

## Lab 4 report

In this scope we will toggle the LED and simulate the code on Tiva\_C Kit that has tm4c123 and arm-cortexM4 processor .

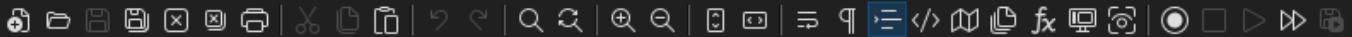
The LED connected to pin3 of PORTF.

From specs :

- ✓ Flash memory starts with address 0x00000000 and has size of 512M.
- ✓ Sram memory starts 0x20000000 and has size of 512M.
- ✓ SYSCTL is system control module that we will use to enable clock for PORTF has base address of 0x400FE000
- ✓ SYSCTL\_RCGC2\_R has offset address of 0x108 under SYSCTL we will assign this register with value of 0x00000020 to enable clock for PORTF
- ✓ GPIO module has base address of 0x40025000 and we will use three registers inside
  - First GPIO\_PORTF\_DIR\_R has offset of 0x400 and we will assign value of 1 in pin3 to define this pin as an output
  - First GPIO\_PORTF\_DEN\_R has offset of 0x51c and we will assign value of 1 in pin3 to enable this pin
- ✓ GPIO\_PORTF\_DR\_R has offset of 0x400 and we will assign value of 1 in pin3 and 0 to toggle the output.

E:\learn\_in\_depth\Unit 3 Embedded\_C\lesson4\_ass\Keil\_uvision\_unit3\_lab4\_project\assignment4\_lab3\main.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

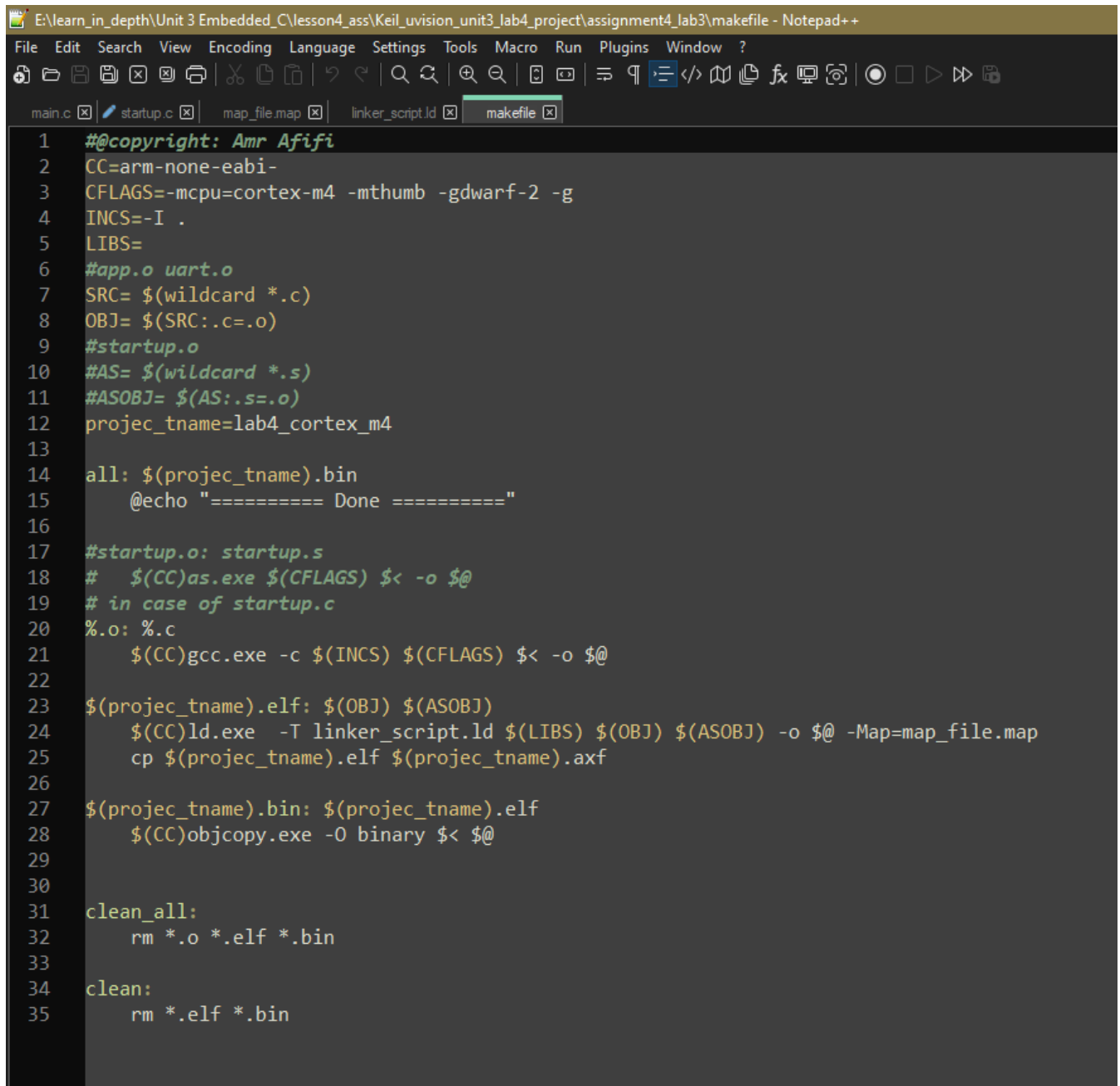


main.c x startup.c x map\_file.map x

```
1 // Amr Afifi
2
3 #define SYSCTRL_RCGC2_R (*((volatile unsigned long *) 0x400FE108))
4 #define GPIO_PORTF_DIR_R (*((volatile unsigned long *) 0x40025400))
5 #define GPIO_PORTF_DEN_R (*((volatile unsigned long *) 0x4002551C))
6 #define GPIO_PORTF_DATA_R (*((volatile unsigned long *) 0x400253FC))
7
8 int main (){
9     volatile unsigned long delay_count; // volatile to avoid the optimizer
10     SYSCTRL_RCGC2_R = 0x00000020 ;
11     //delay to make sure that gpiof is up and running
12     for (delay_count = 0 ; delay_count<200 ; delay_count++);
13
14     GPIO_PORTF_DIR_R |= 1<<3 ; //DIR is output for pin3 port f
15     GPIO_PORTF_DEN_R |= 1<<3 ;
16
17     while (1){
18         GPIO_PORTF_DATA_R |= 1<<3 ;
19         for (delay_count = 0 ; delay_count<200000 ; delay_count++);
20         GPIO_PORTF_DATA_R &= ~(1<<3) ;
21         for (delay_count = 0 ; delay_count<200000 ; delay_count++);
22     }
23
24     return 0;
25 }
```

make file : we will make simple changes

- ✓ we will copy the .axf file to run on kiel micro vision tool
- ✓ change the project name and processor name



```
1  ##copyright: Amr Afifi
2  CC=arm-none-eabi-
3  CFLAGS=-mcpu=cortex-m4 -mthumb -gdwarf-2 -g
4  INCS=-I .
5  LIBS=
6  #app.o uart.o
7  SRC= $(wildcard *.c)
8  OBJ= $(SRC:.c=.o)
9  #startup.o
10 #AS= $(wildcard *.s)
11 #ASOBJ= $(AS:.s=.o)
12 projec_tname=lab4_cortex_m4
13
14 all: $(projec_tname).bin
15     @echo "===== Done ====="
16
17 #startup.o: startup.s
18 # $(CC)as.exe $(CFLAGS) $< -o $@
19 # in case of startup.c
20 %.o: %.c
21     $(CC)gcc.exe -c $(INCS) $(CFLAGS) $< -o $@
22
23 $(projec_tname).elf: $(OBJ) $(ASOBJ)
24     $(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(ASOBJ) -o $@ -Map=map_file.map
25     cp $(projec_tname).elf $(projec_tname).axf
26
27 $(projec_tname).bin: $(projec_tname).elf
28     $(CC)objcopy.exe -O binary $< $@
29
30
31 clean_all:
32     rm *.o *.elf *.bin
33
34 clean:
35     rm *.elf *.bin
```

startup.c ;

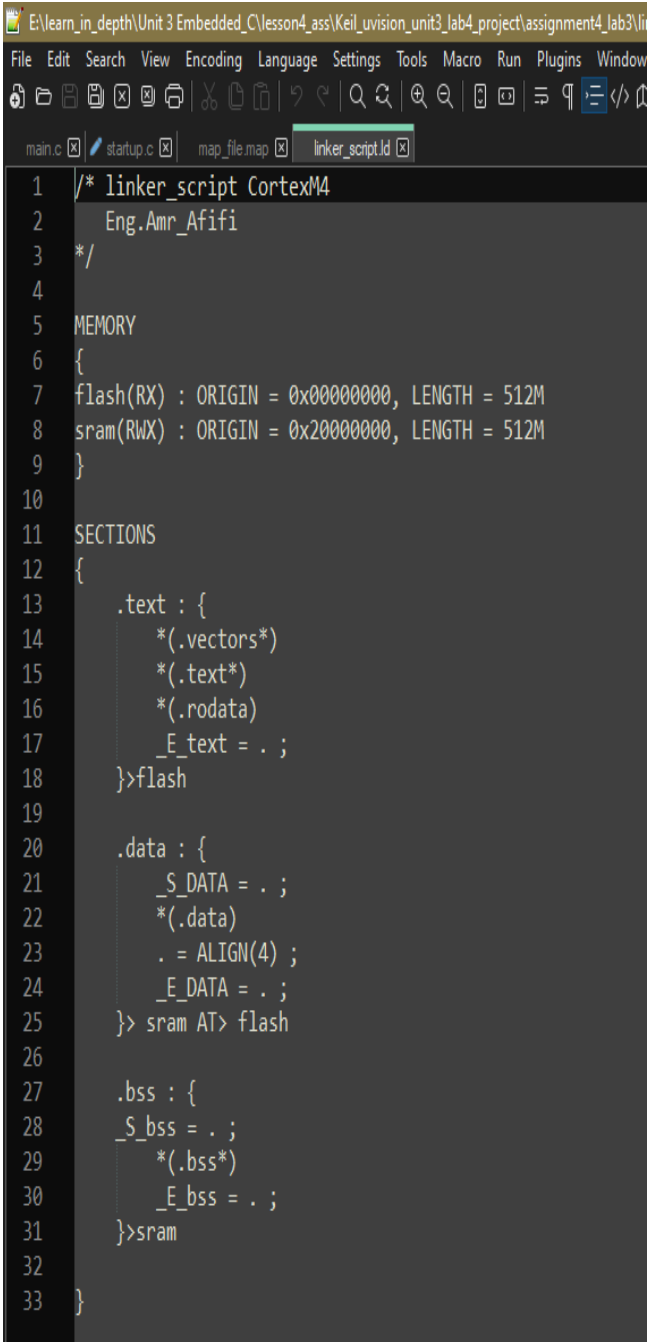
using new approach by initializing the stack pointer in startup.c

- ✓ We will fix SP after 1Kbyte of .bss that by using uninitialized array of int with 256 elements so the total size of the array will be 1024 where the SP will be at the end of the array
- ✓ Array elements will be pointer to function that take nothing and return void
- ✓ The pointers will point to the functions which handle the interrupts according to IVT.

```
*E:\learn_in_depth\Unit 3 Embedded_C\lesson4_ass\Keil_uvision_unit3_lab4_project\assignment4_lab3\startup.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c startup.c map_file.map
1 // startup.c
2 // Eng.Amr Afifi
3 #include <stdint.h>
4
5 extern int main (void);
6
7 void reset_handler();
8
9 void default_handler(){
10     reset_handler();
11 }
12
13 void NMI_Handler(void) __attribute__((weak, alias ("default_handler")));
14 void H_fault_Handler(void) __attribute__((weak, alias ("default_handler")));
15
16 // reserve stack size
17 static unsigned long Stack_top[256] ; // 4*256 =>1024 Bus_fault
18
19 void (*const g_p_f_vectors[]) () __attribute__((section(".vectors"))) =
20 {
21     (void (*)())((unsigned long)Stack_top + sizeof(Stack_top)), // cast to pointer to func that take nothing and return void
22     &reset_handler ,
23     &NMI_Handler,
24     &H_fault_Handler
25 };
26
27 extern unsigned long _E_text ;
28 extern unsigned long _S_DATA ;
29 extern unsigned long _E_DATA ;
30 extern unsigned long _S_bss ;
31 extern unsigned long _E_bss ;
32
33 void reset_handler (){
34
35     // copy the .data section from flash to ram
36     unsigned long data_size = (unsigned char *)&_E_DATA - (unsigned char *)&_S_DATA ;
37     unsigned char* p_src = (unsigned char *)&_E_text ;
38     unsigned char* p_dst = (unsigned char *)&_S_DATA ;
39     unsigned long i=0;
40     for(i=0;i<data_size ; i++){
41         *( (unsigned char *)p_dst++) = *( (unsigned char *)p_src++);
42     }
43     //init .bss
44     unsigned long bss_size = (unsigned char *)&_E_bss - (unsigned char *)&_S_bss ;
45     p_dst = (unsigned char *)&_S_bss ;
46     for(i=0;i<bss_size ; i++){
47         *( (unsigned char *)p_dst++) = (unsigned char )0;
48     }
49
50     // jump to the main
51     main();
52 }
```

## Linker\_script :

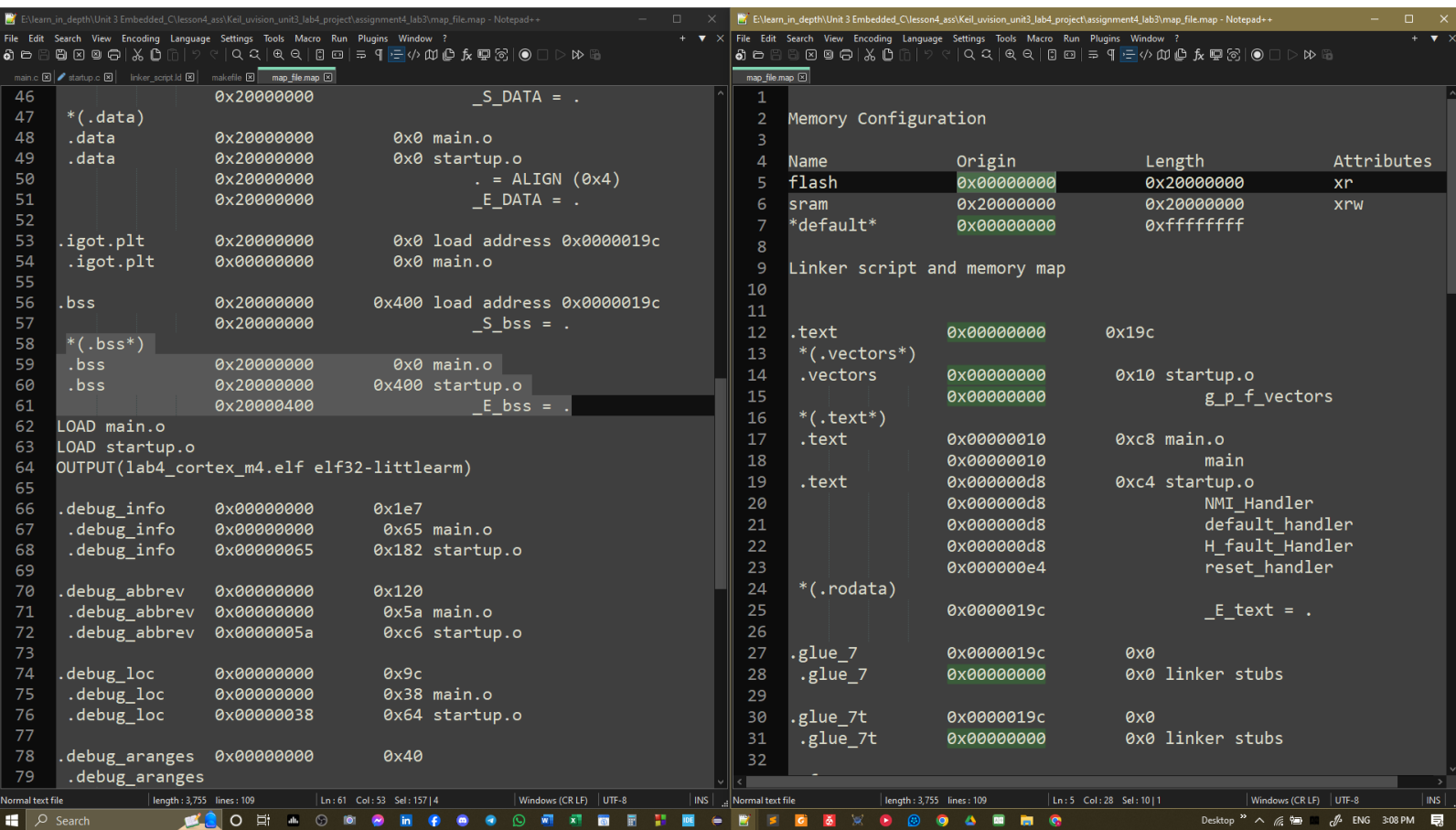
- ✓ delete stack\_top symbol
- ✓ edit the size



```
1  /* linker_script CortexM4
2     Eng.Amr_Afifi
3  */
4
5  MEMORY
6  {
7      flash(RX) : ORIGIN = 0x00000000, LENGTH = 512M
8      sram(RWX) : ORIGIN = 0x20000000, LENGTH = 512M
9  }
10
11  SECTIONS
12  {
13      .text : {
14          *(.vectors*)
15          *(.text*)
16          *(.rodata)
17          _E_text = . ;
18      }>flash
19
20      .data : {
21          _S_DATA = . ;
22          *(.data)
23          . = ALIGN(4) ;
24          _E_DATA = . ;
25      }> sram AT> flash
26
27      .bss : {
28          _S_bss = . ;
29          *(.bss*)
30          _E_bss = . ;
31      }>sram
32
33  }
```

## Map file :

- ✓ Flash starts with 0x00000000
- ✓ .bss starts with 0x20000000 and ends with 0x200000400
- ✓ Note that the incremented is 0x400 which equivalent to 1024 in decimal



The image displays two Notepad++ windows side-by-side, showing the output of a linker script. The left window shows the linker script, and the right window shows the generated memory map.

**Linker Script (Left Window):**

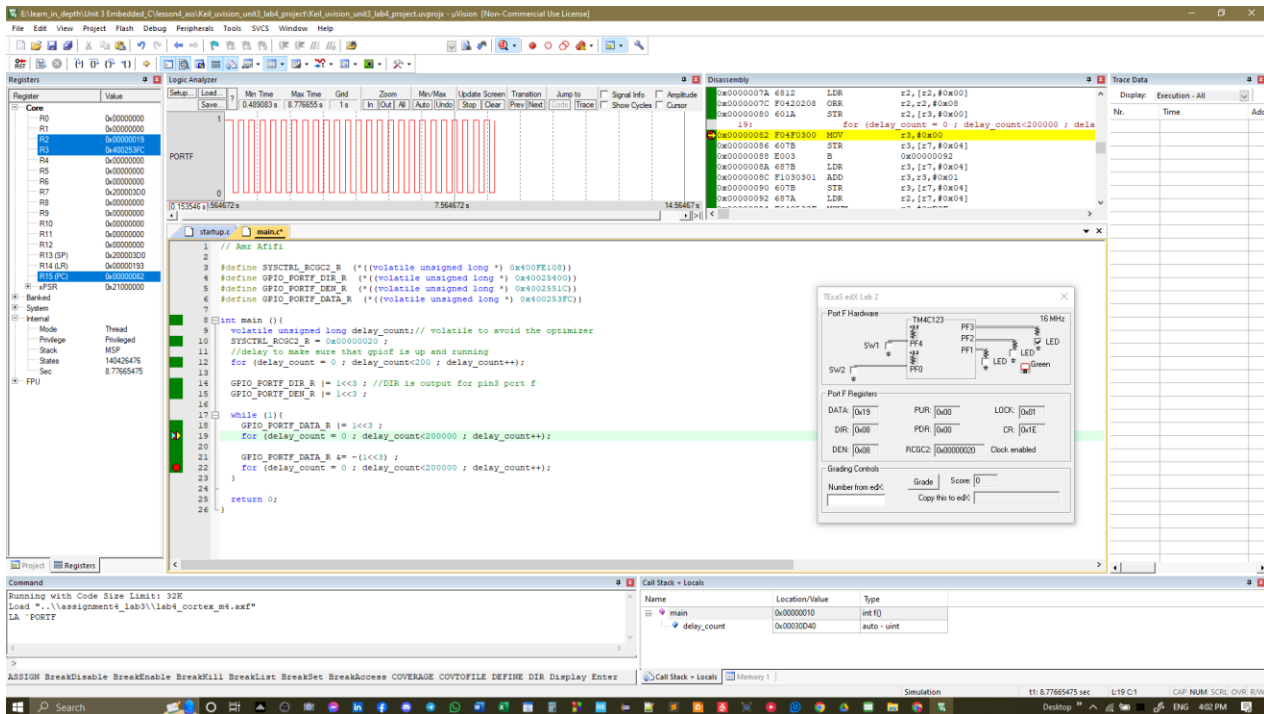
```
46 0x20000000 _S_DATA = .
47 *(.data)
48 .data 0x20000000 0x0 main.o
49 .data 0x20000000 0x0 startup.o
50 0x20000000 . = ALIGN (0x4)
51 0x20000000 _E_DATA = .
52
53 .igot.plt 0x20000000 0x0 load address 0x000019c
54 .igot.plt 0x00000000 0x0 main.o
55
56 .bss 0x20000000 0x400 load address 0x000019c
57 0x20000000 _S_bss = .
58
59 *(.bss*)
59 .bss 0x20000000 0x0 main.o
60 .bss 0x20000000 0x400 startup.o
61 0x20000400 _E_bss = .
62
63 LOAD main.o
64 LOAD startup.o
65
66 .debug_info 0x00000000 0x1e7
67 .debug_info 0x00000000 0x65 main.o
68 .debug_info 0x00000065 0x182 startup.o
69
70 .debug_abbrev 0x00000000 0x120
71 .debug_abbrev 0x00000000 0x5a main.o
72 .debug_abbrev 0x0000005a 0xc6 startup.o
73
74 .debug_loc 0x00000000 0x9c
75 .debug_loc 0x00000000 0x38 main.o
76 .debug_loc 0x00000038 0x64 startup.o
77
78 .debug_aranges 0x00000000 0x40
79 .debug_aranges
```

**Memory Map (Right Window):**

```
1
2 Memory Configuration
3
4 Name Origin Length Attributes
5 flash 0x00000000 0x20000000 xr
6 sram 0x20000000 0x20000000 xrw
7 *default* 0x00000000 0xffffffff
8
9 Linker script and memory map
10
11
12 .text 0x00000000 0x19c
13 *(.vectors*)
14 .vectors 0x00000000 0x10 startup.o
15 0x00000000 g_p_f_vectors
16
17 *(.text*)
17 .text 0x00000010 0xc8 main.o
18 0x00000010 main
19 .text 0x000000d8 0xc4 startup.o
20 0x000000d8 NMI_Handler
21 0x000000d8 default_handler
22 0x000000d8 H_fault_Handler
23 0x000000e4 reset_handler
24
25 *(.rodata)
25 0x00000019c _E_text = .
26
27 .glue_7 0x00000019c 0x0
28 .glue_7 0x00000000 0x0 linker stubs
29
30 .glue_7t 0x00000019c 0x0
31 .glue_7t 0x00000000 0x0 linker stubs
32
```

# Debugging using kiel microvision:

## At high level: led is on



## At low level: led is off

