

## 1 简介

### 1.1 概述

LPC55xx/LPC55Sxx 是一款基于 Arm®Cortex®-M33 的嵌入式应用微控制器。这些设备包括：

- Arm Cortex-M33 协处理器
- CASPER 加密/ FFT 引擎
- 用于 DSP 功能的 PowerQuad 硬件加速器
- 高达 320 KB 的片上 SRAM
- 高达 640 KB 的片上 flash
- PRINCE 模块，用于实时 flash 加密/解密
- 高速和全速 USB 主机和设备接口，无需晶体操作即可实现全速
- SDIO/MMC
- 五个通用计时器
- 一个 SCTimer / PWM
- 一个 RTC/警报计时器
- 一个 24 位多速率计时器 ( MRT )
- 一个窗口式看门狗定时器 ( WWDT )
- 一个高速 SPI ( 50 MHz )
- 九个灵活的串行通信外围设备 ( 可以配置为 USART、SPI、I2C 或 I2S 接口 )
- 可编程逻辑单元 ( PLU )
- 一个 16 位 1.0 Msamples / sec ADC
- 一个比较器
- 一个温度传感器

LPC55xx/LPC55Sxx 提供了两个 Arm Cortex-M33 内核，它们具有以下特性：

- Arm Cortex-M33 内核 ( CPU0 , r0p3 )
  - 以高达 100 MHz 的频率运行
  - 信任区、浮点单元 ( FPU ) 和内存保护单元 ( MPU )
  - Arm Cortex M33 内置嵌套矢量中断控制器 ( NVIC )
  - 不可屏蔽中断 ( NMI ) 输入支持来源选择
  - 具有八个断点和四个监视点的串行线调试，包括用于增强调试功能的串行线输出

### 目录

<b>1</b>	<b>简介</b> .....	<b>1</b>
1.1	概述.....	1
1.2	双核基本机制.....	2
1.3	相关系统资源.....	2
1.4	调试系统.....	2
<b>2</b>	<b>调试环境</b> .....	<b>3</b>
<b>3</b>	<b>MCUXpresso 双核项目配置</b> .....	<b>3</b>
3.1	内存配置.....	4
3.2	架构配置.....	5
3.3	多核配置.....	5
<b>4</b>	<b>在 MCUXpresso 上进行双核项目调试</b> .....	<b>6</b>
4.1	启动调试会话.....	6
4.2	从调试设置.....	7
4.3	双核通信调试.....	8
<b>5</b>	<b>结论</b> .....	<b>11</b>



- 系统计时器
- Arm Cortex-M33 内核 ( CPU1 , r0p3 )
  - 以高达 100 MHz 的频率运行
  - 这个实例的配置不包括 MPU、FPU、DSP、ETM 和 Trustzone
  - 系统计时器

## 1.2 双核基本机制

LPC55xx/LPC55Sxx 中的双核属于非对称体系结构，这意味着一个核 ( CPU0 ) 是主核，另一个核 ( CPU1 ) 是从核。默认情况下，CPU0 被设置为 master，可以正常工作，而 CPU1 被设置为 slave，当芯片启动时，CPU1 被设置为 hold，它的时钟被禁用。要让从核工作，需要主核通过寄存器来释放它和使其时钟。

在双核工作的情况下，它们需要彼此通信，因此 LPC55xx/LPC55Sxx 提供了具有以下特性的 CPU 间邮箱机制。

- 提供处理器间通信，允许多个 CPU 以简单的方式共享资源和彼此通信。
- 每个 CPU 可以给它的伙伴产生多达 32 个用户定义的中断。
- 每个 CPU 可以申请共享资源，如果它是可用的。
- 为通信握手提供互斥配置。

## 1.3 相关系统资源

Arm Cortex M33 包括三个 AHB-Lite 总线、一个系统总线和 I-code、D-code 总线。一条总线专用于指令获取 ( I-code )，一条总线专用于数据访问 ( D-code )。如果并行操作对象是不同的设备，则使用两个核心总线允许同时操作。

这两个 CPU 共享 LPC55xx/LPC55Sxx 中的所有资源(内存和外设)，以在双核使用中获得更好的性能。

LPC55xx/LPC55Sxx 支持 320 KB SRAM，带有用于更高吞吐量的单独总线主访问和用于低功耗操作的单独电源控制。这样可以将两个 CPU 的代码和数据分开来存储和访问。

LPC55xx/LPC55Sxx 通过允许在矩阵的不同从端口上的外设被不同的总线主设备同时访问，以使用多层 AHB 矩阵以灵活的方式将 CPU 总线和其他总线主设备连接到外设，从而优化性能。

## 1.4 调试系统

LPC55xx/LPC55Sxx 的双核调试系统具有以下特点。

- 它支持 arm 串行线调试模式的 CPU0 和 CPU1 ( 如果存在 )。
- 跟踪端口在 CPU0 和 CPU1 上都提供了 Cortex-M33 CPU 指令跟踪功能。通过串行线查看器输出。
- 直接调试访问所有内存、寄存器和外设。
- 调试会话不需要目标资源。
- 断点：CPU0 和 CPU10 包含 8 个指令断点。
- 监视点：CPU0 和 CPU10 M33 包括四个数据监视点，也可以用作触发器。
- 支持 JTAG 边界扫描。
- Macrocell 允许附加的软件控制跟踪 CPU0 和 CPU1。

调试系统包括以下模块。

- JTAG-TAP：NXP 产品和测试工程团队使用测试访问端口。
- DAP：具有串行线端口 ( SWJ-DP ) 的调试访问端口，它解释传入的数据并路由到适当的访问端口 ( AP )。
- CPU0 AP：实例化为 CPU0 的 Cortex-M33 核心的调试访问端口。
- CPU1 AP：实例化为 CPU1 的 Cortex-M33 核心的调试访问端口。CM33 的实例没有安全扩展 ( Armv8-M 的信任区 )。

- DM-AP：调试邮箱的调试访问端口。
  - 此端口总是启用的，外界可以从 ROM 发送和接收数据。
  - 此端口用于执行 NXP 调试认证协议 1.0 版本。

图 1 显示了上层调试端口和连接。

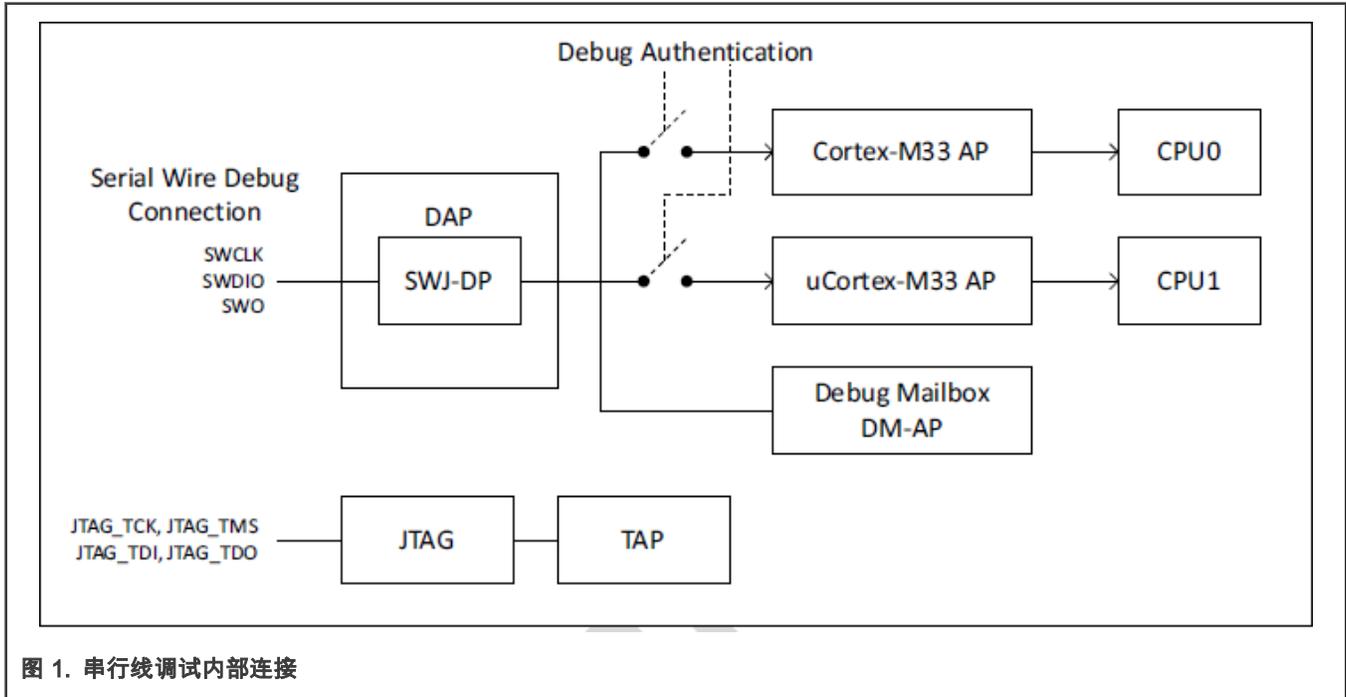


图 1. 串行线调试内部连接

## 2 调试环境

- 硬件：
  - mini/micro USB 电缆
  - LPCXpresso55s69 开发板
  - 个人计算机
- 工具链：
  - MCUXpresso IDE V10.3.0
- 软件：
  - SDK\_2.5.0\_LPCXpresso55s69\_EAR\_3

## 3 MCUXpresso 双核项目配置

双核项目可以以多种方式设计，但是在 MCUXpresso IDE 中存在一个基本的预期，即一个核（主核）将控制（或者至少是启动）其他核（从核）上的代码的执行。MCUXpresso IDE 中的双核项目由两个链接的项目组成：一个项目包含主代码，另一个项目包含从代码。

在启动或复位之后，主核先引导启动，然后负责引导从核。但是，这种关系只适用于引导过程；启动后，应用程序可以将其中一个核作为主内核或从内核。

MCUXpresso IDE 允许轻松创建支持多核 mcu 目标的链接项目。本应用笔记只基于 LPC5500 SDK 里的双核互斥例程（路径是 boards\lpcxpresso55s69\driver\_examples\mailbox\mutex）来介绍双核项目配置。并没有介绍如何创建此配置（关于此信息，可以阅读 MCUXpresso IDE 文件夹下的 MCUXpresso IDE 用户指南）。双核模式下，主从项目配置如下：

1. 存储配置
2. 架构配置
3. 多核配置

首先，启动 MCUXpresso IDE 工作区，安装 LPC55S69 SDK，并在主核和从核上导入邮箱互斥例程。然后可以在 MCUXpresso 中显示 SDK 和例程项目，如 图 2 所示。



### 3.1 内存配置

必须管理内存配置，以避免意外的重叠。在这些项目中，从应用程序完全在主应用程序未使用的 RAM 位置执行。表 1 显示了双核映像的地址分配。

表 1. 双核映像的地址分配

CPU	存储地址	加载地址
主核	0x0	0x0
从核	取决于编译器（在 flash 中）	0x20033000

相应地，图 3 显示了 MCUXpresso 中两个项目的内存配置。突出显示的命名 Ram1（在主项目中别名为 RAM2，在从项目中别名为 RAM）是从应用程序的执行位置。在多核配置中别名 RAM2 被引用（参见图 5）。

**Memory details (LPC55S69)\* -- Master**

Default LinkServer Flash Driver

Type	Name	Alias	Location	Size	Driver
Flash	PROGRAM_FLASH	Flash	0x0	0xa0000	LPC55xx.cfx
RAM	Ram0	RAM	0x20000000	0x31800	
RAM	Ram1	RAM2	0x20033000	0x11000	

**Memory details (LPC55S69)\* -- Slave**

Default LinkServer Flash Driver

Type	Name	Alias	Location	Size	Driver
RAM	Ram1	RAM	0x20033000	0x11000	

图 3. 主从项目中的内存配置

### 3.2 架构配置

如上文介绍，主核和从核都是基于 Arm Cortex-M33 的，尽管两者有一些区别。在主项目和从项目中相应地配置了架构，如 图 4 所示。

**Master:**

Architecture

Floating point

**Slave:**

Architecture

Floating point

图 4. 主从项目中的架构配置

### 3.3 多核配置

多核配置显示了主项目和从项目之间的链接，如 图 5 所示。它表明从应用程序映像（命名为 **M33SLAVE**）集成在主内存区域中。从映像的主内存区域的别名为 RAM2，这与内存配置是一致的（参见 图 3）。



图 5. 主从项目中的多核配置

## 4 在 MCUpresso 上进行双核项目调试

### 4.1 启动调试会话

成功地构建两个项目之后，首先启动主项目的调试会话。在工作区中单击项目名称以选择主项目，然后在 **Quick 面板** 中单击 **Debug**。

**注**

将 USB 线连接到 LPCXpresso55S69 板上标有 **Debug Link** 的 USB 插孔上进行调试。

对于第一次启动，选择 SWD 设备-CPU 核。应为主核/项目选择设备 **0**，为从核/项目选择设备 **1**。图 6 显示了主调试会话的 SWD 配置。

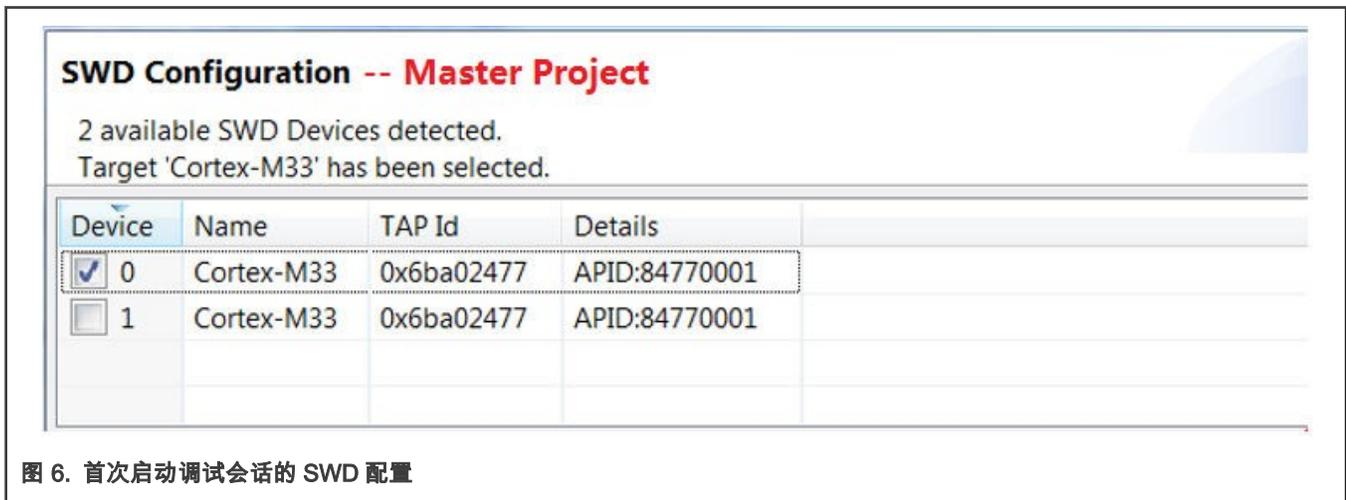


图 6. 首次启动调试会话的 SWD 配置

在完成两个调试会话的启动之后，主调试线程被挂起（基于此项目中的断点设置，主 PC 指针停在代码中的 main()），并且从调试线程无法停止进行调试，因为从核被搁置，需要由主核释放。在图 7 的 **Debug** 窗口中显示了两个调试线程的状态。

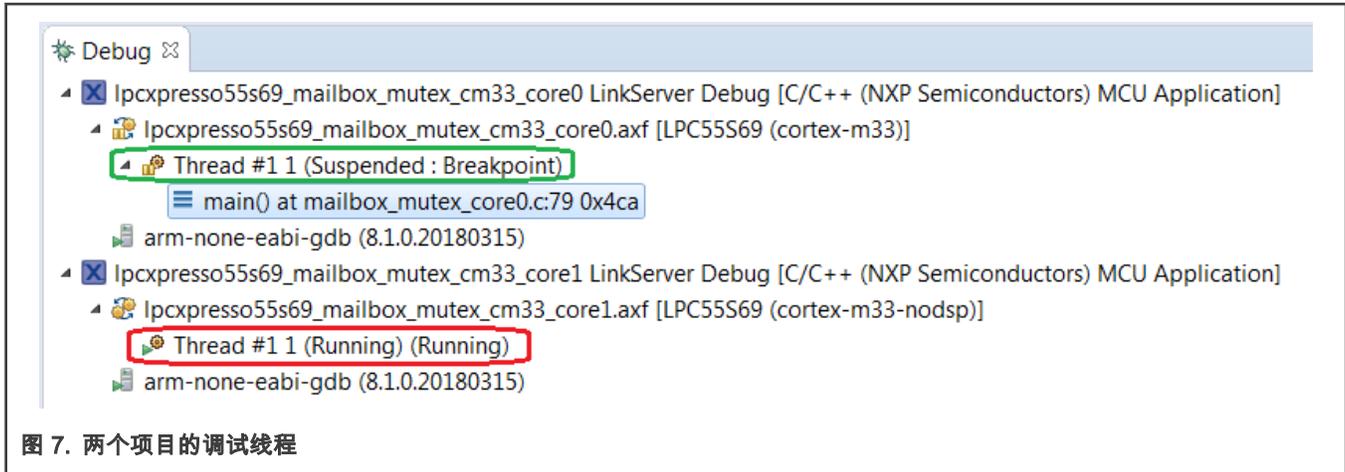


图 7. 两个项目的调试线程

## 4.2 从调试设置

这时可以在主项目上执行调试操作，如单步执行、运行等等。

在图 7 的 Debug 窗口中单击绿色或红色块，以选择要调试的主线程或从线程。在主项目中执行完启动从核的代码行之后（请参见图 8 中绿色突出显示的代码），从核将被释放以执行工作。当前的调试线程将切换到从线程且其 PC 指针将在从代码中的 main() 处停止（参见图 9）。这意味着可以在从项目上执行调试控制（例如单步执行）。

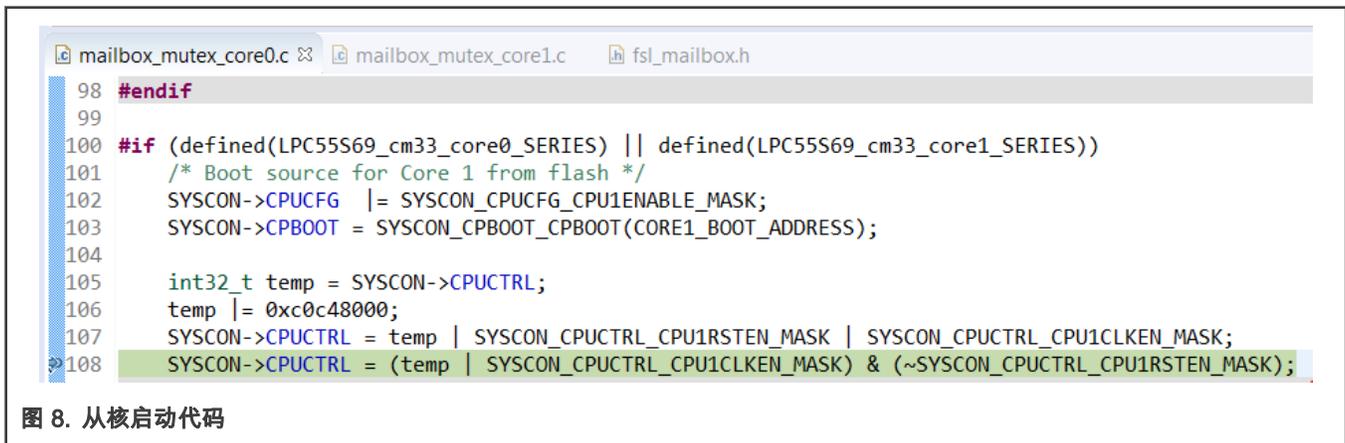
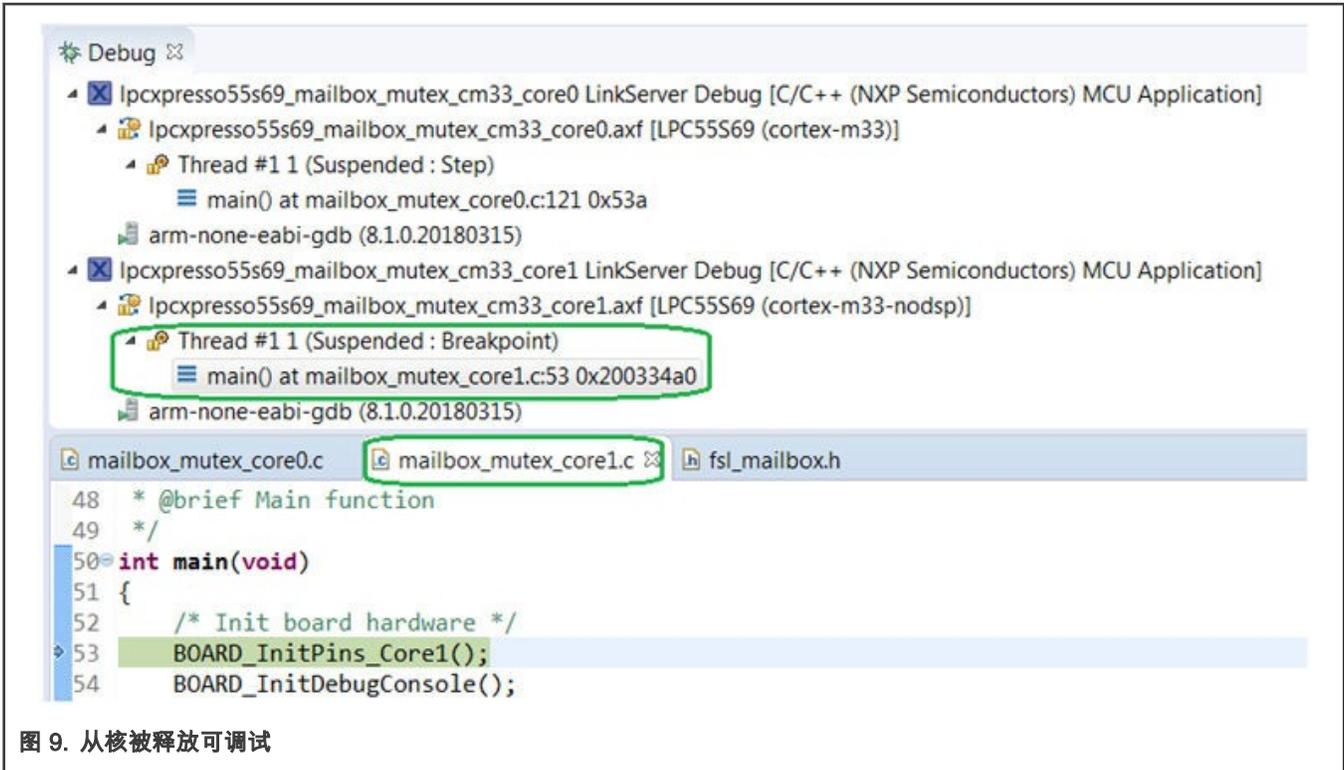
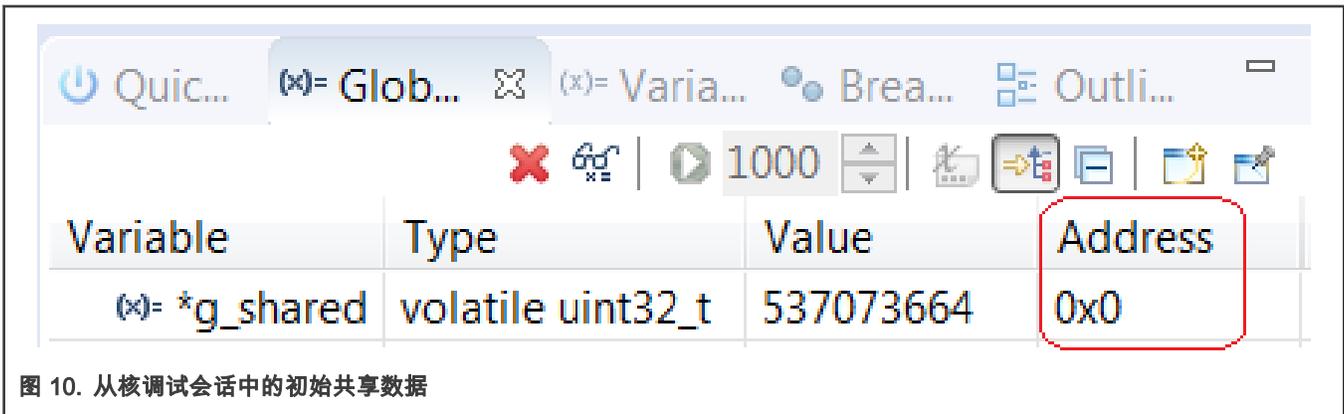


图 8. 从核启动代码



### 4.3 双核通信调试

切换到从调试线程后，在对双核通信过程执行调试控制之前，添加共享变量 \*g\_shared，即双核之间通过邮箱通信的共享数据，到从调试会话的全局变量窗口 (\*g\_shared 是指向 \*g\_shared 地址的指针)。作为初始状态，变量的当前值是随机的，并且如从代码中所定义的当前地址为 0。如 图 10 所示。



开始并继续从代码进入 while(1)循环，其中已实现的函数将获取邮箱互斥锁控制、更新全局变量和设置邮箱互斥锁控制。图 11 显示了循环代码。

```

62     while (1)
63     {
64         /* Get Mailbox mutex */
65         while (MAILBOX_GetMutex(MAILBOX) == 0)
66             ;
67
68         /* The core1 has mutex, can change shared variable g_shared */
69         if (g_shared != NULL)
70         {
71             (*g_shared)++;
72             PRINTF("Core1 has mailbox mutex, update shared variable to: %d\r\n", *g_shared);
73         }
74
75         /* Set mutex to allow access other core to shared variable */
76         MAILBOX_SetMutex(MAILBOX);
77     }
    
```

图 11. 从核上的主循环进程

可以观察到，获取邮箱互斥的代码被执行过了。但是，全局变量 \*g\_shared 将不会被更新，因为初始地址为零 ( NULL )，并且还没有从主核通过邮箱传输。\*g\_shared 的地址值在全局变量窗口中保持为零，如 图 10 所示。

单击 Debug 窗口中的绿色块 ( 如 图 7 所示 ) 转到主调试线程来通过邮箱将共享变量 g\_shared 的地址发送到从核。单击 mailbox\_mutex\_core0.c 文件标签来切换到主代码窗口。在主调试会话中执行调试之前，在全局变量窗口中添加全局变量 g\_shared 以供观察，初始值为 0，地址为 0x20000024，如 图 12 所示。

Variable	Type	Value	Address
(x) g_shared	volatile uint32_t	0	0x20000024

图 12. 主核调试会话中的初始共享变量

执行过代码行，来通过邮箱将共享变量 g\_shared 的地址发送到从核中 ( 如 图 13 所示 )，单击调试窗口中的红色块返回到从调试线程 ( 如 图 7 所示 )，并返回从代码的 main()。

```

mailbox_mutex_core0.c | mailbox_mutex_core1.c
128     /* Send address of shared variable to CM33 core1 by Mailbox*/
129     MAILBOX_SetValue(MAILBOX, kMAILBOX_CM33_Core1, (uint32_t)&g_shared);
    
```

图 13. 主核代码行发送共享变量的地址

可以看到，只要刷新窗口显示，例如执行 'setp over' 操作，全局变量窗口中的共享变量 \*g\_shared 就会与主核具有相同的值 (0) 和同步的地址 (0x20000024) ( 图 14 所示 )。这表示从核已经通过邮箱成功接收到共享数据的地址。

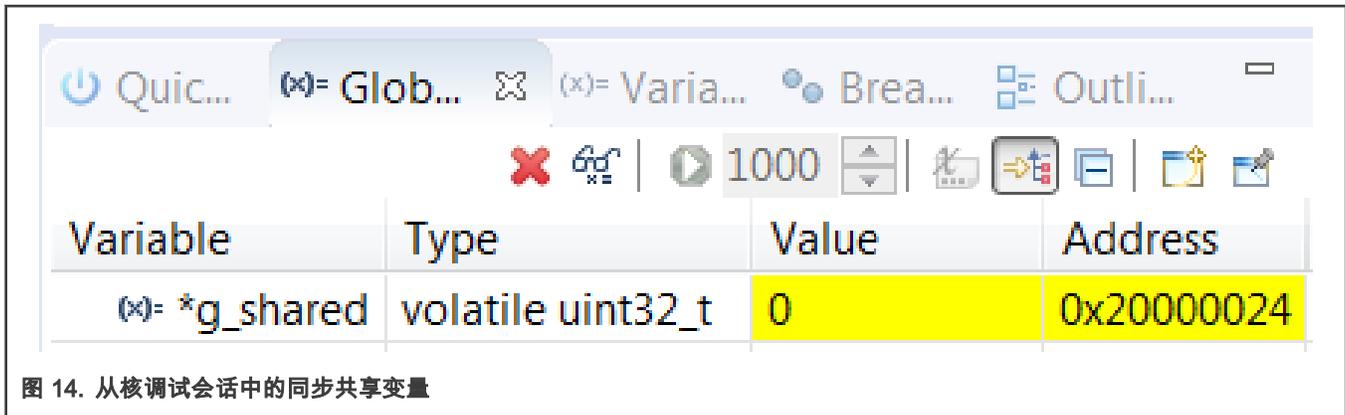


图 14. 从核调试会话中的同步共享变量

执行过 `(*g_shared)++;` 代码后，可以观察到共享变量 `*g_shared` 的值变为 1（如图 15 所示）。

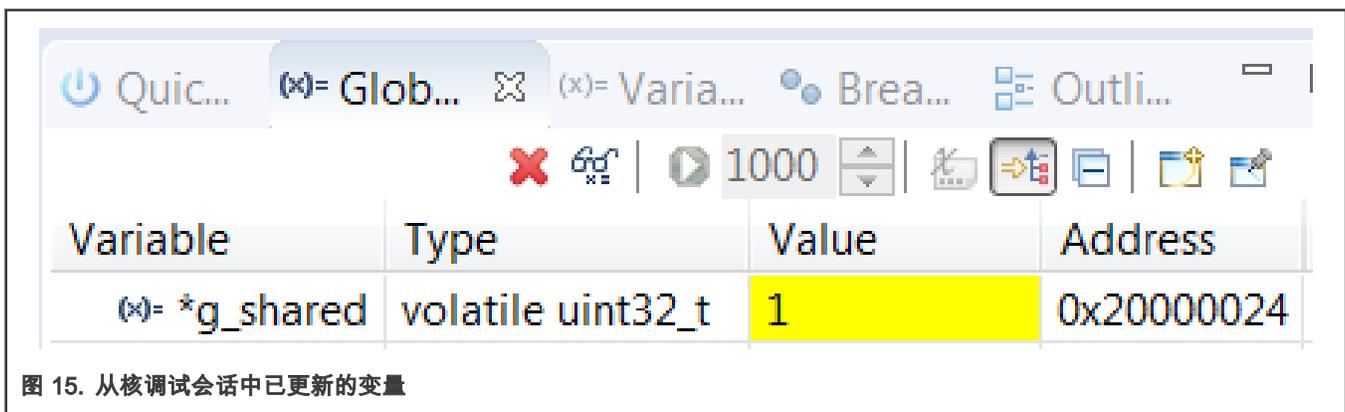


图 15. 从核调试会话中已更新的变量

执行过设置互斥量 (`MAILBOX_SetMutex(MAILBOX);`) 的代码后，意味着两个核都可以访问共享资源。目前，切换到主线程调试主 `while(1)` 循环的功能与从线程类似：获取邮箱互斥锁，更新共享变量 `g_shared`，设置邮箱互斥锁（参见图 16）。获取互斥锁的代码将在执行 **Step over** 步骤时被执行通过。同时，**全局变量**窗口中的共享变量 `g_shared` 的值将刷新为 1（参见图 17），因为它已经被从核更改。

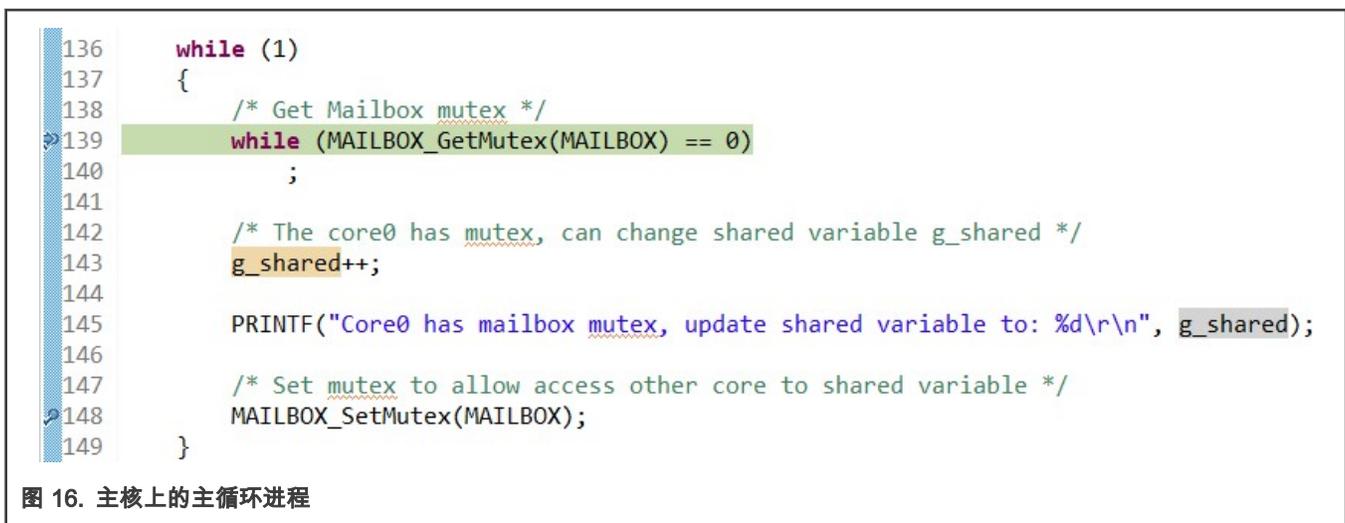
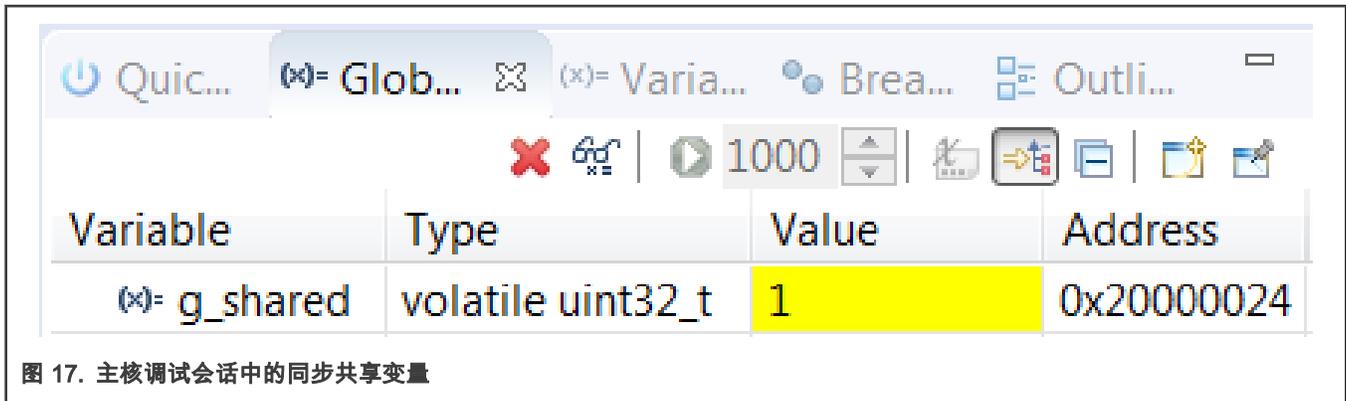


图 16. 主核上的主循环进程



执行过代码 `g_shared++` 之后，可以观察到全局变量窗口中的共享变量 `g_shared` 的值变为 2。在执行主代码中设置互斥量的代码（`MAILBOX_SetMutex(MAILBOX)`）之前，切换到从调试线程并做单步执行，它会保持停留在获取互斥量的代码上，因为互斥量访问还没有被主核释放。切换到主调试线程并单步执行设置互斥锁的代码来释放互斥锁控制。

然后，返回到从调试线程并执行单步执行。这时，可以观察到代码被执行跨过，因为可以获得互斥锁控制。类似地，如果在获取互斥锁的代码完成后，设置互斥锁的代码没有执行，那么主调试线程就会在获取互斥锁的代码上卡住。

接下来，如果按照上面的操作在主调试线程和从调试线程之间重复切换来进行调试控制，就会发现共享变量在主核和从核之间依次增加。只有成功获得互斥锁控制的核才能对共享变量进行更改。这意味着，如果在调试期间保持运行一个核的代码，共享变量只会被这个核增加。

## 5 结论

本应用笔记简单介绍了 LPC55xx / LPC55Sxx 中的双核机制和调试系统，然后简要介绍了与双核项目相关的一些配置，并基于 LPC5500 SDK 中的 `mailbox_mutex` 驱动程序示例，阐述了如何在 MCUXpresso IDE 中调试双核项目。

用户在 MCUXpresso 中调试双核项目非常方便，可以启动两个内核的调试会话，并且可以在一个工作区中完成调试过程。只需在“调试”窗口中单击“主线程”或“从线程”，即可切换双核项目中的任何一个进行调试。对各个核的调试操作，例如设置断点、变量视图，都与单核的是一样的。

### 注

在主核启动从核之前要先运行主调试线程。

**How To Reach Us**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

**Limited warranty and liability** — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

**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.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2019-2021.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 5 March, 2019

Document identifier: AN12358

