

# AN101: GCT301S Quick Start

---

I.	Adding the GCT301S Device Family Pack .....	2
II.	New Project Using Device Family Pack .....	3
III.	Programming the Flash with J-LINK.....	8
IV.	Debugging.....	11

## I. Adding the GCT301S Device Family Pack

Two direct ways to install the pack:

- (1) Double-click on the file SZGC.GCT301S.1.0.2.pack to start the installation; OR
- (2) Right click on SZGC.GCT301S.1.0.2.pack, select **Open With** and then select Pack Unzip.

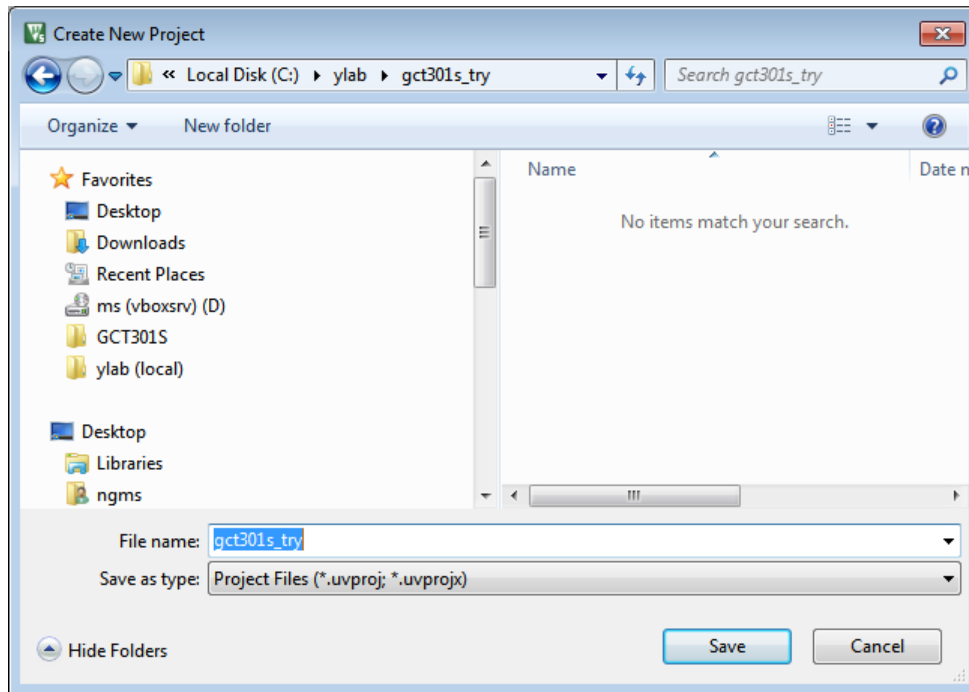
When the following dialog box appears, click Next to complete the installation.

Note: KEIL uVision5 is required.

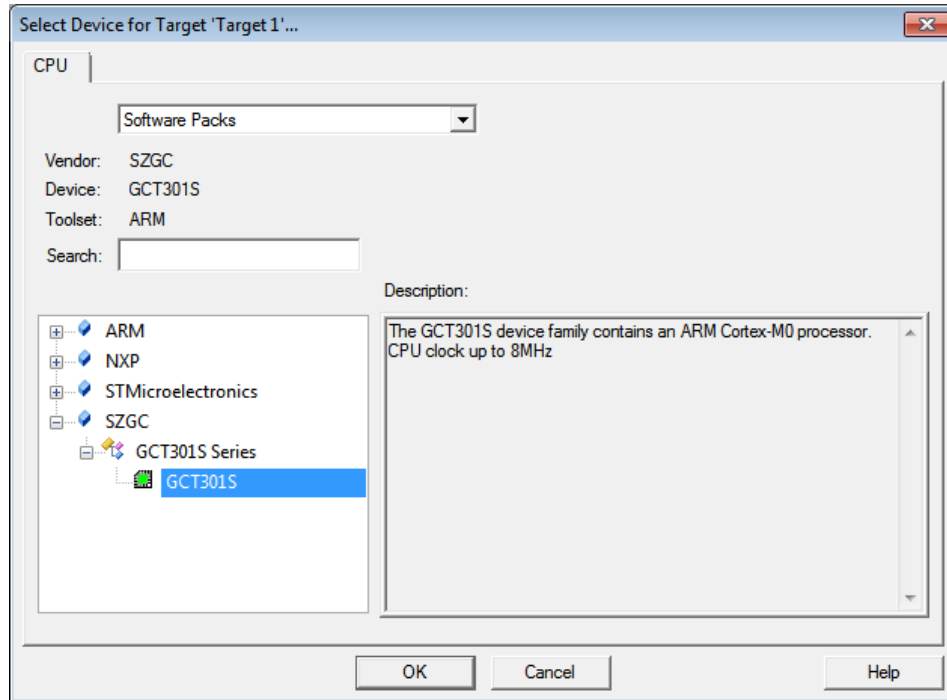


## II. New Project Using Device Family Pack

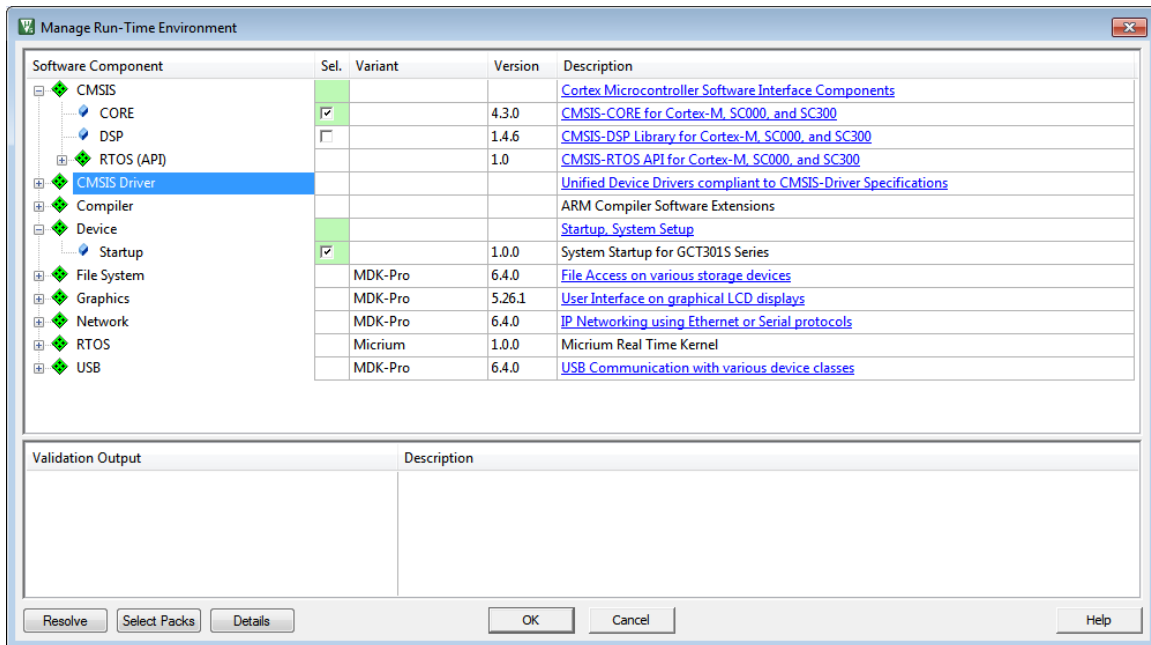
- (1) Create a working folder gct301s\_try.
- (2) From the menu, select Project → New uVision Project.
- (3) Move to the gct301s\_try folder, name the project gct301s\_try in the Filename textbox, and save the new project.



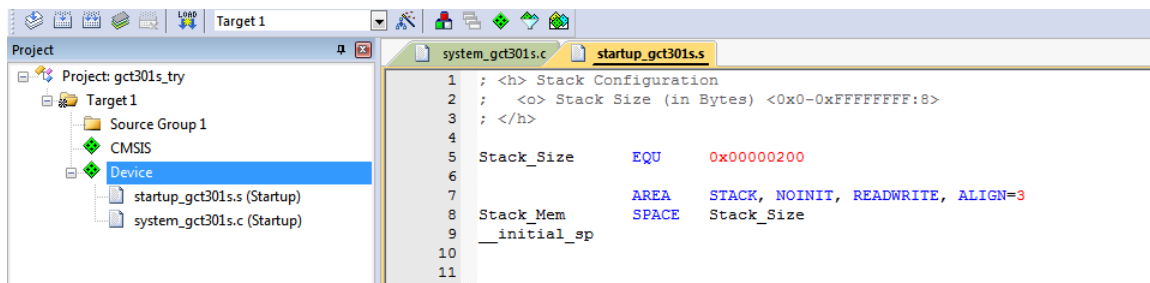
- (4) In the Select Device dialog box, choose SZGC→GCT301S Series → GCT301S. Click OK to complete the selection.



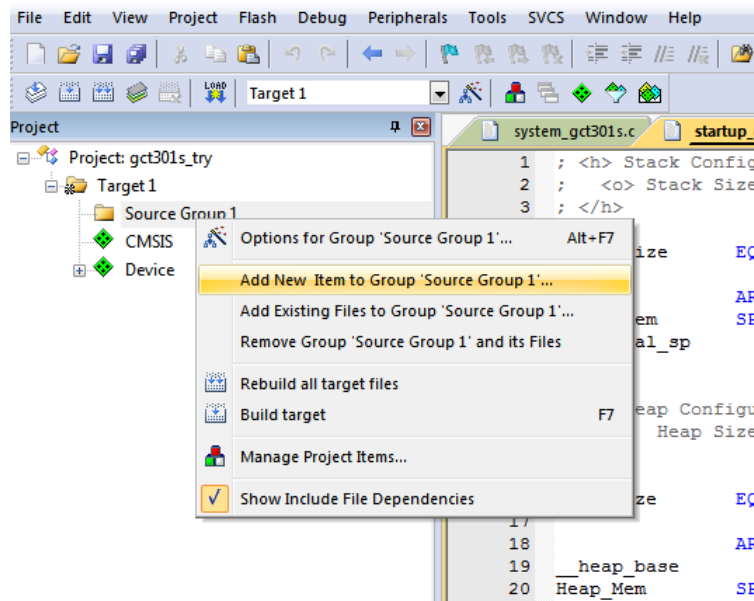
- (5) In the Manage Run-Time Environment dialog box, do the following and then click OK to complete the selections.
- Expand the CMSIS option, and check the CORE option.
  - Expand the Device option, and check the Startup option.



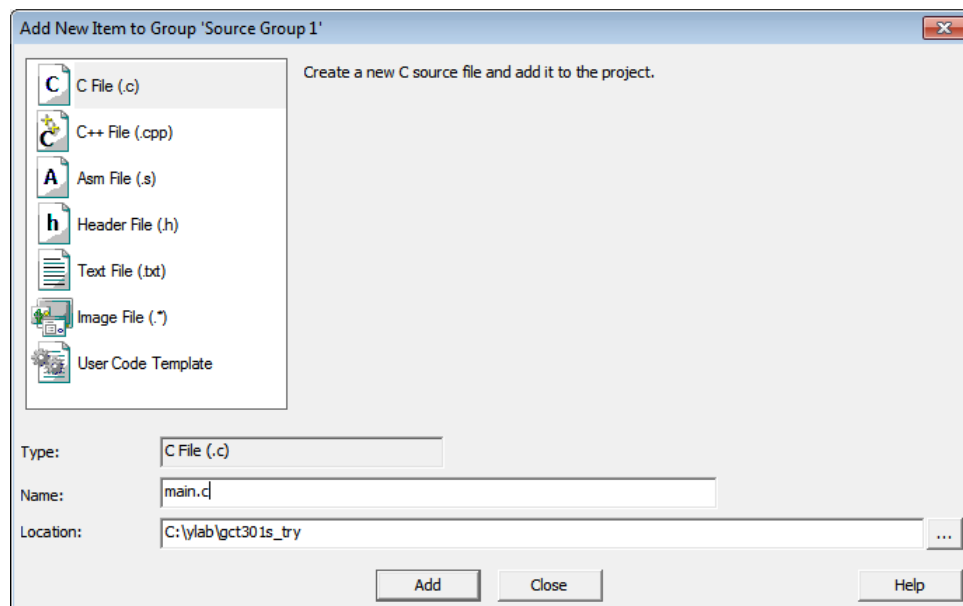
- (6) The project browser should be similar to the following screen capture.



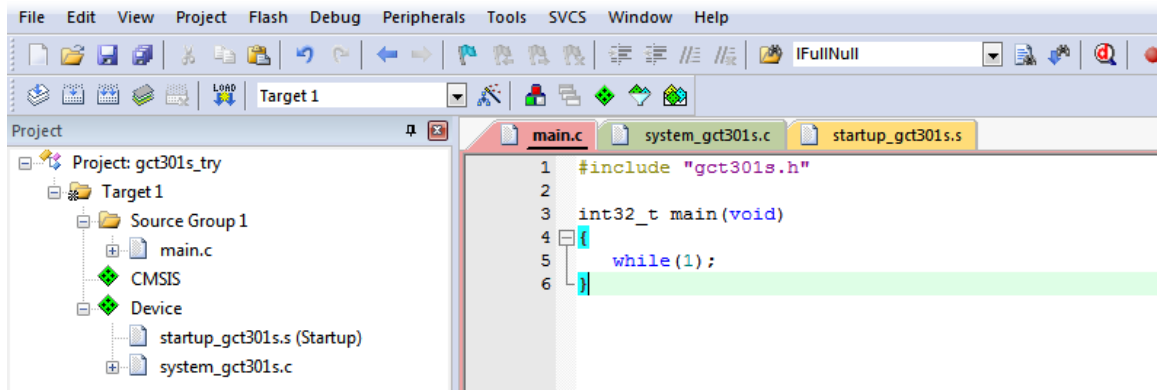
- (7) Let us add the main.c source file into the project. Right click on Source Group 1, the select Add New Item to Group 'Source Group 1...'



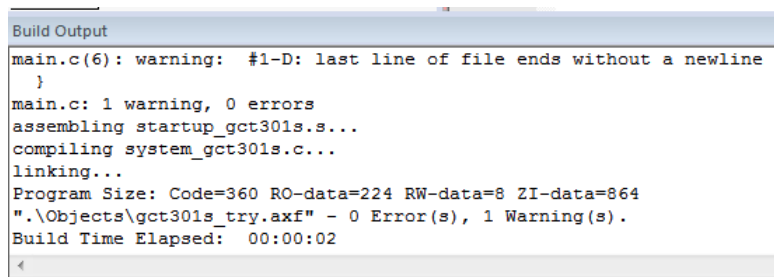
- (8) In the Add New Item to Group 'Source Group 1', select the C File (.c) file type, enter the filename as main.c in the Name textbox. Click Add to add the file into the project.



- (9) Let us code a simple infinite loop in main.c as shown in the following screen capture. Note the C include directive, where the gct301s.h header file is included. Also note that the use of the new standard C data type int32\_t is not mandatory but is recommended for source code portability.



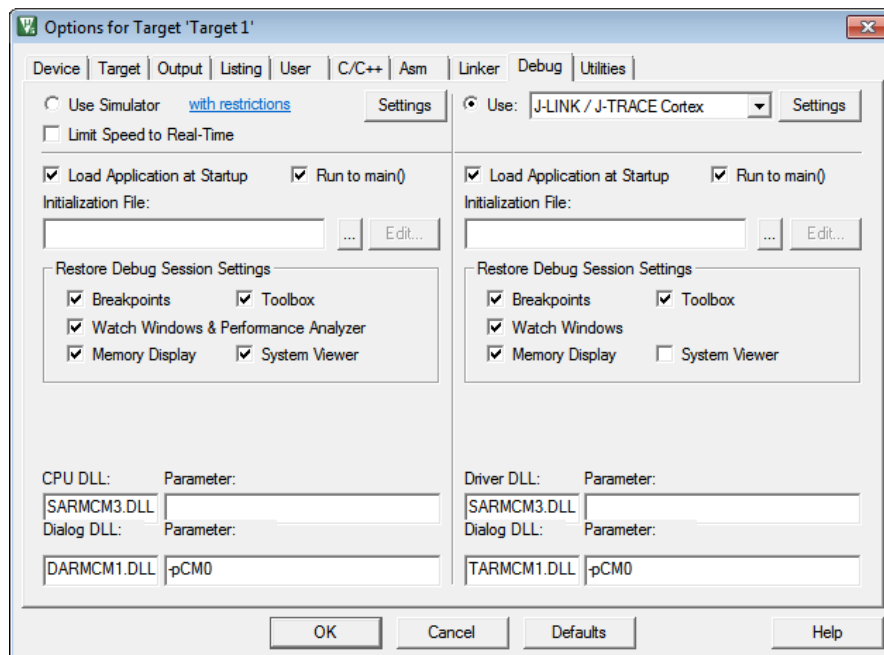
- (10) From the main menu, select Project → Build Target to build the project. This can also be accomplished with pressing the F7 key. There should be no error.



### III. Programming the Flash with J-LINK

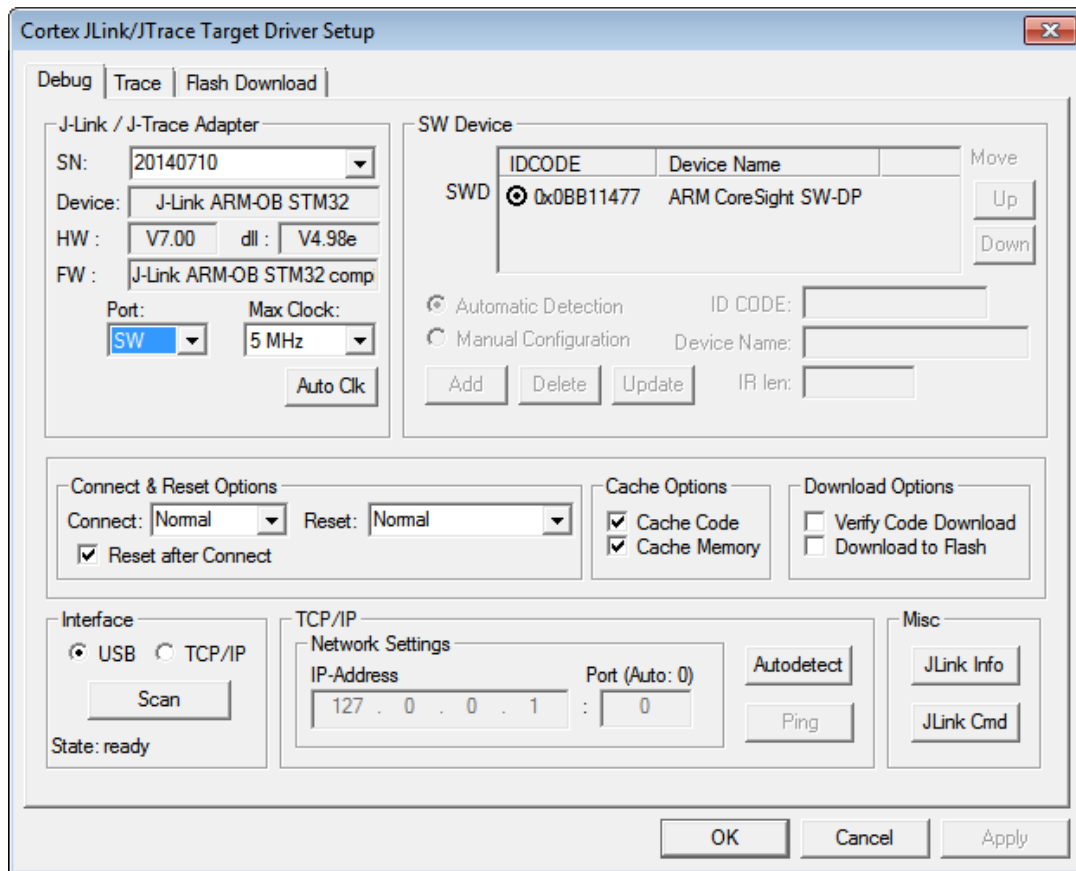
J-LINK is recommended because it has a command line debugger that is useful when doing low speed debugging. Briefly, the debugger allows the user to set the Serial Wire Debug clock speed to communicate with the target device. When a target device is in idle or sleep modes, the system clock is the low speed RC oscillator or 32,768Hz crystal oscillator. If the debugger attempts to communicate with the target device with high speed clock, the target device on-chip debug module may not be able to communicate properly, leading to failed debug operations.

- (1) From the main menu, select Project→Options for Target 'Target 1', and choose the Debug tab. Change the debugger to J-LINK/J-TRACE Cortex.
- (2) Click the corresponding Settings button. This will most likely bring up a dialog box that says "GCT301S is unknown to this version of the J-LINK software. Click No to ignore the message.

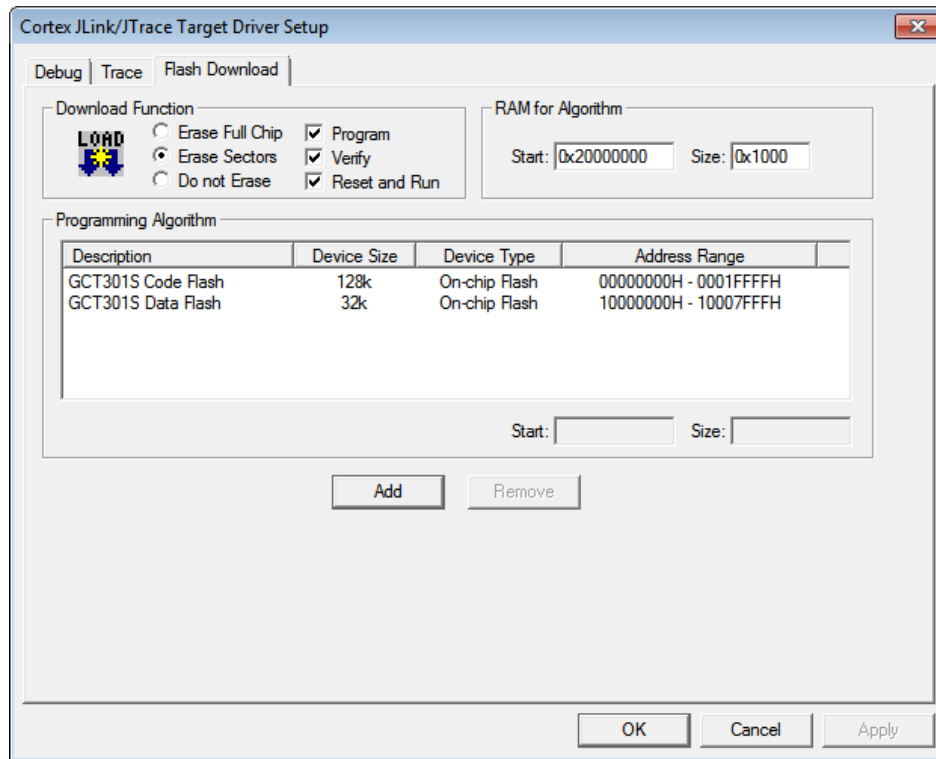




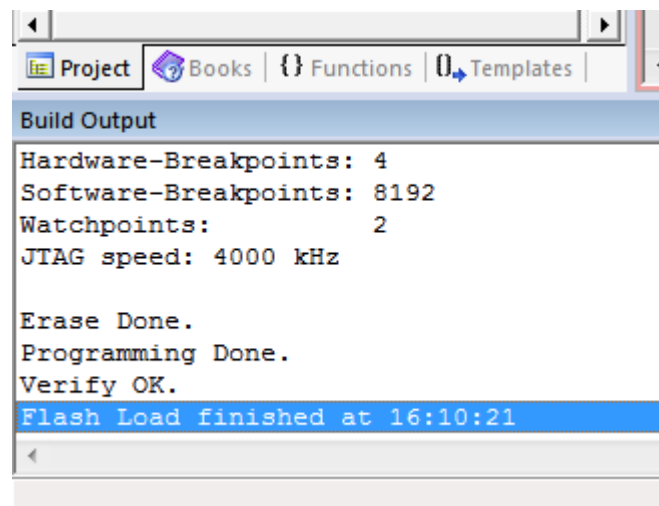
- (3) In the Cortex JLink/JTrace Target Driver Setup dialog box, change the Port to SW. If the GCT301S device is connected to the debugger, the device will be detected as shown in the screen capture below.



- (4) Choose the Flash Download tab. Make sure that the GCT301S Code Flash and GCT301S Data Flash programming algorithms are available as shown below.
- (5) To reset and run the program after the flash is programmed, check the Reset and Run option.
- (6) Click OK to exit the dialog box.
- (7) Click OK again to return the source code editing mode.



- (8) From the main menu, select Debug→Download to flash the program. There Build Output box should show the messages that Erase, Program and Verify are OK.

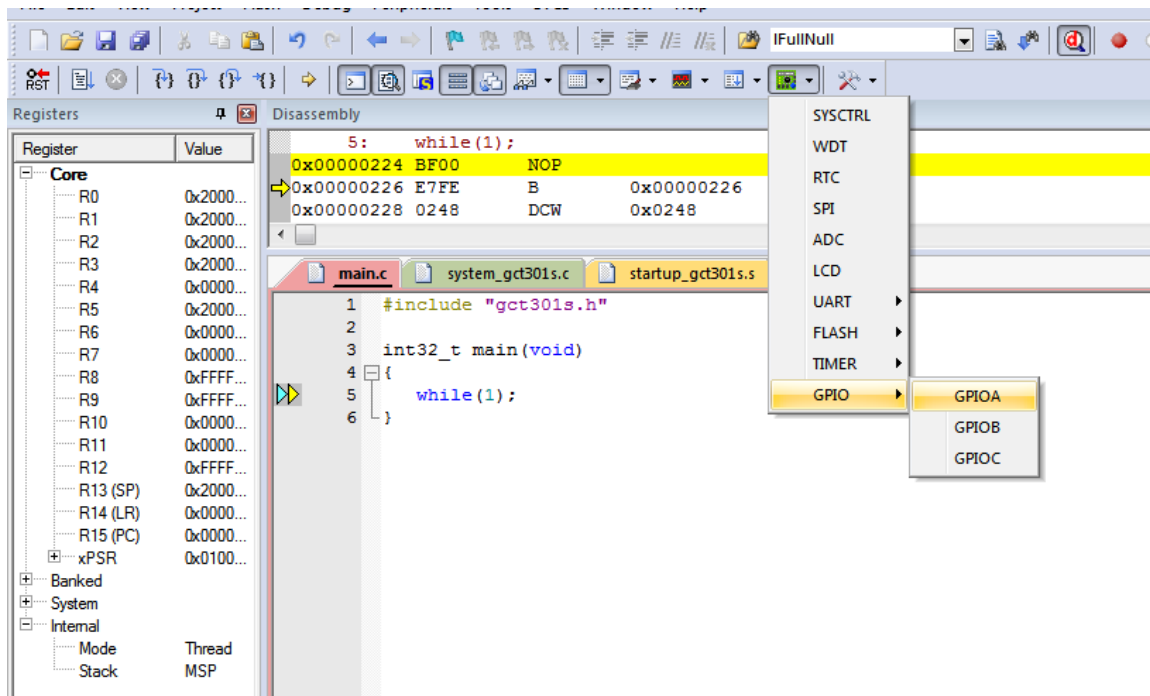


## IV. Debugging

The KEIL uVision has the common debug features.

For GCT301S, there is also a System Viewer to simplify the access to peripheral registers during debugging.

- (1) To enter debug mode, from the main menu, select Debug → Start/Stop Debug Session.
- (2) The debugger may run the program and stop at the first statement in the main.c.
- (3) The System Viewer is available from the main menu, Peripheral → System Viewer as shown in the following screen capture. The next two images show the System Control and GPIOA registers in the System Viewer.
- (4) Note that changing the values in the system viewer will trigger a write to the corresponding register. However, if the value is the same as the one shown, the write is not trigger. This behavior is due to KEIL and not specifically due to GCT301S.



SYSCTRL	
Property	Value
SYS_RST_STA	0
SYS_CLK_STA	0
SWD_ERA_KEY	0
SWD_ERA_CMD	0
RST_CR	0
CLK_DIVSEL	0x00000001
CLK_SRCSEL	0
NMI_CTL	0
HSRC_CTL	0x00000040
LSRC_CTL	0x00000020
PWR_CTL	0
DPLL_CTL	0
CHIP_CFG	0x00005844
SYS_CPL	0
HXTAL_CTL	0
RTC_CTL	0
WDT_CTL	0

GPIOA	
Property	Value
DATAIN	0x000000FF
DATAOUT	0
ANDISSET	0x0000FCFF
ANDISCLR	0x0000FCFF
OUTENSET	0
OUTENCLR	0
ALTFUNCSET	0
ALTFUNCCLR	0
INTENSET	0
INTENCLR	0
INTTYPESET	0
INTTYPECLR	0
INTPOLSET	0
INTPOLCLR	0
INTSTATUS	0
INTCLR	0
BSCR	0x00000000
PUDISSET	0
PUDISCLR	0
<b>DATAOUT</b> [Bits 31..0] RW (@ 0x40080004) Output latch	