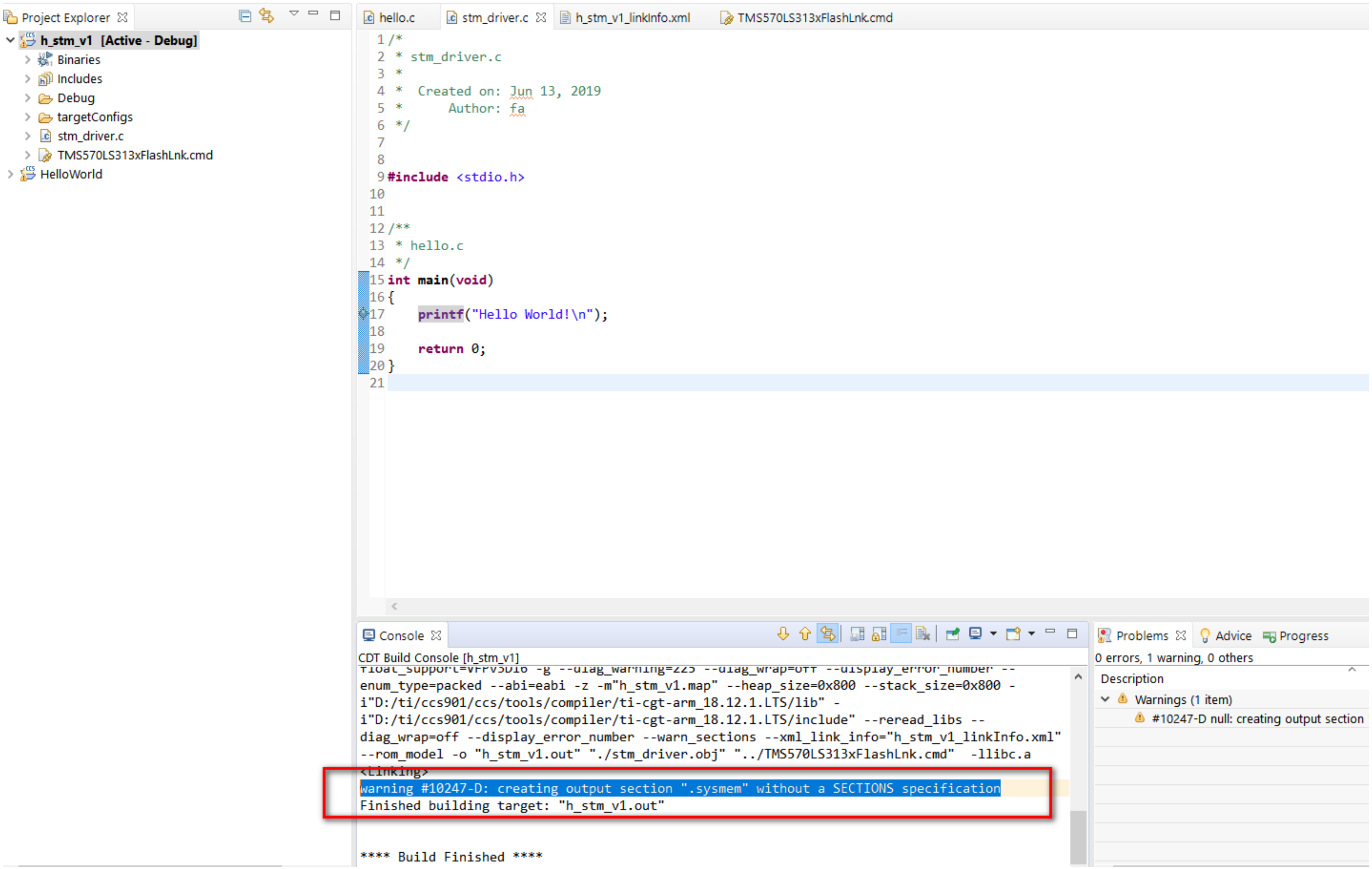
```
warning #10247-D: creating output section ".sysmem" without a SECTIONS specification
```



```c
/*
 * stm_driver.c
 *
 *  Created on: Jun 13, 2019
 *      Author: fa
 */


#include <stdio.h>


/**
 * hello.c
 */
int main(void)
{
    printf("Hello World!\n");

    return 0;
}

```

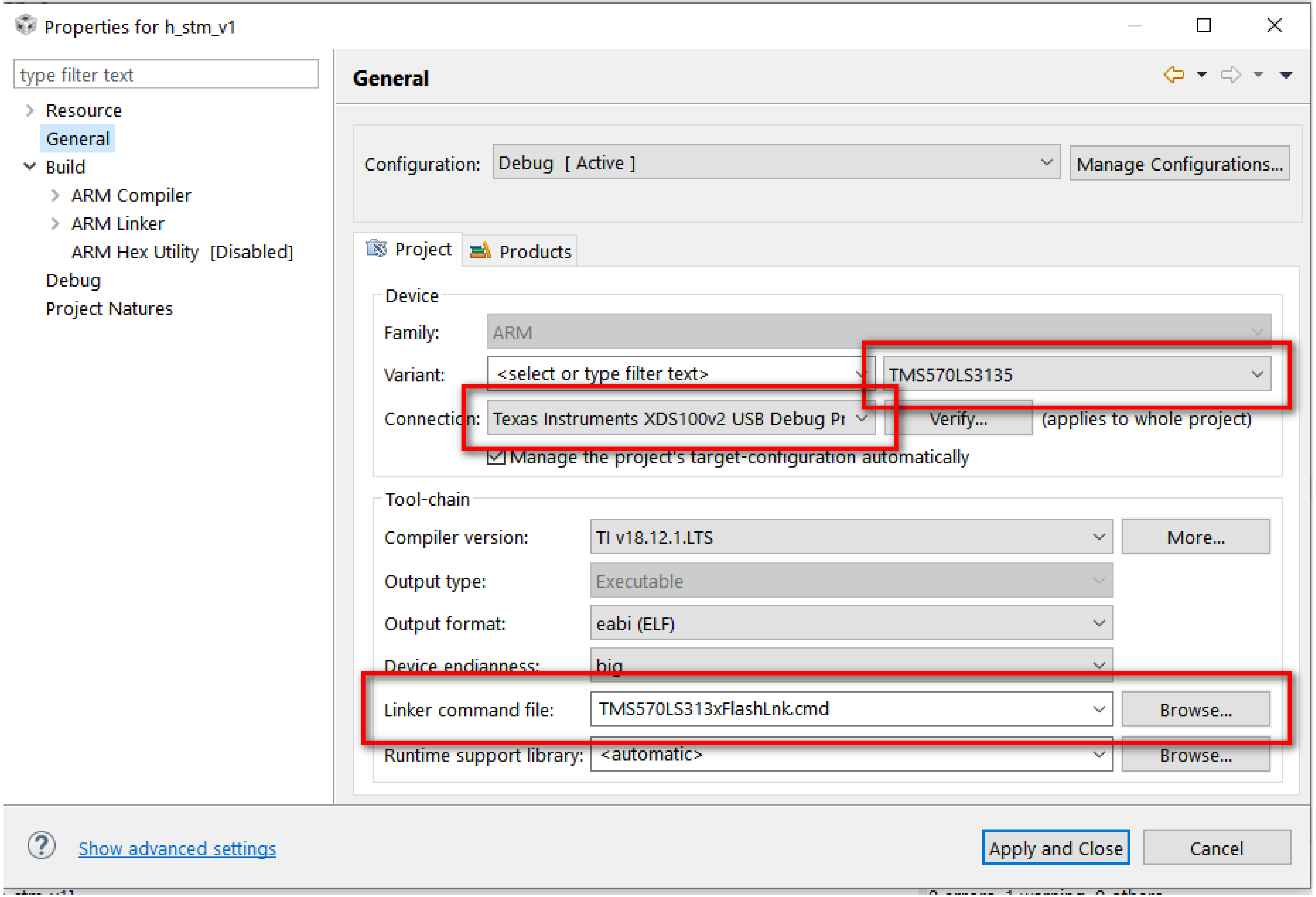
CDT Build Console [h_stm_v1]

```
float_support=VFPv3D16 -g --diag_warning=225 --diag_wrap=off --display_error_number --
enum_type=packed --abi=eabi -z -m"h_stm_v1.map" --heap_size=0x800 --stack_size=0x800 -
i"D:/ti/ccs901/ccs/tools/compiler/ti-cgt-arm_18.12.1.LTS/lib" -
i"D:/ti/ccs901/ccs/tools/compiler/ti-cgt-arm_18.12.1.LTS/include" --reread_libs --
diag_wrap=off --display_error_number --warn_sections --xml_link_info="h_stm_v1_linkInfo.xml"
--rom_model -o "h_stm_v1.out" "./stm_driver.obj" "../TMS570LS313xFlashLnk.cmd"  -llibc.a
<Linking>
warning #10247-D: creating output section ".sysmem" without a SECTIONS specification
Finished building target: "h_stm_v1.out"


**** Build Finished ****
```

0 errors, 1 warning, 0 others

| Description |
|---|
| Warnings (1 item) |
| #10247-D null: creating output section |

then:



Properties for h_stm_v1

General

Configuration: Debug [ Active ]

Device

Family: ARM

Variant: <select or type filter text> TMS570LS3135

Connection: Texas Instruments XDS100v2 USB Debug Pr (applies to whole project)

☑ Manage the project's target-configuration automatically

Tool-chain

Compiler version: TI v18.12.1.LTS

Output type: Executable

Output format: eabi (ELF)

Device endianness: big

Linker command file: TMS570LS313xFlashLnk.cmd

Runtime support library: <automatic>

For the HDK development kit TMS570LS31x ICE Board, with USB xds100v2 jtag I have chosen: TMS570LS3135 (not sure).
1. Does this match with the development kit?

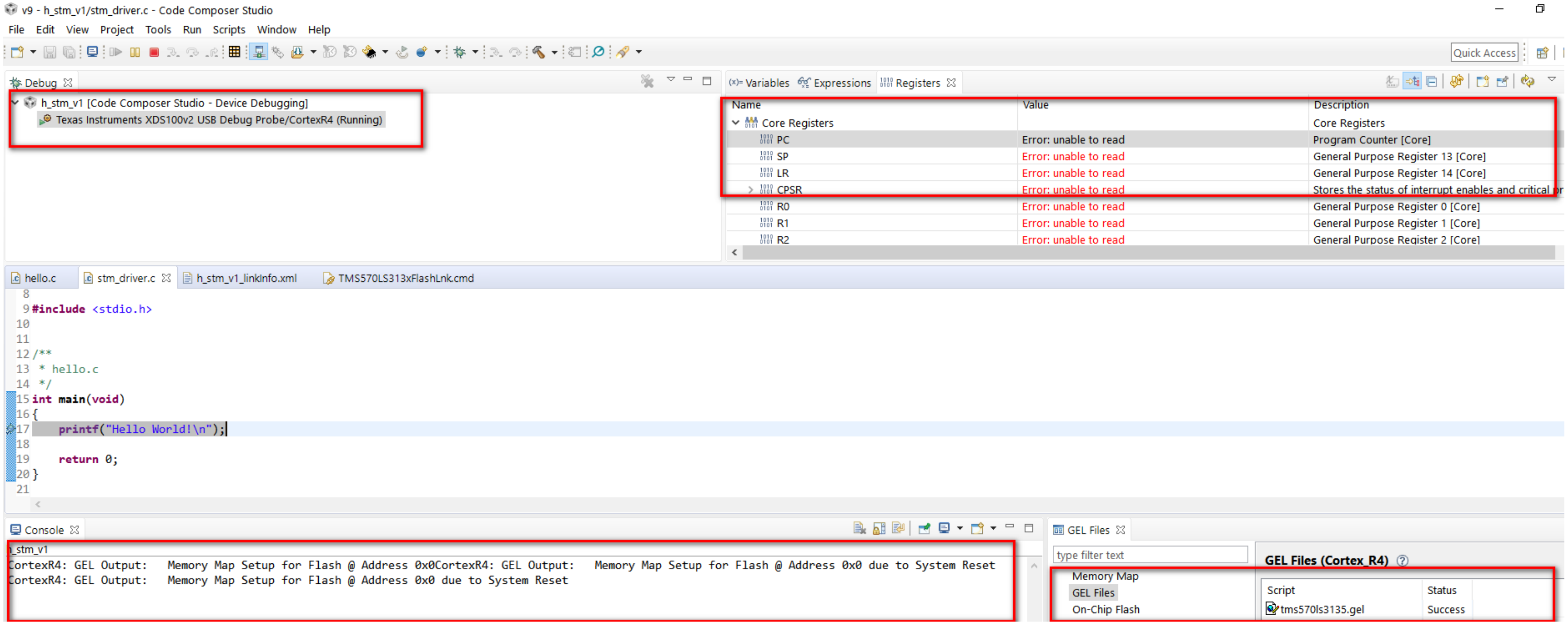
When verifying the connection:
[Start]

```
Execute the command:

%ccs_base%/common/uscif/dbgjtag -f %boarddatafile% -rv -o -F inform,logfile=yes -S pathlength -S integrity

[Result]


-----[Print the board config pathname(s)]------------------------------------

C:\Users\fa\AppData\Local\TEXASI~1\CCS\ccs901\
    0\0\BrdDat\testBoard.dat

-----[Print the reset-command software log-file]-----------------------------

This utility has selected a 100- or 510-class product.
This utility will load the adapter 'jioserdesusb.dll'.
The library build date was 'Mar 25 2019'.
The library build time was '17:36:26'.
The library package version is '8.1.0.00007'.
The library component version is '35.35.0.0'.
The controller does not use a programmable FPGA.
The controller has a version number of '4' (0x00000004).
The controller has an insertion length of '0' (0x00000000).
This utility will attempt to reset the controller.
This utility has successfully reset the controller.

-----[Print the reset-command hardware log-file]-----------------------------

The scan-path will be reset by toggling the JTAG TRST signal.
The controller is the FTDI FT2232 with USB interface.
The link from controller to target is direct (without cable).
The software is configured for FTDI FT2232 features.
The controller cannot monitor the value on the EMU[0] pin.
The controller cannot monitor the value on the EMU[1] pin.
The controller cannot control the timing on output pins.
The controller cannot control the timing on input pins.
The scan-path link-delay has been set to exactly '0' (0x0000).

-----[The log-file for the JTAG TCLK output generated from the PLL]----------

There is no hardware for programming the JTAG TCLK frequency.
```

```
-----[Measure the source and frequency of the final JTAG TCLKR input]--------

There is no hardware for measuring the JTAG TCLK frequency.

-----[Perform the standard path-length test on the JTAG IR and DR]-----------

This path-length test uses blocks of 64 32-bit words.

The test for the JTAG IR instruction path-length succeeded.
The JTAG IR instruction path-length is 6 bits.

The test for the JTAG DR bypass path-length succeeded.
The JTAG DR bypass path-length is 1 bits.

-----[Perform the Integrity scan-test on the JTAG IR]------------------------

This test will use blocks of 64 32-bit words.
This test will be applied just once.

Do a test using 0xFFFFFFFF.
Scan tests: 1, skipped: 0, failed: 0
Do a test using 0x00000000.
Scan tests: 2, skipped: 0, failed: 0
Do a test using 0xFE03E0E2.
Scan tests: 3, skipped: 0, failed: 0
Do a test using 0x01FC1F1D.
Scan tests: 4, skipped: 0, failed: 0
Do a test using 0x5533CCAA.
Scan tests: 5, skipped: 0, failed: 0
Do a test using 0xAACC3355.
Scan tests: 6, skipped: 0, failed: 0
All of the values were scanned correctly.

The JTAG IR Integrity scan-test has succeeded.

-----[Perform the Integrity scan-test on the JTAG DR]------------------------

This test will use blocks of 64 32-bit words.
This test will be applied just once.

Do a test using 0xFFFFFFFF.
```

```
Scan tests: 1, skipped: 0, failed: 0
Do a test using 0x00000000.
Scan tests: 2, skipped: 0, failed: 0
Do a test using 0xFE03E0E2.
Scan tests: 3, skipped: 0, failed: 0
Do a test using 0x01FC1F1D.
Scan tests: 4, skipped: 0, failed: 0
Do a test using 0x5533CCAA.
Scan tests: 5, skipped: 0, failed: 0
Do a test using 0xAACC3355.
Scan tests: 6, skipped: 0, failed: 0
All of the values were scanned correctly.

The JTAG DR Integrity scan-test has succeeded.

[End]
```

then when I would debug:



h_stm_v1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Debug Probe/CortexR4 (Running)

| Name | Value | Description |
|---|---|---|
| Core Registers | | Core Registers |
| PC | Error: unable to read | Program Counter [Core] |
| SP | Error: unable to read | General Purpose Register 13 [Core] |
| LR | Error: unable to read | General Purpose Register 14 [Core] |
| CPSR | Error: unable to read | Stores the status of interrupt enables and critical pr |
| R0 | Error: unable to read | General Purpose Register 0 [Core] |
| R1 | Error: unable to read | General Purpose Register 1 [Core] |
| R2 | Error: unable to read | General Purpose Register 2 [Core] |

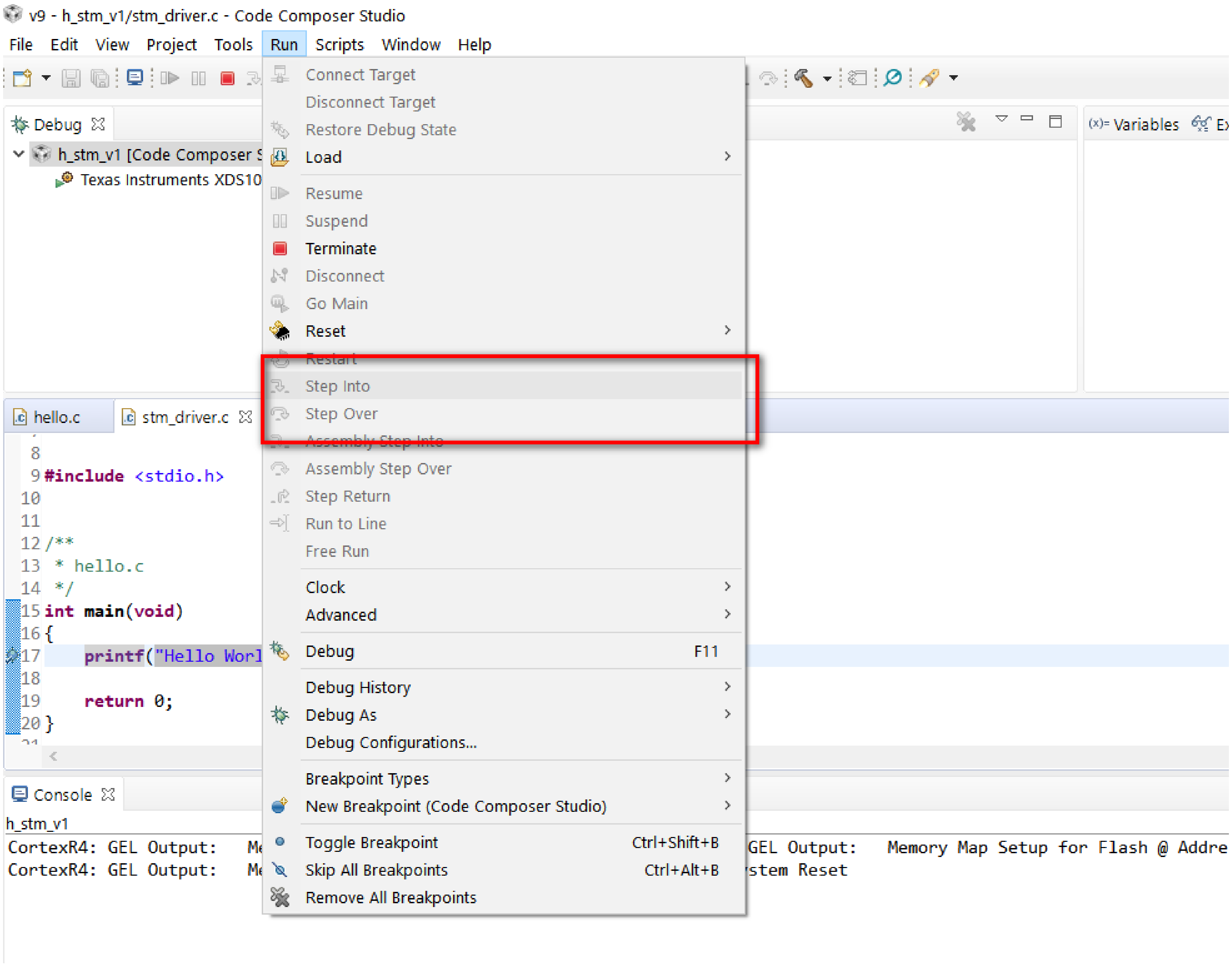
```
#include <stdio.h>


/**
 * hello.c
 */
int main(void)
{
    printf("Hello World!\n");

    return 0;
}

```

```
h_stm_v1
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0 due to System Reset
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0 due to System Reset
```

GEL Files (Cortex_R4)

| Script | Status |
|---|---|
| tms570ls3135.gel | Success |

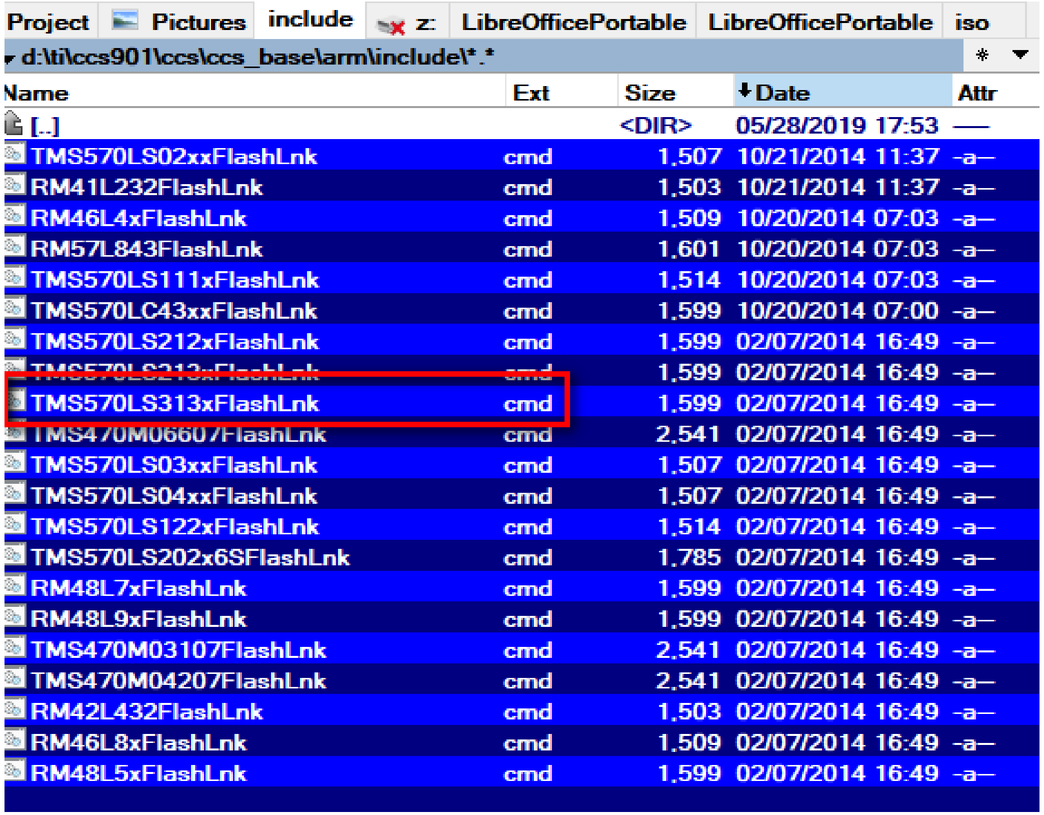
2. As I am not able to step through (see below), please let me know what I should to have the debug running on HDK board?



```
#include <stdio.h>


/**
 * hello.c
 */
int main(void)
{
    printf("Hello Worl

    return 0;
}
```

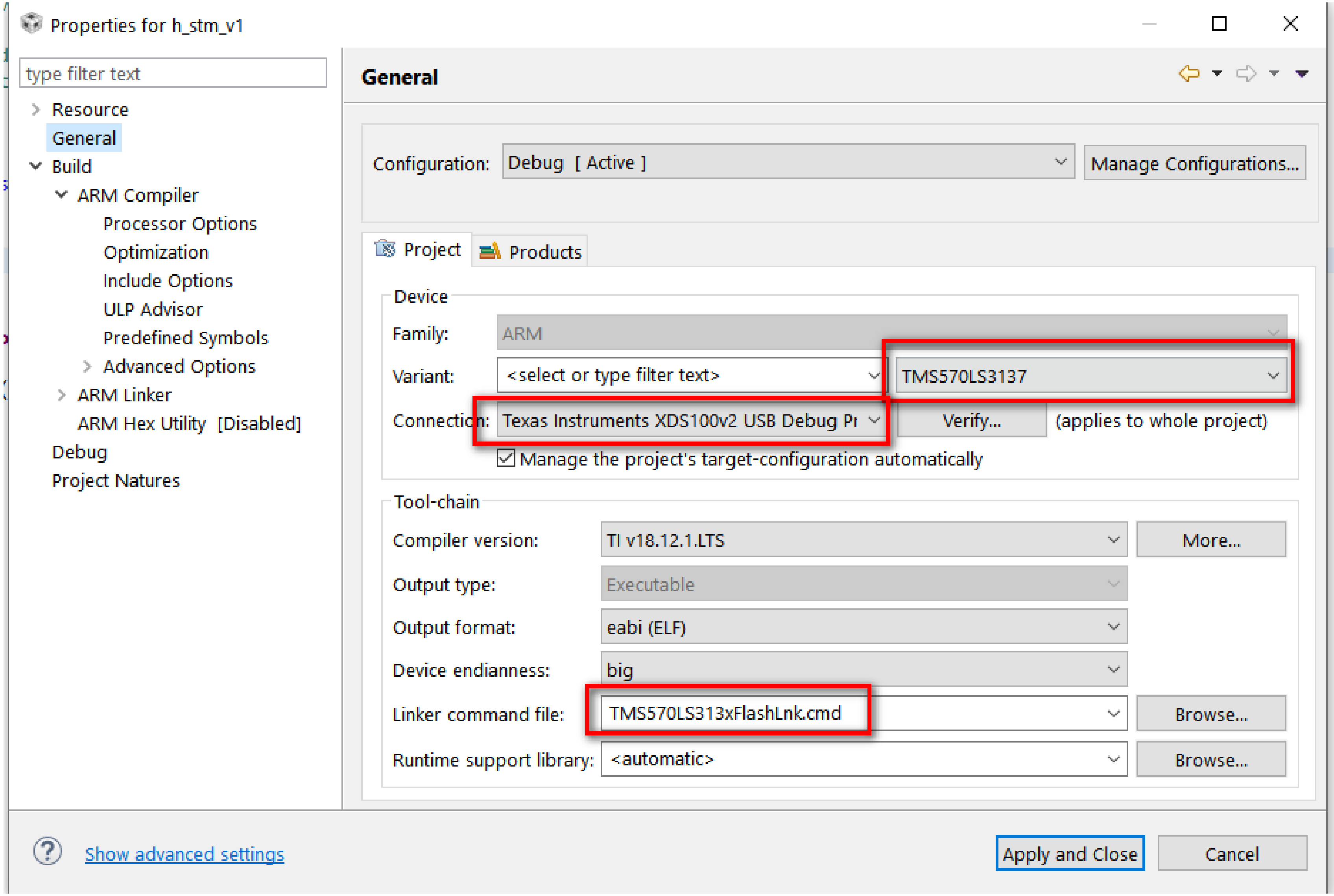
```
h_stm_v1
CortexR4: GEL Output:    Me                                             GEL Output:    Memory Map Setup for Flash @ Addre
CortexR4: GEL Output:    Me                                             stem Reset
```

2nd round:
The only place where I’ve got linker files:



| Name | Ext | Size | Date | Attr |
|---|---|---|---|---|
| [..] | | <DIR> | 05/28/2019 17:53 | —— |
| TMS570LS02xxFlashLnk | cmd | 1,507 | 10/21/2014 11:37 | -a— |
| RM41L232FlashLnk | cmd | 1,503 | 10/21/2014 11:37 | -a— |
| RM46L4xFlashLnk | cmd | 1,509 | 10/20/2014 07:03 | -a— |
| RM57L843FlashLnk | cmd | 1,601 | 10/20/2014 07:03 | -a— |
| TMS570LS111xFlashLnk | cmd | 1,514 | 10/20/2014 07:03 | -a— |
| TMS570LC43xxFlashLnk | cmd | 1,599 | 10/20/2014 07:00 | -a— |
| TMS570LS212xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |
| TMS570LS213xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |
| TMS570LS313xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |
| TMS470M06607FlashLnk | cmd | 2,541 | 02/07/2014 16:49 | -a— |
| TMS570LS03xxFlashLnk | cmd | 1,507 | 02/07/2014 16:49 | -a— |
| TMS570LS04xxFlashLnk | cmd | 1,507 | 02/07/2014 16:49 | -a— |
| TMS570LS122xFlashLnk | cmd | 1,514 | 02/07/2014 16:49 | -a— |
| TMS570LS202x6SFlashLnk | cmd | 1,785 | 02/07/2014 16:49 | -a— |
| RM48L7xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |
| RM48L9xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |
| TMS470M03107FlashLnk | cmd | 2,541 | 02/07/2014 16:49 | -a— |
| TMS470M04207FlashLnk | cmd | 2,541 | 02/07/2014 16:49 | -a— |
| RM42L432FlashLnk | cmd | 1,503 | 02/07/2014 16:49 | -a— |
| RM46L8xFlashLnk | cmd | 1,509 | 02/07/2014 16:49 | -a— |
| RM48L5xFlashLnk | cmd | 1,599 | 02/07/2014 16:49 | -a— |

Current project configuration switched to TMS570LS3137, added the linker command file from above:



Properties for h_stm_v1

General

Configuration: Debug [ Active ]

Device

Family: ARM

Variant: <select or type filter text> TMS570LS3137

Connection: Texas Instruments XDS100v2 USB Debug Pr

(applies to whole project)

☑ Manage the project's target-configuration automatically

Tool-chain

Compiler version: TI v18.12.1.LTS

Output type: Executable

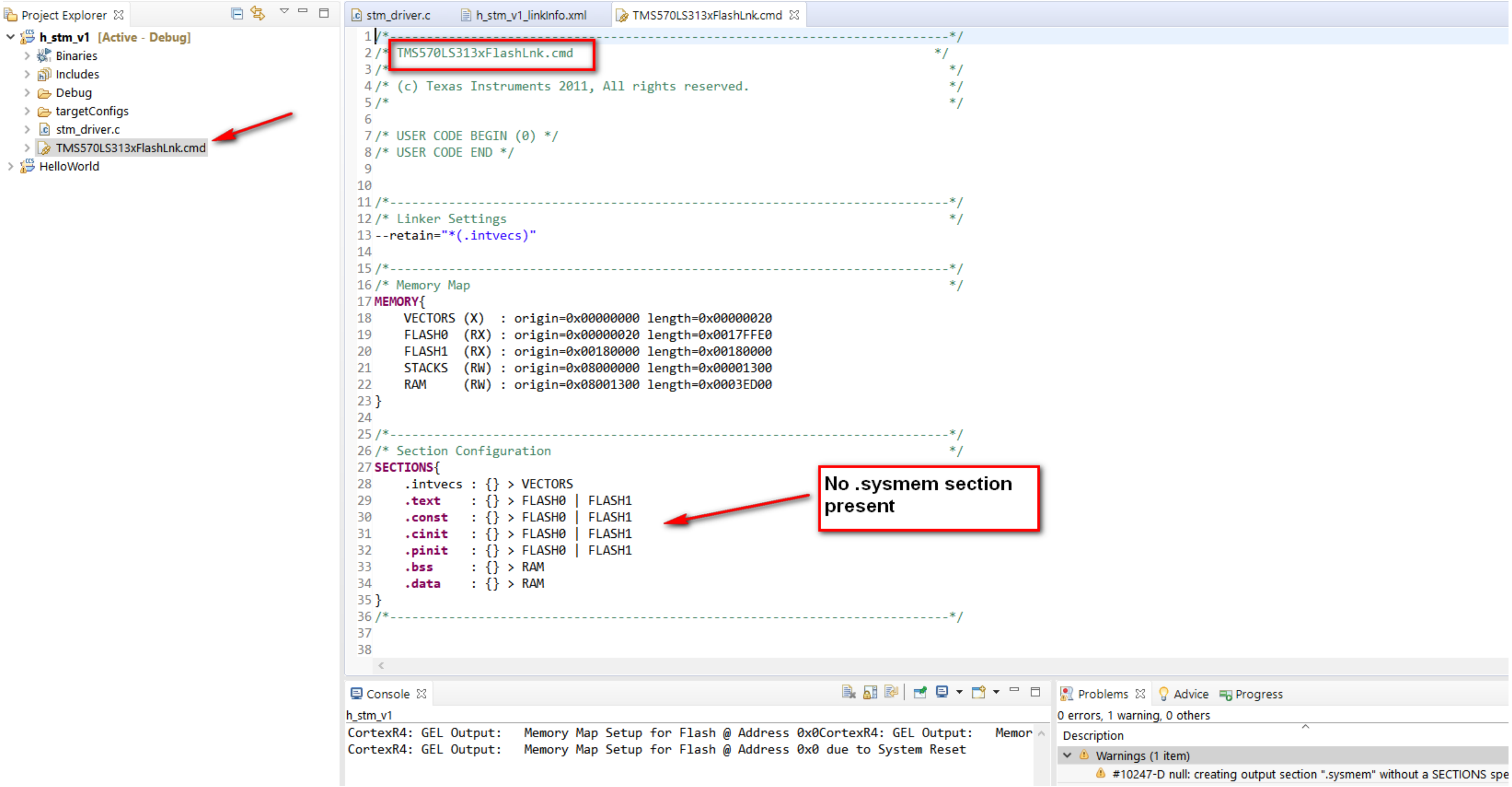
Output format: eabi (ELF)

Device endianness: big

Linker command file: TMS570LS313xFlashLnk.cmd

Runtime support library: <automatic>

Now the linker file content is:



```
/*----------------------------------------------------------------------------*/
/* TMS570LS313xFlashLnk.cmd                                                 */
/*                                                                            */
/* (c) Texas Instruments 2011, All rights reserved.                           */
/*                                                                            */

/* USER CODE BEGIN (0) */
/* USER CODE END */


/*----------------------------------------------------------------------------*/
/* Linker Settings                                                            */
--retain="*(.intvecs)"

/*----------------------------------------------------------------------------*/
/* Memory Map                                                                 */
MEMORY{
    VECTORS (X)  : origin=0x00000000 length=0x00000020
    FLASH0  (RX) : origin=0x00000020 length=0x0017FFE0
    FLASH1  (RX) : origin=0x00180000 length=0x00180000
    STACKS  (RW) : origin=0x08000000 length=0x00001300
    RAM     (RW) : origin=0x08001300 length=0x0003ED00
}

/*----------------------------------------------------------------------------*/
/* Section Configuration                                                      */
SECTIONS{
    .intvecs : {} > VECTORS
    .text    : {} > FLASH0 | FLASH1
    .const   : {} > FLASH0 | FLASH1
    .cinit   : {} > FLASH0 | FLASH1
    .pinit   : {} > FLASH0 | FLASH1
    .bss     : {} > RAM
    .data    : {} > RAM
}
/*----------------------------------------------------------------------------*/


```

No .sysmem section present

```
h_stm_v1
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0CortexR4: GEL Output:   Memor
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0 due to System Reset
```

0 errors, 1 warning, 0 others

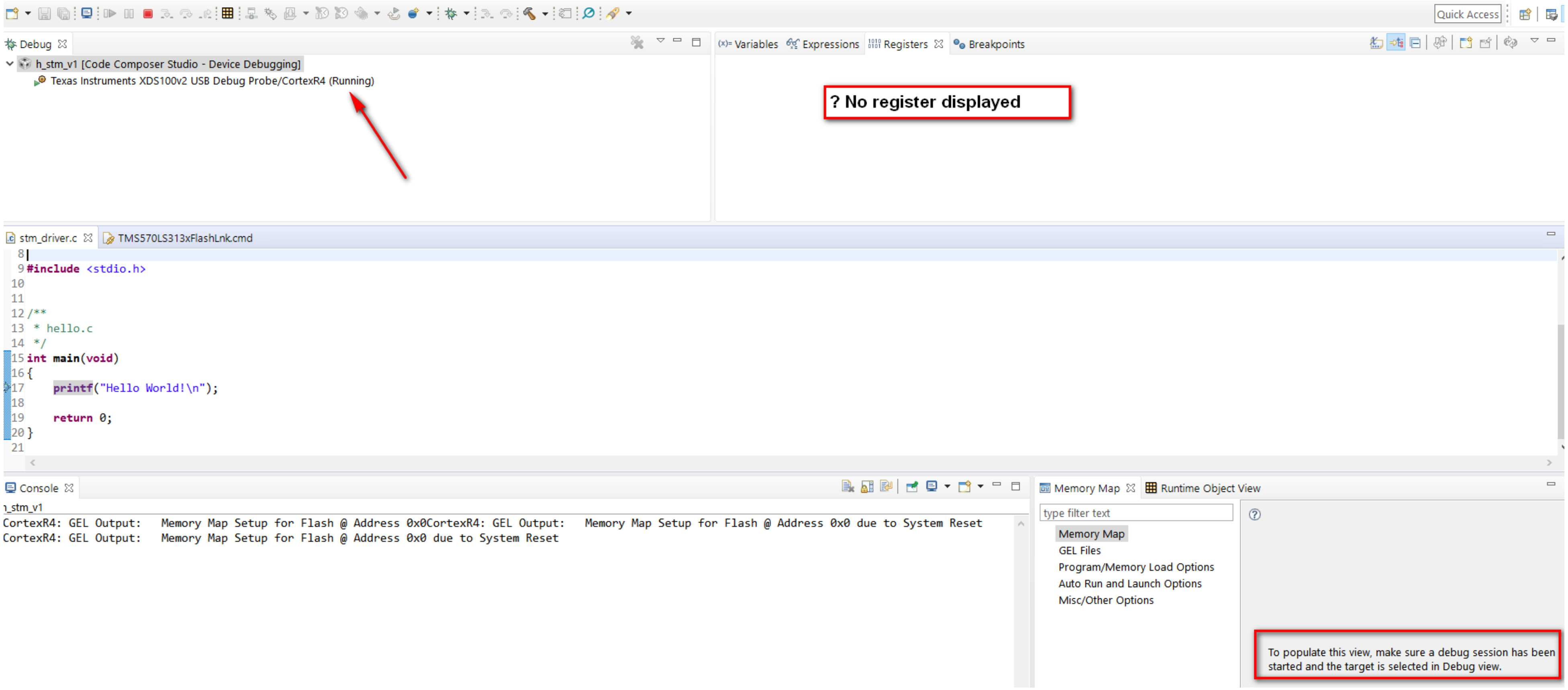
Description

Warnings (1 item)

#10247-D null: creating output section ".sysmem" without a SECTIONS spe

I have searched through the cmd files, no .sysmem section in any of the linker files.

Now when I am debugging:



h_stm_v1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Debug Probe/CortexR4 (Running)

? No register displayed

```

#include <stdio.h>


/**
 * hello.c
 */
int main(void)
{
    printf("Hello World!\n");

    return 0;
}

```

```
ı_stm_v1
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0 due to System Reset
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address 0x0 due to System Reset
```

To populate this view, make sure a debug session has been started and the target is selected in Debug view.

The program doesn’t stop at the breakpoint.