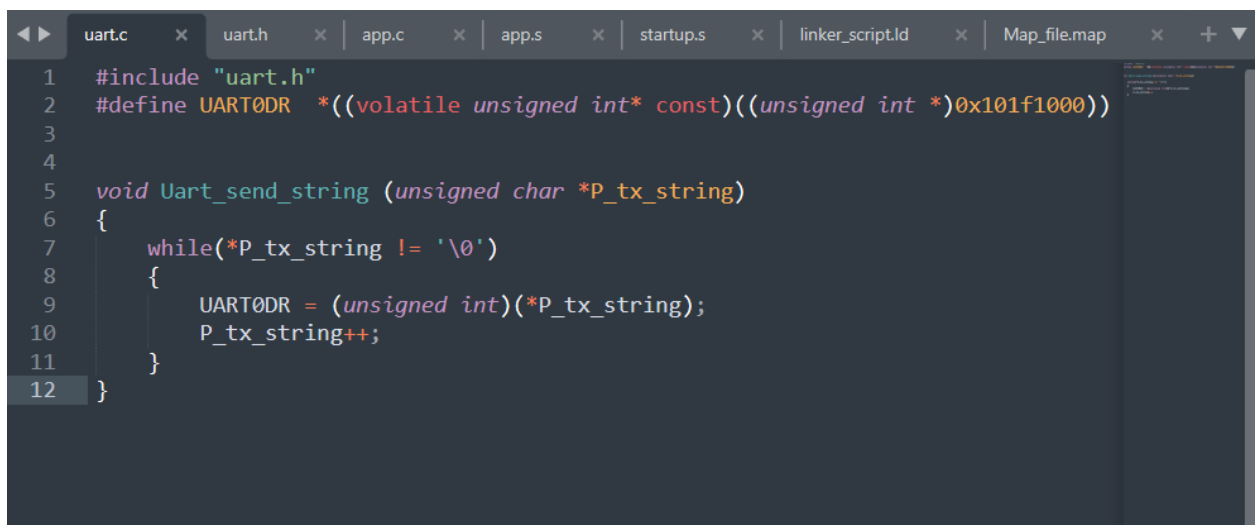


Assignment-2

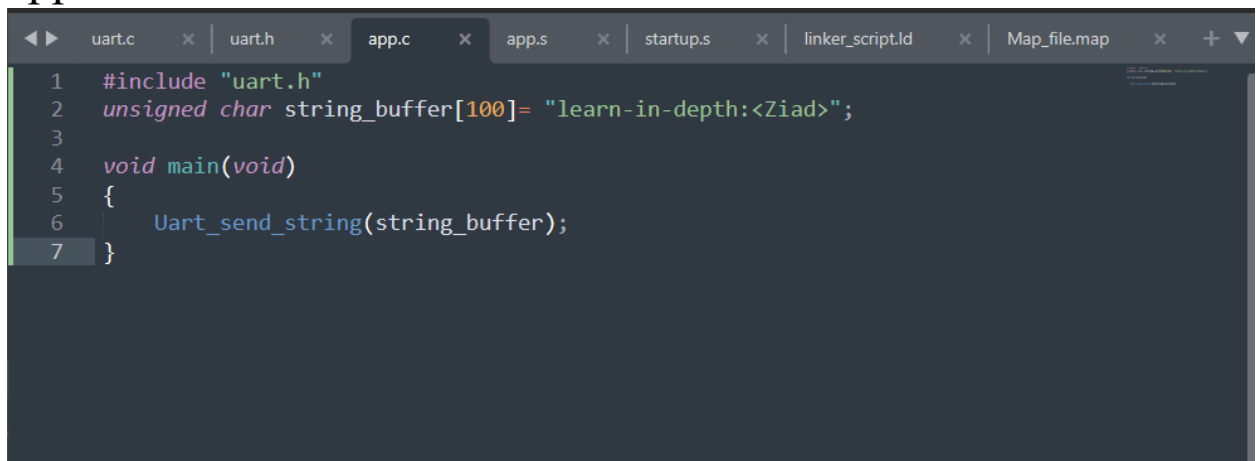
- The codes :

1. uart.c



```
1  #include "uart.h"
2  #define UART0DR *((volatile unsigned int* const)((unsigned int *)0x101f1000))
3
4
5  void Uart_send_string (unsigned char *P_tx_string)
6  {
7      while(*P_tx_string != '\0')
8      {
9          UART0DR = (unsigned int)(*P_tx_string);
10         P_tx_string++;
11     }
12 }
```

2. app.c



```
1  #include "uart.h"
2  unsigned char string_buffer[100]= "learn-in-depth:<Ziad>";
3
4  void main(void)
5  {
6      Uart_send_string(string_buffer);
7  }
```

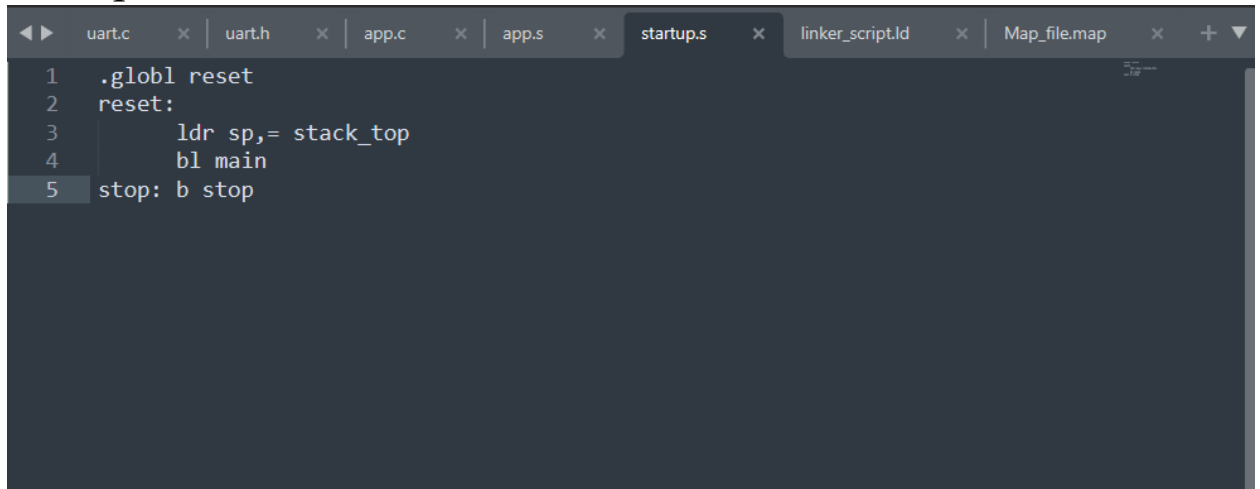
3. uart.h



The screenshot shows an IDE with several open files: uart.c, uart.h, app.c, app.s, startup.s, linker_script.ld, and Map_file.map. The 'uart.h' file is selected and displays the following code:

```
1  #ifndef _UART_H_
2  #define _UART_H_
3
4  void Uart_send_string (unsigned char *P_tx_string);
5
6  #endif
```

4. startup.s



The screenshot shows the same IDE with the 'startup.s' file selected. The code in the file is as follows:

```
1  .globl reset
2  reset:
3      ldr sp,= stack_top
4      bl main
5  stop: b stop
```

5. linker_script.ld

```
uart.c  x  uart.h  x  app.c  x  app.s  x  startup.s  x  linker_script.ld  x  Map_file.map  x  +  ▼

1  ENTRY(reset)
2  MEMORY
3  {
4      Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
5  }
6  SECTIONS
7  {
8      . = 0x10000;
9      .startup : :
10     {
11         startup.o(.text)
12     }>Mem
13     .text :
14     {
15         *(.text) *(.rodata)
16     }>Mem
17     .data :
18     {
19         *(.data)
20     }>Mem
21     .bss :
22     {
23         *(.bss) *(COMMON)
24     }>Mem
25     . = . + 0x1000;
26     stack_top = . ;
27 }
```

- obj files:

1. app.o

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ arm-none-eabi-objdump.exe -h app.o
```

app.o: file format elf32-littlearm

Sections:

Idx	Name	Size	VMA	LMA	File off	Algn
0	.text	00000018	00000000	00000000	00000034	2**2
	CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE					
1	.data	00000064	00000000	00000000	0000004c	2**2
	CONTENTS, ALLOC, LOAD, DATA					
2	.bss	00000000	00000000	00000000	000000b0	2**0
	ALLOC					
3	.debug_info	0000006c	00000000	00000000	000000b0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
4	.debug_abbrev	0000005a	00000000	00000000	0000011c	2**0
	CONTENTS, READONLY, DEBUGGING					
5	.debug_loc	0000002c	00000000	00000000	00000176	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_aranges	00000020	00000000	00000000	000001a2	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
7	.debug_line	00000035	00000000	00000000	000001c2	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_str	00000088	00000000	00000000	000001f7	2**0
	CONTENTS, READONLY, DEBUGGING					
9	.comment	00000012	00000000	00000000	0000027f	2**0
	CONTENTS, READONLY					
10	.ARM.attributes	00000032	00000000	00000000	00000291	2**0
	CONTENTS, READONLY					
11	.debug_frame	0000002c	00000000	00000000	000002c4	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					

2. uart.o

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ arm-none-eabi-objdump.exe -h uart.o

uart.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000050  00000000  00000000  00000034  2**2
                CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data          00000000  00000000  00000000  00000084  2**0
                CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000  00000000  00000000  00000084  2**0
                ALLOC
 3 .debug_info    0000005c  00000000  00000000  00000084  2**0
                CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev  00000051  00000000  00000000  000000e0  2**0
                CONTENTS, READONLY, DEBUGGING
 5 .debug_loc     0000002c  00000000  00000000  00000131  2**0
                CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020  00000000  00000000  0000015d  2**0
                CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line    0000003d  00000000  00000000  0000017d  2**0
                CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str     0000008a  00000000  00000000  000001ba  2**0
                CONTENTS, READONLY, DEBUGGING
 9 .comment       00000012  00000000  00000000  00000244  2**0
                CONTENTS, READONLY
10 .ARM.attributes 00000032  00000000  00000000  00000256  2**0
                CONTENTS, READONLY
11 .debug_frame   00000028  00000000  00000000  00000288  2**2
                CONTENTS, RELOC, READONLY, DEBUGGING
```

3. startup.o

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:    file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000010  00000000  00000000  00000034  2**2
                CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000000  00000000  00000000  00000044  2**0
                CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000  00000000  00000000  00000044  2**0
                ALLOC
 3 .ARM.attributes 00000022  00000000  00000000  00000044  2**0
                CONTENTS, READONLY
```

- To show sections for learn-in-depth.elf

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00000010  00000000  00000000  00000034  2**2
               CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data          00000000  00000000  00000000  00000044  2**0
               CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  00000044  2**0
               ALLOC
  3 .ARM.attributes 00000022  00000000  00000000  00000044  2**0
               CONTENTS, READONLY
```

- To show symbol table for learn-in-depth.elf

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
000110dc D stack_top
00010008 t stop
00010078 D string_buffer
00010028 T Uart_send_string
```

- burn binary file on board using qemu

```
ziade@LAPTOP-S0076FNK MINGW64 /d/Embedded system diploma/github_repo/Unit_3_Embedded_C/Lesson2_Assignment (main)
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn-in-depth:<Ziad>
```