1 Creating a project

The TI compiler that comes with CCS has a code size limit. For this reason, I suggest using the GNU compiler that comes with CCS.

Select the "CCS Project" template as shown in figure 1.

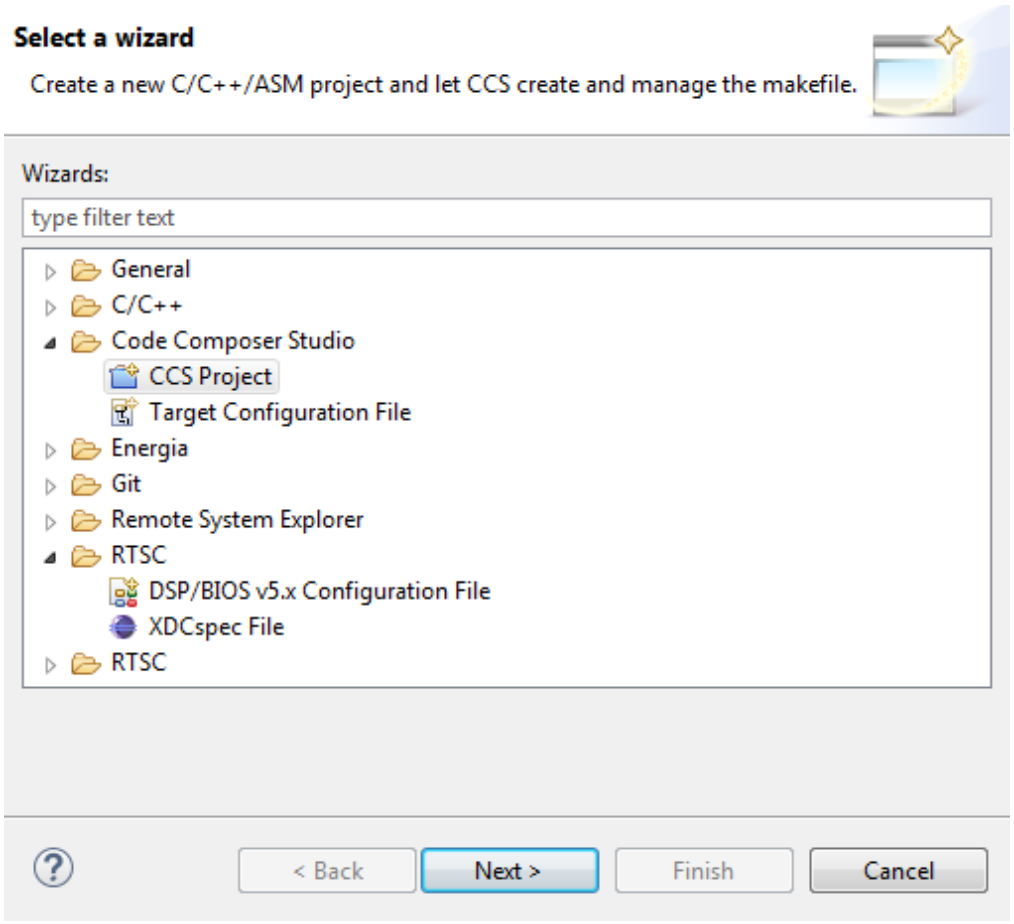

Figure 1

Select the target, board, and connection as in figure 2. Select the GNU compiler and choose a template project. The TI-RTOS templates have minimal code and are a good starting point. I recommend using the Typical example as it allows dynamic allocation of tasks inside of your code rather than only static allocation inside of the config file.

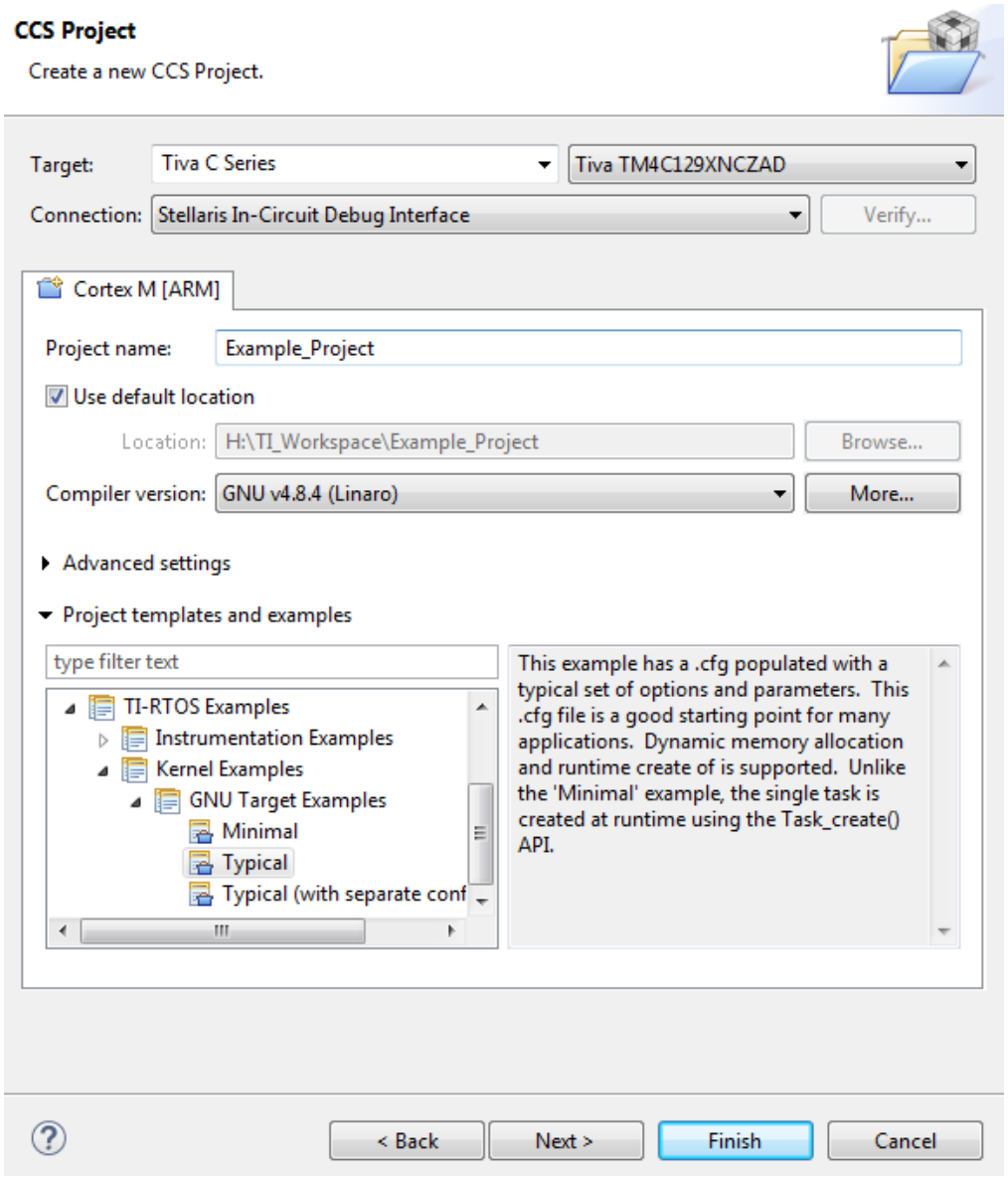

Figure 2

2 Memory issues with printf family

If you use any of the printf() functions (sprintf, etc), you may run into problems with your code halting when it reaches the printf() call. This is due to a lack of stack size as the printf() functions are quite large. You can either use the usprintf() function or increase the stack size in the config file as shown in figure 3. You will probably have to modify the stack and heap size regardless if you end up using lots of recursive functions or dynamically allocated variables.

```
/*
 * The BIOS module will create the default heap for the system.
 * Specify the size of this default heap.
 */
BIOS.heapSize = 0x1000;

/*
 * Build a custom SYS/BIOS library from sources.
 */
BIOS.libType = BIOS.LibType_Custom;

/* System stack size (used by ISRs and Swis) */
Program.stack = 0x400;
```

Figure 3

3 Adding the graphics library to a TI-RTOS project

The graphics library for the screen has been pre-compiled and provided by TI for use, but it requires some additional setup to be used.

The **gr** library has to be linked to provide the functions used. To do this, open the project properties by right clicking on the project and selecting **Properties**. Under **Build**, **GNULinker**, **Libraries**, add **"gr"** and **"driver"** to the libraries and the following paths to the library search path:

```
"${COM_TI_RTSC_TIRTOSTIVAC_INSTALL_DIR}/products/TivaWare_C_Series-2.1.0.12573c/grlib/gcc"
"${COM_TI_RTSC_TIRTOSTIVAC_INSTALL_DIR}/products/TivaWare_C_Series-2.1.0.12573c/driverlib/gcc"
```

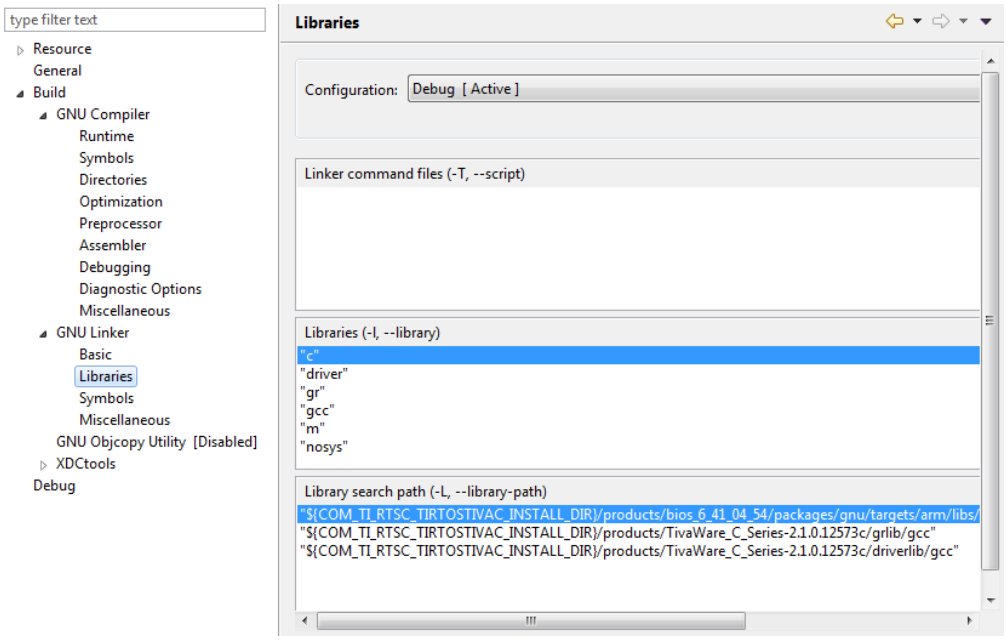

Figure 4

You also need to tell the compiler where to find all the graphics and driver library header files. Add the following paths

to the project's include paths:

```
"${COM_TI_RTSC_TIRTOSTIVAC_INSTALL_DIR}/products/TivaWare_C_Series-2.1.0.12573c"
"${PROJECT_LOC}"
```

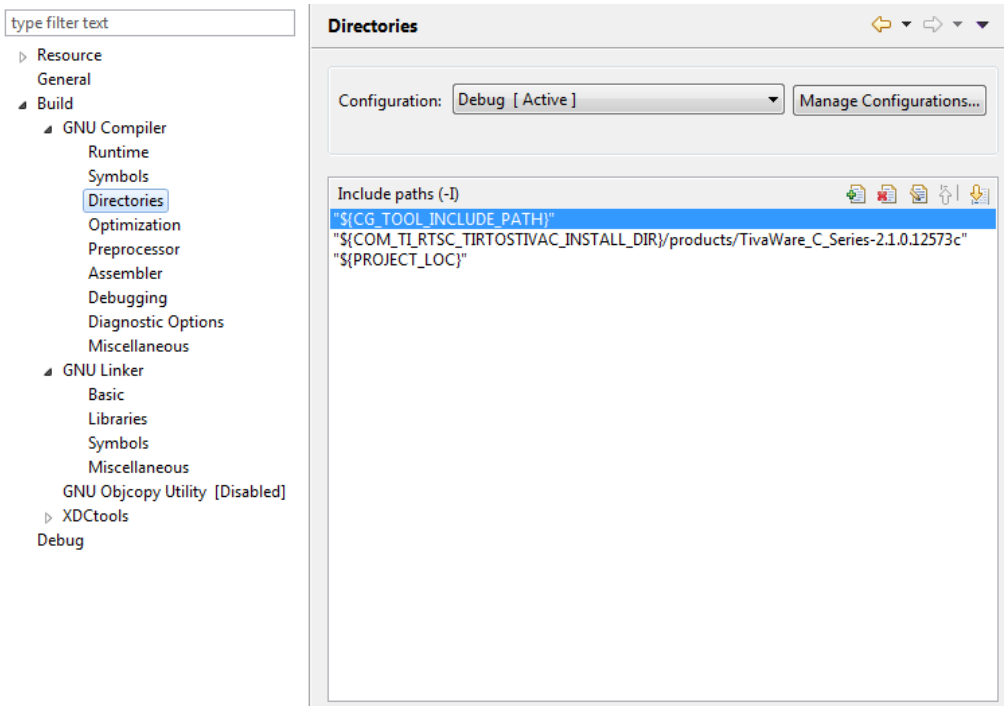

Figure 5

Open the app.cfg file and add the following lines to the end of the file, do not copy them from this PDF document.

```
var TIRTOS = xdc.useModule('ti.tirtos.TIRTOS');
TIRTOS.useEMAC = true;
TIRTOS.useGPIO = true;
Hwi.create(33, '&TouchScreenIntHandler');
```

To initialise the screen, copy the **"drivers"** folder with the **frame.c**, **frame.h**, **kentec320x240x16_ssd2119.c**, **kentec320x240x16_ssd2119.h**, **pinout.h**, **pinout.c**, **touch.c** and **touch.h** files from the **C:\ti\TivaWare_C_Series-2.0.1.11577\examples\boards\ek-tm4c129x\drivers** folder.

You will have to modify the **PinoutSet** function to remove references to the **ROM** functions. Simply remove the **ROM_** prefix from all the functions.

4 Adding networking capabilities to a TI-RTOS project

Add the following path to your project's include paths:

```
"${COM_TI_RTSC_TIRTOSTIVAC_INSTALL_DIR}/products/ndk_2_24_02_31/packages/ti/ndk/inc"
"${COM_TI_RTSC_TIRTOSTIVAC_INSTALL_DIR}/products/ndk_2_24_02_31/packages/ti/ndk/inc/bsd"
```

You do not need to add anything to the library search paths.

Open the project's configuration file and go to the **TI-RTOS - System Overview** window.

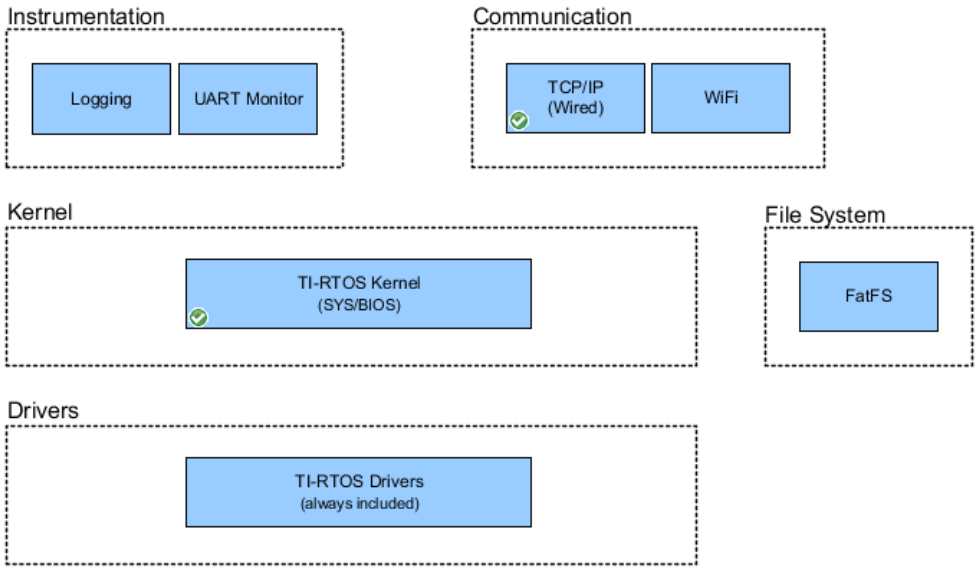

Figure 6

Click on **TCP/IP (Wired)** and check the **Add NDK/Global to my configuration** checkbox.

Go to the **Hooks** tab and set the **Network open hook** to the function that creates your network task.

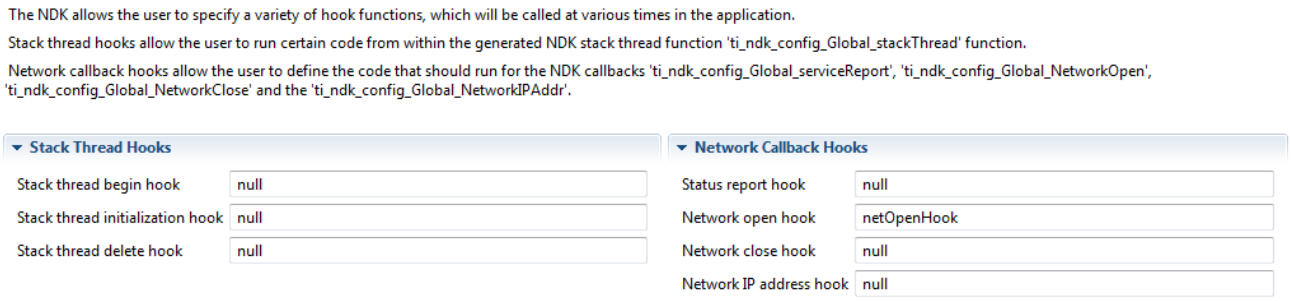

Figure 7

Go to the **System Overview** tab and add the **UDP**, and **IP** modules to the configuration.

Go to the **TI-RTOS Kernel (SYS/BIOS)** module, go to the **Runtime** tab. Increase **Heap size** to a much larger value, such as 20480.

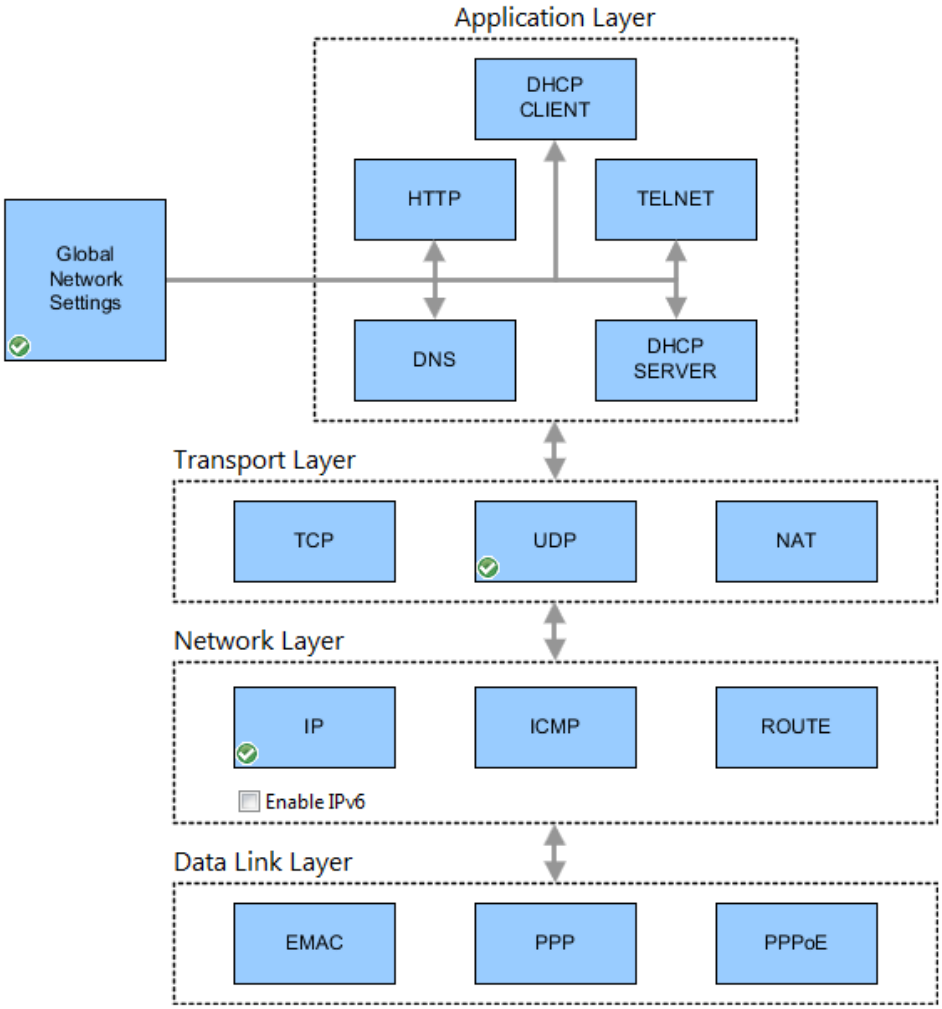

Figure 8

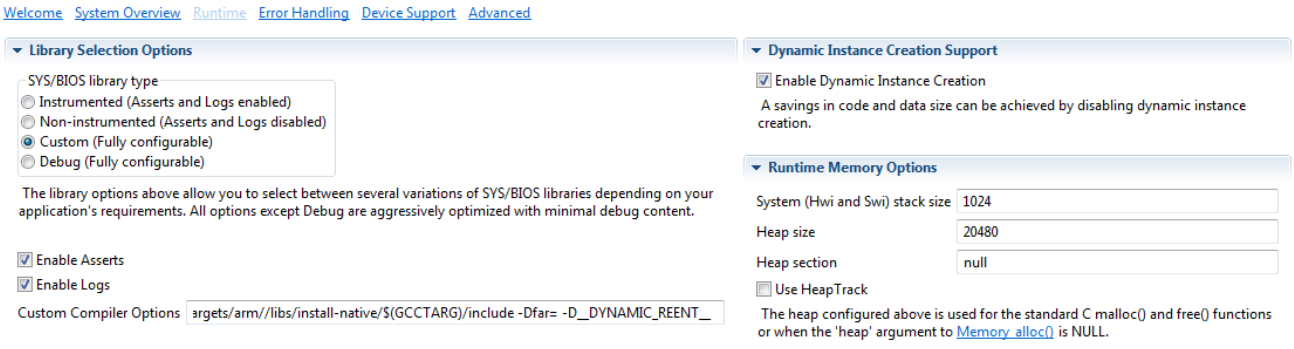

Figure 9