PRESUMIBILMENTE CON RISPOSTA

LPC43xx Cortex-M4 / M0 Multicore Applications

Domanda posta da **LPCware Support** in 22-apr-2016

Basic principles

The LPC43xx family of MCUs contain a Cortex-M4 ‘Master’ core and one (or more) Cortex-M0 ‘Slave’ core. After a power-on or Reset, the Master core boots and is then responsible for booting the Slave core(s); hence the names Master and Slave. In reality Master and Slave only applies to the booting process; after boot, your application may treat any of the cores as the Master or Slave.

The LPCXpresso IDE allows for the easy creation of 'linked’ projects that support the targeting of Multicore MCU including the LPC43xx family. The principle is that the Slave application(s) must be completely self contained and built into an application image (axf file). The Slave application(s) are then converted and included within the Master application, resulting in a single application image containing the Master and Slave applications.

Projects

To create a Multicore application, a project for each core must be created. Each project should be self-contained and complete, so must include a vector table and a main() routine. Additionally, the project for the Master core will also include code to boot the Slave(s) and should (optionally, but usually) include the Slave applications code.
Note that although possible, sharing of code between the Master and Slave is not recommended for the following reasons:

- Running code for the Master and Slave will lead to serious memory contention, potentially significantly slowing down all cores.
- As a Cortex-M0 has a smaller instruction set, compared to a Cortex-M4, any shared code must be compiled for the Cortex-M0. This code will be less optimal than code compiled for a Cortex-M4.

LPC43xx Project Wizards

The Project Wizard mechanism now contains a number of sets of LPC43xx related project wizards:

- **LPC43xx (basic)**
  - Wizards as per earlier releases of the Code Red IDE, including wizards for creating library projects
- **LPC43xx Multicore M0APP**
  - Wizards for creating M0APP slave applications
- **LPC43xx Multicore M0SUB**
  - Wizards for creating M0SUB slave applications. Note that M0SUB project are only applicable for LPC4370.
- **LPC43xx Multicore M4**
  - Wizards for creating M4 master applications (which will pull in any appropriate slave application images).

When creating a multicore application, you should first of all create the projects for any M0 slaves, then create the M4 master project. This will then allow you to associate the slave applications with the master, as you run through the project wizard to create the master application.

Slave Memory Location

The large number of different blocks of memory available on LPC43xx parts (including parts with and without internal flash) means that there a wide number of possibilities as to where the master and slave application code may be placed within the overall MCU memory map. But the key point is that the slave memory setting in the master (M4) project should match how the slave project itself was built.
If you have not set up the master and slave projects such that the address that the master locates the slave code into does not match the location that the slave project was built for, then you will see the corresponding build error message below:

```
M0APP execute address differs from address provided in source image
```

```
M0SUB execute address differs from address provided in source image
```

You will then need to modify your project settings for either the slave or master project, as detailed below, so that the slave code is placed into the same memory by both projects.

In order to keep things simple within the project wizards, the following memory defaults are used when creating multicore projects:

Slave Projects

By default, the memory setup for a slave project will match the default memory setup for the selected part. This will normally need to be modified, either by selecting an appropriate memory configuration file as you run through the wizard, or by manually editing the memory configuration in the MCU setting of Project Properties after the project has been created.

Note - the supplied **LPC43x7-M0_BankB.xml** can be used when creating M0APP projects so as to cause the M0APP code into bank B of the internal flash of LPC43x7 parts. This can be found within the product install at **<install_dir>\<ide_name>\Wizards\MemConfigs\NXP.**

For more information on memory configuration editing please see the LPCXpresso User Guide.

Master Projects

By default, the M0APP slave image will be placed in the 2nd RAM block of the master (M4) project's memory map. The multicore M4 project wizards also allow you to reconfigure the project to automatically use the 2nd flash block in dual bank flash MCU. Thus for example with an LPC43x7 part, the M4 application can be placed in bank A (at 0x1a000000) and the M0APP application can be placed in bank B (at 0x1b000000).

Also, if you associate an M0SUB slave project with an M4 master project, then the M0SUB project should normally be placed in the `RamM0Sub16` RAM block of the M4's memory map. Thus you will need to ensure that the M0SUB project has been configured such that this RAM block is first in the memory block list displayed by the memory configuration editor in the MCU settings of Project Properties. Alternatively modify the slave settings of the MCU Linker/Target tab in the M4 master project's Project Properties (see below).

What else do the multicore wizards do ?

First of all, appropriate startup code will be created in the project for the master / slave the project was generated for.In addition the M4 master project includes files containing functions to release M0 slaves from reset:

- **cr_start_m0.c**, plus the associate **cr_start_m0.h**

Finally, the main() function generated by the M4 master wizard contains code to conditionally call functions to release M0 slaves from reset. This is done conditionally using defines that are created, based on which slaves

are linked to the master when running through the project wizard. For more information, see the FAQ " Compiler defines for LPC43xx Multicore projects".

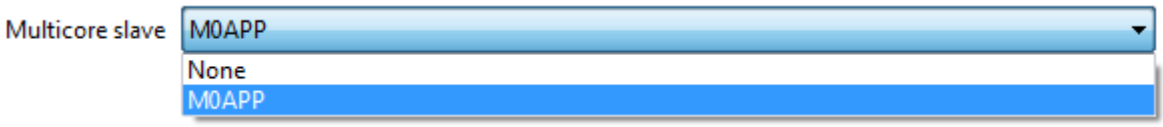
```
     // Start M0APP slave processor
#if defined (LPC43_MULTICORE_M0APP)
     cr_start_m0(SLAVE_M0APP,&__core_m0app_START__);
#endif
     // Start M0SUB slave processor
#if defined (LPC43_MULTICORE_M0SUB)
     cr_start_m0(SLAVE_M0SUB,&__core_m0sub_START__);
#endif
```

Multicore settings in Project Properties

The following settings are provided in project properties in relation to multicore projects. If you create your multicore master and slave projects using the multicore project wizards, then these settings will be configured automatically for you, with appropriate default values. However you may still need to modify these if the defaults are not appropriate for your application (for example the memory blocks used for the slave images), or if you want to configure a project generated using the "basic" LPC43xx project wizards for multicore use.

Slave applications

The Slave application must be completely self contained and linked to run at the appropriate location with the Slaves memory map. To support inclusion with the Master application image, a new option has been added to the MCU Linker/Target tab. This option allows selection of the type of Slave:

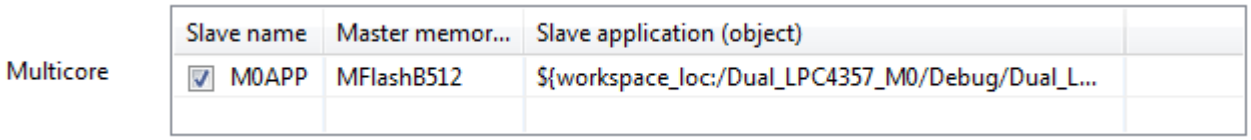


- None: This application is not going to be used in a Multicore system
- M0APP: This application is going to run in the M0 Slave "Application" core of the LPC43xx MCU
- M0SUB: This application is going to run in the "Subsystem" core of the LPC4370 MCU

These options will only be present when the project is configured to use a Multicore MCU.When this option is used, a new object file (project.axf.o) is created in the same directory as the application image. This object file contains just the binary image data of the application and is used to link with the Master application (see below).

Master application

The Master application is responsible for loading the Slave code into the appropriate location in the Memory map and for booting the Slave.To support creation of a single application image containing code for Master and Slaves, a new option has been added to the MCU Linker/Target tab.



- Slave name (checkbox)
  - The Slave that this applies to. The checkbox is used to enable/disable inclusion of this image
- Master link section
  - The memory region in the Master memory map in which to place the Slave image. To avoid possible issues with alignment, the image is always placed at the **start** of this memory region. Multiple Slaves should not be placed in the same memory region (although this is not currently validated).
- Slave application (object)
  - The object file containing the Slave binary image data

Debugging

Debugging via Red Probe+ or LPC-Link

When debugging an LPC43xx based system using Red Probe+ or LPC-Link, it is only possible to debug either the Cortex-M4 master or (one of) the Cortex-M0 slave(s).

However whichever CPU is to be debugged, the actual image must be download to the system via the Cortex-M4 master project (so that the master can then release the slaves from reset). The launch configuration for any M0 slave applications will have the "Attach Only" option set to "True" by default, which will prevent any attempt to download code direct to the M0 target.

First of all launch a debug session for the **M4** project to download the combined M4/M0 image into flash. When you make this connection you need to make sure that you actually connect to the correct core within the MCU.
The first time you start an M4 debug session, you will receive a pop up dialog offering you a choice on two cores on the LPC43xx . Ensure that you select the device which has the TAP Id = 0x4ba00477 - which is the M4. This selection is then stored in the *.jtag file that will be placed in the Debug/Release directory. Note that you can cause the pop up dialog to reappear again by deleting the *.jtag file.

Having connected to the M4, close down the debug session.

Now launch a debug session for the **M0** project. The first time you start an M0 debug session, you will receive the JTAG selection pop up dialog again . Ensure that you select the device which has the TAP Id = 0x0ba01477 - which is the M0.

Debugging via LPC-Link2

When using LPC-Link2 (with "Redlink" firmware and connecting over JTAG, not SWD), it is possible to debug all cores in an LPC43xx part from within the same IDE instance in parallel.

To do this:

1. From the M4 Master project, start a debug connection and execute the master code such that the M0 slave core(s) have been enabled.
2. Now, without closing down the debug connection to the M4 master, start a debug connection from the M0 slave project. If you then pause execution of the slave's debug session, you will be able to step, view registers etc.

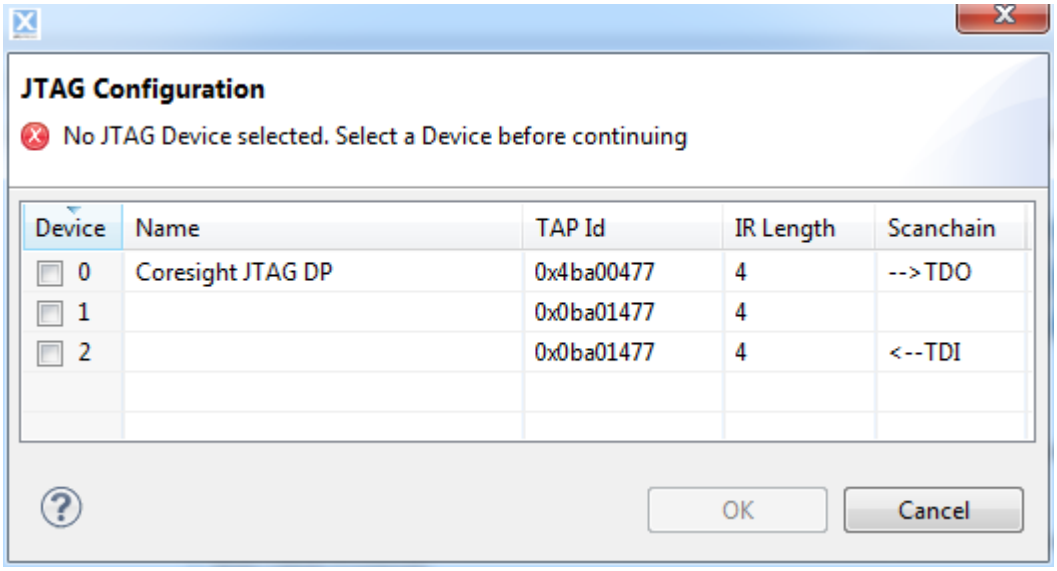
There is a demonstration of this in action in this video of one of the NXP sessions at DesignWest 2013:

http://www.youtube.com/watch?v=THqkAcZvMXI

[skip through to time 26:28 of the video for the LPC43 part]

Note about debugging LPC4370

Both Cortex-M0 cores in the LPC4370 have the same JTAG TAP id (0x0ba01477). Thus when you start a debug connection to the MCU you will see a JTAG selection window.



Device 1 is the M0SUB core. The device closest to TDI (device 2 here) is then the M0APP core. Make sure that you select the correct core for each debug connection !

Nessun altro ha questa domanda

Visibilità: LPCXpresso IDE FAQs • **6485 visualizzazioni**

Ultima modifica effettuata il 17-mag-2016 5.48

Tag: projects multi-core lpcxpresso lpc43xx

Categorie: Activation / Installation / Licensing Debugging IDE usage and settings

0 risposte

Contenuti correlati

- Compiler defines for LPC43xx Multicore projects
- Where is the Linker/Target tab in MCUXpresso?
- MCU Linker Issue for Multicore project
- Compiling multicore examples in LPCXpresso v7
- board.h Error when I use new MCUExpresso 10.0.2

Contenuto consigliato

- How to enter into ISP mode through 'reinvoke ISP' in LPC1759
- LPC55S69 documentation
- LPC8N04 i2c Demos from SDK 2.5.0 not working
- Can't build LPC8N04 / OM40002 example code in "release" mode
- ISP over I2C - How?

Collegamenti alla discussione

- Re: Need a multicore demo for Lpc4367
- Re: MCU Linker Issue for Multicore project
- Where is the Linker/Target tab in MCUXpresso?
- Where is the Linker/Target tab in MCUXpresso?
- Re: Which settings do I edit when changing from LPC4357 to LPC4333