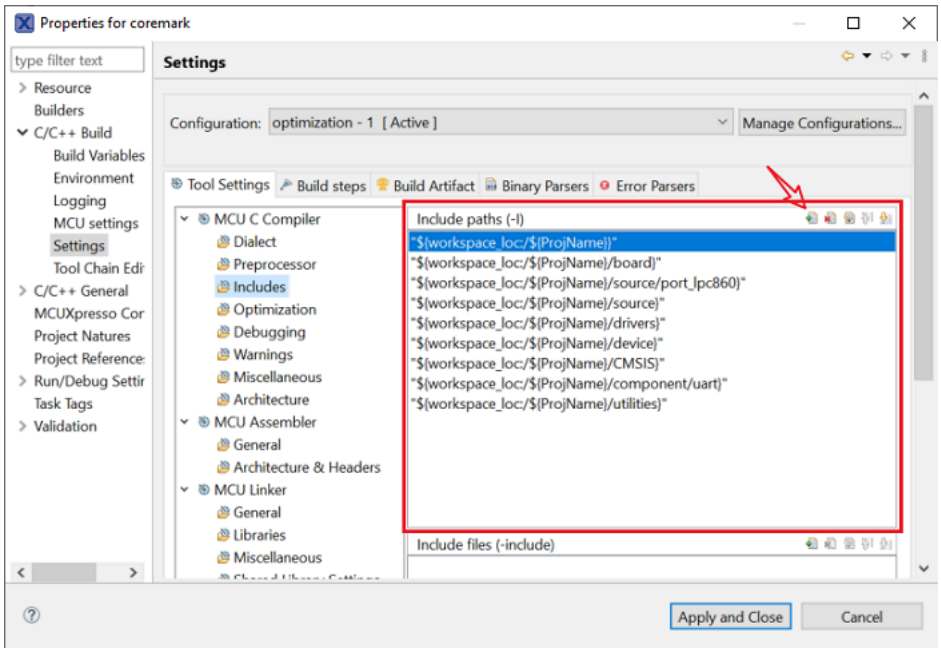

Figure 11. MCUXpresso compiler include paths

2.3.3 Optimization setting

To improve CoreMark score, the optimization level should be set higher. However, when benchmarking the power consumption of the MCU, the optimization level should be set to "None".

For Keil MDK:

Select **Project** > **Options for Target 'xxx'** > **C/C++ (AC6) tab** > **Optimization**. Then select **-O3** for CoreMark test and **-O0** for power consumption test.

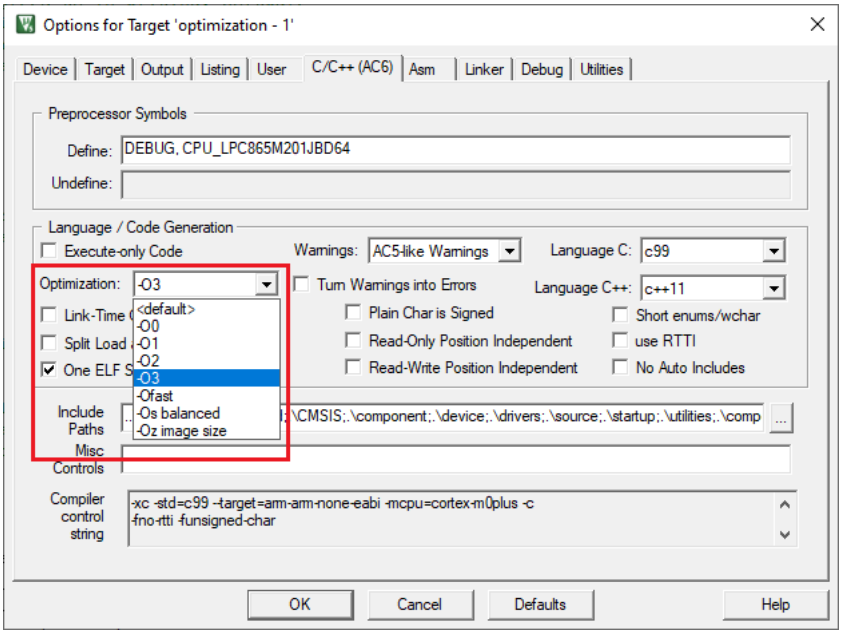

Figure 12. Keil MDK optimization setting

For IAR EWARM:

Select **Project** > **Options** > **C/C++ Compiler** > **optimizations tab** > **Level**. Set the optimization level to "High", select "Speed" from the list, and select the **No size constraints** checkbox for CoreMark test. Select "None" for power consumption test.

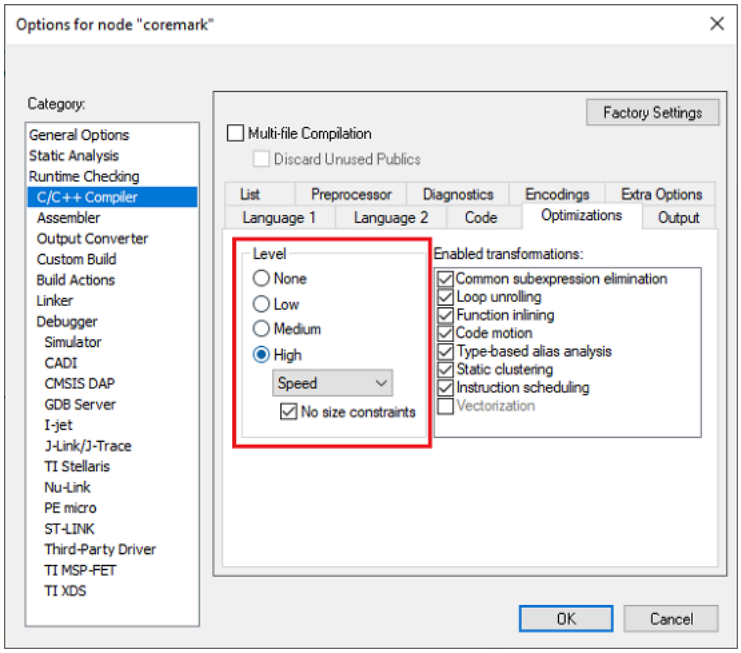

Figure 13. IAR EWARM optimization setting

For MCUXpresso:

Select **Project** > **Properties** > **C/C++ Build** > **Setting** > **Optimization tab** > **Optimization** Level. Set the optimization level to **Optimize most (-O3)** for CoreMark test and select **None (-O0)** for power consumption test.

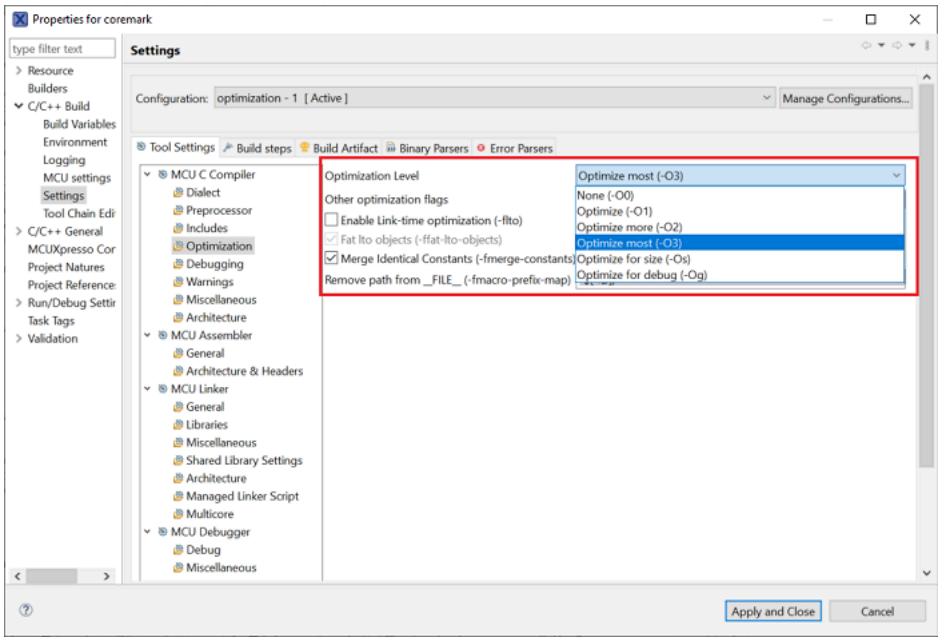

Figure 14. MCUXpresso optimization setting

2.3.4 Create another build configuration

The user can create more than one configuration in a project. Different configurations may contain different source files and execute different options. Therefore, creating another configuration can make testing more convenient.

Here, there are two build configurations – "optimization - 1" (optimization level: -O3) and "optimization - 0" (optimization level: -O0) to measure power consumption easily.

For Keil MDK:

Configuration can be created by clicking the arrow pointing icon. The active configurations can be selected via the dropdown list, as shown in Figure 15.

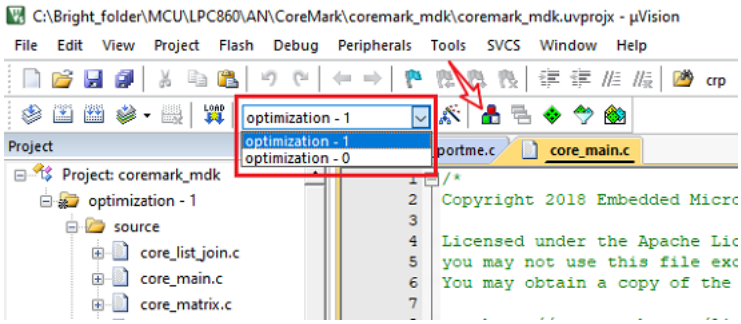

Figure 15. Creating Keil MDK configuration

For IAR EWARM:

Under **Project** > **Edit Configurations**, configurations can be created and edited. In addition, the user can select the active configuration the dropdown list, as shown in Figure 16.

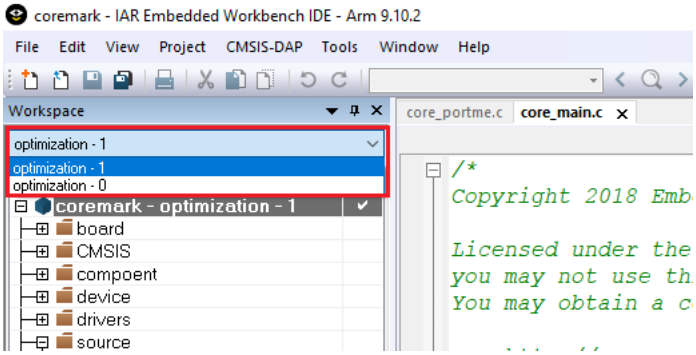

Figure 16. Creating IAR EWARM configuration

For MCUXpresso:

Under **Project** > **Build Configurations** > **Manage**, configurations can be created and edited. In addition, the user can send the configuration to active from **Project** > **Build Configurations** > **Set Active**, as shown in Figure 17.

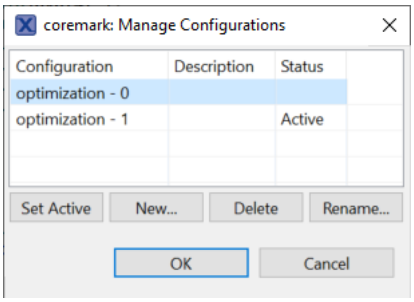

Figure 17. Creating MCUXpresso configuration

3 Measuring on board

This chapter describes the steps to measure CoreMark and power consumption on the board.

3.1 LPCXpresso860-MAX board

The LPCXpresso860-MAX board is a development board based on LPC865M201, as shown in Figure 18. The board supports a debug and serial port connection via J4.

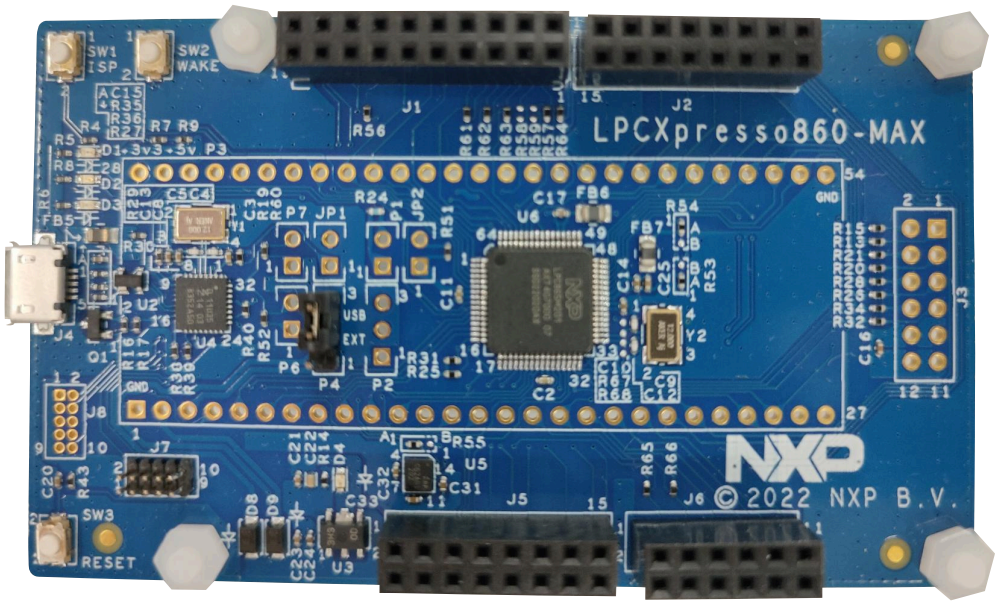

Figure 18. The LPCXpresso860-MAX board

The board ships with CMSIS_DAP debug firmware programmed. For more information on CMSIS_DAP debug firmware, visit https://www.nxp.com/downloads/en/software/lpc_driver_setup.exe.

For debugging and terminal debug messages, connect a USB cable to J4 USB connector. Board schematics are available on www.nxp.com.

3.2 Setup before measurement

3.2.1 Debug terminal setup

To observe debug messages from the board, a UART debug terminal (here, Tera Term) should be opened. First, set the terminal program to the appropriate COM port, and then use the setting "9600, 8, none, 1, none", as shown in Figure 19.

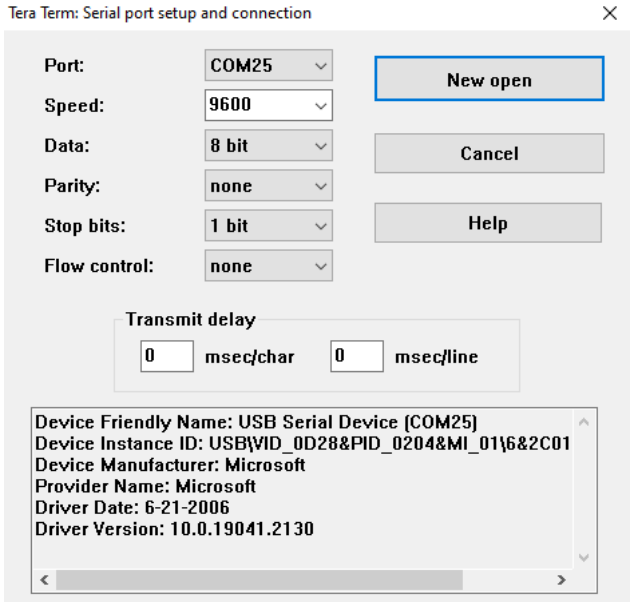

Figure 19. Debug terminal setup

3.2.2 Board setup

To measure the LPC86x power consumption, remove R51 to reserve interface for multimeter connection, and then connect multimeter across JP2, as shown in Figure 20.

Figure 20. Power consumption setup

Note: *To measure the real power consumption of the MCU, LEDs including D1/D2/D3 should be removed before the test.*

Note: *When the multimeter is not connected, connect pin jumper across JP2.*

3.3 Code running

First, use a USB cable to connect J4 with PC. The PC recognizes the serial port and displays the information in Device Manager, as shown in Figure 21.

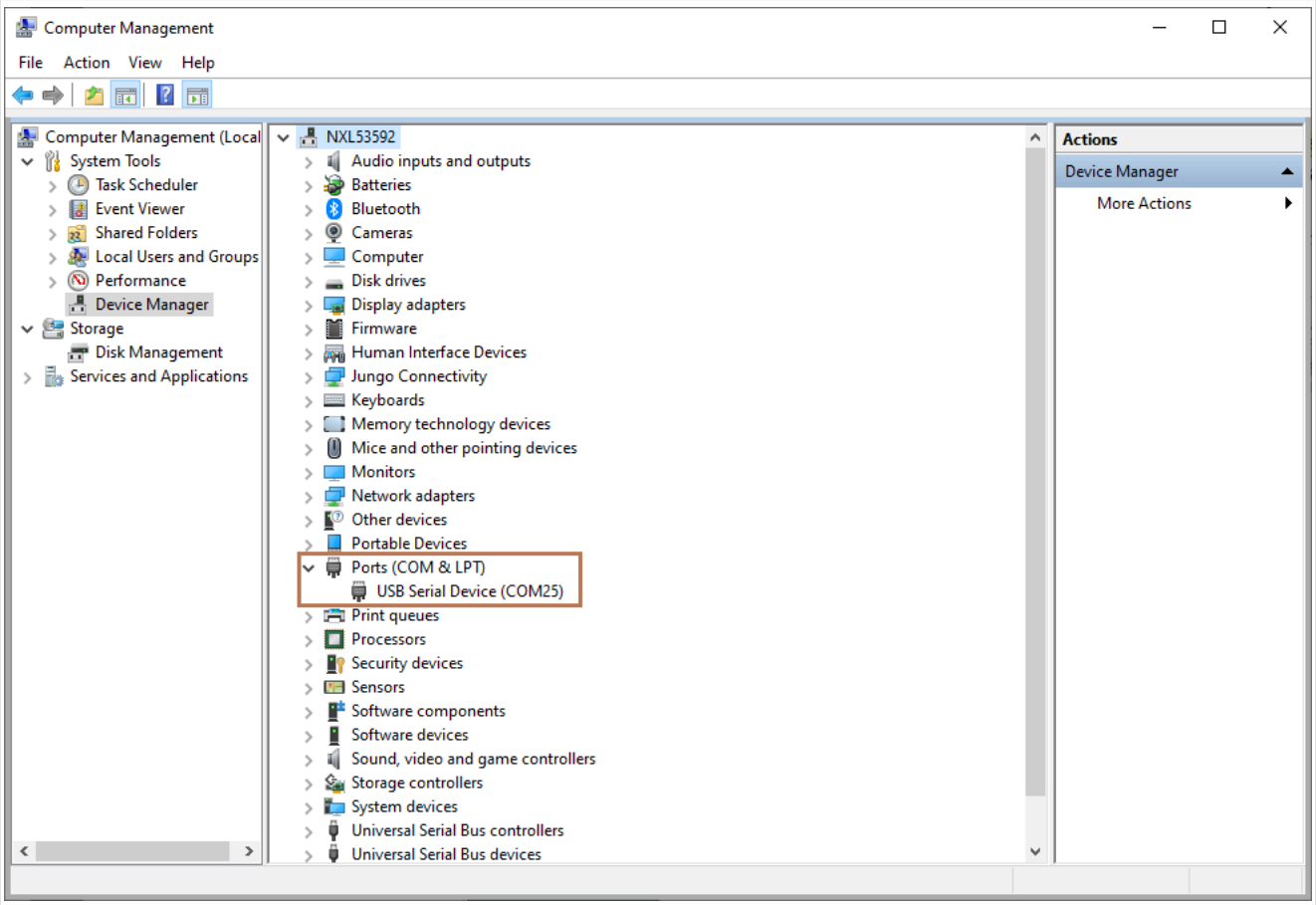

Figure 21. USB serial port on PC

Open a UART debug terminal (here, SSCOM) and configure, as described in Section 3.2.1.

3.3.1 CoreMark test

CoreMark test can be divided into two sections: CoreMark score measurement and CoreMark power consumption measurement.

Before the CoreMark score measurement, the project under the configuration of "optimization - 1" should be downloaded to the board. Then, click the reset button on board, as shown in Figure 22. The terminal displays the prompt information. The user can input 1 from the PC keyboard to select the CoreMark Test mode, and then input 1, 2, 3, and so on, to select the core frequency of LPC86x such as 18 MHz FRO, 24 MHz FRO, 30 MHz FRO, and so on.

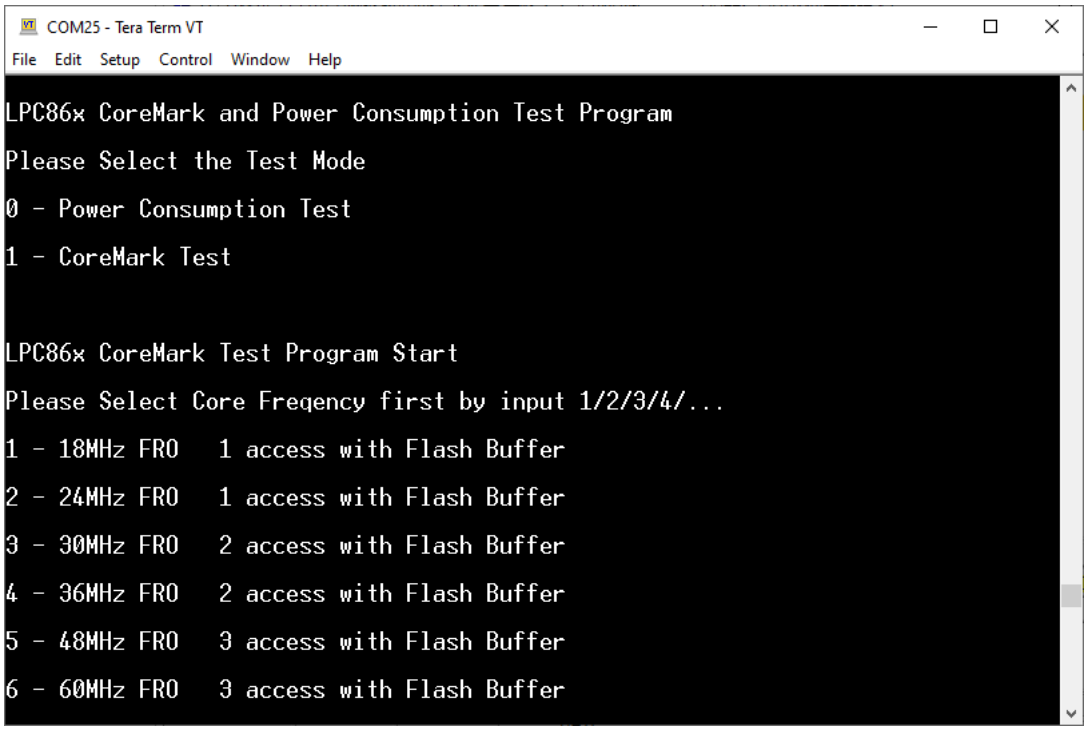

Figure 22. CoreMark score measurement choose menu

Once a character is input, the Coremark test program starts immediately and displays the information of system configuration, then waits for 10 seconds or more. The CoreMark benchmark then prints on the terminal after a few seconds, as shown in Figure 24.

Before the CoreMark power consumption measurement, the project under the configuration of "optimization - 0" should be downloaded to the board, and a multimeter should be connected across JP2. Then, click the reset button on board and repeat the operation above. At the same time, read the value on multimeter.

3.3.2 Active power consumption test

Before the power consumption test, the project under the configuration of "optimization - 0" should be downloaded to the board, and a multimeter should be connected across JP2. Then, click the reset button on the board, as shown in Figure 23. The terminal then displays the prompt information. The user can input 0 from PC keyboard to select the Power Consumption Test mode, and then input 0, 1, 2, and so on, to select the core frequency of LPC86x such as 1 MHz LPOSC, 18 MHz FRO, 24 MHz FRO, and so on.

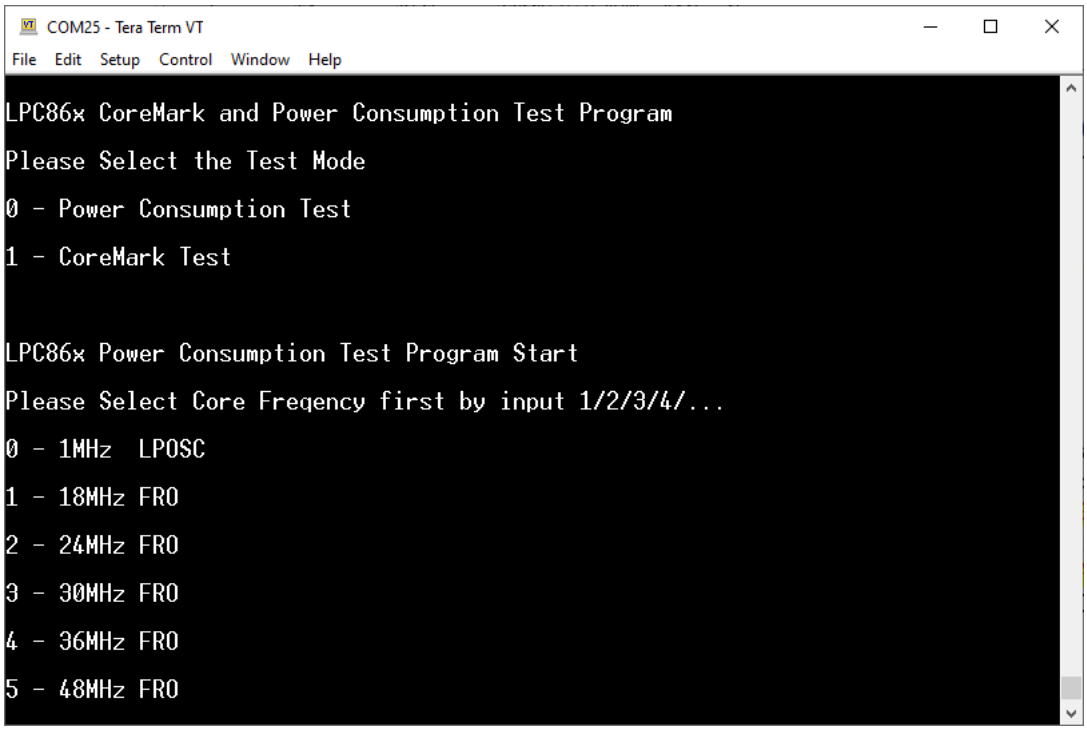

Figure 23. Active power consumption measurement choose menu

4 Result

This chapter describes the steps to obtain test results.

4.1 CoreMark test result

The CoreMark benchmark score is the number of iterations per second.

For CoreMark score measurement, for example, on IAR EWARM:

Under the condition "60 MHz FRO 3 access with Flash Buffer", the CoreMark score is 92.71, as shown in Figure 24, and the CoreMark/MHz score is 92.71/60 = 1.55 CoreMark/MHz.

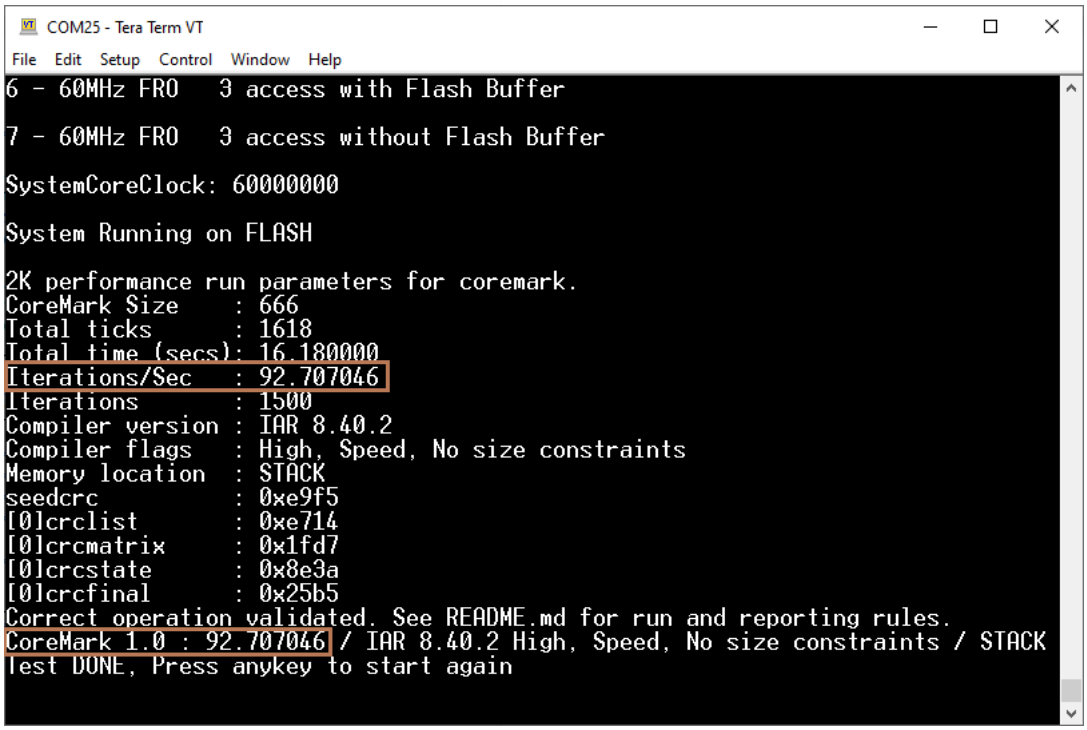

Figure 24. CoreMark score measurement test result

For CoreMark power consumption measurement:

At first, a multimeter should be connected across JP2 to measure the power consumption, as shown in Figure 20, and then fill in the calculation results in the table.

Table 1 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "18 MHz FRO, 1 access, with Flash Buffer, and Cache".

Table 1. LPCXpresso860-MAX board CoreMark test result (18 MHz FRO with flash buffer)

Frequency(MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
18	IAR EWARM	43.86	2.44	4.14	230.00
	Keil MDK	36.50	2.03	4.24	235.56
	MCUXpresso	34.60	1.92	4.23	235.00

Table 2 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "24 MHz FRO, 1 access, with Flash Buffer, and Cache".

Table 2. LPCXpresso860-MAX board CoreMark test result (24 MHz FRO with flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
24	IAR EWARM	58.48	2.44	5.39	224.58
	Keil MDK	48.66	2.03	5.53	230.42
	MCUXpresso	46.12	1.92	5.49	228.75

Table 3 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "30 MHz FRO, 2 access, with Flash Buffer, and Cache".

Table 3. LPCXpresso860-MAX board CoreMark test result (30 MHz FRO with flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
30	IAR EWARM	59.30	1.98	6.06	202.00
	Keil MDK	50.28	1.68	6.37	212.33
	MCUXpresso	49.78	1.66	6.26	208.67

Table 4 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "36 MHz FRO, 2 access, with Flash Buffer, and Cache".

Table 4. LPCXpresso860-MAX board CoreMark test result (36 MHz FRO with flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
36	IAR EWARM	71.17	1.98	7.09	196.94
	Keil MDK	60.35	1.68	7.45	206.94
	MCUXpresso	59.74	1.66	7.32	203.33

Table 5 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "48 MHz FRO, 3 access, with Flash Buffer, and Cache".

Table 5. LPCXpresso860-MAX board CoreMark test result (48 MHz FRO with flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
48	IAR EWARM	73.92	1.54	8.21	171.04
	Keil MDK	64.16	1.34	8.84	184.17
	MCUXpresso	66.23	1.38	8.62	179.58

Table 6 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "60 MHz FRO, 3 access, with Flash Buffer, and Cache".

Table 6. LPCXpresso860-MAX board CoreMark test result (60 MHz FRO with flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
60	IAR EWARM	92.71	1.55	10.12	168.67
	Keil MDK	80.21	1.34	10.90	181.67
	MCUXpresso	82.78	1.38	10.62	177.00

Table 7 shows typical CoreMark score when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "60 MHz FRO, 3 access, without Flash Buffer and Cache".

Table 7. LPCXpresso860-MAX board CoreMark test result (60 MHz FRO without flash buffer)

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
60	IAR EWARM	78.04	1.30	9.51	158.50

Table 7. LPCXpresso860-MAX board CoreMark test result (60 MHz FRO without flash buffer)*...continued*

Frequency (MHz)	IDE	CoreMark	CoreMark/MHz	Power Consumption(mA)	µA/MHz
	Keil MDK	67.42	1.12	10.01	166.83
	MCUXpresso	60.39	1.01	9.87	164.50

4.2 Active power consumption test result

To measure the LPC86x active power consumption, connect the multimeter across JP2, as shown in Figure 20.

Note: *The current data on the board may be slightly higher than the data sheet because the board has more other components that may cost more power.*

Table 8 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "1 MHz LPOSC".

Table 8. LPCXpresso860-MAX board active power consumption test result (1 MHz LPOSC)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
1	IAR EWARM	0.37	370.00
	Keil MDK	0.37	370.00
	MCUXpresso	0.36	360.00

Table 9 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "18 MHz FRO".

Table 9. LPCXpresso860-MAX board active power consumption test result (18 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
18	IAR EWARM	2.95	163.89
	Keil MDK	3.05	169.44
	MCUXpresso	2.85	158.33

Table 10 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "24 MHz FRO".

Table 10. LPCXpresso860-MAX board active power consumption test result (24 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
24	IAR EWARM	3.83	159.58
	Keil MDK	3.93	163.75
	MCUXpresso	3.68	153.33

Table 11 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "30 MHz FRO".

Table 11. LPCXpresso860-MAX board active power consumption test result (30 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
30	IAR EWARM	4.70	156.67
	Keil MDK	4.86	162.00

Table 11. LPCXpresso860-MAX board active power consumption test result (30 MHz FRO)*...continued*

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
	MCUXpresso	4.52	150.67

Table 12 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "36 MHz FRO".

Table 12. LPCXpresso860-MAX board active power consumption test result (36 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
36	IAR EWARM	5.47	151.94
	Keil MDK	5.67	157.50
	MCUXpresso	5.26	146.11

Table 13 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "48 MHz FRO".

Table 13. LPCXpresso860-MAX board active power consumption test result (48 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
48	IAR EWARM	7.16	149.17
	Keil MDK	7.34	152.92
	MCUXpresso	6.87	143.13

Table 14 shows power consumption when benchmarked on Keil MDK/IAR EWARM/MCUXpresso under the condition "60 MHz FRO".

Table 14. LPCXpresso860-MAX board active power consumption test result (60 MHz FRO)

Frequency(MHz)	IDE	Power Consumption(mA)	µA/MHz
60	IAR EWARM	8.82	147.00
	Keil MDK	9.03	150.50
	MCUXpresso	8.47	141.17

5 Conclusion

In this document, CoreMark and active power consumption are measured on the LPC86x with different IDEs (Keil MDK, IAR EWARM, and MCUXpresso). It describes how to port and edit the code, and at the same time, it introduces the process of measurement.

The CoreMark results measured on board LPCXpresso860-MAX show that many factors (including core frequency, flash memory access time, flash buffer, and flash cache) affect CoreMark score and power consumption. Usually, the higher the core frequency, the higher the CoreMark score, but the higher the power consumption. The best CoreMark number is 92.71. It is achieved by using IAR EWARM running at 60 MHz, and the lowest power consumption is 4.14 mA, achieved by using IAR EWARM running at 18 MHz.

The active power consumption results measured on board LPCXpresso860-MAX show that the higher the core frequency, the higher the power consumption. However, the µA/MHz decreases when the core frequency increases. The lowest active power consumption is 0.36 mA, achieved by using MCUXpresso running at 1 MHz.

6 Reference

- *LPC55(S)0x CoreMark Porting Guide* (document AN13035)
- *LPC86x User manual* (document UM11607)
- *LPC86x Data Sheet* (document LPC86x)

7 Revision history

Table 15 summarizes the changes done to this document since the initial release.

Revision history

Revision number	Date	Substantive changes
0	08 May 2023	Initial release

8 Legal information

8.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. - NXP B.V. is not an operating company and it does not distribute or sell products.

8.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

For more information, please visit: http://www.nxp.com

Date of release: 8 May 2023
Document identifier: AN13790