Mastering Embedded System

Unit3 Lesson3

Lab_2

CORTEX-M3 STM32F103C6

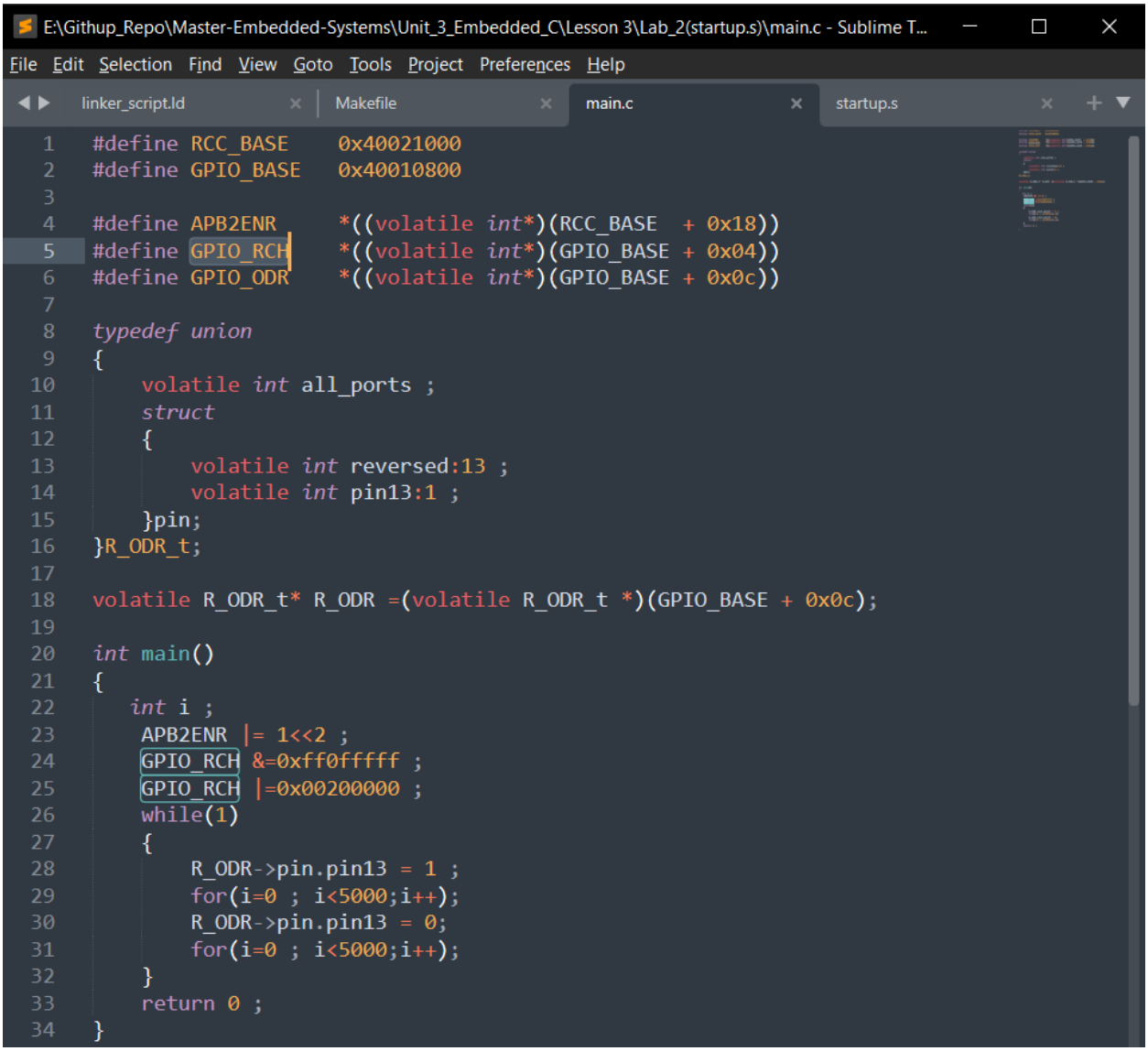
I will write bare-metal software for a SM32F103C6 to toggle the led on part A pin 13.

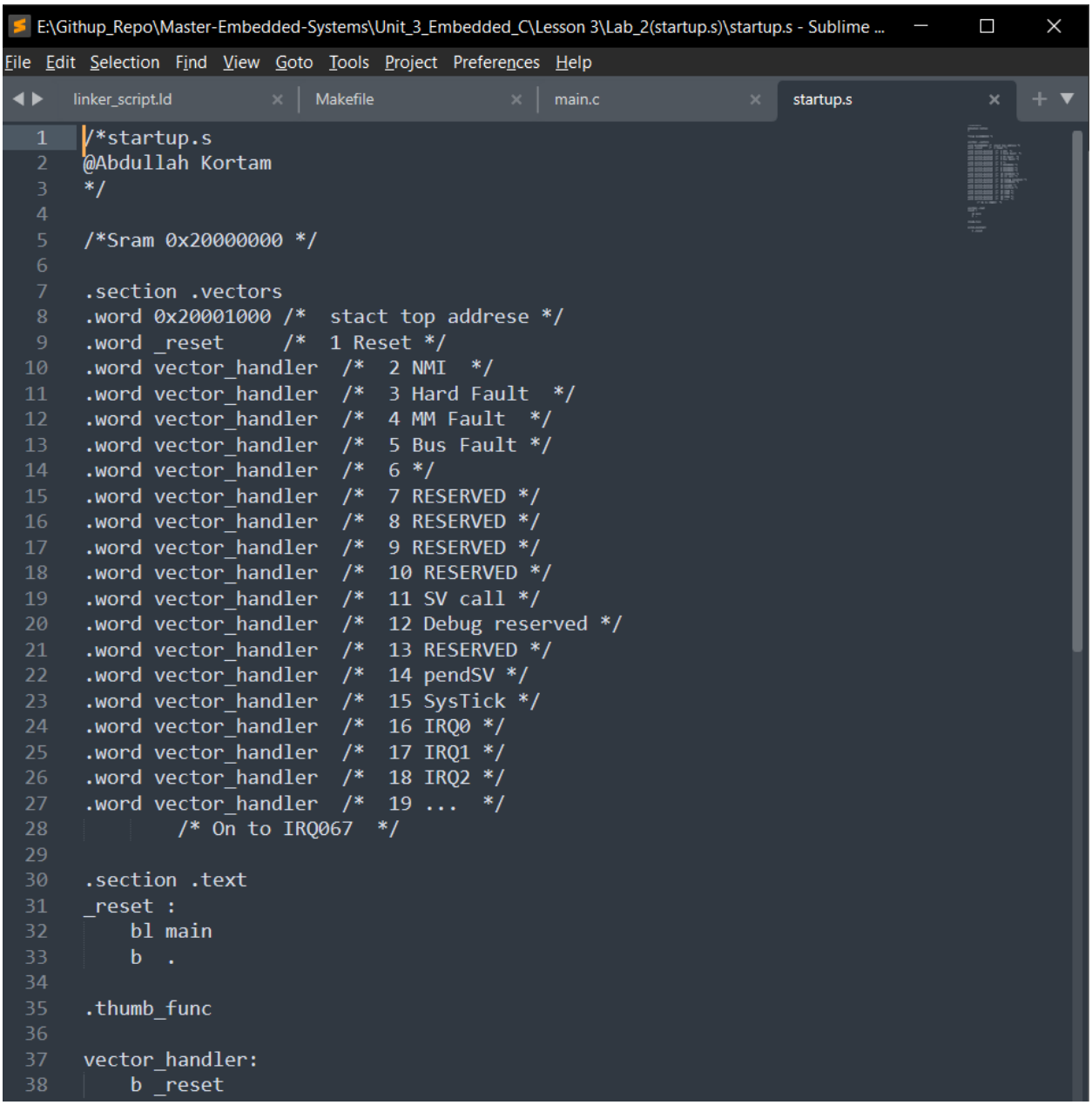
Peripherals:

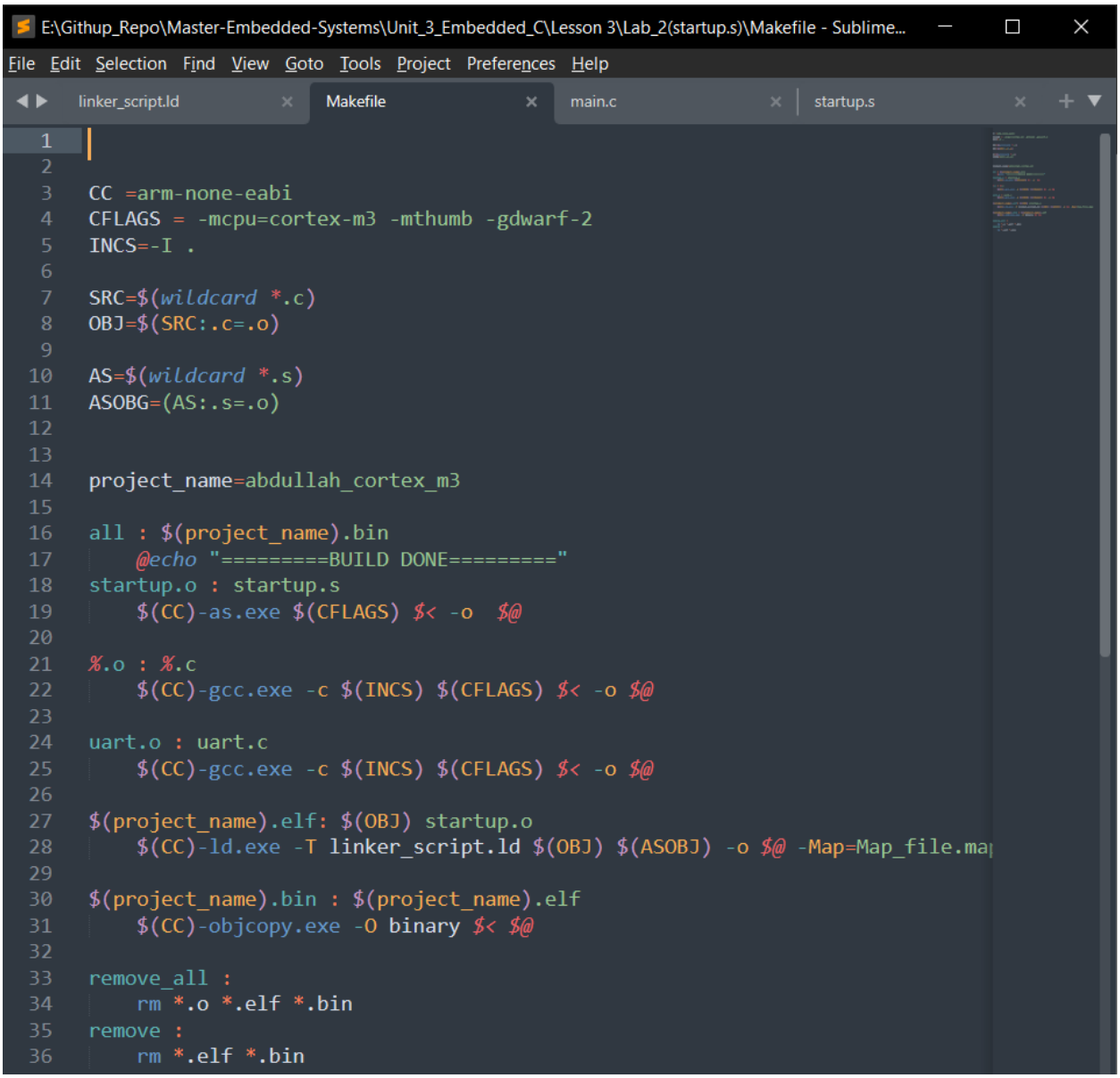
RCC stands for reset and clock control, and its base address is 0x40021000. While APB2ENR has an offset from the base of 0x18 ,set its value with 1.

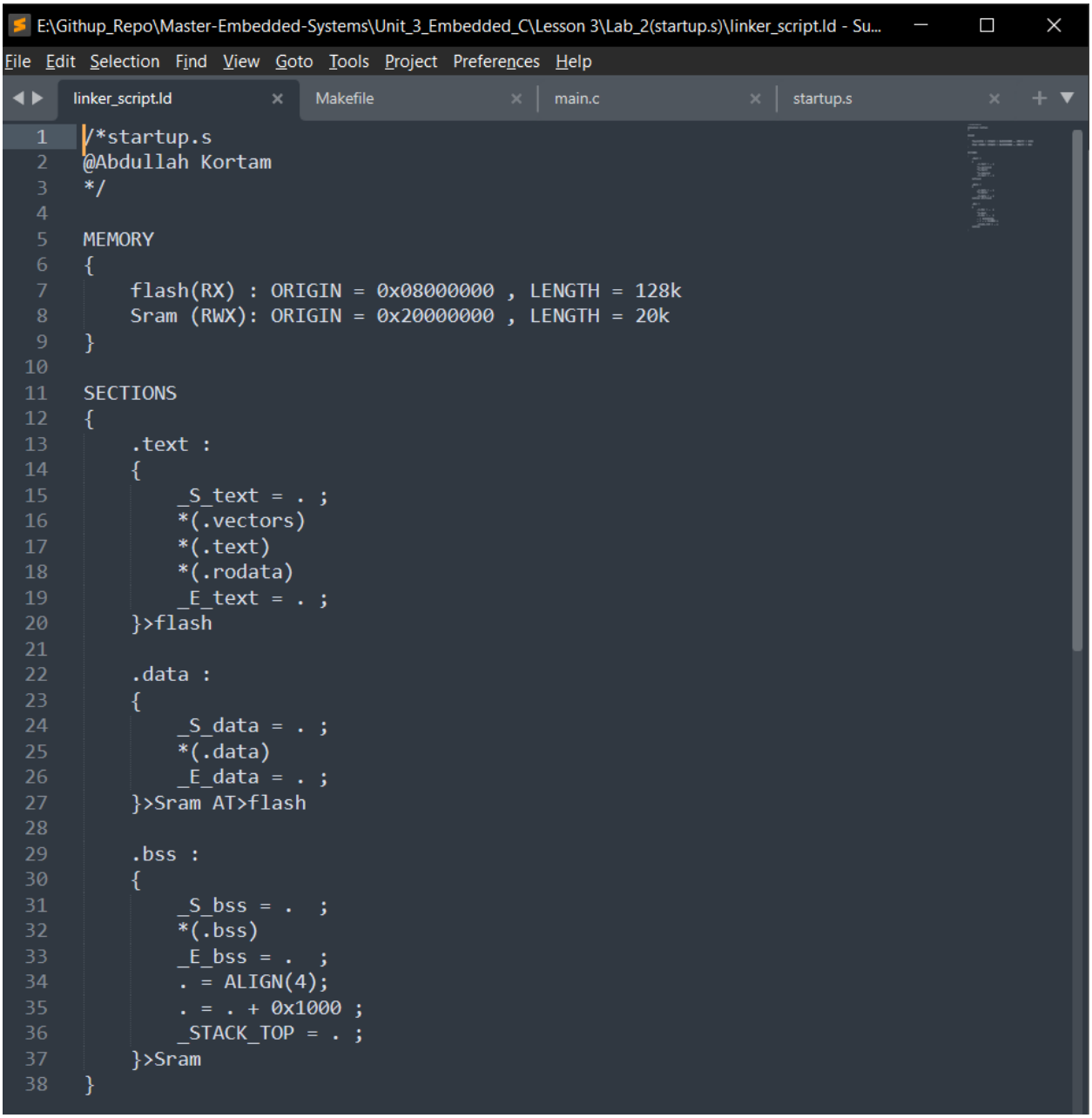
GPIO stands for general purpose input output, and its base address is 0x40010800.
It has 2 registers:

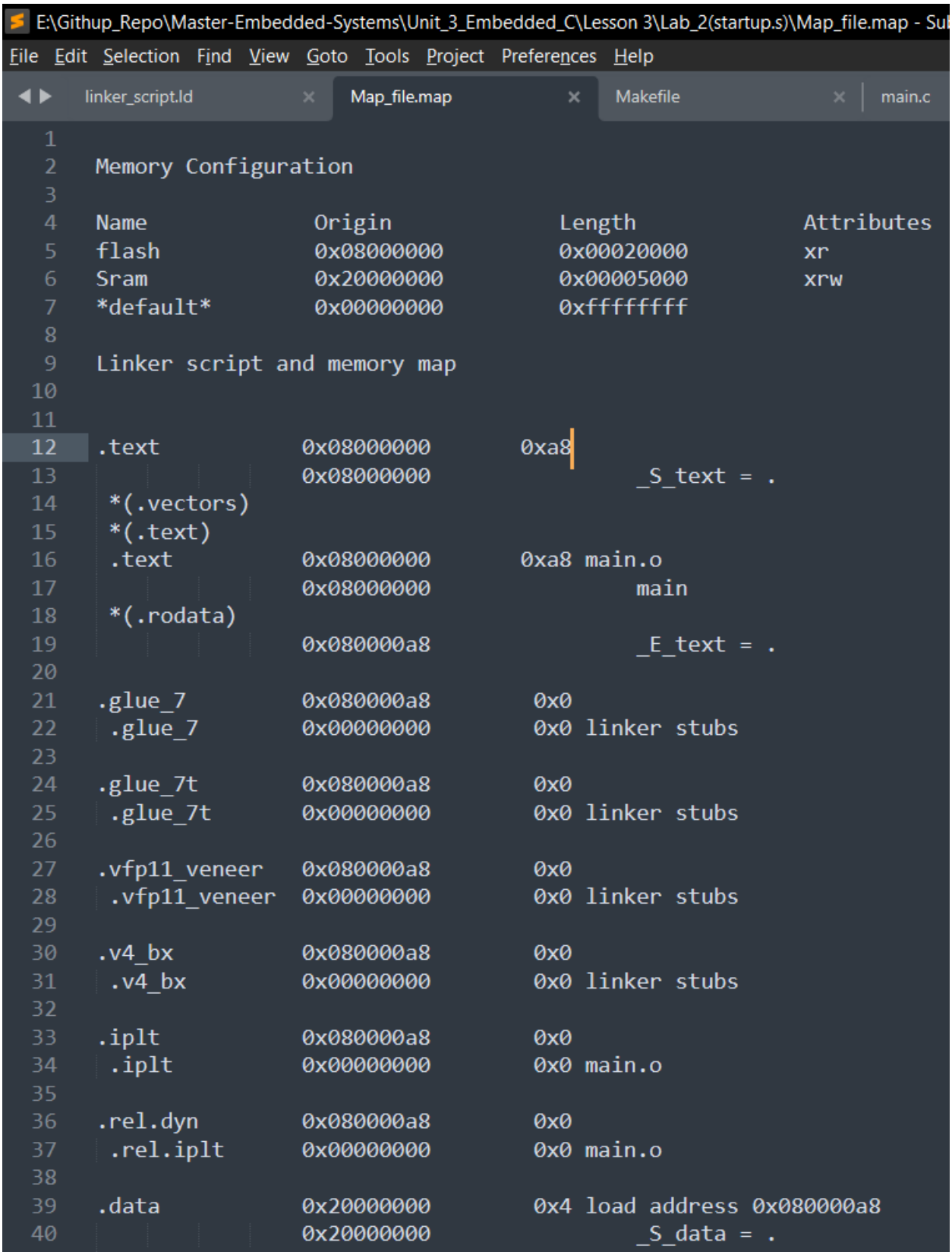
1) GPIO_RCH has an offset from the base 0x04, set its bits (20:24) by 0x2.
2) GPIO_ODR has an offset from the base 0x0c, toggle value on pin13 to toggle the led.

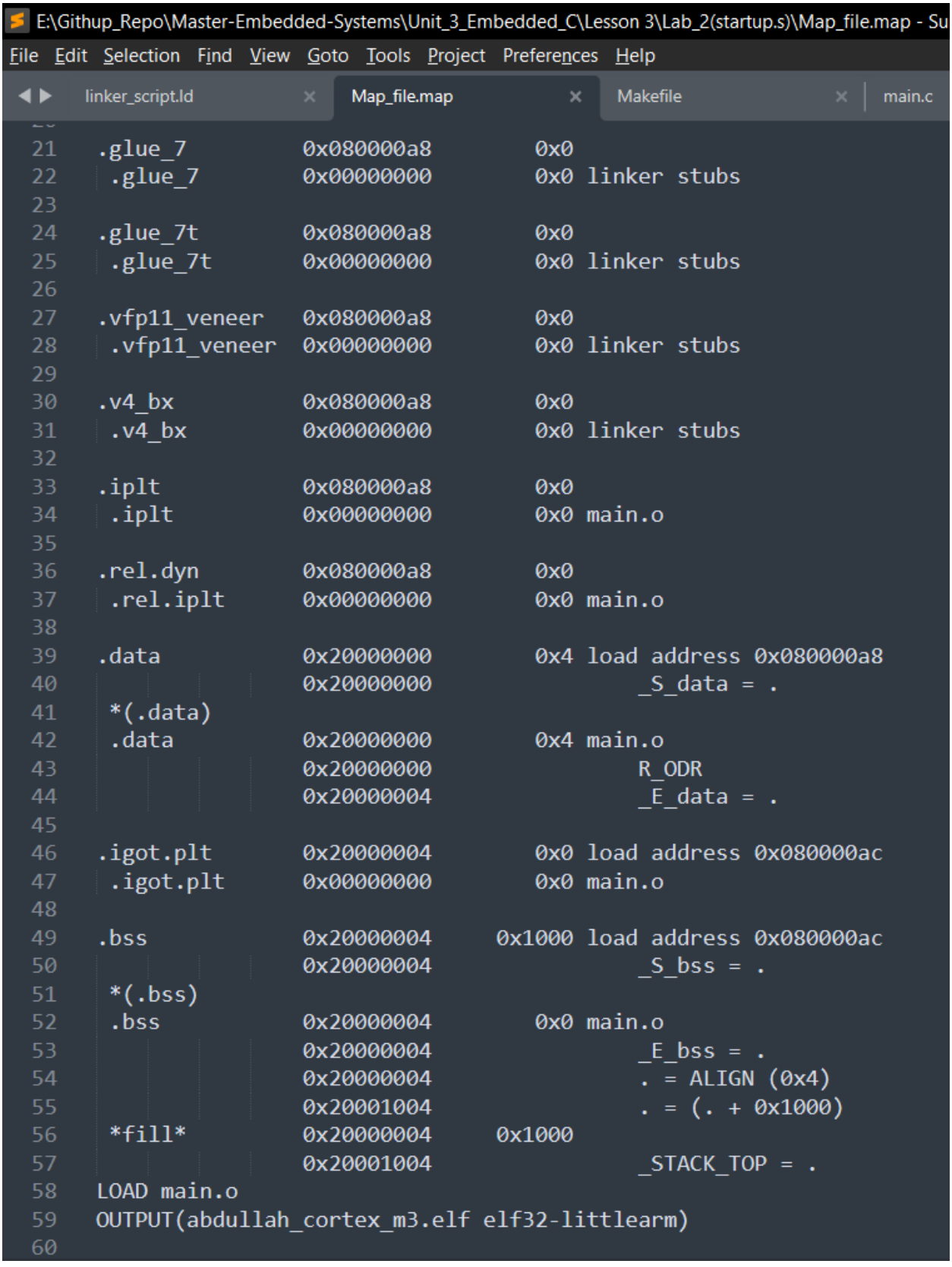


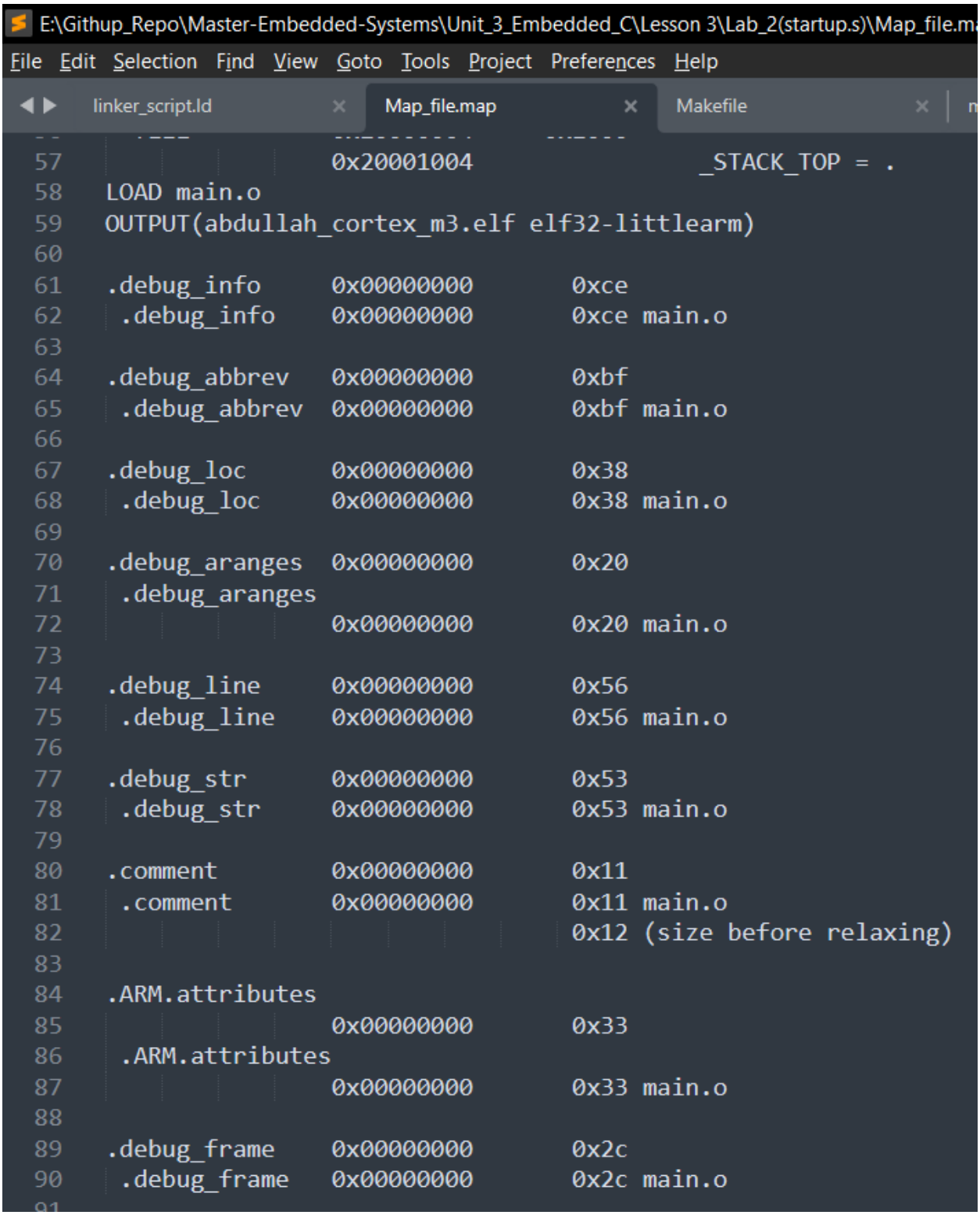

E:\Githup_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\startup.s - Sublime ...
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Makefile
main.c
startup.s
/*startup.s
@Abdullah Kortam
*/
/*Sram 0x20000000 */
.section .vectors
.word 0x20001000 /* stact top addrese */
.word _reset /* 1 Reset */
.word vector_handler /* 2 NMI */
.word vector_handler /* 3 Hard Fault */
.word vector_handler /* 4 MM Fault */
.word vector_handler /* 5 Bus Fault */
.word vector_handler /* 6 */
.word vector_handler /* 7 RESERVED */
.word vector_handler /* 8 RESERVED */
.word vector_handler /* 9 RESERVED */
.word vector_handler /* 10 RESERVED */
.word vector_handler /* 11 SV call */
.word vector_handler /* 12 Debug reserved */
.word vector_handler /* 13 RESERVED */
.word vector_handler /* 14 pendSV */
.word vector_handler /* 15 SysTick */
.word vector_handler /* 16 IRQ0 */
.word vector_handler /* 17 IRQ1 */
.word vector_handler /* 18 IRQ2 */
.word vector_handler /* 19 ... */
/* On to IRQ067 */
.section .text
_reset :
bl main
b .
.thumb_func
vector_handler:
b _reset


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\Makefile - Sublime...
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Makefile
main.c
startup.s
CC =arm-none-eabi
CFLAGS = -mcpu=cortex-m3 -mthumb -gdwarf-2
INCS=-I .
SRC=$(wildcard *.c)
OBJ=$(SRC:.c=.o)
AS=$(wildcard *.s)
ASOBG=(AS:.s=.o)
project_name=abdullah_cortex_m3
all : $(project_name).bin
@echo "=========BUILD DONE========="
startup.o : startup.s
$(CC)-as.exe $(CFLAGS) $< -o $@
%.o : %.c
$(CC)-gcc.exe -c $(INCS) $(CFLAGS) $< -o $@
uart.o : uart.c
$(CC)-gcc.exe -c $(INCS) $(CFLAGS) $< -o $@
$(project_name).elf: $(OBJ) startup.o
$(CC)-ld.exe -T linker_script.ld $(OBJ) $(ASOBJ) -o $@ -Map=Map_file.ma
$(project_name).bin : $(project_name).elf
$(CC)-objcopy.exe -O binary $< $@
remove_all :
rm *.o *.elf *.bin
remove :
rm *.elf *.bin


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\linker_script.ld - Su...
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Makefile
main.c
startup.s
/*startup.s
@Abdullah Kortam
*/
MEMORY
{
flash(RX) : ORIGIN = 0x08000000 , LENGTH = 128k
Sram (RWX): ORIGIN = 0x20000000 , LENGTH = 20k
}
SECTIONS
{
.text :
{
_S_text = . ;
*(.vectors)
*(.text)
*(.rodata)
_E_text = . ;
}>flash
.data :
{
_S_data = . ;
*(.data)
_E_data = . ;
}>Sram AT>flash
.bss :
{
_S_bss = . ;
*(.bss)
_E_bss = . ;
. = ALIGN(4);
. = . + 0x1000 ;
_STACK_TOP = . ;
}>Sram
}


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\Map_file.map - Su
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Map_file.map
Makefile
main.c
Memory Configuration
Name Origin Length Attributes
flash 0x08000000 0x00020000 xr
Sram 0x20000000 0x00005000 xrw
default 0x00000000 0xffffffff
Linker script and memory map
.text 0x08000000 0xa8
0x08000000 _S_text = .
*(.vectors)
*(.text)
.text 0x08000000 0xa8 main.o
0x08000000 main
*(.rodata)
0x080000a8 _E_text = .
.glue_7 0x080000a8 0x0
.glue_7 0x00000000 0x0 linker stubs
.glue_7t 0x080000a8 0x0
.glue_7t 0x00000000 0x0 linker stubs
.vfp11_veneer 0x080000a8 0x0
.vfp11_veneer 0x00000000 0x0 linker stubs
.v4_bx 0x080000a8 0x0
.v4_bx 0x00000000 0x0 linker stubs
.iplt 0x080000a8 0x0
.iplt 0x00000000 0x0 main.o
.rel.dyn 0x080000a8 0x0
.rel.iplt 0x00000000 0x0 main.o
.data 0x20000000 0x4 load address 0x080000a8
0x20000000 _S_data = .


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\Map_file.map - Su
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Map_file.map
Makefile
main.c
.glue_7 0x080000a8 0x0
.glue_7 0x00000000 0x0 linker stubs
.glue_7t 0x080000a8 0x0
.glue_7t 0x00000000 0x0 linker stubs
.vfp11_veneer 0x080000a8 0x0
.vfp11_veneer 0x00000000 0x0 linker stubs
.v4_bx 0x080000a8 0x0
.v4_bx 0x00000000 0x0 linker stubs
.iplt 0x080000a8 0x0
.iplt 0x00000000 0x0 main.o
.rel.dyn 0x080000a8 0x0
.rel.iplt 0x00000000 0x0 main.o
.data 0x20000000 0x4 load address 0x080000a8
0x20000000 _S_data = .
*(.data)
.data 0x20000000 0x4 main.o
0x20000000 R_ODR
0x20000004 _E_data = .
.igot.plt 0x20000004 0x0 load address 0x080000ac
.igot.plt 0x00000000 0x0 main.o
.bss 0x20000004 0x1000 load address 0x080000ac
0x20000004 _S_bss = .
*(.bss)
.bss 0x20000004 0x0 main.o
0x20000004 _E_bss = .
0x20000004 . = ALIGN (0x4)
0x20001004 . = (. + 0x1000)
fill 0x20000004 0x1000
0x20001004 _STACK_TOP = .
LOAD main.o
OUTPUT(abdullah_cortex_m3.elf elf32-littlearm)


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.s)\Map_file.ma
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Map_file.map
Makefile
0x20001004 _STACK_TOP = .
LOAD main.o
OUTPUT(abdullah_cortex_m3.elf elf32-littlearm)
.debug_info 0x00000000 0xce
.debug_info 0x00000000 0xce main.o
.debug_abbrev 0x00000000 0xbf
.debug_abbrev 0x00000000 0xbf main.o
.debug_loc 0x00000000 0x38
.debug_loc 0x00000000 0x38 main.o
.debug_aranges 0x00000000 0x20
.debug_aranges
0x00000000 0x20 main.o
.debug_line 0x00000000 0x56
.debug_line 0x00000000 0x56 main.o
.debug_str 0x00000000 0x53
.debug_str 0x00000000 0x53 main.o
.comment 0x00000000 0x11
.comment 0x00000000 0x11 main.o
0x12 (size before relaxing)
.ARM.attributes
0x00000000 0x33
.ARM.attributes
0x00000000 0x33 main.o
.debug_frame 0x00000000 0x2c
.debug_frame 0x00000000 0x2c main.o


E:\Github_Repo\Master-Embedded-Systems\Unit_3_Embedded_C\Lesson 3\Lab_2(startup.c)\startup.c -
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
Map_file.map
startup.c
* @author : Abdullah Kortam
#include <stdint.h>
extern uint32_t _STACK_TOP ;
extern uint32_t main() ;
void Reset_Handler();
void Default_Hundler()
{
Reset_Handler();
}
void NMI_Handler () __attribute__((weak,alias("Default_Hundler")));
void H_Fault_Handler() __attribute__((weak,alias("Default_Hundler")));
void Bus_Fault() __attribute__((weak,alias("Default_Hundler")));
void Usage_Fault_Handler() __attribute__((weak,alias("Default_Hundler")));
uint32_t vector[] __attribute__((section(".vextors")))={
(uint32_t) &_STACK_TOP ,
(uint32_t) &Reset_Handler,
(uint32_t)&NMI_Handler,
(uint32_t)&H_Fault_Handler,
(uint32_t)&Bus_Fault,
(uint32_t)&Usage_Fault_Handler
};
extern uint32_t _E_TEXT ;
extern uint32_t _S_DATA ;
extern uint32_t _E_DATA ;
extern uint32_t _S_BSS ;
extern uint32_t _E_BSS ;
uint32_t i ;
void Reset_Handler ()
{
uint32_t DATA_SIZE = (uint8_t*)&_E_DATA - (uint8_t*)&_S_DATA ;
uint8_t P_src = (uint8_t*) &_E_TEXT ;
uint8_t P_des = (uint8_t*) &_S_DATA ;
for(i=0;i<DATA_SIZE;i++)
{
((uint8_t)P_des++) = *((uint8_t*)P_src++) ;
}
uint32_t BSS_Size = (uint8_t*)&_E_BSS - (uint8_t*)&_S_BSS ;
P_des = (uint8_t*)&_S_BSS ;
for (i = 0; i < BSS_Size; ++i)
{
((uint8_t)P_des++) = (uint8_t)0 ;
}
main();
}