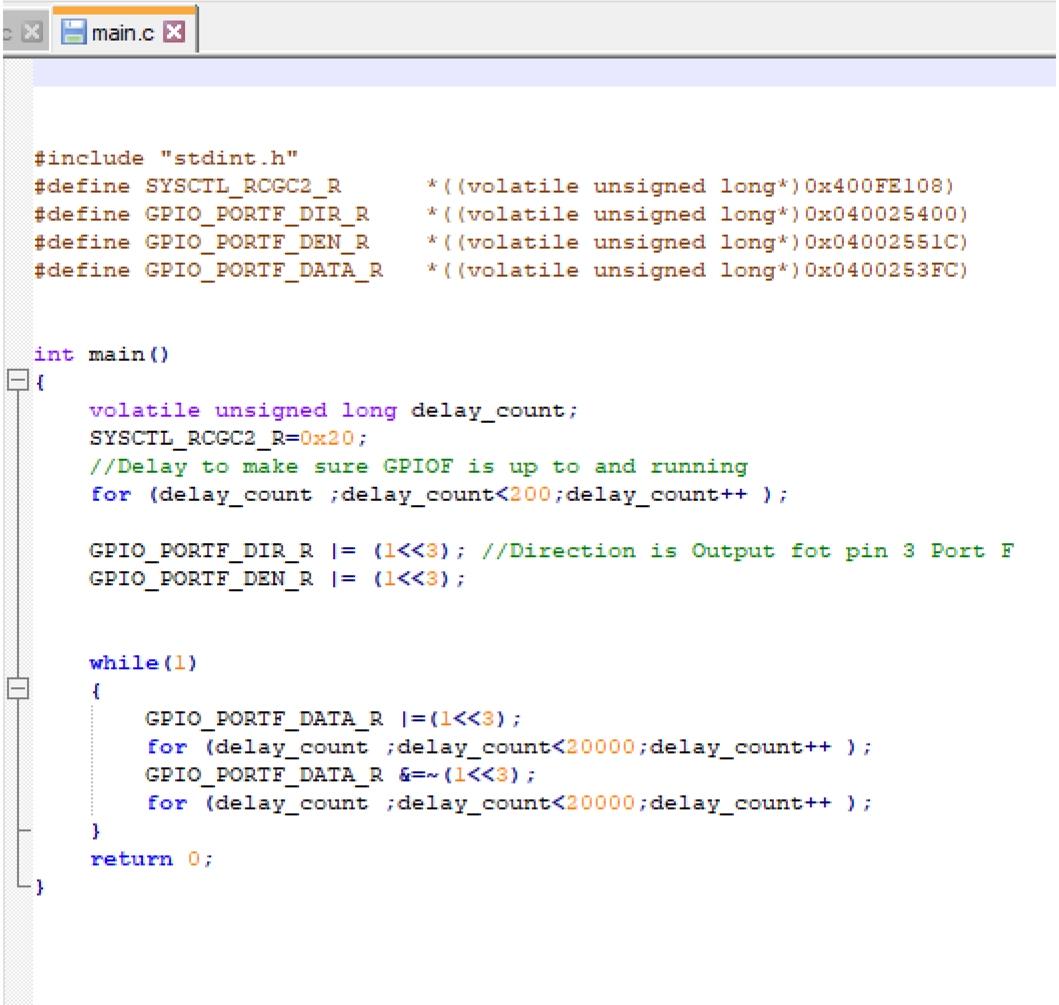
Unit 3 Lesson 4

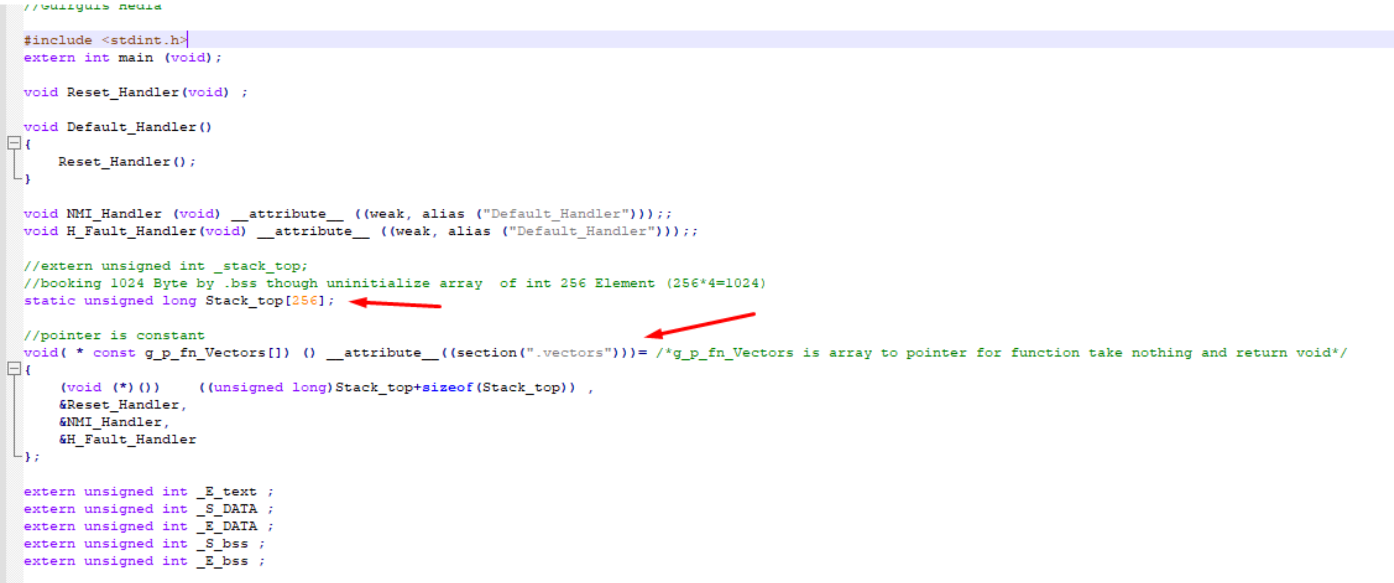
Guirguis Hedia

- main.c File

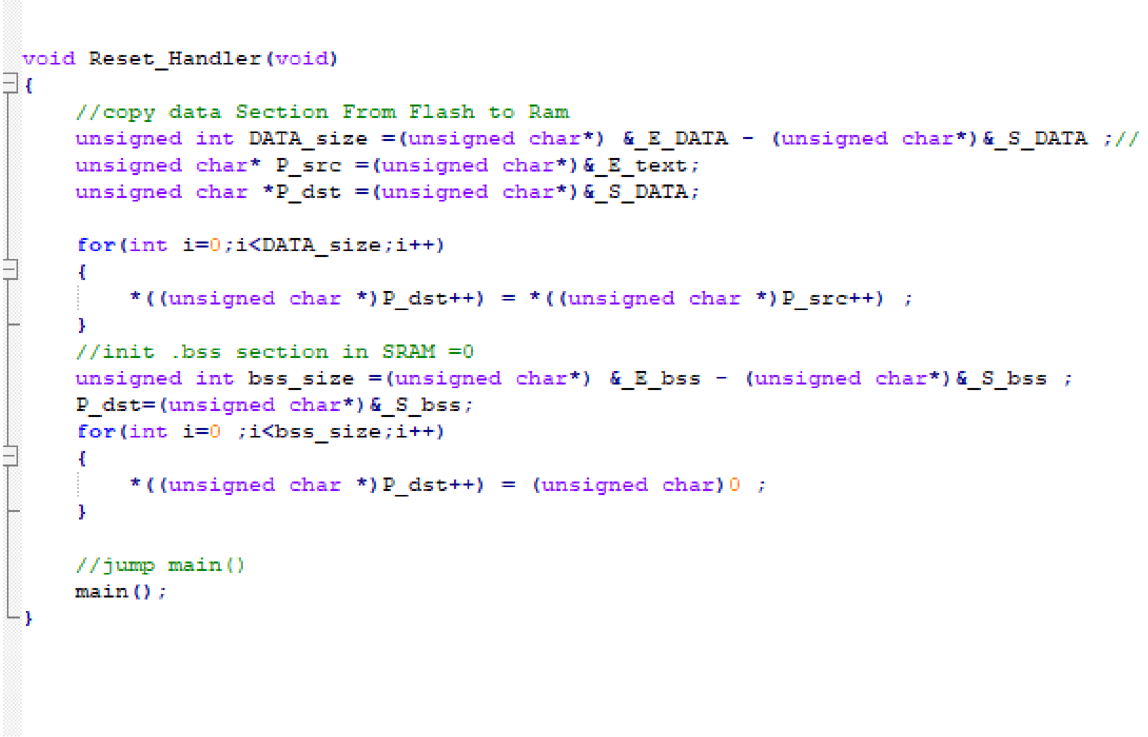


```
#include "stdint.h"
#define SYSCTL_RCGC2_R      *((volatile unsigned long*)0x400FE108)
#define GPIO_PORTF_DIR_R    *((volatile unsigned long*)0x040025400)
#define GPIO_PORTF_DEN_R    *((volatile unsigned long*)0x04002551C)
#define GPIO_PORTF_DATA_R   *((volatile unsigned long*)0x0400253FC)


int main()
{
    volatile unsigned long delay_count;
    SYSCTL_RCGC2_R=0x20;
    //Delay to make sure GPIOF is up to and running
    for (delay_count ;delay_count<200;delay_count++ );

    GPIO_PORTF_DIR_R |= (1<<3); //Direction is Output fot pin 3 Port F
    GPIO_PORTF_DEN_R |= (1<<3);


    while(1)
    {
        GPIO_PORTF_DATA_R |=(1<<3);
        for (delay_count ;delay_count<20000;delay_count++ );
        GPIO_PORTF_DATA_R &=~(1<<3);
        for (delay_count ;delay_count<20000;delay_count++ );
    }
    return 0;
}
```

- Startup File

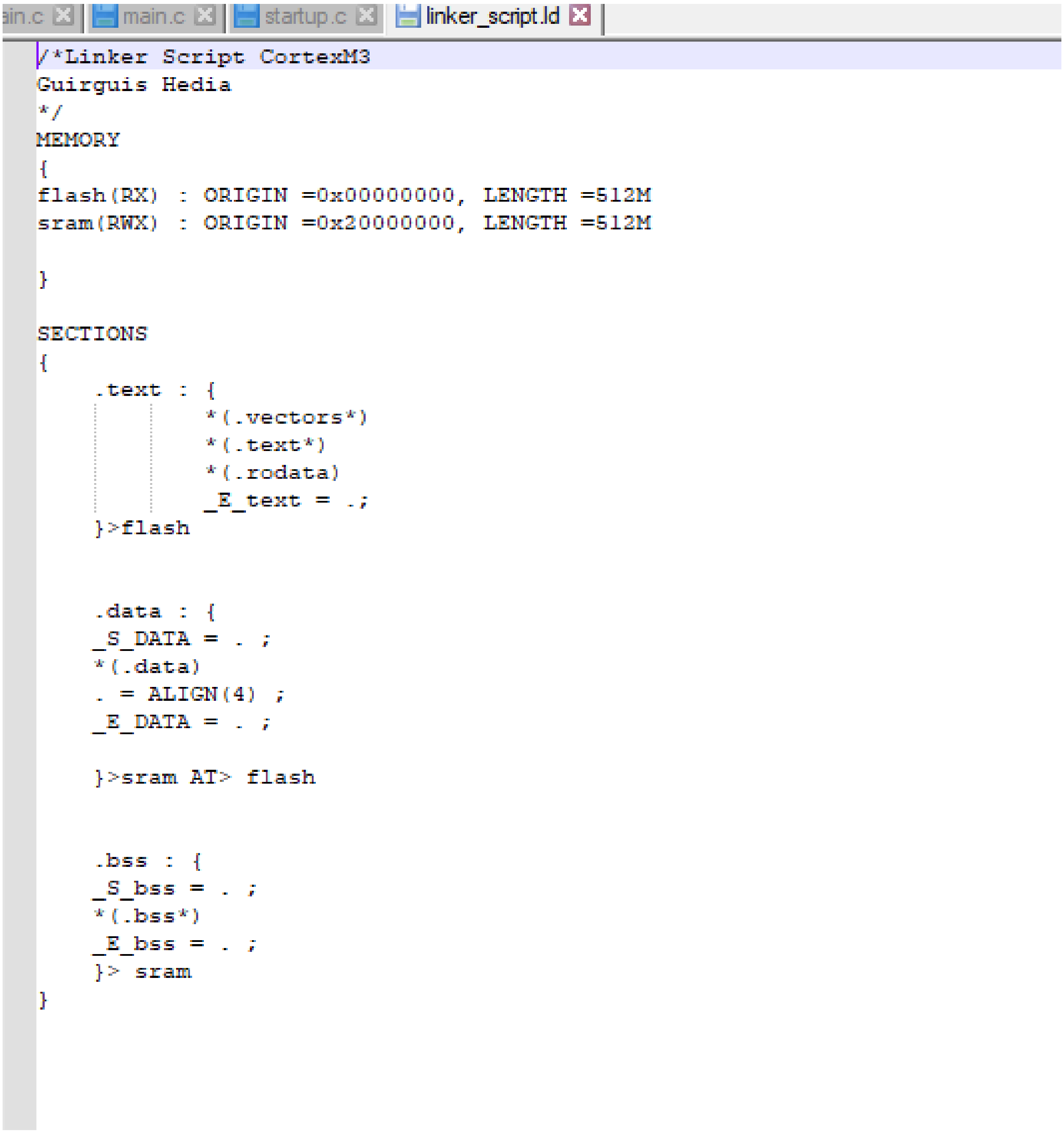


```
//Guirguis Media

#include <stdint.h>
extern int main (void);

void Reset_Handler(void) ;

void Default_Handler()
{
    Reset_Handler();
}

void NMI_Handler (void) __attribute__ ((weak, alias ("Default_Handler")));;
void H_Fault_Handler(void) __attribute__ ((weak, alias ("Default_Handler")));;

//extern unsigned int _stack_top;
//booking 1024 Byte by .bss though uninitialize array  of int 256 Element (256*4=1024)
static unsigned long Stack_top[256];

//pointer is constant
void( * const g_p_fn_Vectors[]) () __attribute__((section(".vectors")))= /*g_p_fn_Vectors is array to pointer for function take nothing and return void*/
{
    (void (*)())    ((unsigned long)Stack_top+sizeof(Stack_top)) ,
    &Reset_Handler,
    &NMI_Handler,
    &H_Fault_Handler
};

extern unsigned int _E_text ;
extern unsigned int _S_DATA ;
extern unsigned int _E_DATA ;
extern unsigned int _S_bss ;
extern unsigned int _E_bss ;
```



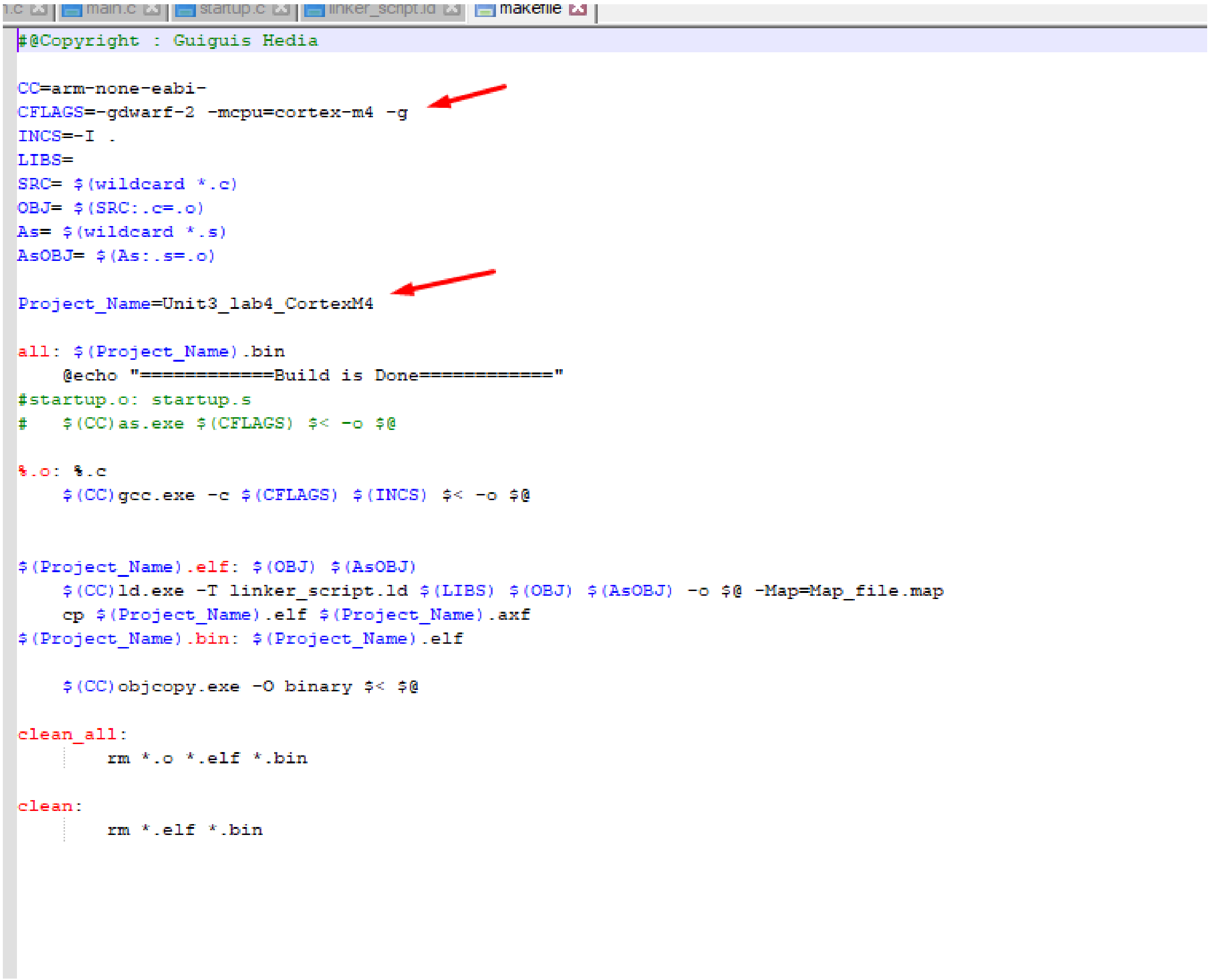
```
void Reset_Handler(void)
{
    //copy data Section From Flash to Ram
    unsigned int DATA_size =(unsigned char*) &_E_DATA - (unsigned char*)&_S_DATA ;//
    unsigned char* P_src =(unsigned char*)&_E_text;
    unsigned char *P_dst =(unsigned char*)&_S_DATA;

    for(int i=0;i<DATA_size;i++)
    {
        *((unsigned char *)P_dst++) = *((unsigned char *)P_src++) ;
    }
    //init .bss section in SRAM =0
    unsigned int bss_size =(unsigned char*) &_E_bss - (unsigned char*)&_S_bss ;
    P_dst=(unsigned char*)&_S_bss;
    for(int i=0 ;i<bss_size;i++)
    {
        *((unsigned char *)P_dst++) = (unsigned char)0 ;
    }

    //jump main()
    main();
}
```

- LinkerScript File



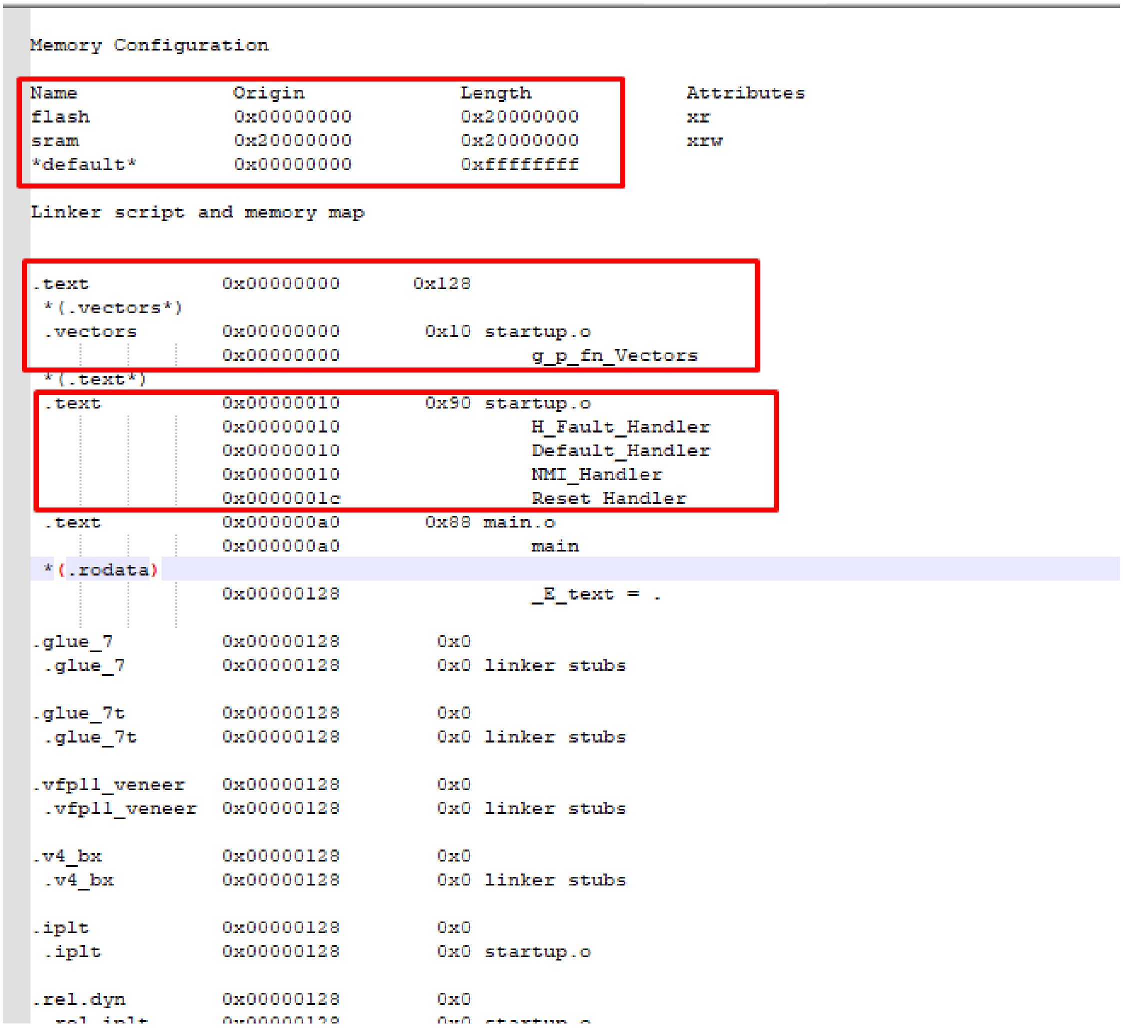
```
/*Linker Script CortexM3
Guirguis Hedia
*/
MEMORY
{
flash(RX) : ORIGIN =0x00000000, LENGTH =512M
sram(RWX) : ORIGIN =0x20000000, LENGTH =512M

}

SECTIONS
{
    .text : {
            *(.vectors*)
            *(.text*)
            *(.rodata)
            _E_text = .;
    }>flash


    .data : {
    _S_DATA = . ;
    *(.data)
    . = ALIGN(4) ;
    _E_DATA = . ;

    }>sram AT> flash


    .bss : {
    _S_bss = . ;
    *(.bss*)
    _E_bss = . ;
    }> sram
}
```

- MakeFile



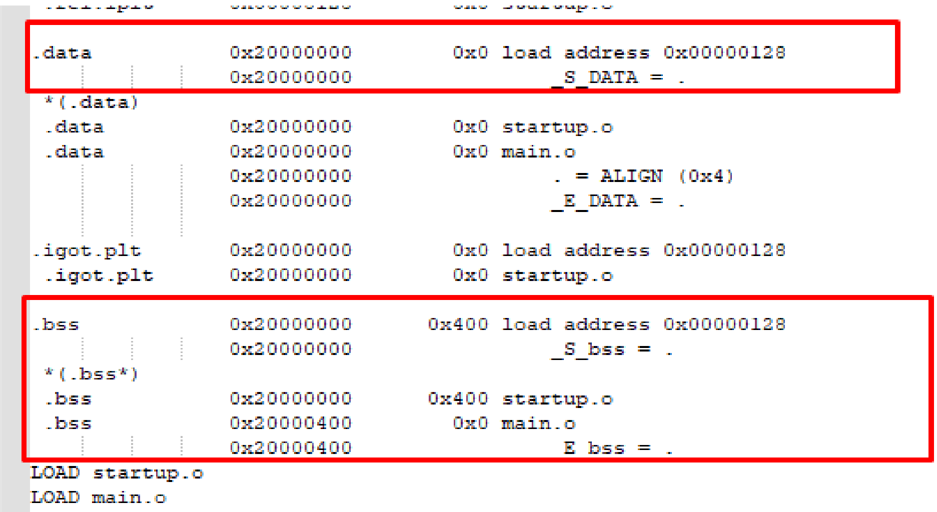
```
#@Copyright : Guiguis Hedia

CC=arm-none-eabi-
CFLAGS=-gdwarf-2 -mcpu=cortex-m4 -g
INCS=-I .
LIBS=
SRC= $(wildcard *.c)
OBJ= $(SRC:.c=.o)
As= $(wildcard *.s)
AsOBJ= $(As:.s=.o)

Project_Name=Unit3_lab4_CortexM4

all: $(Project_Name).bin
    @echo "============Build is Done============"
#startup.o: startup.s
#   $(CC)as.exe $(CFLAGS) $< -o $@

%.o: %.c
    $(CC)gcc.exe -c $(CFLAGS) $(INCS) $< -o $@


$(Project_Name).elf: $(OBJ) $(AsOBJ)
    $(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(AsOBJ) -o $@ -Map=Map_file.map
    cp $(Project_Name).elf $(Project_Name).axf
$(Project_Name).bin: $(Project_Name).elf

    $(CC)objcopy.exe -O binary $< $@

clean_all:
        rm *.o *.elf *.bin

clean:
        rm *.elf *.bin
```

- MapFile



```
Memory Configuration

Name             Origin             Length             Attributes
flash            0x00000000         0x20000000         xr
sram             0x20000000         0x20000000         xrw
*default*        0x00000000         0xffffffff

Linker script and memory map


.text           0x00000000       0x128
 *(.vectors*)
 .vectors       0x00000000        0x10 startup.o
                0x00000000                 g_p_fn_Vectors
 *(.text*)
 .text          0x00000010        0x90 startup.o
                0x00000010                 H_Fault_Handler
                0x00000010                 Default_Handler
                0x00000010                 NMI_Handler
                0x0000001c                 Reset Handler
 .text          0x000000a0        0x88 main.o
                0x000000a0                 main
 *(.rodata)
                0x00000128                 _E_text = .

.glue_7         0x00000128         0x0
 .glue_7        0x00000128         0x0 linker stubs

.glue_7t        0x00000128         0x0
 .glue_7t       0x00000128         0x0 linker stubs

.vfp11_veneer   0x00000128         0x0
 .vfp11_veneer  0x00000128         0x0 linker stubs

.v4_bx          0x00000128         0x0
 .v4_bx         0x00000128         0x0 linker stubs

.iplt           0x00000128         0x0
 .iplt          0x00000128         0x0 startup.o

.rel.dyn        0x00000128         0x0
 .rel.iplt      0x00000128         0x0 startup.o
```

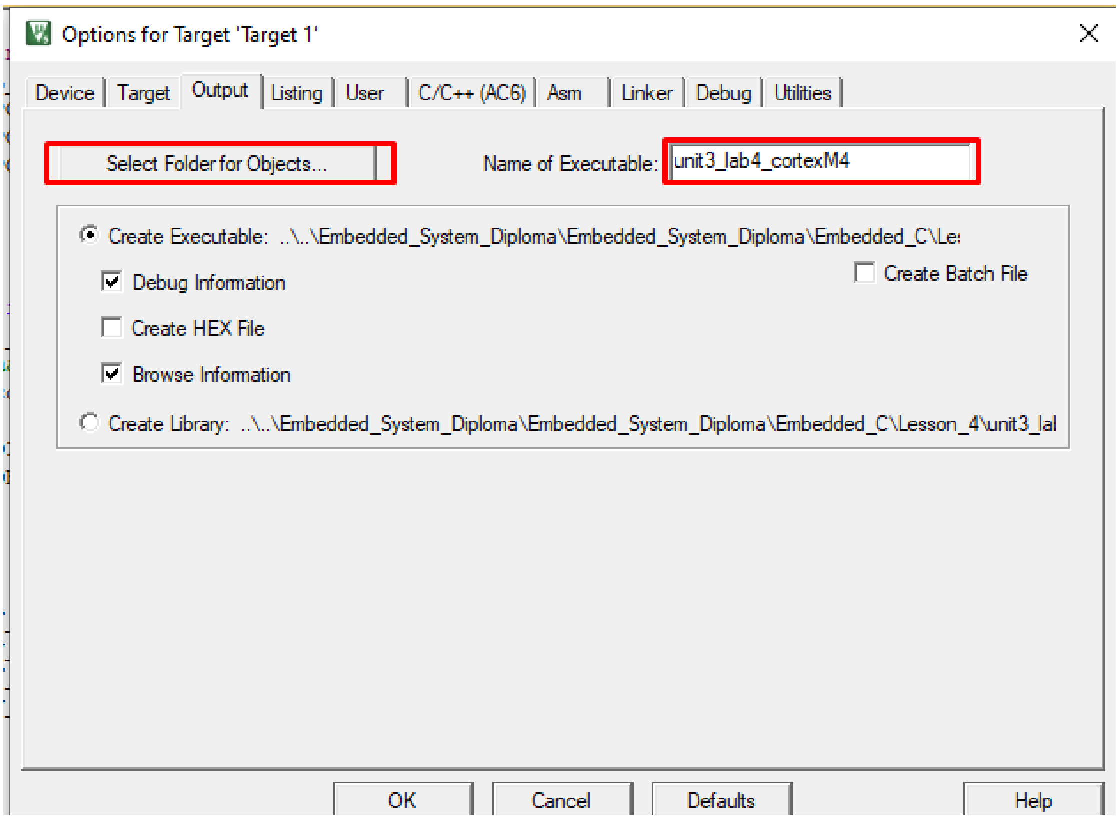
- MapFile Part2



```
 .rel.iplt      0x00000128         0x0 startup.o

.data           0x20000000         0x0 load address 0x00000128
                0x20000000                 _S_DATA = .
 *(.data)
 .data          0x20000000         0x0 startup.o
 .data          0x20000000         0x0 main.o
                0x20000000                 . = ALIGN (0x4)
                0x20000000                 _E_DATA = .

.igot.plt       0x20000000         0x0 load address 0x00000128
 .igot.plt      0x20000000         0x0 startup.o

.bss            0x20000000       0x400 load address 0x00000128
                0x20000000                 _S_bss = .
 *(.bss*)
 .bss           0x20000000       0x400 startup.o
 .bss           0x20000400         0x0 main.o
                0x20000400                 E bss = .
LOAD startup.o
LOAD main.o
```

Simulation Steps in Keil_uvisionil :

- Choose The Folder Which Has The file You Want To Simulated and Write The Executable file Name , and Choose Use Simulator



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities

Select Folder for Objects...

Name of Executable: unit3_lab4_cortexM4

● Create Executable: ..\..\Embedded_System_Diploma\Embedded_System_Diploma\Embedded_C\Le:

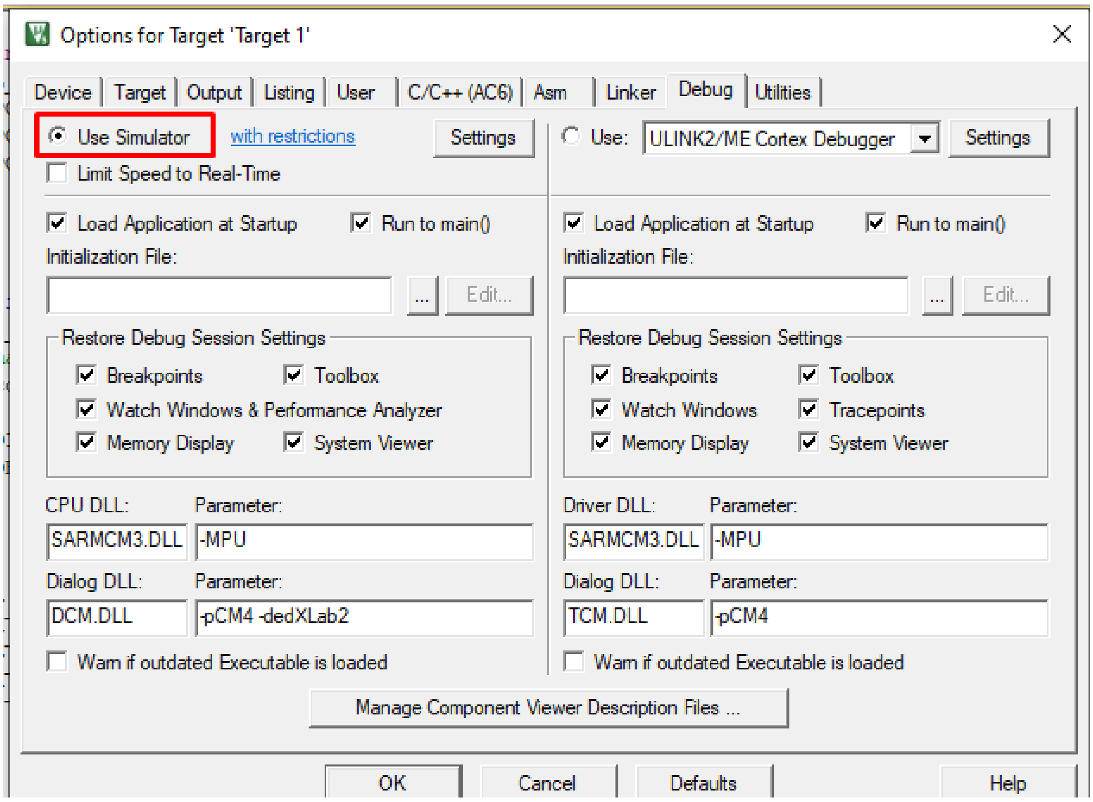
☑ Debug Information

☐ Create Batch File

☐ Create HEX File

☑ Browse Information

○ Create Library: ..\..\Embedded_System_Diploma\Embedded_System_Diploma\Embedded_C\Lesson_4\unit3_lal



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities

● Use Simulator with restrictions Settings

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4 -dedXLab2

☐ Warn if outdated Executable is loaded

○ Use: ULINK2/ME Cortex Debugger Settings

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows ☑ Tracepoints

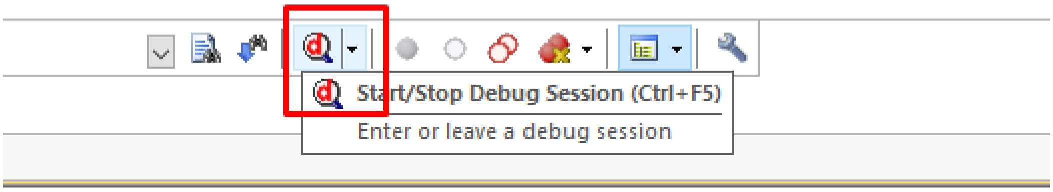
☑ Memory Display ☑ System Viewer

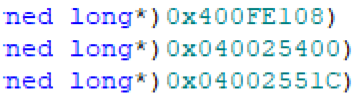
Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

☐ Warn if outdated Executable is loaded

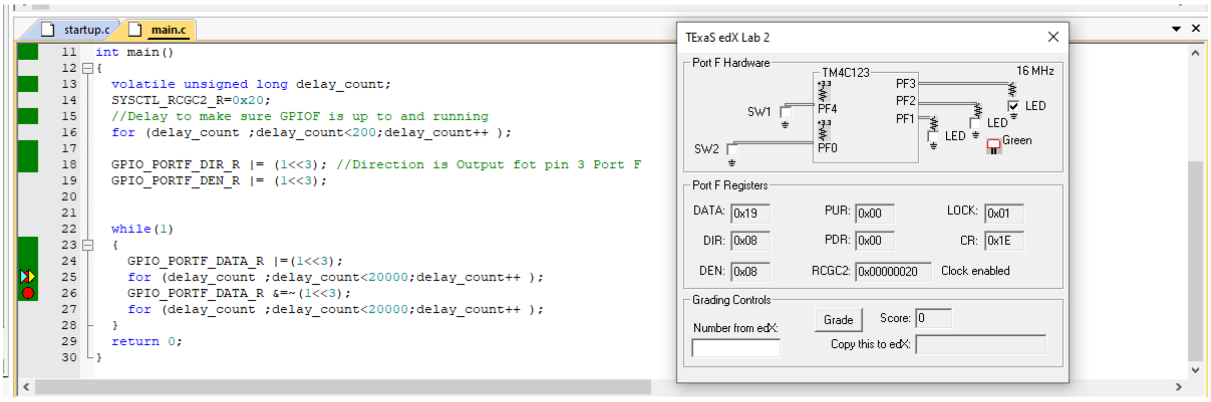
- Start Debugging


Start/Stop Debug Session (Ctrl+F5)
Enter or leave a debug session



```
ned long*)0x400FE108)
ned long*)0x040025400)
ned long*)0x04002551C)
```

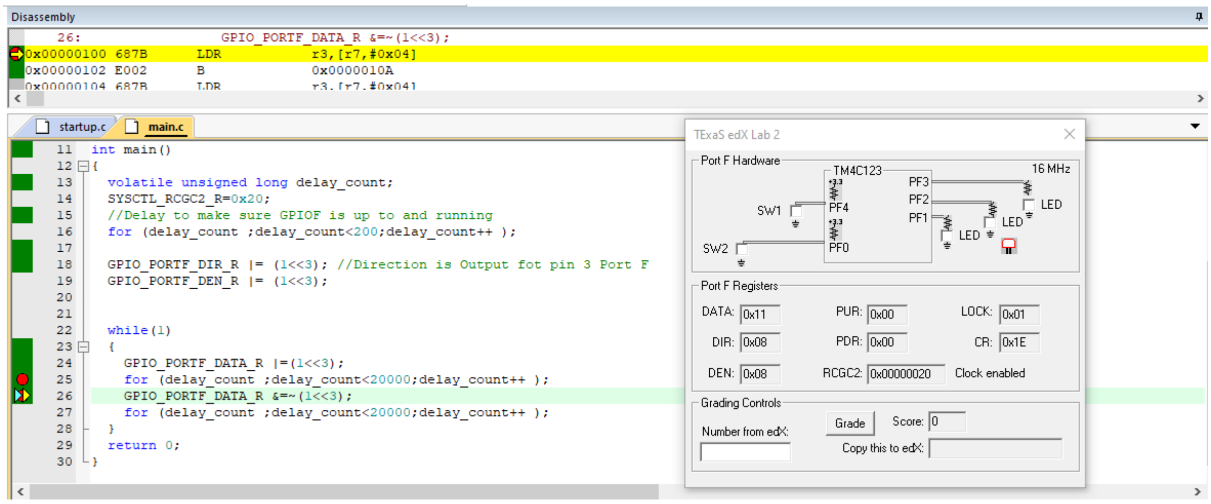
- Check of The output



```
int main()
{
  volatile unsigned long delay_count;
  SYSCTL_RCGC2_R=0x20;
  //Delay to make sure GPIOF is up to and running
  for (delay_count ;delay_count<200;delay_count++ );

  GPIO_PORTF_DIR_R |= (1<<3); //Direction is Output fot pin 3 Port F
  GPIO_PORTF_DEN_R |= (1<<3);


  while(1)
  {
    GPIO_PORTF_DATA_R |=(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
    GPIO_PORTF_DATA_R &=~(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
  }
  return 0;
}
```

DATA: 0x19 PUR: 0x00 LOCK: 0x01
DIR: 0x08 PDR: 0x00 CR: 0x1E
DEN: 0x08 RCGC2: 0x00000020 Clock enabled

Number from edX: Score: 0
Copy this to edX:



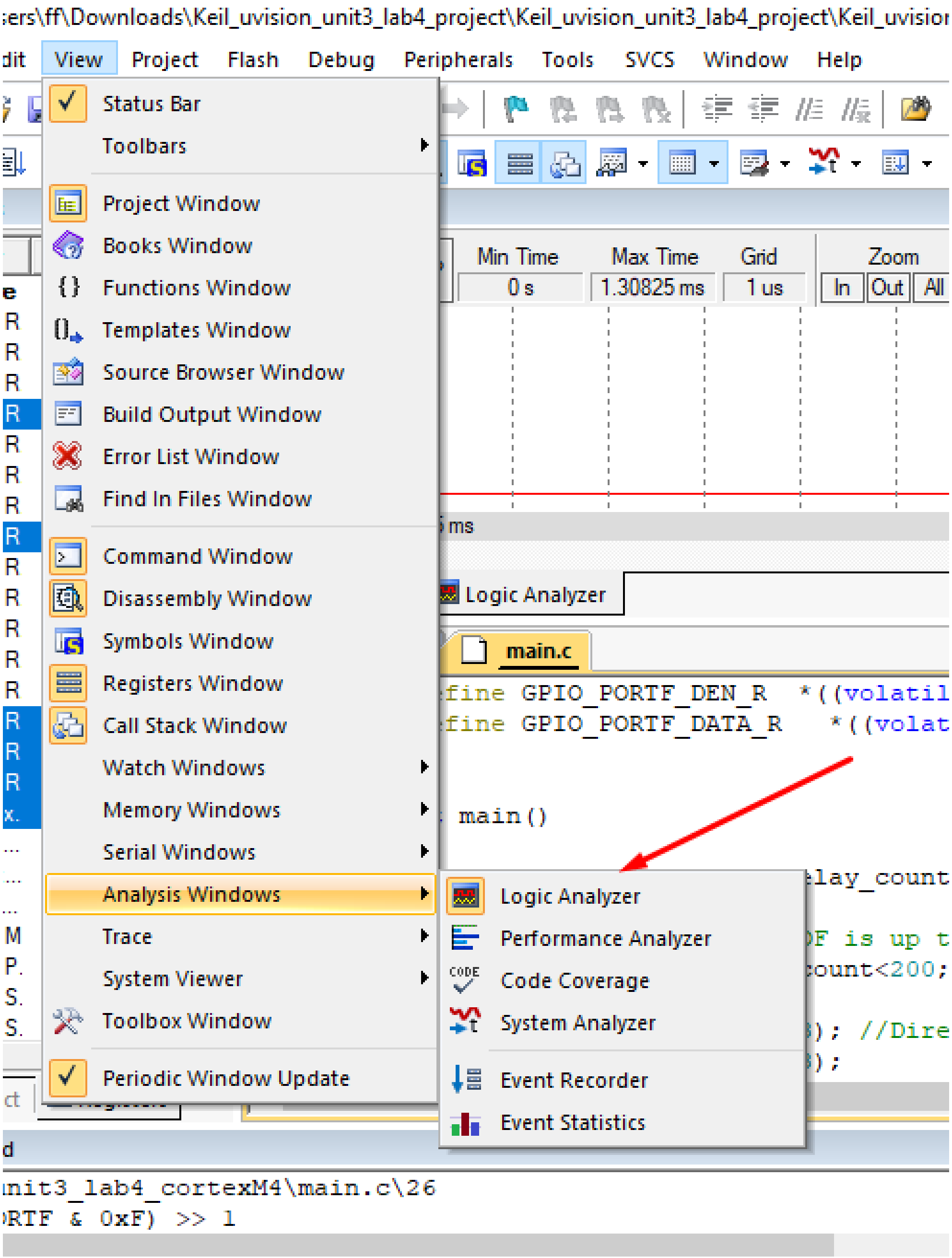
```
    26:              GPIO_PORTF_DATA_R &=~(1<<3);
0x00000100 687B      LDR           r3,[r7,#0x04]
0x00000102 E002      B             0x0000010A
0x00000104 687B      LDR           r3,[r7,#0x04]
```

```
int main()
{
  volatile unsigned long delay_count;
  SYSCTL_RCGC2_R=0x20;
  //Delay to make sure GPIOF is up to and running
  for (delay_count ;delay_count<200;delay_count++ );

  GPIO_PORTF_DIR_R |= (1<<3); //Direction is Output fot pin 3 Port F
  GPIO_PORTF_DEN_R |= (1<<3);


  while(1)
  {
    GPIO_PORTF_DATA_R |=(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
    GPIO_PORTF_DATA_R &=~(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
  }
  return 0;
}
```

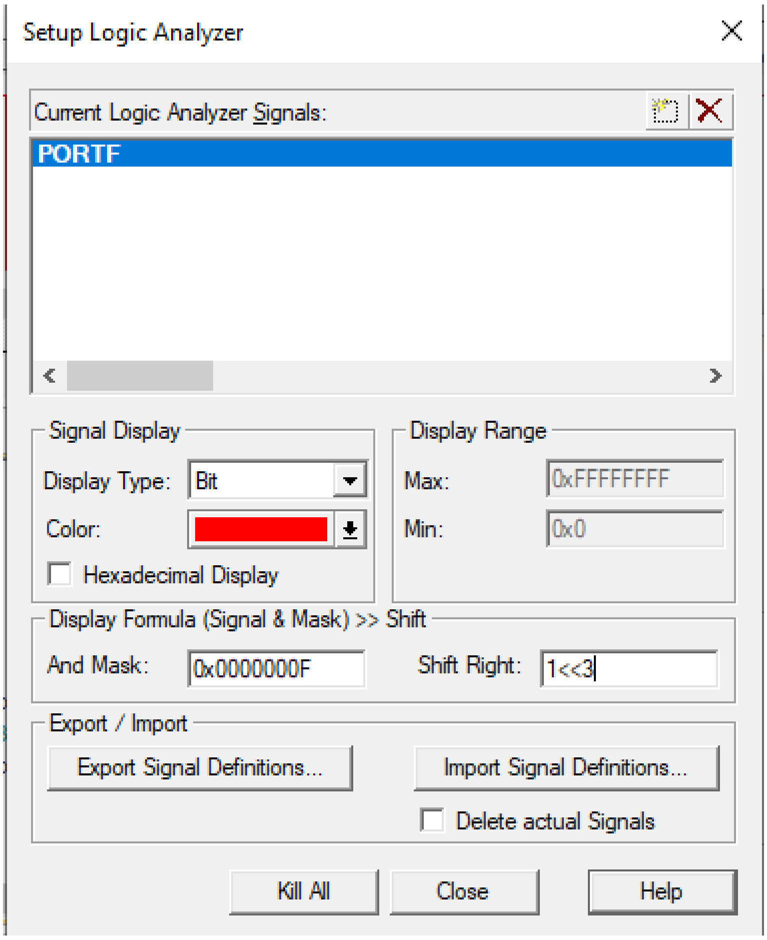
DATA: 0x11 PUR: 0x00 LOCK: 0x01
DIR: 0x08 PDR: 0x00 CR: 0x1E
DEN: 0x08 RCGC2: 0x00000020 Clock enabled

Number from edX: Score: 0
Copy this to edX:

- Use Logical Analyzer



- Set Configuration For Logical Analyzer



Setup Logic Analyzer

Current Logic Analyzer Signals:

PORTF

Signal Display

Display Type: Bit

Color:

□ Hexadecimal Display

Display Range

Max: 0xFFFFFFFF

Min: 0x0

Display Formula (Signal & Mask) >> Shift

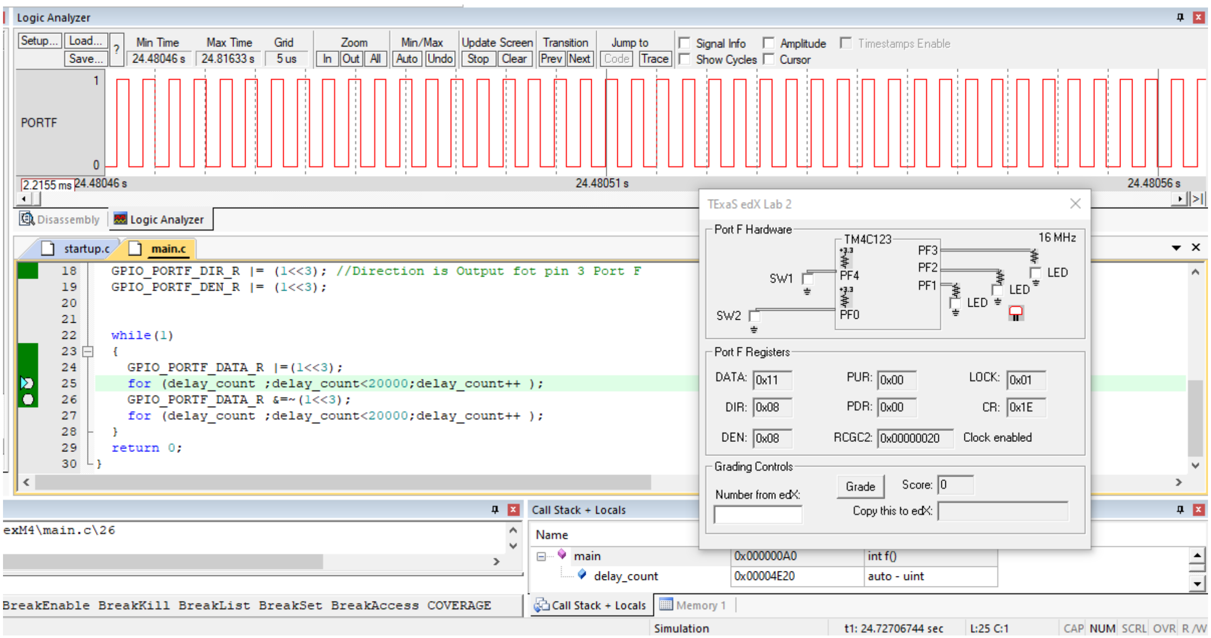
And Mask: 0x0000000F

Shift Right: 1<<3

Export / Import

□ Delete actual Signals

- Output of The Logical Analyzer



Logic Analyzer

Min Time: 24.48046 s

Max Time: 24.81633 s

Grid: 5 us

```
  GPIO_PORTF_DIR_R |= (1<<3); //Direction is Output fot pin 3 Port F
  GPIO_PORTF_DEN_R |= (1<<3);


  while(1)
  {
    GPIO_PORTF_DATA_R |=(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
    GPIO_PORTF_DATA_R &=~(1<<3);
    for (delay_count ;delay_count<20000;delay_count++ );
  }
  return 0;
}
```

TExaS edX Lab 2

Port F Hardware

Port F Registers

DATA: 0x11
PUR: 0x00
LOCK: 0x01
DIR: 0x08
PDR: 0x00
CR: 0x1E
DEN: 0x08
RCGC2: 0x00000020 Clock enabled

Grading Controls

Number from edX:

Score: 0

Copy this to edX:

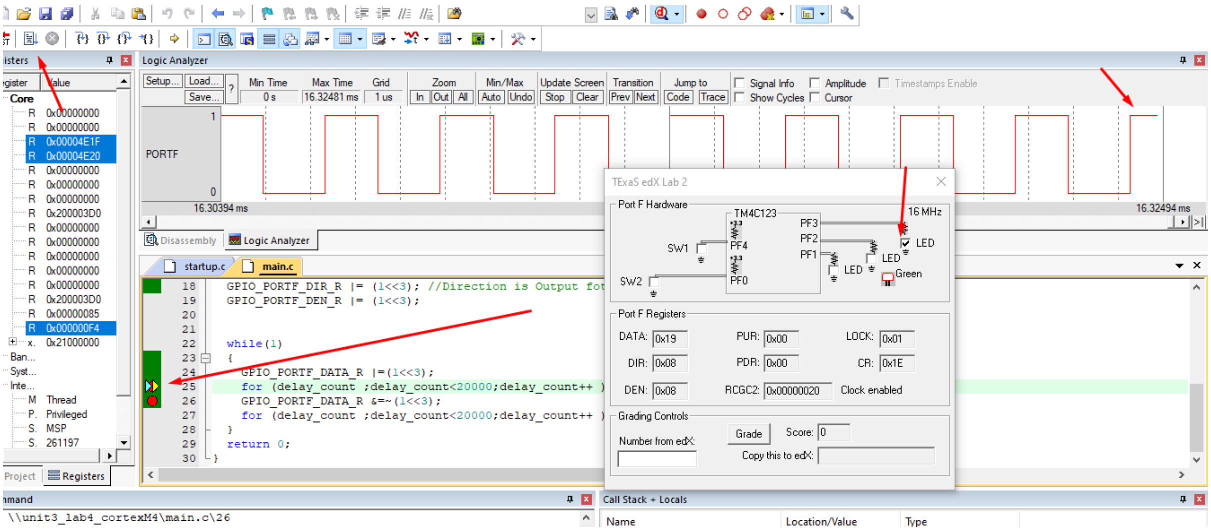
```
exM4\main.c\26
```

```
BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Call Stack + Locals

Name		
main	0x000000A0	int f()
delay_count	0x00004E20	auto - uint

Simulation t1: 24.72706744 sec L:25 C:1 CAP NUM SCRL OVR R /W



```
Core
  R   0x00000000
  R   0x00000000
  R   0x00004E1F
  R   0x00004E20
  R   0x00000000
  R   0x00000000
  R   0x00000000
  R   0x200003D0
  R   0x00000000
  R   0x00000000
  R   0x00000000
  R   0x00000000
  R   0x00000000
  R   0x200003D0
  R   0x00000085
  R   0x000000F4
+ x.  0x21000000
Ban...
Syst...
Inte...
  M   Thread
  P.  Privileged
  S.  MSP
  S.  261197
```

```c
    GPIO_PORTF_DIR_R |= (1<<3); //Direction is Output fot
    GPIO_PORTF_DEN_R |= (1<<3);


    while(1)
    {
      GPIO_PORTF_DATA_R |=(1<<3);
      for (delay_count ;delay_count<20000;delay_count++ )
      GPIO_PORTF_DATA_R &=~(1<<3);
      for (delay_count ;delay_count<20000;delay_count++ )
    }
    return 0;
}
```

```
\\unit3_lab4_cortexM4\main.c\26
```

DATA: 0x19 PUR: 0x00 LOCK: 0x01

DIR: 0x08 PDR: 0x00 CR: 0x1E

DEN: 0x08 RCGC2: 0x00000020 Clock enabled

Number from edX:

Score: 0

Copy this to edX: