Learning Report – Embedded C

Global Engineering Academy

Genesis

L&T Technology Services

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
1	30th Sept, 2020	Milind Mohapatra		To be approved by Dr. Vivek Kaundal, Bhargav N	
2	5th Oct, 2020	Milind Mohapatra		To be approved by Dr. Vivek Kaundal, Bhargav N	

CONTENTS

TABLE OF FIGURES

Activity 1: Linker Script

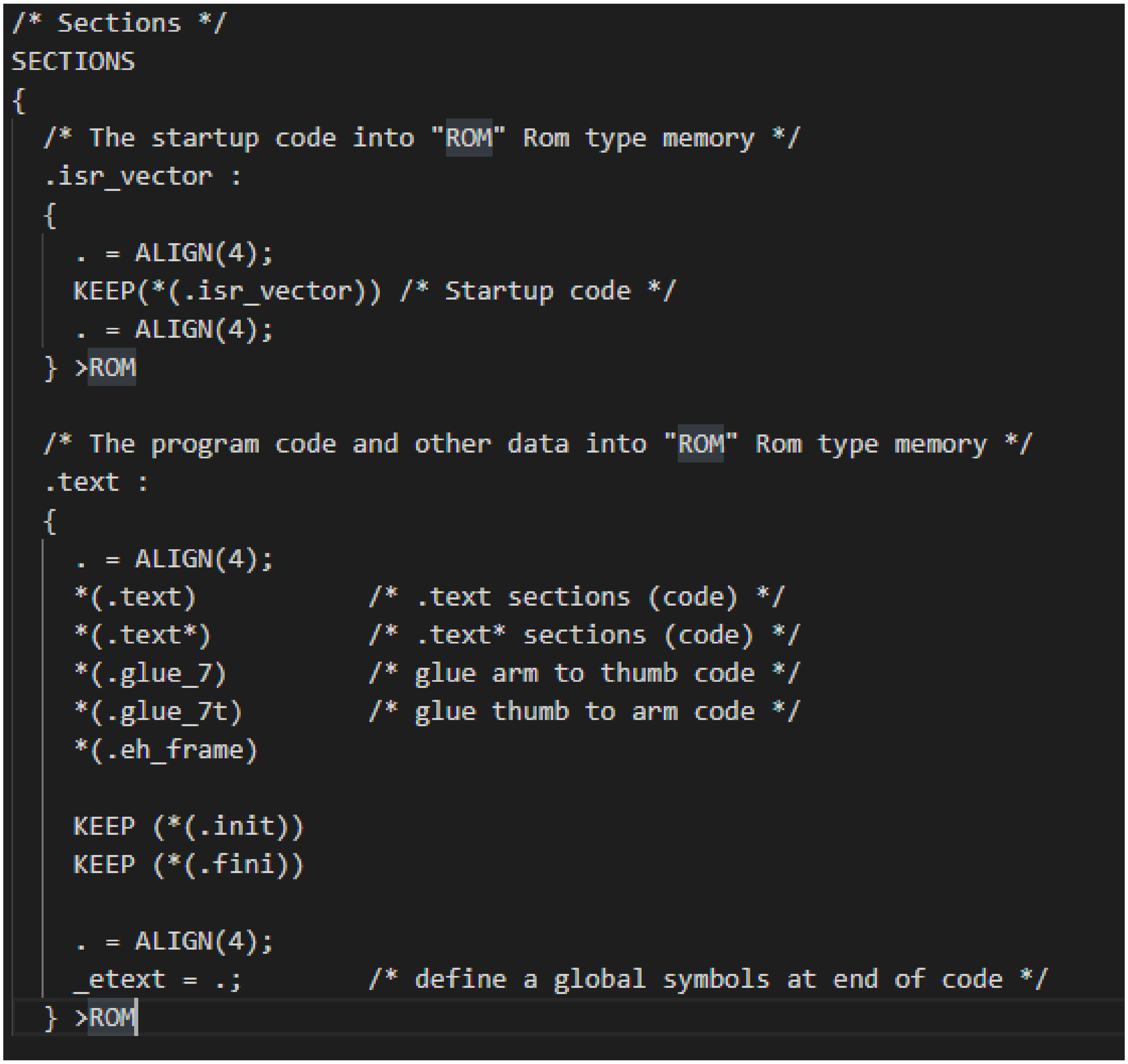
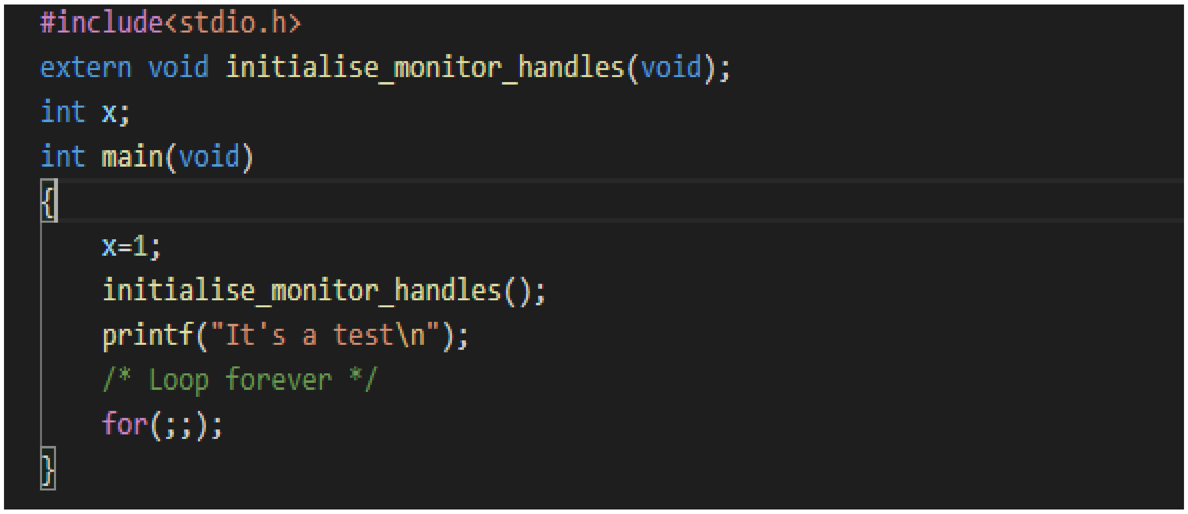

```
/* Sections */
SECTIONS
{
  /* The startup code into "ROM" Rom type memory */
  .isr_vector :
  {
    . = ALIGN(4);
    KEEP(*(.isr_vector)) /* Startup code */
    . = ALIGN(4);
  } >ROM

  /* The program code and other data into "ROM" Rom type memory */
  .text :
  {
    . = ALIGN(4);
    *(.text)           /* .text sections (code) */
    *(.text*)          /* .text* sections (code) */
    *(.glue_7)         /* glue arm to thumb code */
    *(.glue_7t)        /* glue thumb to arm code */
    *(.eh_frame)

    KEEP (*(.init))
    KEEP (*(.fini))

    . = ALIGN(4);
    _etext = .;        /* define a global symbols at end of code */
  } >ROM
```

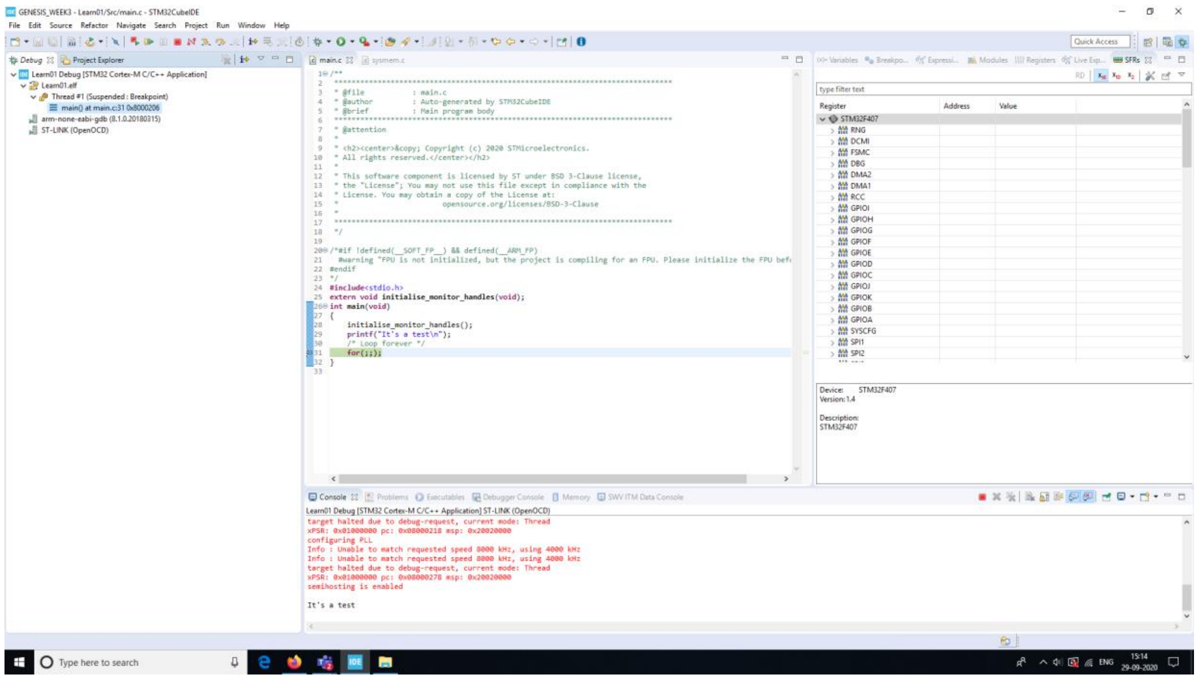
Figure 1: Linker section

Activity 2: Semi Hosting



```c
#include<stdio.h>
extern void initialise_monitor_handles(void);
int x;
int main(void)
{
    x=1;
    initialise_monitor_handles();
    printf("It's a test\n");
    /* Loop forever */
    for(;;);
}
```

Figure 2: Code Snippet for semi hosting

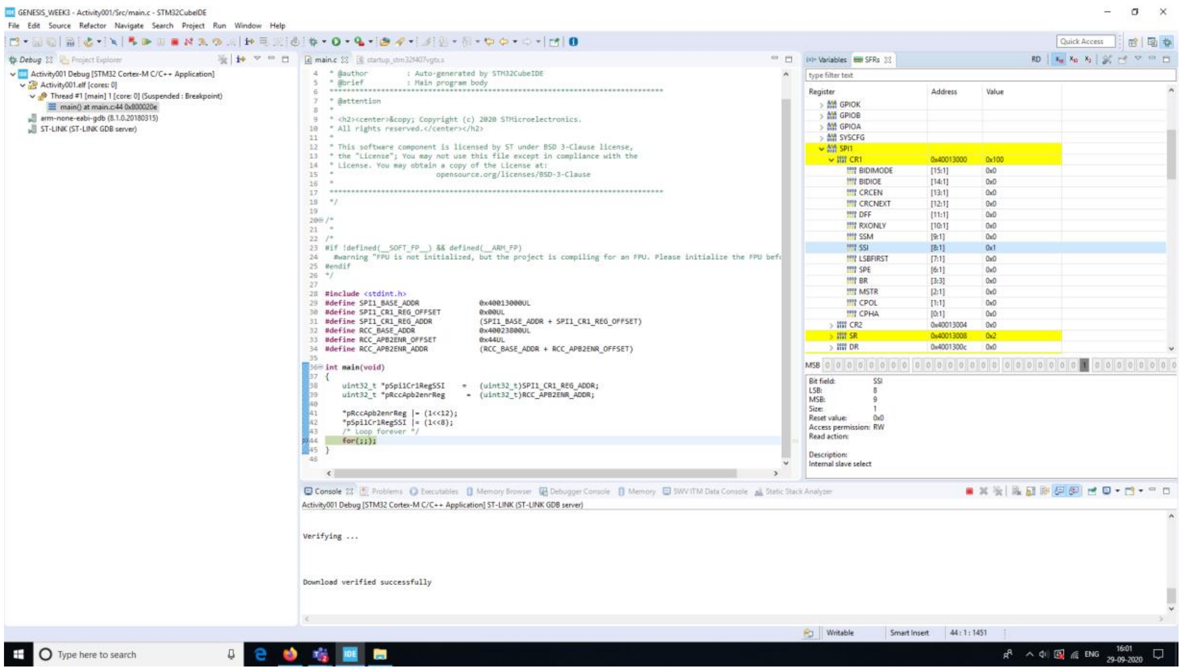


```c
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */

/*#if !defined(__SOFT_FP__) && defined(__ARM_FP)
  #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU befo
#endif
*/
#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{
    initialise_monitor_handles();
    printf("It's a test\n");
    /* Loop forever */
    for(;;);
}
```

```
Learn01 Debug [STM32 Cortex-M C/C++ Application] ST-LINK (OpenOCD)
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000218 msp: 0x20020000
configuring PLL
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000278 msp: 0x20020000
semihosting is enabled

It's a test
```

Figure 3: Semi hosting enabled

Activity 3: Changing bit state of SPI1 SPE



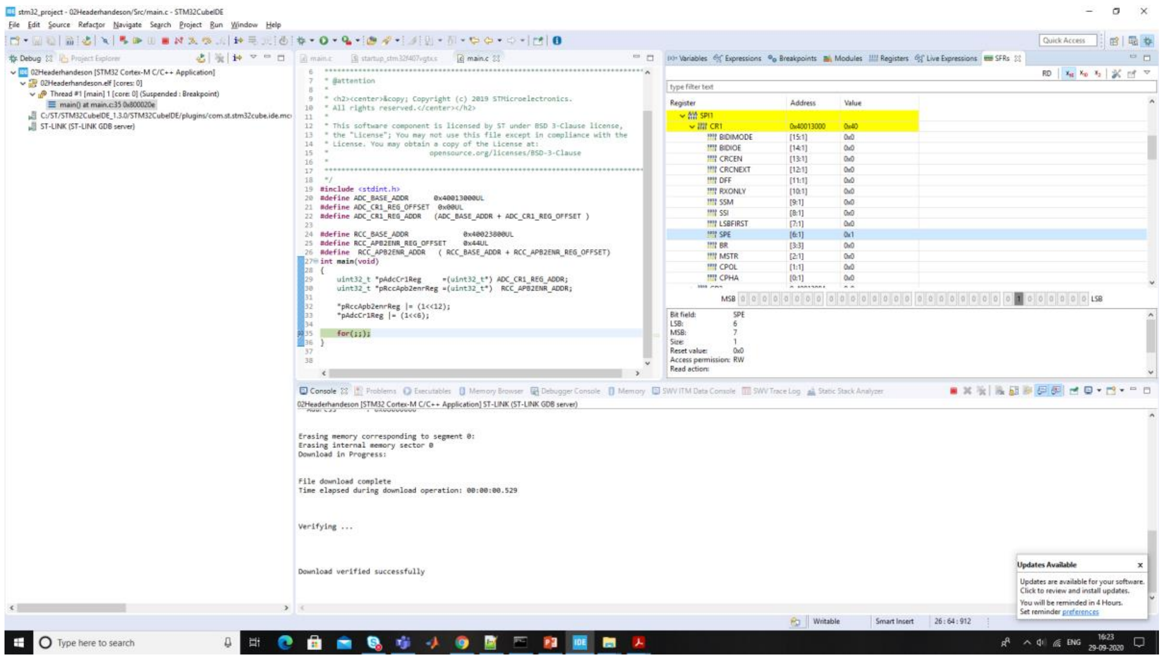
```c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */

/*
 *
/*
#if !defined(__SOFT_FP__) && defined(__ARM_FP)
  #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU bef
#endif
*/

#include <stdint.h>
#define SPI1_BASE_ADDR            0x40013000UL
#define SPI1_CR1_REG_OFFSET       0x00UL
#define SPI1_CR1_REG_ADDR         (SPI1_BASE_ADDR + SPI1_CR1_REG_OFFSET)
#define RCC_BASE_ADDR             0x40023800UL
#define RCC_APB2ENR_OFFSET        0x44UL
#define RCC_APB2ENR_ADDR          (RCC_BASE_ADDR + RCC_APB2ENR_OFFSET)

int main(void)
{
    uint32_t *pSpi1Cr1RegSSI   =  (uint32_t)SPI1_CR1_REG_ADDR;
    uint32_t *pRccApb2enrReg    =  (uint32_t)RCC_APB2ENR_ADDR;

    *pRccApb2enrReg |= (1<<12);
    *pSpi1Cr1RegSSI |= (1<<8);
    /* Loop forever */
    for(;;);
}
```

Register	Address	Value
GPIOK		
GPIOB		
GPIOA		
SYSCFG		
SPI1		
CR1	0x40013000	0x100
BIDIMODE	[15:1]	0x0
BIDIOE	[14:1]	0x0
CRCEN	[13:1]	0x0
CRCNEXT	[12:1]	0x0
DFF	[11:1]	0x0
RXONLY	[10:1]	0x0
SSM	[9:1]	0x0
SSI	[8:1]	0x1
LSBFIRST	[7:1]	0x0
SPE	[6:1]	0x0
BR	[3:3]	0x0
MSTR	[2:1]	0x0
CPOL	[1:1]	0x0
CPHA	[0:1]	0x0
CR2	0x40013004	0x0
SR	0x40013008	0x2
DR	0x4001300c	0x0

```
Bit field:          SSI
LSB:                8
MSB:                9
Size:               1
Reset value:        0x0
Access permission:  RW
Read action:

Description:
Internal slave select
```

```
Activity001 Debug [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server)

Verifying ...


Download verified successfully
```

Figure 4: Bit modification of SPI1 SSI



```c
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2019 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */
#include <stdint.h>
#define ADC_BASE_ADDR       0x40013000UL
#define ADC_CR1_REG_OFFSET  0x00UL
#define ADC_CR1_REG_ADDR    (ADC_BASE_ADDR + ADC_CR1_REG_OFFSET )

#define RCC_BASE_ADDR             0x40023800UL
#define RCC_APB2ENR_REG_OFFSET    0x44UL
#define  RCC_APB2ENR_ADDR   ( RCC_BASE_ADDR + RCC_APB2ENR_REG_OFFSET)
int main(void)
{
    uint32_t *pAdcCr1Reg     =(uint32_t*) ADC_CR1_REG_ADDR;
    uint32_t *pRccApb2enrReg =(uint32_t*)  RCC_APB2ENR_ADDR;

    *pRccApb2enrReg |= (1<<12);
    *pAdcCr1Reg |= (1<<6);

    for(;;);
}
```

Register	Address	Value
SPI1		
CR1	0x40013000	0x40
BIDIMODE	[15:1]	0x0
BIDIOE	[14:1]	0x0
CRCEN	[13:1]	0x0
CRCNEXT	[12:1]	0x0
DFF	[11:1]	0x0
RXONLY	[10:1]	0x0
SSM	[9:1]	0x0
SSI	[8:1]	0x0
LSBFIRST	[7:1]	0x0
SPE	[6:1]	0x1
BR	[3:3]	0x0
MSTR	[2:1]	0x0
CPOL	[1:1]	0x0
CPHA	[0:1]	0x0

```
Bit field:          SPE
LSB:                6
MSB:                7
Size:               1
Reset value:        0x0
Access permission:  RW
Read action:
```

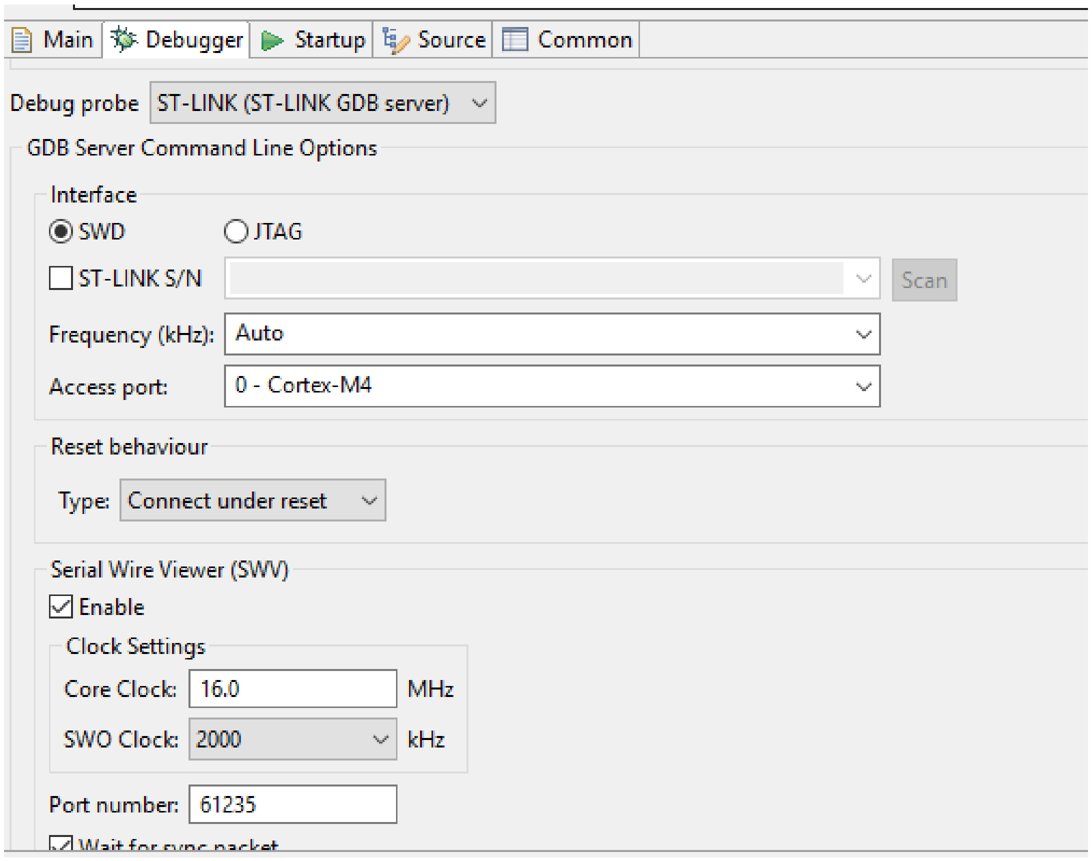
```
02Headerhandeson [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server)

Erasing memory corresponding to segment 0:
Erasing internal memory sector 0
Download in Progress:


File download complete
Time elapsed during download operation: 00:00:00.529



Verifying ...



Download verified successfully
```

Updates Available
Updates are available for your software. Click to review and install updates.
You will be reminded in 4 Hours.
Set reminder preferences

Figure 5: Bit modification of SPI1 SPE

Activity 4: Debugging techniques

Serial wire viewer and data tracing



Debug probe: ST-LINK (ST-LINK GDB server)

GDB Server Command Line Options

Interface

[✓ SWD] JTAG

☐ ST-LINK S/N

Frequency (kHz): Auto

Access port: 0 - Cortex-M4

Reset behaviour

Type: Connect under reset

Serial Wire Viewer (SWV)

☑ Enable

Clock Settings

Core Clock: 16.0 MHz

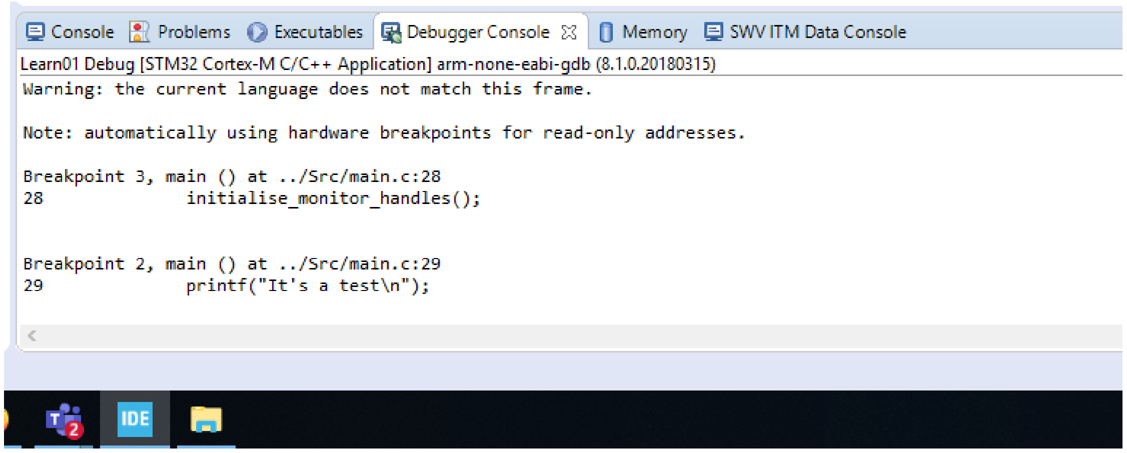
SWO Clock: 2000 kHz

Port number: 61235

☑ Wait for sync packet

Figure 6: Enable serial wire viewer

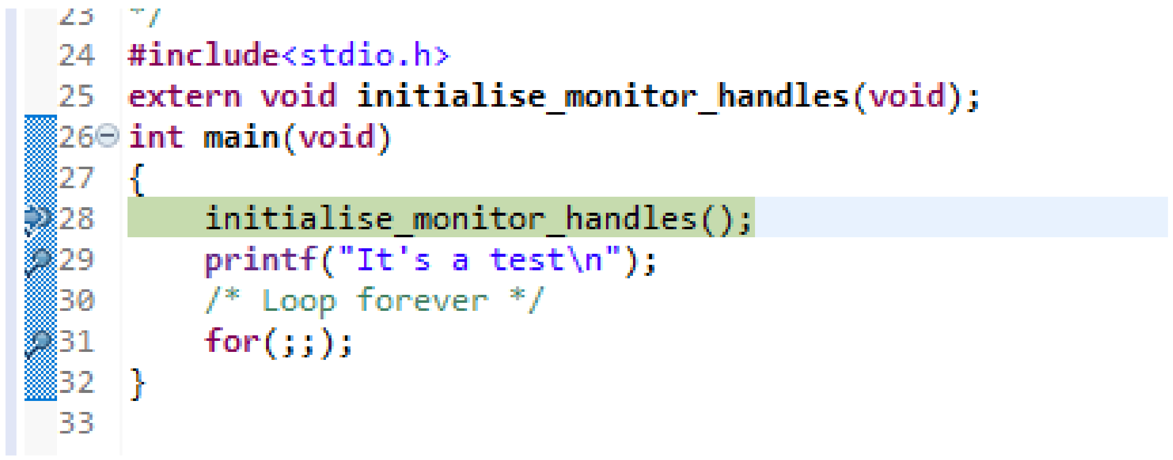
Single stepping, stepping over and stepping out



```
Learn01 Debug [STM32 Cortex-M C/C++ Application] arm-none-eabi-gdb (8.1.0.20180315)
Warning: the current language does not match this frame.

Note: automatically using hardware breakpoints for read-only addresses.

Breakpoint 3, main () at ../Src/main.c:28
28            initialise_monitor_handles();


Breakpoint 2, main () at ../Src/main.c:29
29            printf("It's a test\n");
```

Figure 7: Breakpoint stepping

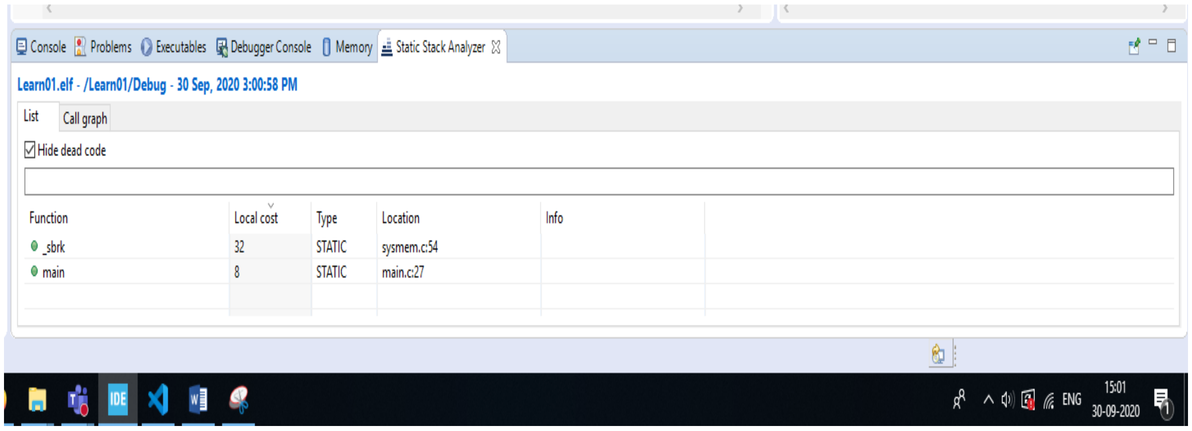
Breakpoints



```c
*/
#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{
    initialise_monitor_handles();
    printf("It's a test\n");
    /* Loop forever */
    for(;;);
}
```

Figure 8: Multiple breakpoints

Call stack (Static stack analyzer)

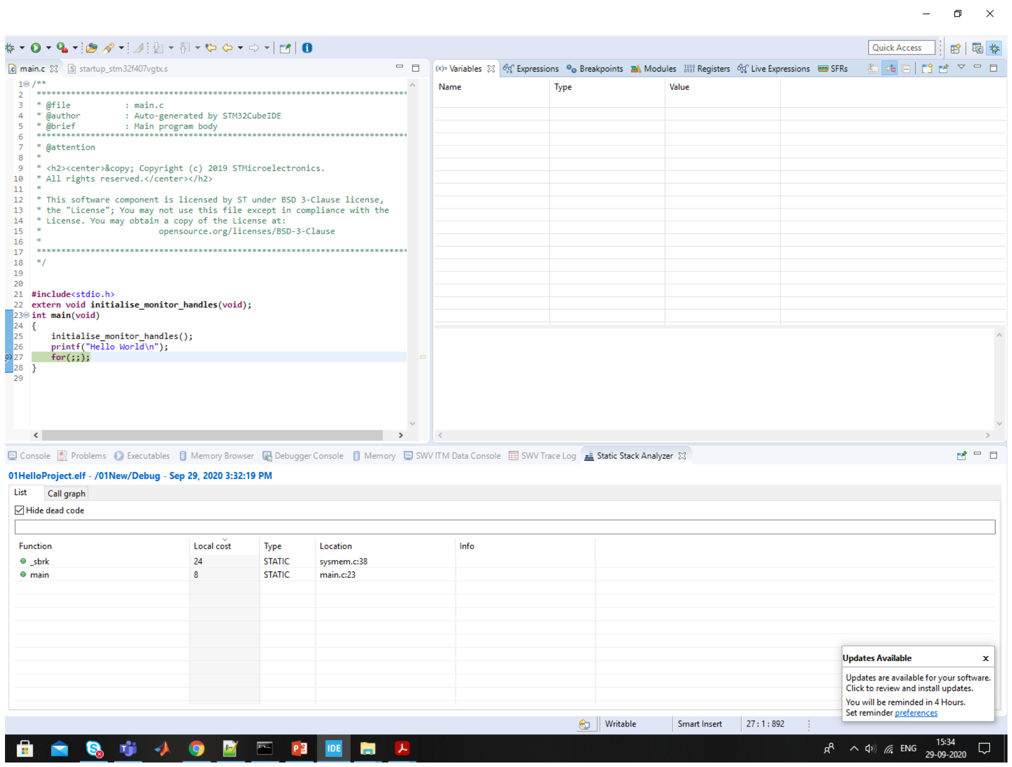


Learn01.elf - /Learn01/Debug - 30 Sep, 2020 3:00:58 PM

Function	Local cost	Type	Location	Info
_sbrk	32	STATIC	sysmem.c:54	
main	8	STATIC	main.c:27	

Figure 9: Static stack analyzer

Expression and Variable window



```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2019 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */


#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{
    initialise_monitor_handles();
    printf("Hello World\n");
    for(;;);
}
```

01HelloProject.elf - /01New/Debug - Sep 29, 2020 3:32:19 PM

☑ Hide dead code

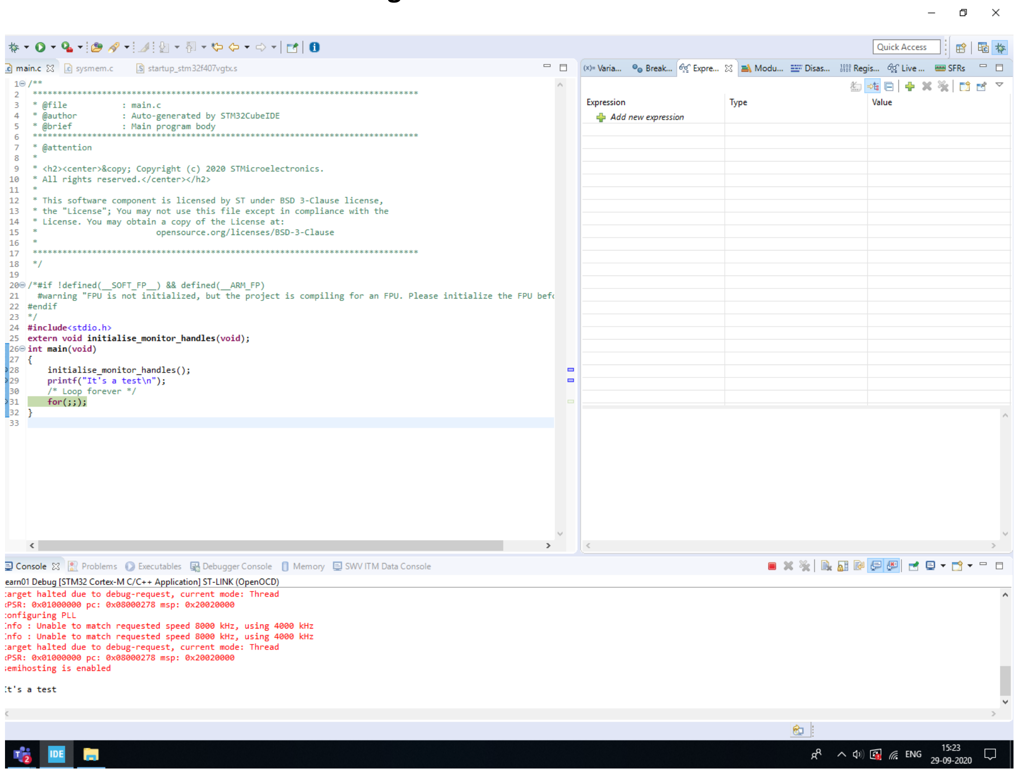
Function	Local cost	Type	Location	Info
_sbrk	24	STATIC	sysmem.c:38	
main	8	STATIC	main.c:23	

Updates Available

Updates are available for your software. Click to review and install updates.

You will be reminded in 4 Hours.
Set reminder preferences

Figure 10: Variable window

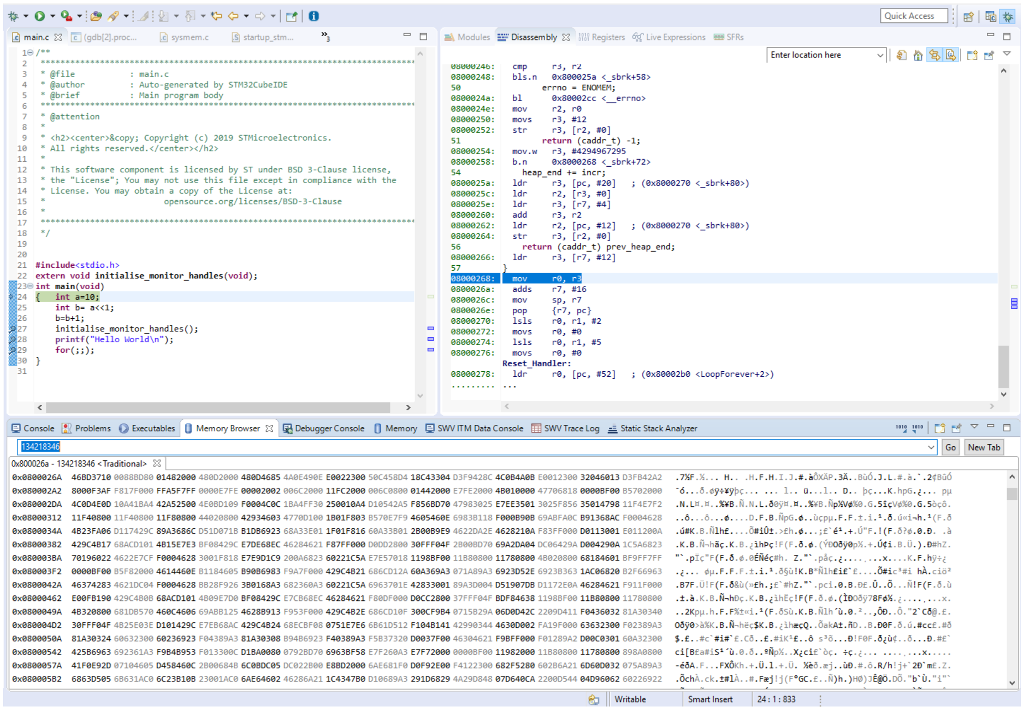


```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */

/*#if !defined(__SOFT_FP__) && defined(__ARM_FP)
  #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU befo
#endif
*/
#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{
    initialise_monitor_handles();
    printf("It's a test\n");
    /* Loop forever */
    for(;;);
}
```

```
earn01 Debug [STM32 Cortex-M C/C++ Application] ST-LINK (OpenOCD)
:arget halted due to debug-request, current mode: Thread
:PSR: 0x01000000 pc: 0x08000278 msp: 0x20020000
:onfiguring PLL
:nfo : Unable to match requested speed 8000 kHz, using 4000 kHz
:nfo : Unable to match requested speed 8000 kHz, using 4000 kHz
:arget halted due to debug-request, current mode: Thread
:PSR: 0x01000000 pc: 0x08000278 msp: 0x20020000
;emihosting is enabled

:t's a test
```

Figure 11: Expression window

Memory browser



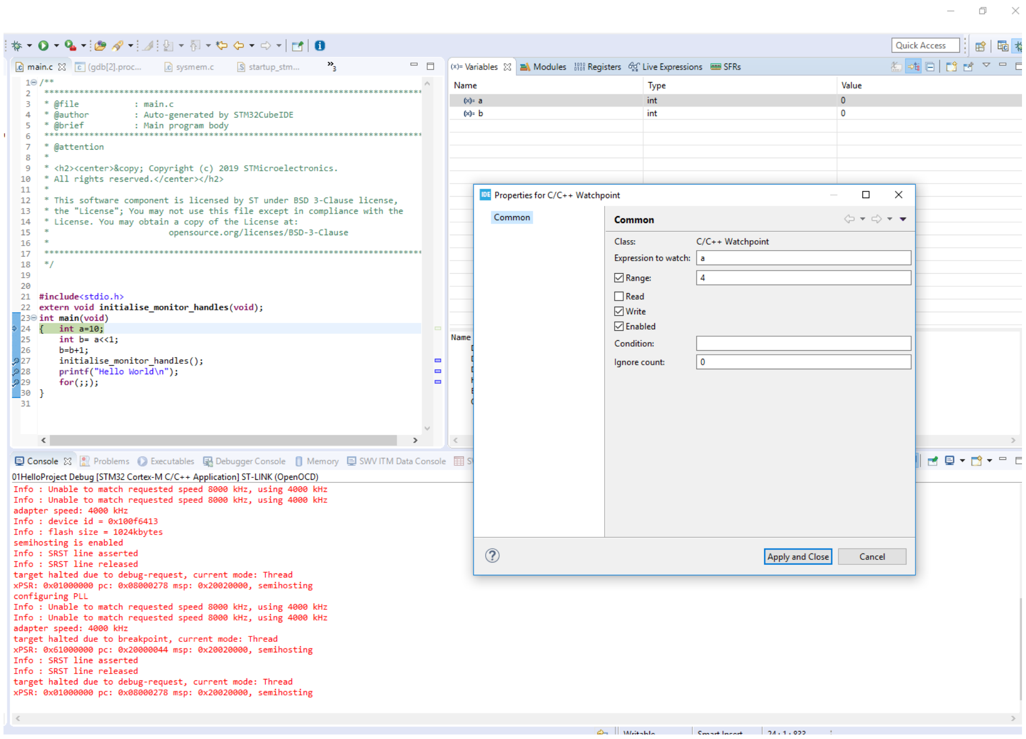
```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2019 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */


#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{   int a=10;
    int b= a<<1;
    b=b+1;
    initialise_monitor_handles();
    printf("Hello World\n");
    for(;;);
}
```

```
08000246:   cmp     r3, r2
08000248:   bls.n   0x800025a <_sbrk+58>
50              errno = ENOMEM;
0800024a:   bl      0x80002cc <__errno>
0800024e:   mov     r2, r0
08000250:   movs    r3, #12
08000252:   str     r3, [r2, #0]
51              return (caddr_t) -1;
08000254:   mov.w   r3, #4294967295
08000258:   b.n     0x8000268 <_sbrk+72>
54            heap_end += incr;
0800025a:   ldr     r3, [pc, #20]   ; (0x8000270 <_sbrk+80>)
0800025c:   ldr     r2, [r3, #0]
0800025e:   ldr     r3, [r7, #4]
08000260:   add     r3, r2
08000262:   ldr     r2, [pc, #12]   ; (0x8000270 <_sbrk+80>)
08000264:   str     r3, [r2, #0]
56            return (caddr_t) prev_heap_end;
08000266:   ldr     r3, [r7, #12]
57          }
08000268:   mov     r0, r3
0800026a:   adds    r7, #16
0800026c:   mov     sp, r7
0800026e:   pop     {r7, pc}
08000270:   lsls    r0, r1, #2
08000272:   movs    r0, #0
08000274:   lsls    r0, r1, #5
08000276:   movs    r0, #0
          Reset_Handler:
08000278:   ldr     r0, [pc, #52]   ; (0x80002b0 <LoopForever+2>)
.........   ...
```

```
134218346
0x800026a - 134218346 <Traditional>
0x0800026A  46BD3710 0088BD80 01482000 480D2000 480D4685 4A0E490E E0022300 50C458D4 18C43304 D3F9428C 4C0B4A0B E0012300 32046013 D3FB42A2  .7½F.½... H.. .H.F.H.I.J.#.àÔXÄP.3Ä..BùÓ.J.L.#.à.`.2¢BûÓ
0x080002A2  8000F3AF F817F000 FFA5F7FF 0000E7FE 00002002 006C2000 11FC2000 006C0800 01442000 E7FE2000 4B010000 47706818 0000BF00 B5702000  ¯ó...ð.øÿ÷¥ÿþç... ... l.. ü...l.. D.. þç...K.hpG.¿... pµ
0x080002DA  4C0D4E0D 10A41BA4 42A52500 4E0BD109 F0004C0C 1BA4FF30 250010A4 D10542A5 F856BD70 47983025 E7EE3501 3025F856 35014798 11F4E7F2  .N.L¤.¤..%¥B.Ñ.N.L.ð0ÿ¤.¤..%¥B.Ñp½Vø%0.G.5îçVø%0.G.5òçô.
0x08000312  11F40800 11F40800 11F80800 44020800 42934603 4770D100 1B01F803 B570E7F9 4605460E 6983B118 F000B90B 69ABFA0C B91368AC F0004628  ..ô...ô...ø....D.F.B.ÑpG.ø..ùçpµ.F.F.±.i.¹.ð.ú«i¬h.¹(F.ð
0x0800034A  4B23FA06 D117429C 89A3686C D51D071B B1DB6923 68A33E01 1F01F816 60A33B01 2B00B9E9 4622DA2E 4628210A F83FF000 D0113001 E011200A  .ú#K.B.Ñlh£....Õ#iÛ±.>£h.ø...;£`é¹.+.Ú"F.!(F.ð?ø.0.Ð. .à
0x08000382  429C4B17 68ACD101 4B15E7E3 BF08429C E7DE68EC 46284621 F87FF000 D0DD2800 30FFF04F 2B00BD70 69A2DA04 DC06429A D004290A 1C5A6823  .K.B.Ñ¬hãç.K.B.¿ìhÞç!F(F.ð.ø.(ÝÐOðÿ0p½.+.Ú¢i.B.Ü.).Ð#hZ.
0x080003BA  70196022 4622E7CF F0004628 3001F818 E7E9D1C9 200A6823 60221C5A E7E57018 1198BF00 11B80800 11780800 4B020800 68184601 BF9FF7FF  "`.pÏç"F(F.ð.ø.0ÉÑéç#h. Z."`.påç.¿.......x....K.F.hÿ÷¿
0x080003F2  0000BF00 B5F82000 4614460E B1184605 B90B6983 F9A7F000 429C4B21 686CD12A 60A369A3 071A89A3 6923D52E 6923B363 1AC06820 B2F66963  .¿... øµ.F.F.F.±.i.¹.ð§ù!K.B*Ñlh£i£`£....Õ#ic³#i hÀ.ciö²
0x0800042A  46374283 4621DC04 F0004628 BB28F926 3B0168A3 682360A3 60221C5A 6963701E 42833001 89A3D004 D51907DB D1172E0A 46284621 F911F000  .B7F.Ü!F(F.ð&ù(»£h.;£`#hZ."`.pci.0.B.Ð£.Û..Õ...Ñ!F(F.ð.ù
0x08000462  E00FB190 429C4B0B 68ACD101 4B09E7D0 BF08429C E7CB68EC 46284621 F80DF000 D0CC2800 37FFF04F BDF84638 1198BF00 11B80800 11780800  .±.à.K.B.Ñ¬hÐç.K.B.¿ìhËç!F(F.ð.ø.(ÌÐOðÿ78Fø½.¿.......x.
0x0800049A  4B320800 681DB570 460C4606 69ABB125 4628B913 F953F000 429C4B2E 686CD10F 300CF9B4 0715B29A 06D0D42C 2209D411 F0436032 81A30340  ..2Kpµ.h.F.F%±«i.¹(F.ðSù.K.B.Ñlh´ù.0.².,ÔÐ..Ô."2`Cð@.£.
0x080004D2  30FFF04F 4B25E03E D101429C E7EB68AC 429C4B24 68ECBF08 0751E7E6 6B61D512 F104B141 42990344 4630D002 FA19F000 63632300 F02389A3  Oðÿ0>à%K.B.Ñ¬hëç$K.B.¿ìhæçQ..ÕakA±.ñD..B.Ð0F.ð.ú.#cc£.#ð
0x0800050A  81A30324 60632300 60236923 F04389A3 81A30308 B94B6923 F40389A3 F5B37320 D0037F00 46304621 F9BFF000 F01289A2 D00C0301 60A32300  $.£..#c`#i#`£.Cð..£.#iK¹£..ô s³õ...Ð!F0F.ð¿ù¢..ð...Ð.#£`
0x08000542  425B6963 692361A3 F9B4B953 F013300C D1BA0080 0792BD70 6963BF58 E7F260A3 E7F72000 0000BF00 11982000 11B80800 11780800 898A0800  ci[B£a#iS¹´ù.0.ð..ºÑp½..X¿cič`òç. ÷ç.¿... .......x.....
0x0800057A  41F0E92D 07104605 D458460C 2B00684B 6C0BDC05 DC022B00 E8BD2000 6AE681F0 D0F92E00 F4122300 682F5280 602B6A21 6D60D032 075A89A3  -éðA.F...FXÔKh.+.Ü.l.+.Ü. ½èð.æj..ùÐ.#.ô.R/h!j+`2Ð`m£.Z.
0x080005B2  6863D505 6B631AC0 6C23B10B 23001AC0 6AE64602 46286A21 1C4347B0 D10689A3 291D6829 4A29D848 07D640CA 2200D544 04D96062 60226922  .ÕchÀ.ck.±#lÀ..#.Fæj!j(F°GC.£..Ñ)h.)HØ)JÊ@Ö.DÕ."b`Ù."i"`
```

Figure 12: Memory browser

Data watch points



```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2019 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */


#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{   int a=10;
    int b= a<<1;
    b=b+1;
    initialise_monitor_handles();
    printf("Hello World\n");
    for(;;);
}
```

Name	Type	Value
a	int	0
b	int	0

Properties for C/C++ Watchpoint

Common

Class: C/C++ Watchpoint

Expression to watch: a

☑ Range: 4

☐ Read

☑ Write

☑ Enabled

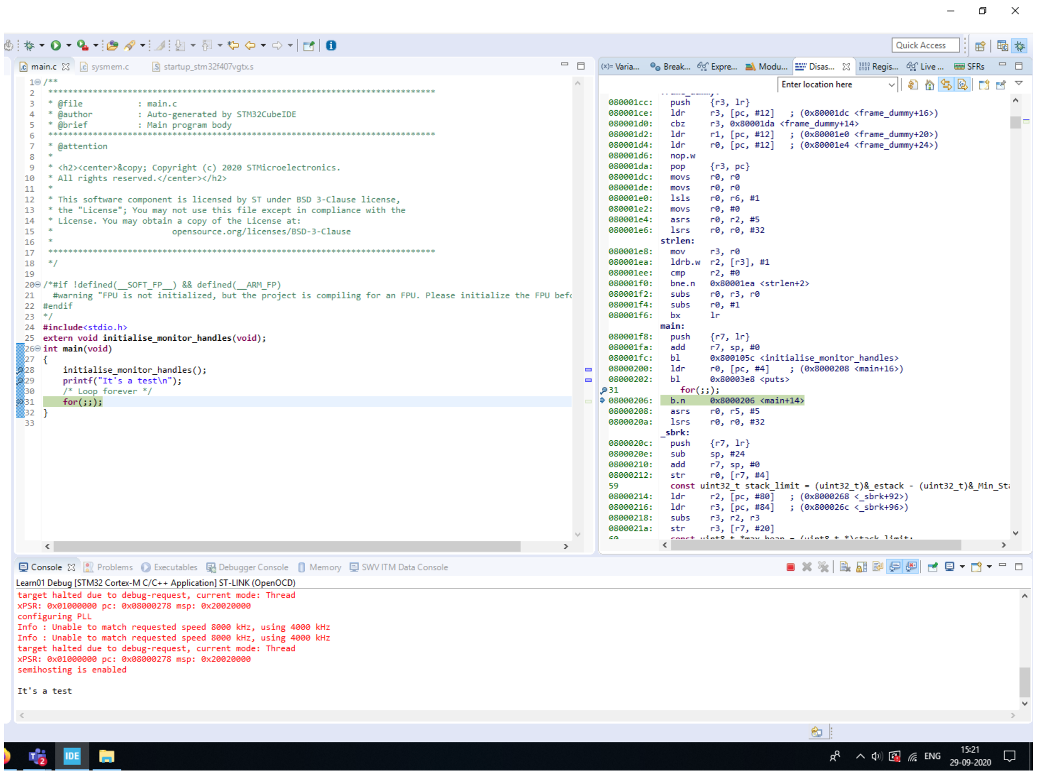
Condition:

Ignore count: 0

```
01HelloProject Debug [STM32 Cortex-M C/C++ Application] ST-LINK (OpenOCD)
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
adapter speed: 4000 kHz
Info : device id = 0x100f6413
Info : flash size = 1024kbytes
semihosting is enabled
Info : SRST line asserted
Info : SRST line released
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000278 msp: 0x20020000, semihosting
configuring PLL
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
adapter speed: 4000 kHz
target halted due to breakpoint, current mode: Thread
xPSR: 0x61000000 pc: 0x20000044 msp: 0x20020000, semihosting
Info : SRST line asserted
Info : SRST line released
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000278 msp: 0x20020000, semihosting
```

Figure 13: Assigning watch-point

Disassembly



```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Auto-generated by STM32CubeIDE
 * @brief          : Main program body
 ******************************************************************************
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *                        opensource.org/licenses/BSD-3-Clause
 *
 ******************************************************************************
 */

/*#if !defined(__SOFT_FP__) && defined(__ARM_FP)
  #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU bef
#endif
*/
#include<stdio.h>
extern void initialise_monitor_handles(void);
int main(void)
{
    initialise_monitor_handles();
    printf("It's a test\n");
    /* Loop forever */
    for(;;);
}
```

```
080001cc:   push    {r3, lr}
080001ce:   ldr     r3, [pc, #12]   ; (0x80001dc <frame_dummy+16>)
080001d0:   cbz     r3, 0x80001da <frame_dummy+14>
080001d2:   ldr     r1, [pc, #12]   ; (0x80001e0 <frame_dummy+20>)
080001d4:   ldr     r0, [pc, #12]   ; (0x80001e4 <frame_dummy+24>)
080001d6:   nop.w
080001da:   pop     {r3, pc}
080001dc:   movs    r0, r0
080001de:   movs    r0, r0
080001e0:   lsls    r0, r6, #1
080001e2:   movs    r0, #0
080001e4:   asrs    r0, r2, #5
080001e6:   lsrs    r0, r0, #32
          strlen:
080001e8:   mov     r3, r0
080001ea:   ldrb.w  r2, [r3], #1
080001ee:   cmp     r2, #0
080001f0:   bne.n   0x80001ea <strlen+2>
080001f2:   subs    r0, r3, r0
080001f4:   subs    r0, #1
080001f6:   bx      lr
          main:
080001f8:   push    {r7, lr}
080001fa:   add     r7, sp, #0
080001fc:   bl      0x800105c <initialise_monitor_handles>
08000200:   ldr     r0, [pc, #4]    ; (0x8000208 <main+16>)
08000202:   bl      0x80003e8 <puts>
31            for(;;);
08000206:   b.n     0x8000206 <main+14>
08000208:   asrs    r0, r5, #5
0800020a:   lsrs    r0, r0, #32
          _sbrk:
0800020c:   push    {r7, lr}
0800020e:   sub     sp, #24
08000210:   add     r7, sp, #0
08000212:   str     r0, [r7, #4]
59          const uint32_t stack_limit = (uint32_t)&_estack - (uint32_t)&_Min_St
08000214:   ldr     r2, [pc, #80]   ; (0x8000268 <_sbrk+92>)
08000216:   ldr     r3, [pc, #84]   ; (0x800026c <_sbrk+96>)
08000218:   subs    r3, r2, r3
0800021a:   str     r3, [r7, #20]
```

Learn01 Debug [STM32 Cortex-M C/C++ Application] ST-LINK (OpenOCD)

```
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000278 msp: 0x20020000
configuring PLL
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
Info : Unable to match requested speed 8000 kHz, using 4000 kHz
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000278 msp: 0x20020000
semihosting is enabled

It's a test
```

Figure 14: Disassembly window

Activity 5: MCU specific header file

Link to repository: Link

Activity 6- Code Quality MISRA C Standards

6.1. Reliability

- Initialize areas and use them by taking their sizes into consideration.
    - Use areas after initializing them.
    - Describe initializations without excess or deficiency
    - Pay attention to the range of the area pointed by a pointer.

- Use data by taking their ranges, sizes and internal representations into consideration.
    - Make comparisons that do not depend on internal representations.
    - When values such as logical values are defined as a range, do not make a judgment by finding whether a value is equivalent to any value (representative value that is implemented) within this range
    - Use the same data type to perform operations or comparisons.
    - Describe code by taking operation precision into consideration.
    - Do not use operations that have the risk of information loss.
    - Use types that can represent the target data.
    - Pay attention to pointer types.
    - Write in a way that will enable the compiler to check that there are no conflicting declarations, usages and definitions.

- Write in a way that ensures intended behavior
    - Prevent operations that may cause runtime error from falling into error cases.
    - Check the interface restrictions when a function is called.
    - Do not perform recursive calls.
    - Pay attention to branch conditions and describe how to handle cases that do not follow the predefined conditions when they occur.
    - Pay attention to the order of evaluation.
    - Be careful with how to access the shared data in programs that use threads or signals.

6.2 Maintainability

- Keep in mind that others will read the program.
    - Do not leave unused descriptions.
    - Do not writing confusingly.
    - Do not write in an unconventional style.
    - Write in a style that clearly specifies the operator precedence.
    - Explicitly describe the operations that are likely to cause misunderstanding when they are omitted.
    - Use one area for one purpose.
    - Do not reuse names.
    - Do not use language specifications that are likely to cause misunderstanding.
    - When writing in an unconventional style, explicitly state its intention.
    - Do not embed magic numbers.
    - Explicitly state the area attributes.
    - Correctly describe the statements even if they are not compiled.

- Write in a style that can prevent modification errors.
    - Clarify the grouping of structured data and blocks.
    - Localize access ranges and related data.

- Write programs simply.
    - Do structured programming.
    - Limit the number of side effects per statement to one.
    - Write expressions that differ in purpose separately.
    - Do not use complicated pointer operations.

- Write in a unified style.
    - Unify the coding styles.
    - Unify the style of writing comments.
    - Unify the naming conventions.
    - Unify the contents to be described in a file and the order of describing them.
    - Unify the style of writing declarations.
    - Unify the style of writing null pointers.
    - Unify the style of writing pre-processor directives.

- Write in a style that makes testing easy.
    - Write in a style that makes it easy to investigate the causes of problems when they occur.
    - Be careful when using dynamic memory allocations.

6.3 Efficiency

- Write in a style that takes account of resource and time efficiencies.

6.4 Portability

- Write in a style that is not dependent on the compiler
    - Do not use functionalities that are advanced features or implementation-defined
    - Use only the characters and escape sequences defined in the language standard.
    - Confirm and document data type representations behavioral specifications of advanced functionalities and implementation- dependent parts.
    - For source file inclusion confirm the implementation dependent parts and write in a style that is not implementation- dependent.
    - Write in a style that does not depend on the environment used for compiling.
- Localize the code that has a problem with portability.