

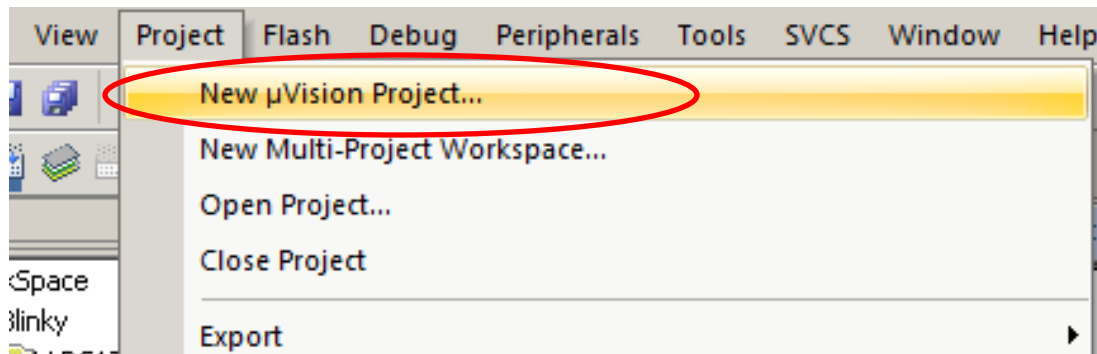
Introduction Keil μ Vision IDE

Electrical and Computer Engineering
University of Waterloo

Irene Huang

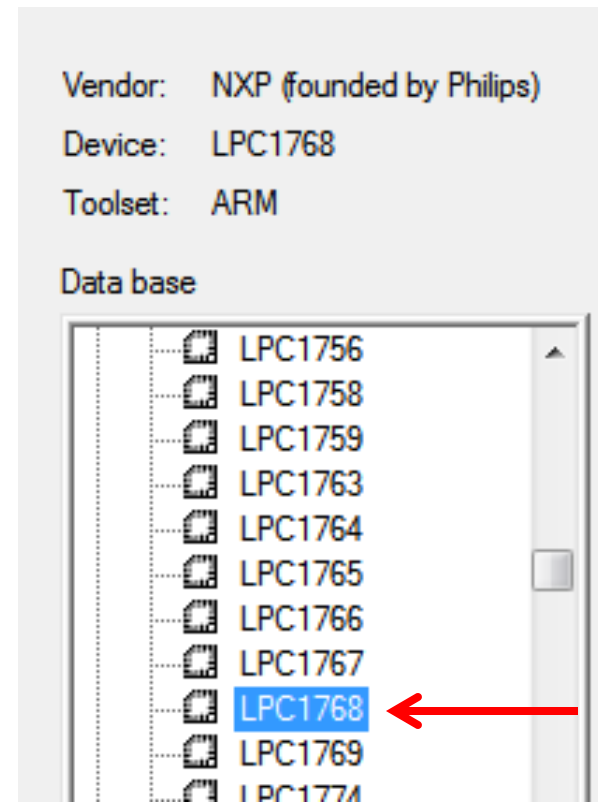
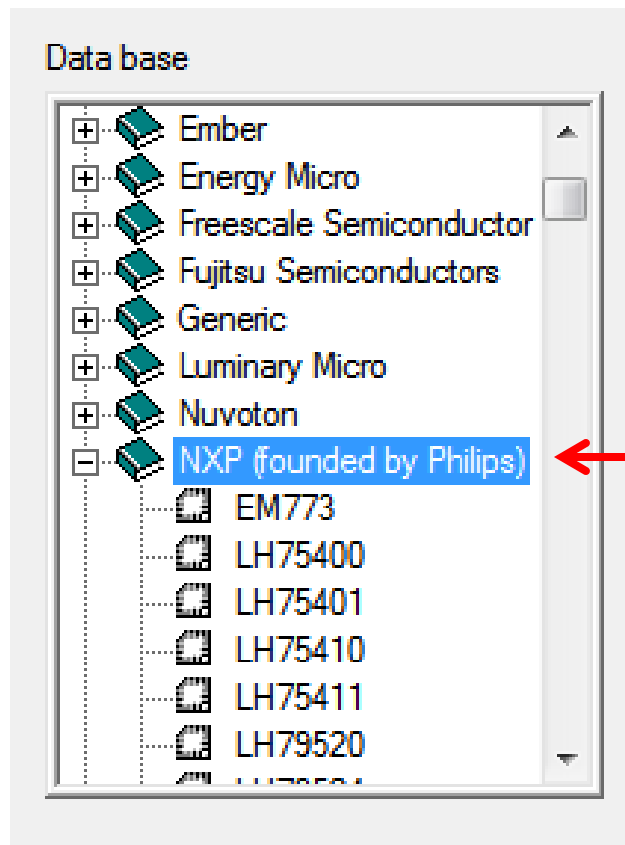
Creating a New Project

- Create a new folder and name it “Hello”
- Copy the following files to the “Hello” folder
 - Keil\Startup\system_LPC17xx.c
 - Keil\GLCD*.*
- Create a new μ Vision by click
 - Project \rightarrow New μ Vision Project



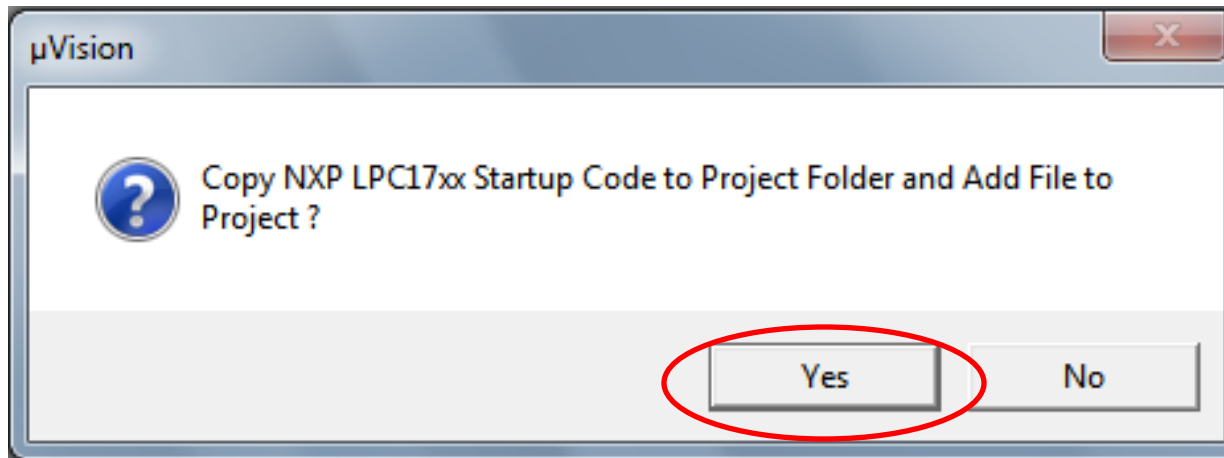
Choosing the Device

- Choose NXP(Founded by Philips) → LPC1768



Copying the startup Code

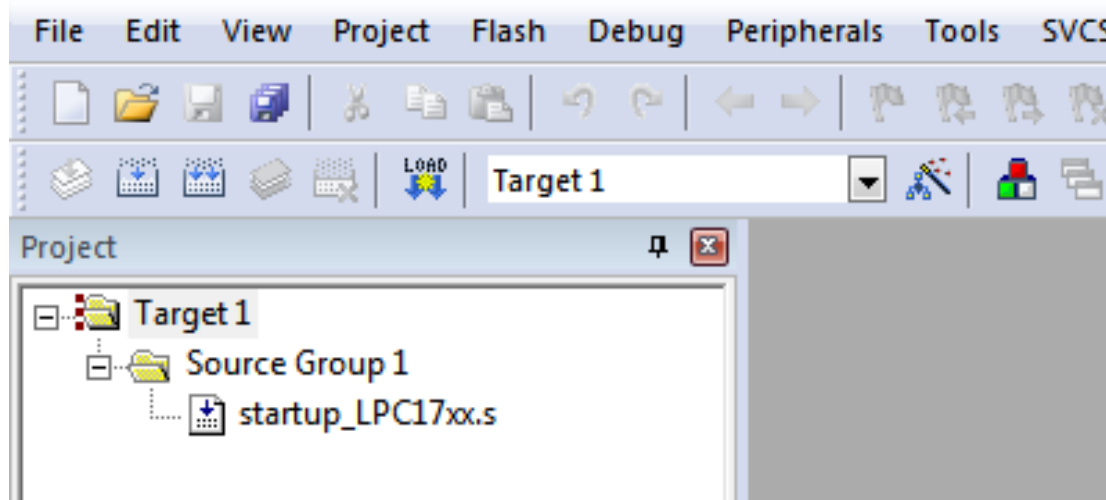
- Answer “**Yes**” to copy the startup code



- If you answer “No”, you need to copy the file manually from the example code folder.
 - Keil\Startup\startup_LPC17xx.s

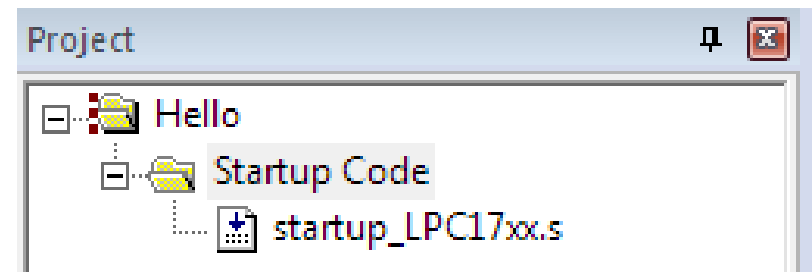
Default Project Components

- You will see the default project setup as follows:



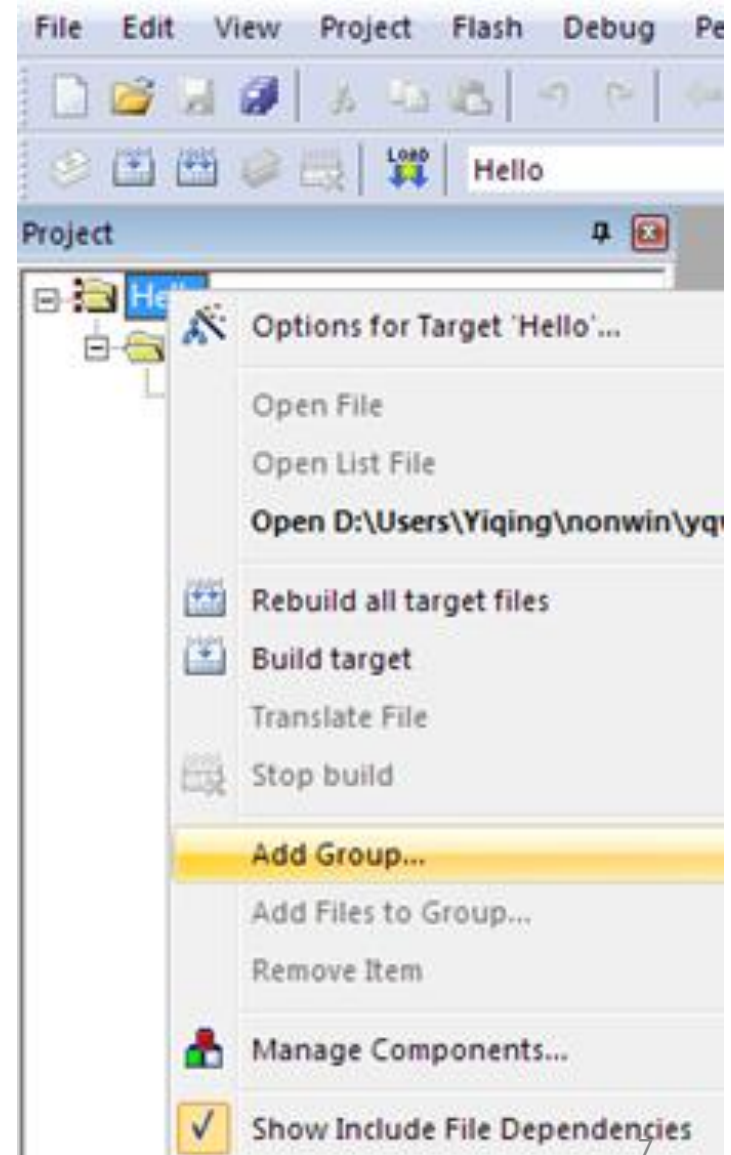
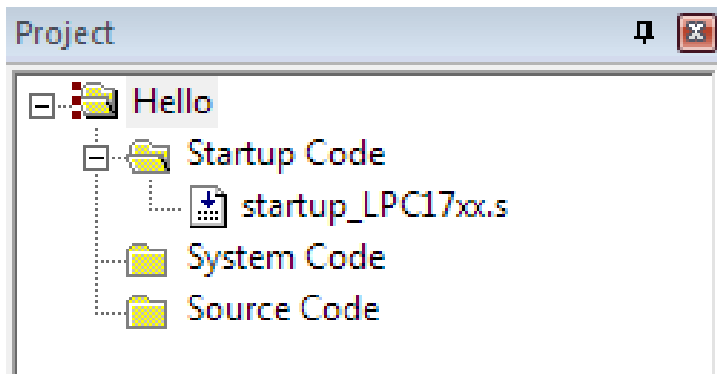
Renaming Project Components

- Rename the default target name
 - Click the target name to highlight it
 - Click the highlighted name to input a new target name, say “Hello”
- Rename the default source group name
 - Click the default source group to highlight it
 - Click the highlighted name to input a new name, say “Startup Code”



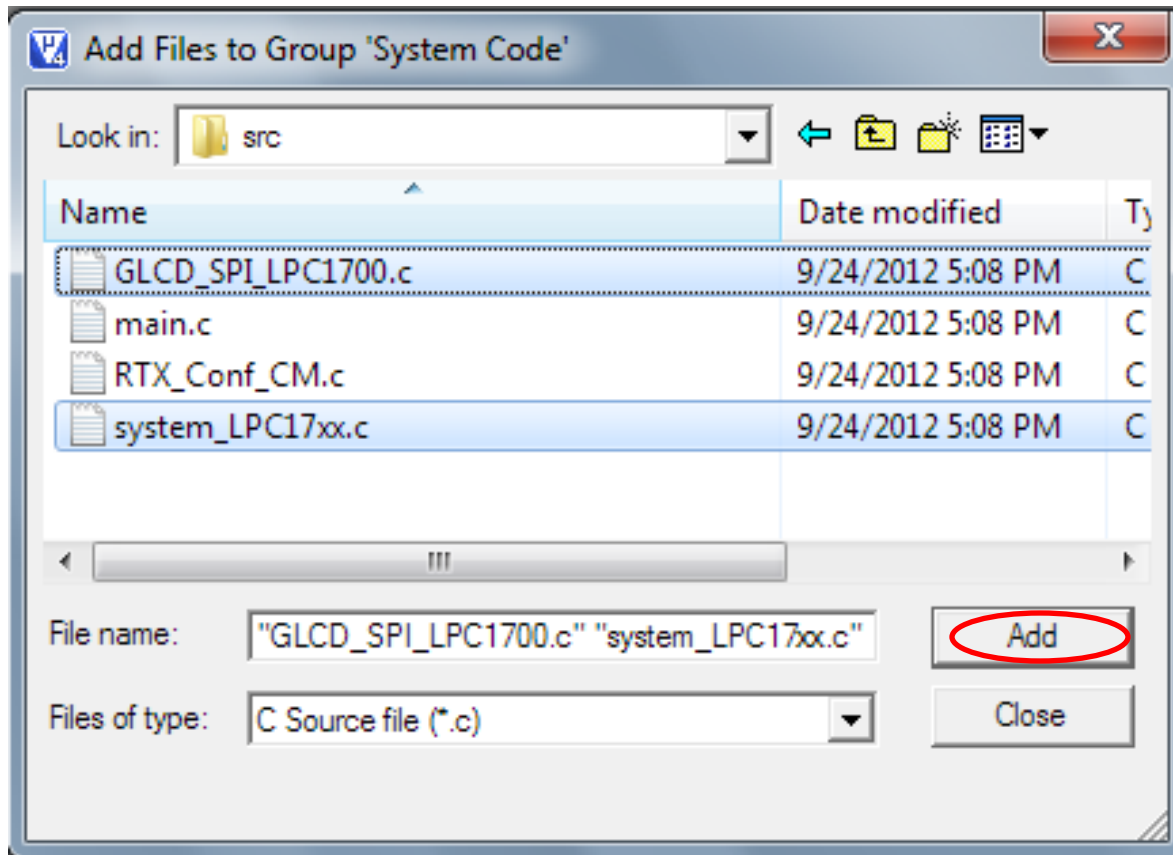
Adding New Source Groups

- Right click the target name and add the following source groups
 - System Code
 - Source Code
- You will see the following



Adding Files to Source Groups

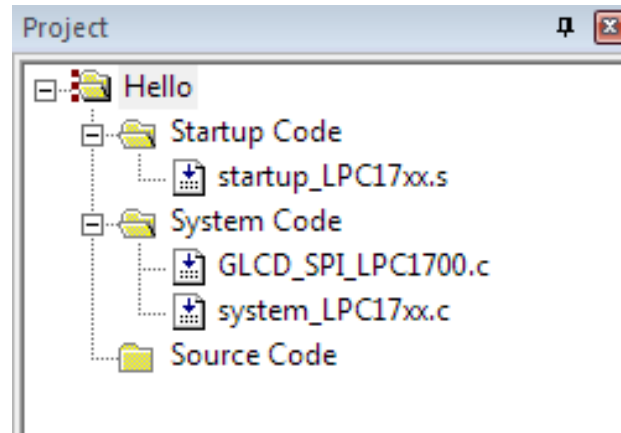
- Double click the “System Code” group and add the following files to it.
 - **system_LPC17xx.c** and **GLCD_SPI_LPC1700.c**



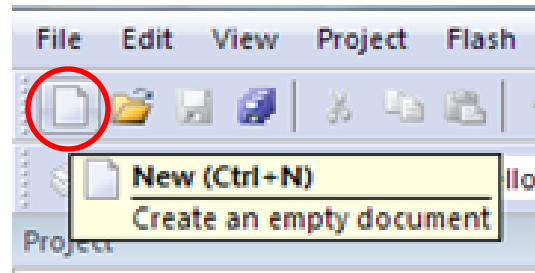
Double click file names to add
or
highlight file names and click "Add" button.

Creating Your Own Source Code

- Your project would now look like this.

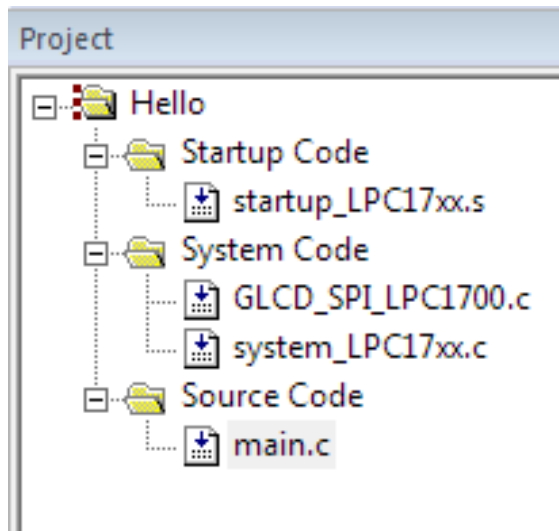


- Click New button to Create a `main.c` file.



Adding main.c to Project

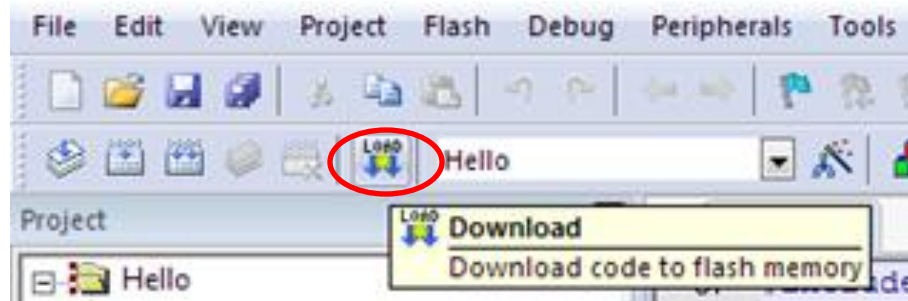
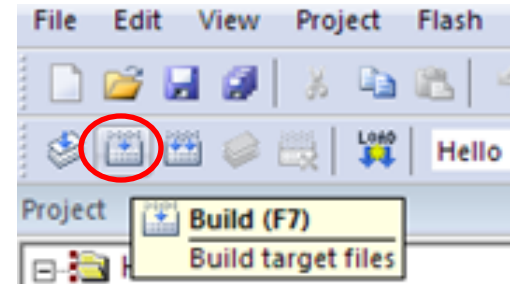
- Add the `main.c` to the “Source Code” group.



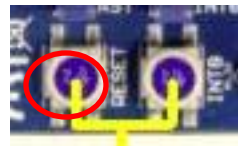
```
main.c
01 #include <LPC17xx.h>
02 #include "GLCD.h"
03
04 int main()
05 {
06     SystemInit();
07     GLCD_Init();
08     GLCD_Clear(Yellow);
09     GLCD_DisplayString(1,1,1,"Hello World!");
10     return 0;
11 }
12
```

Build, Download and Run

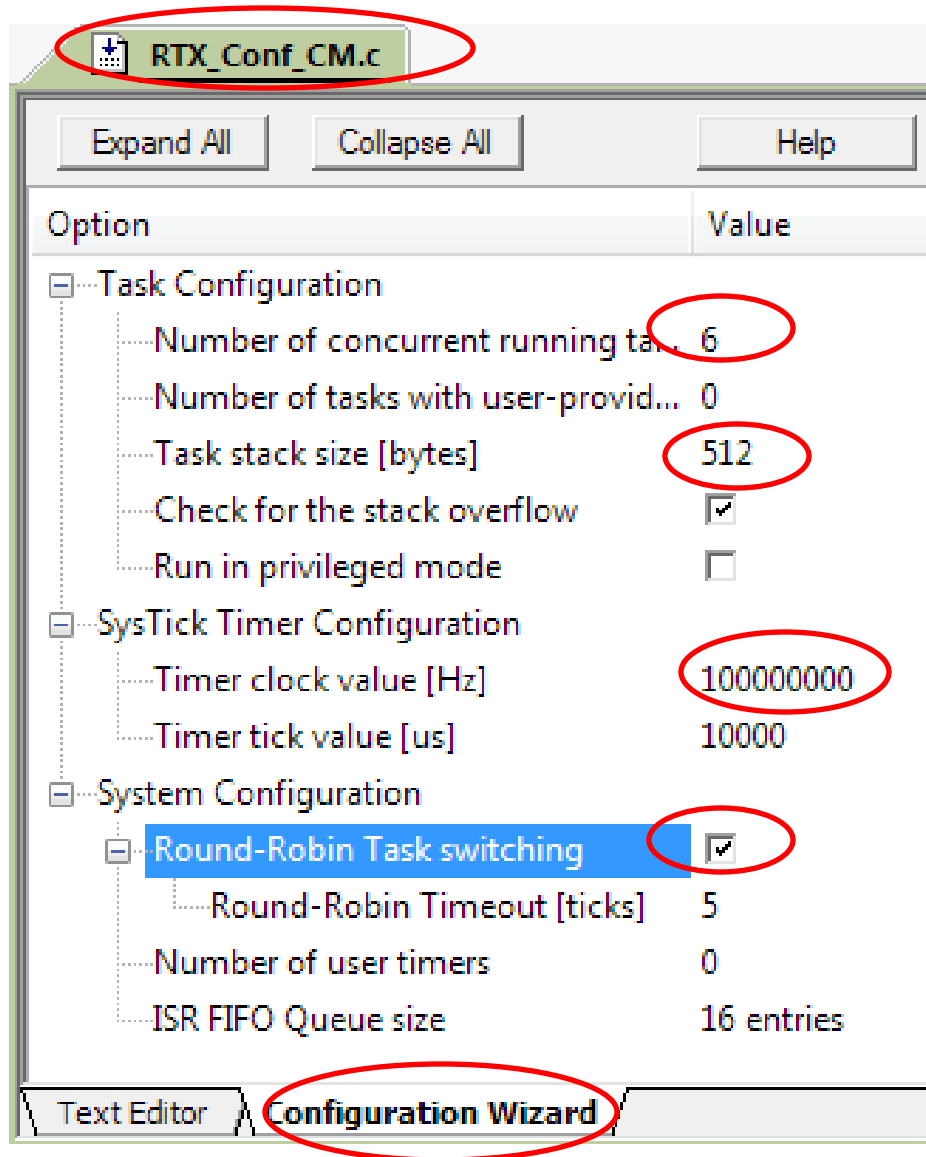
- Click the Build button
- Click the Download button



- Press the “Reset” button on the board to run



Changing it to an RTX Application



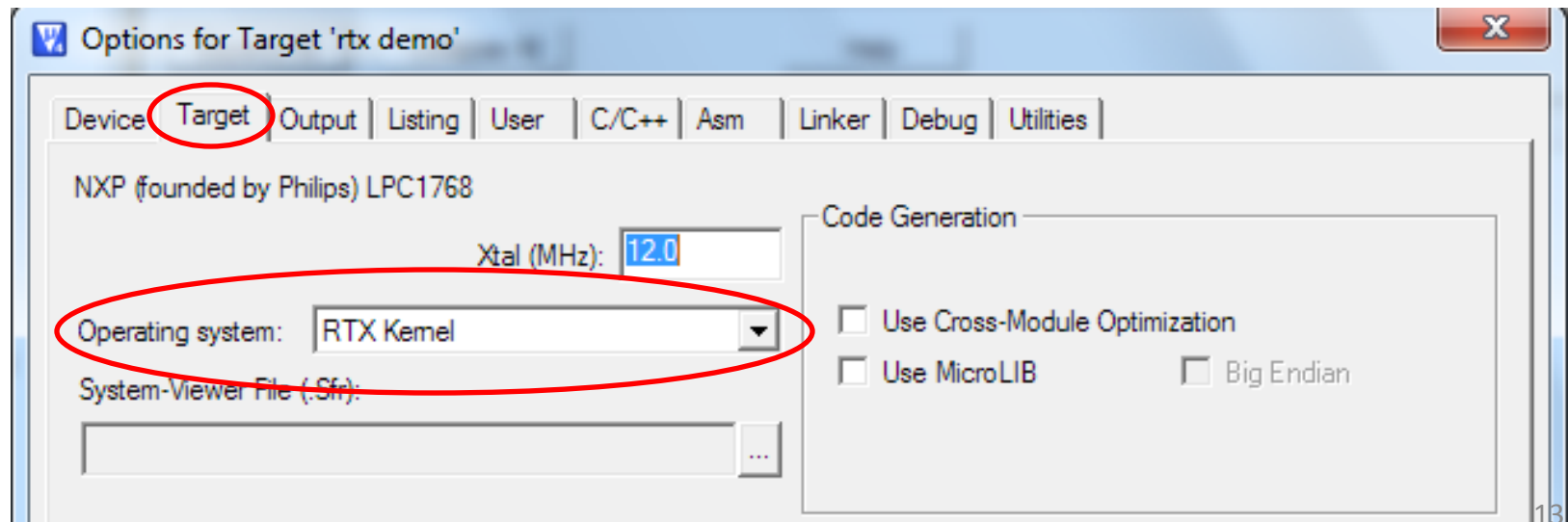
- Add the following files to the project
 - Keil\Startup\RTX_Conf_CM.c
- Configure the RTX

Configure RTX in Target Option

- Click the “Target Option” button



- Select “RTX Kernel” as the operating system



Writing an RTX Application

- You need to include the **RTL.h** file

```
#include <RTL.h>
```

- Call `os_sys_init(taskname)` to initialize the OS and start the first task

```
os_sys_init(init)
```

- Each task needs to start with `__task` keyword.

```
__task void init(void);
```

- The RTX_demo project uses LED device functions. You will need to add the following to the project.

```
- Keil\LED\*.*
```

Debugger

The screenshot shows the uVision4 IDE interface with several windows and annotations:

- Registers Window:** Displays the state of CPU registers. Annotations include:
 - Reset: Reset the CPU** (pointing to the Reset button)
 - Run(F5)** (pointing to the Run button)
 - Stop: Stop Code Execution** (pointing to the Stop button)
 - Step(F11): Step one line** (pointing to the Step Into button)
 - Step Over(F10)** (pointing to the Step Over button)
 - Step Out(Ctrl+F11)** (pointing to the Step Out button)
- Disassembly Window:** Shows the assembly code being executed. Annotations include:
 - Start/Stop Debug Session (Ctrl+F5)** (pointing to the Start/Stop button)
 - Insert/Remove Break Points(F9)** (pointing to the Breakpoint button)
 - Enable/Disable Break Points(Ctrl+F9)** (pointing to the Breakpoint button)
 - Disable All Break Points** (pointing to the Breakpoint button)
 - Kill All Break Points (Ctrl+Shift+F9)** (pointing to the Breakpoint button)
 - Run to Cursor line (Ctrl+F10)** (pointing to the Run to Cursor button)
 - Show Next Statement: Show statement in PC** (pointing to the Next Statement button)
- Serial Window:** Displays serial communication data. Annotations include:
 - Serial Windows to show or hide** (pointing to the Serial Window icon)
- Local Variables Window:** Displays local variables. Annotations include:
 - Local Variables Window** (pointing to the Locals window)
- Command Window:** Displays the command line. Annotations include:
 - Simulator is used for debugging** (pointing to the Simulation button)

The bottom status bar shows the simulation time: **t1: 0.00004850 sec**, **L1: C:1**, and **CAP NUM SCRL OVR**.