

# GENESIS - Learning Report Embedded c



LTTTS  
GLOBAL  
ENGINEERING  
ACADEMY



*L&T Technology Services*



**Details**

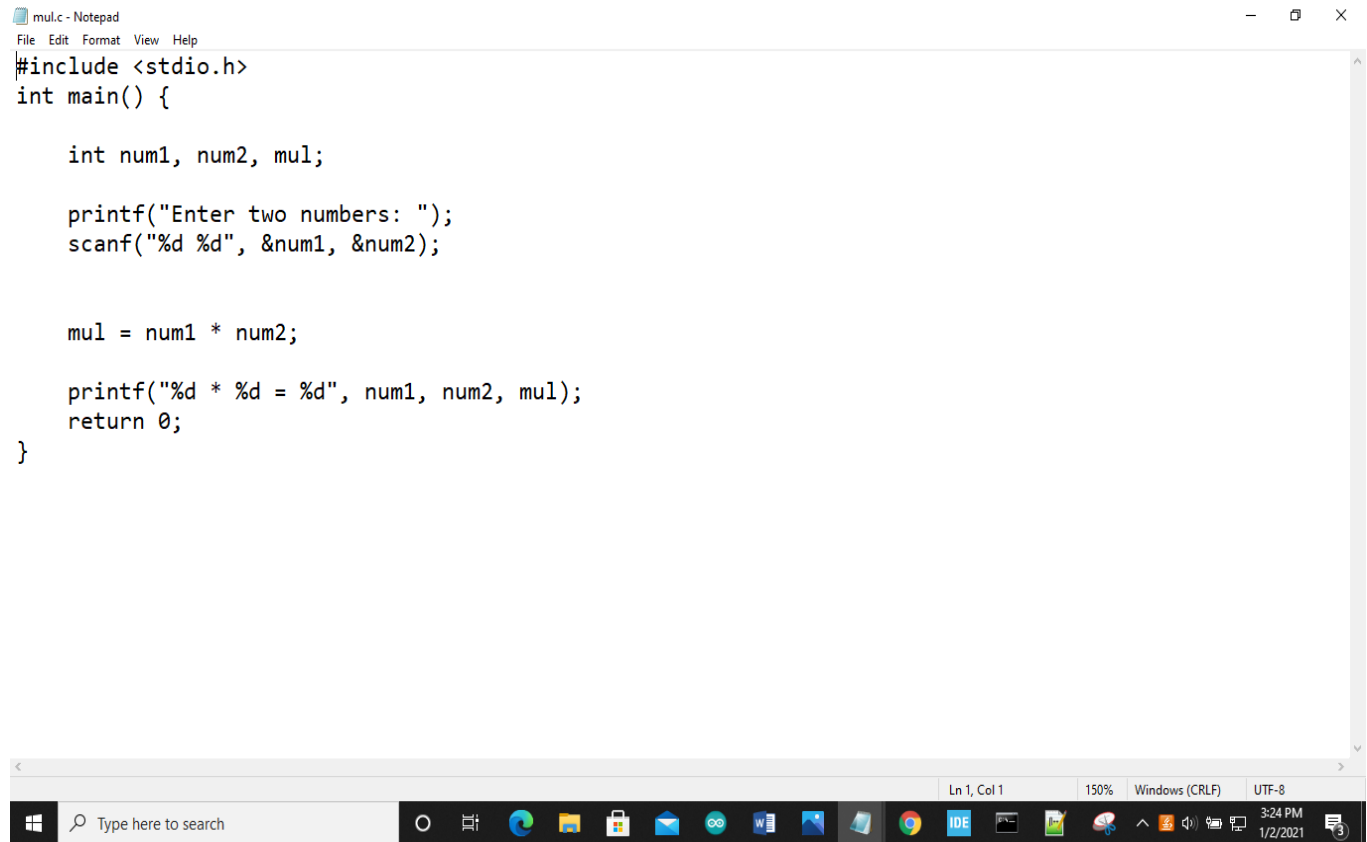
Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1	03/01/2021	Naveen Kumar Pappuru			

## CONTENTS

<b>Activity 1: Build Process</b>	<b>4</b>
<b>Activity 2: Linker Script</b>	<b>6</b>
<b>Activity 3: Startup Code</b>	<b>7</b>
<b>Activity 4: Make file</b>	<b>8</b>
<b>Mini Project Main logic function</b>	<b>9</b>

**Activity-1:**

Build System which is able to create the final executable and compiling it in make file using command prompt of window.



```
mul.c - Notepad
File Edit Format View Help
#include <stdio.h>
int main() {

    int num1, num2, mul;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    mul = num1 * num2;

    printf("%d * %d = %d", num1, num2, mul);
    return 0;
}
```

The screenshot shows a Windows Notepad window titled 'mul.c - Notepad'. The window contains a C program that prompts the user to enter two numbers, calculates their product, and displays the result. The code is as follows:

```
#include <stdio.h>
int main() {

    int num1, num2, mul;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    mul = num1 * num2;

    printf("%d * %d = %d", num1, num2, mul);
    return 0;
}
```

The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several application icons including Edge, File Explorer, Mail, Teams, Word, and IDE. The system tray on the right shows the time as 3:24 PM on 1/2/2021.

**Make file for build process:**

```
Command Prompt
C:\Users\99003181\Desktop\neven>dir
Volume in drive C is Windows
Volume Serial Number is 1E01-2866

Directory of C:\Users\99003181\Desktop\neven

01/02/2021  03:21 PM  <DIR>          .
01/02/2021  03:21 PM  <DIR>          ..
01/02/2021  03:17 PM                248 mul.c
12/26/2020  11:48 AM                 54 mul.h
01/02/2021  03:21 PM            1,160 mul.o
               3 File(s)            1,462 bytes
               2 Dir(s)  162,328,248,320 bytes free

C:\Users\99003181\Desktop\neven>arm-none-eabi-gcc -S mul.c -o mul.s

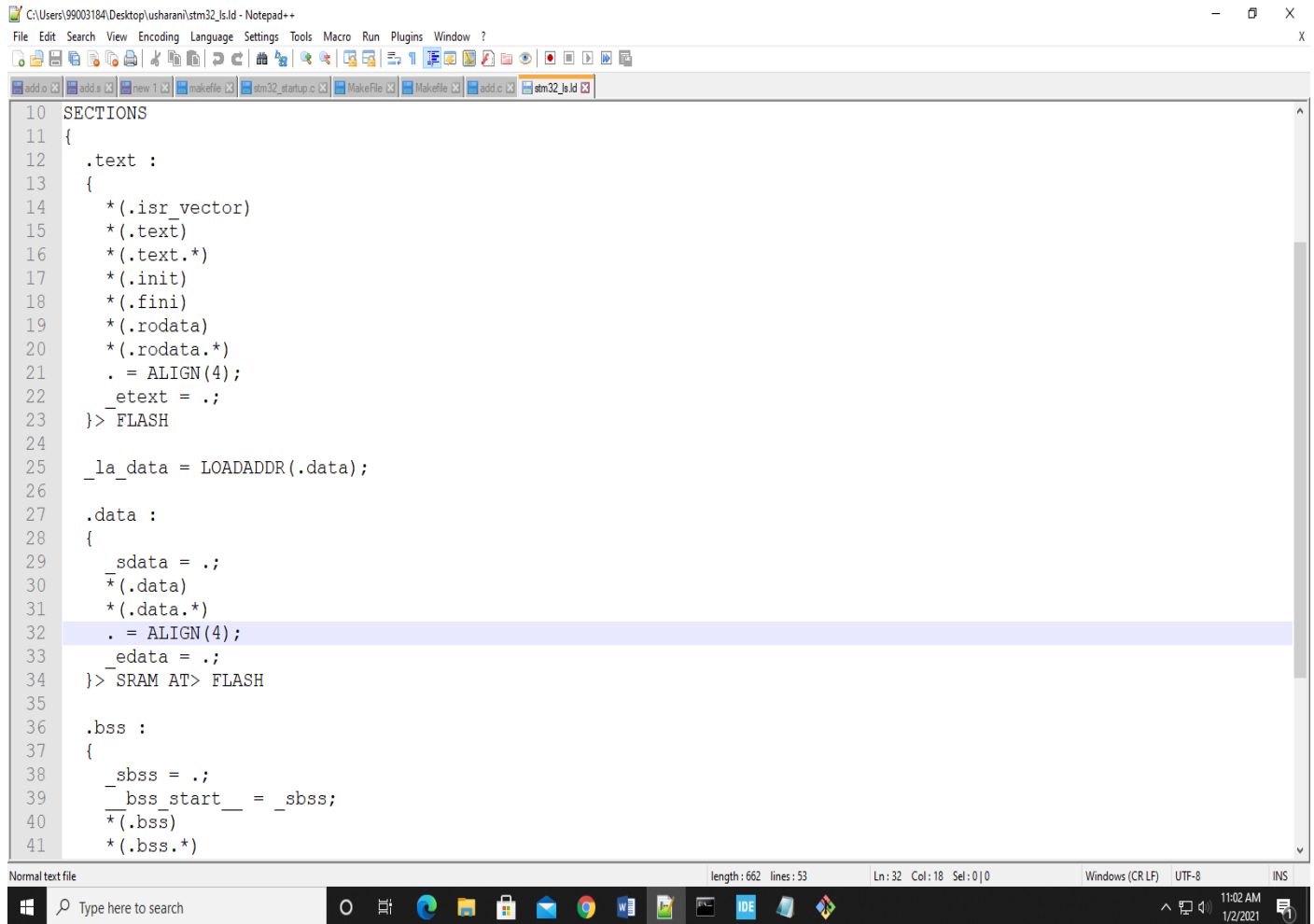
C:\Users\99003181\Desktop\neven>dir
Volume in drive C is Windows
Volume Serial Number is 1E01-2866

Directory of C:\Users\99003181\Desktop\neven

01/02/2021  03:23 PM  <DIR>          .
01/02/2021  03:23 PM  <DIR>          ..
01/02/2021  03:23 PM            140 Makefile
01/02/2021  03:17 PM                248 mul.c
12/26/2020  11:48 AM                 54 mul.h
01/02/2021  03:21 PM            1,160 mul.o
01/02/2021  03:23 PM            1,226 mul.s
               5 File(s)            2,828 bytes
               2 Dir(s)  162,327,080,960 bytes free

C:\Users\99003181\Desktop\neven>make
make: `mul.o' is up to date.

C:\Users\99003181\Desktop\neven>
```

**Activity-2:****Linker Script:**

```
10 SECTIONS
11 {
12     .text :
13     {
14         *(.isr_vector)
15         *(.text)
16         *(.text.*)
17         *(.init)
18         *(.fini)
19         *(.rodata)
20         *(.rodata.*)
21         . = ALIGN(4);
22         _etext = .;
23     }> FLASH
24
25     _la_data = LOADADDR(.data);
26
27     .data :
28     {
29         _sdata = .;
30         *(.data)
31         *(.data.*)
32         . = ALIGN(4);
33         _edata = .;
34     }> SRAM AT> FLASH
35
36     .bss :
37     {
38         _sbss = .;
39         __bss_start__ = _sbss;
40         *(.bss)
41         *(.bss.*)
```

### Activity-3:

### Start-up Code:

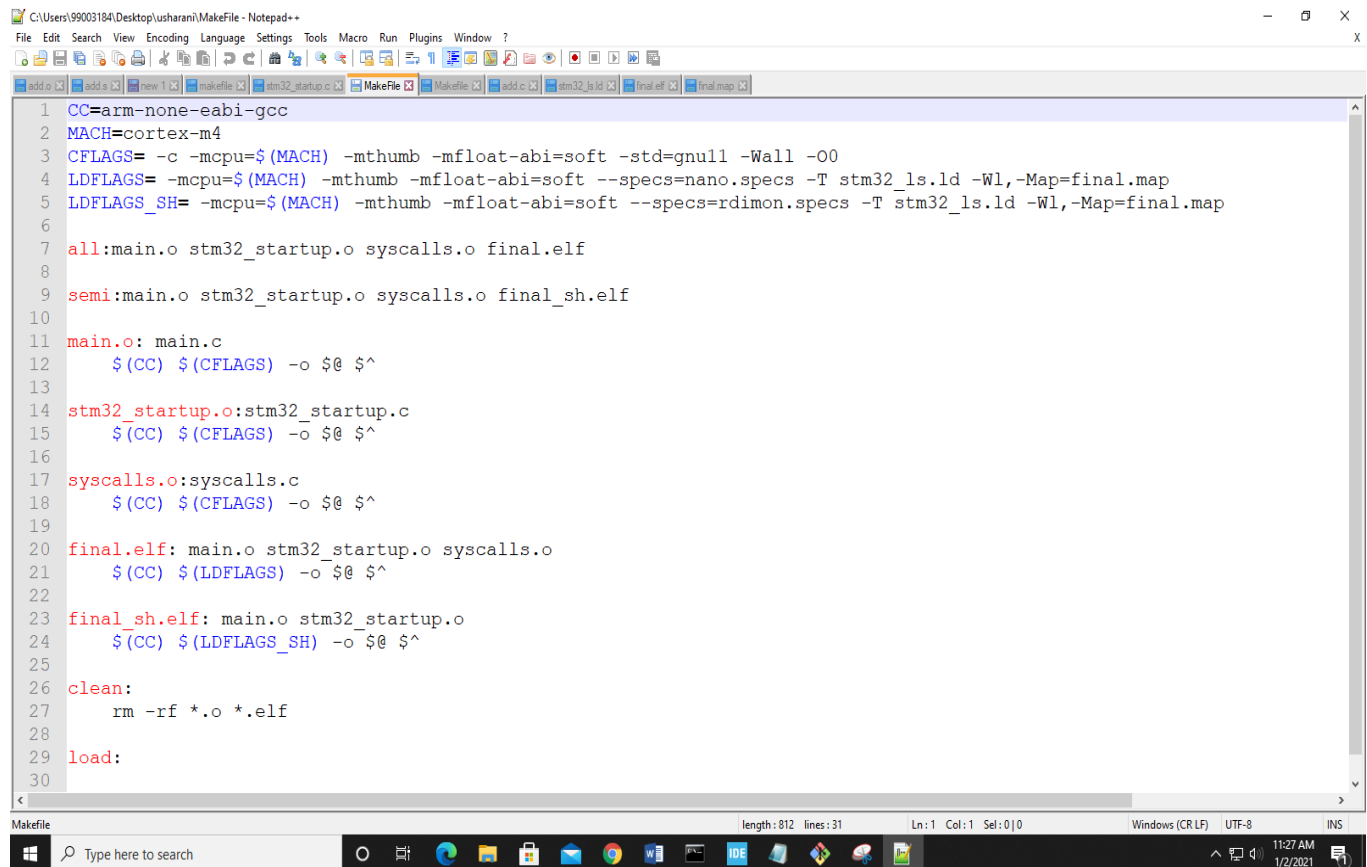
### Various Address of Interrupt Subroutine Vectors (ISR) and its priorities of ISR's:

```
stm32_startup - Notepad
File Edit Format View Help

void DMA2_Stream7_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void USART6_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void I2C3_EV_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void I2C3_ER_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void OTG_HS_EP1_OUT_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void OTG_HS_EP1_IN_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void OTG_HS_WKUP_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void OTG_HS_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void DCMI_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void CRYP_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void HASH_RNG_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));
void FPU_IRQHandler (void) __attribute__((weak, alias("Default_Handler")));

uint32_t vectors[] __attribute__((section(".isr_vector"))) = {
    STACK_START,
    (uint32_t)Reset_Handler,
    (uint32_t)NMI_Handler,
    (uint32_t)HardFault_Handler,
    (uint32_t)MemManage_Handler,
    (uint32_t)BusFault_Handler,
    (uint32_t)UsageFault_Handler,
    0,
    0,
    0,
    0,
    (uint32_t)SVC_Handler,
    (uint32_t)DebugMon_Handler,
    0,
    (uint32_t)PendSV_Handler,
    (uint32_t)SysTick_Handler,
    (uint32_t)WWDG_IRQHandler,
    (uint32_t)PVD_IRQHandler,
    (uint32_t)TAMP_STAMP_IRQHandler,
    (uint32_t)RTC_WKUP_IRQHandler,
    -
}

Ln 3, Col 32 120% Windows (CRLF) UTF-8 11:24 AM 1/2/2021
```

**Activity-4:****Make file:**

```
1 CC=arm-none-eabi-gcc
2 MACH=cortex-m4
3 CFLAGS= -c -mcpu=$(MACH) -mthumb -mfloat-abi=soft -std=gnu11 -Wall -O0
4 LDFLAGS= -mcpu=$(MACH) -mthumb -mfloat-abi=soft --specs=nano.specs -T stm32_ls.ld -Wl,-Map=final.map
5 LDFLAGS_SH= -mcpu=$(MACH) -mthumb -mfloat-abi=soft --specs=rdimon.specs -T stm32_ls.ld -Wl,-Map=final.map
6
7 all:main.o stm32_startup.o syscalls.o final.elf
8
9 semi:main.o stm32_startup.o syscalls.o final_sh.elf
10
11 main.o: main.c
12     $(CC) $(CFLAGS) -o $@ $^
13
14 stm32_startup.o:stm32_startup.c
15     $(CC) $(CFLAGS) -o $@ $^
16
17 syscalls.o:syscalls.c
18     $(CC) $(CFLAGS) -o $@ $^
19
20 final.elf: main.o stm32_startup.o syscalls.o
21     $(CC) $(LDFLAGS) -o $@ $^
22
23 final_sh.elf: main.o stm32_startup.o
24     $(CC) $(LDFLAGS_SH) -o $@ $^
25
26 clean:
27     rm -rf *.o *.elf
28
29 load:
30
```



## Mini Project

### Main logic function:

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    buffer_tx = HAL_GPIO_ReadPin(PIR_Read_Pin_GPIO_Port, PIR_Read_Pin_Pin);
    if(buffer_tx == 1)
    {
        HAL_GPIO_WritePin(Orange_led_GPIO_Port, Orange_led_Pin, 1);
    }
    else
    {
        HAL_GPIO_WritePin(Orange_led_GPIO_Port, Orange_led_Pin, 0);
    }
    HAL_SPI_Transmit(&hspi1, &buffer_tx, 1, 100);
}
/* USER CODE END 3 */
}
```

/\*\*

