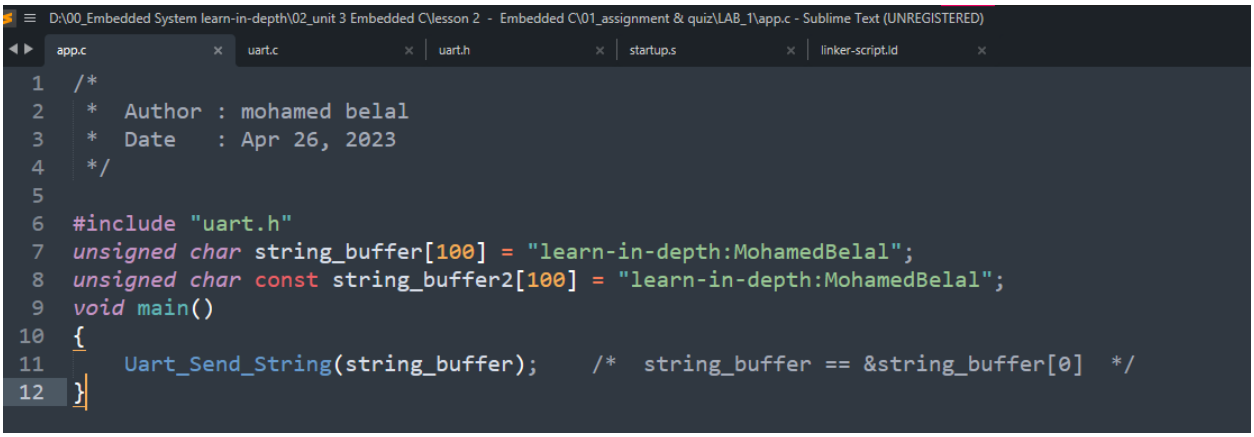
ARM VersatilePB Lab Report

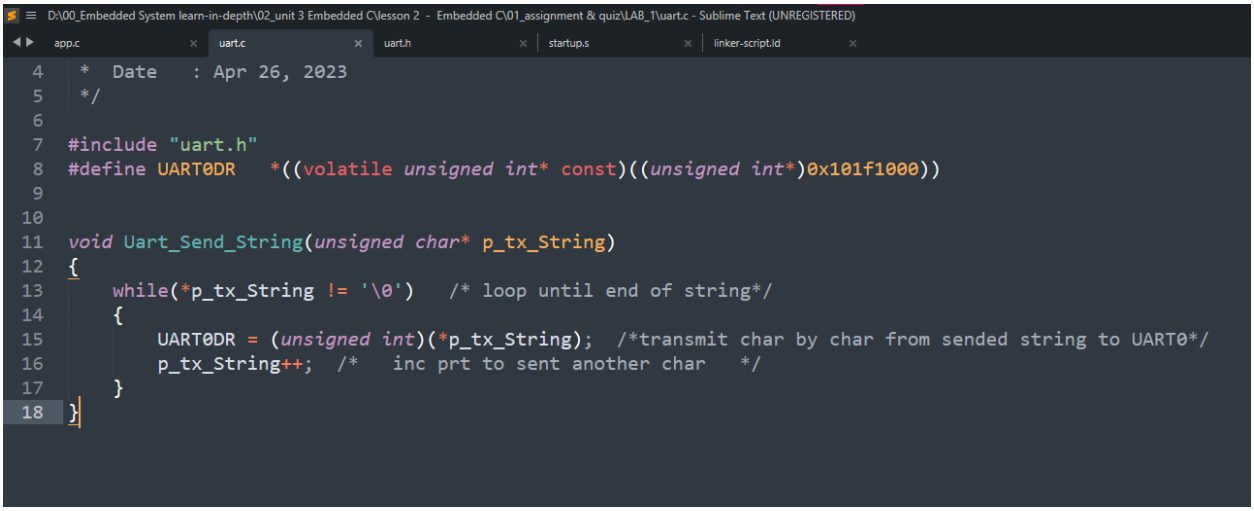
Name:	Mohamed belal
Unit:	Embedded C
Assignment:	2
Topic:	create a BareMetal software to send “learn-in-depth:MohamedBelal” using UART.

1 Source Code

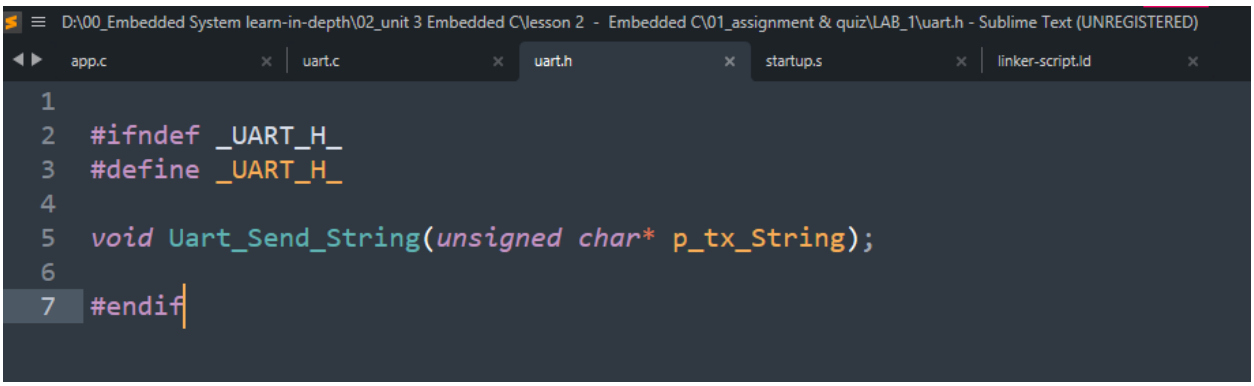
1.1 app.c



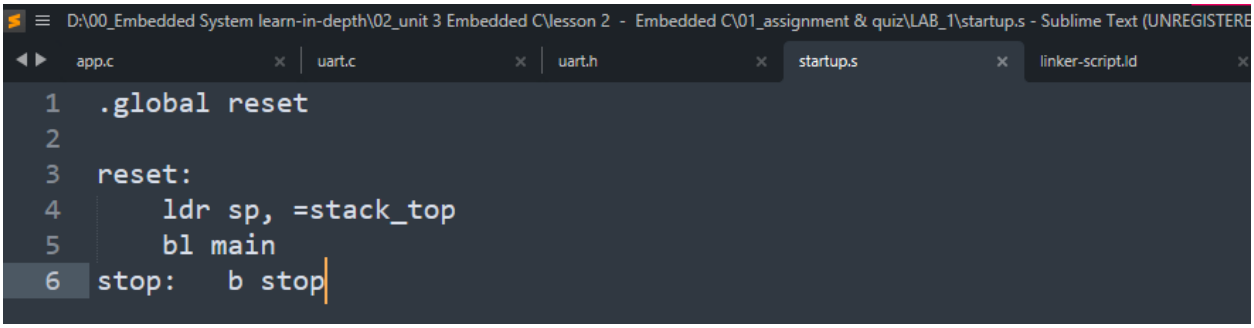
1.2 Uart.c



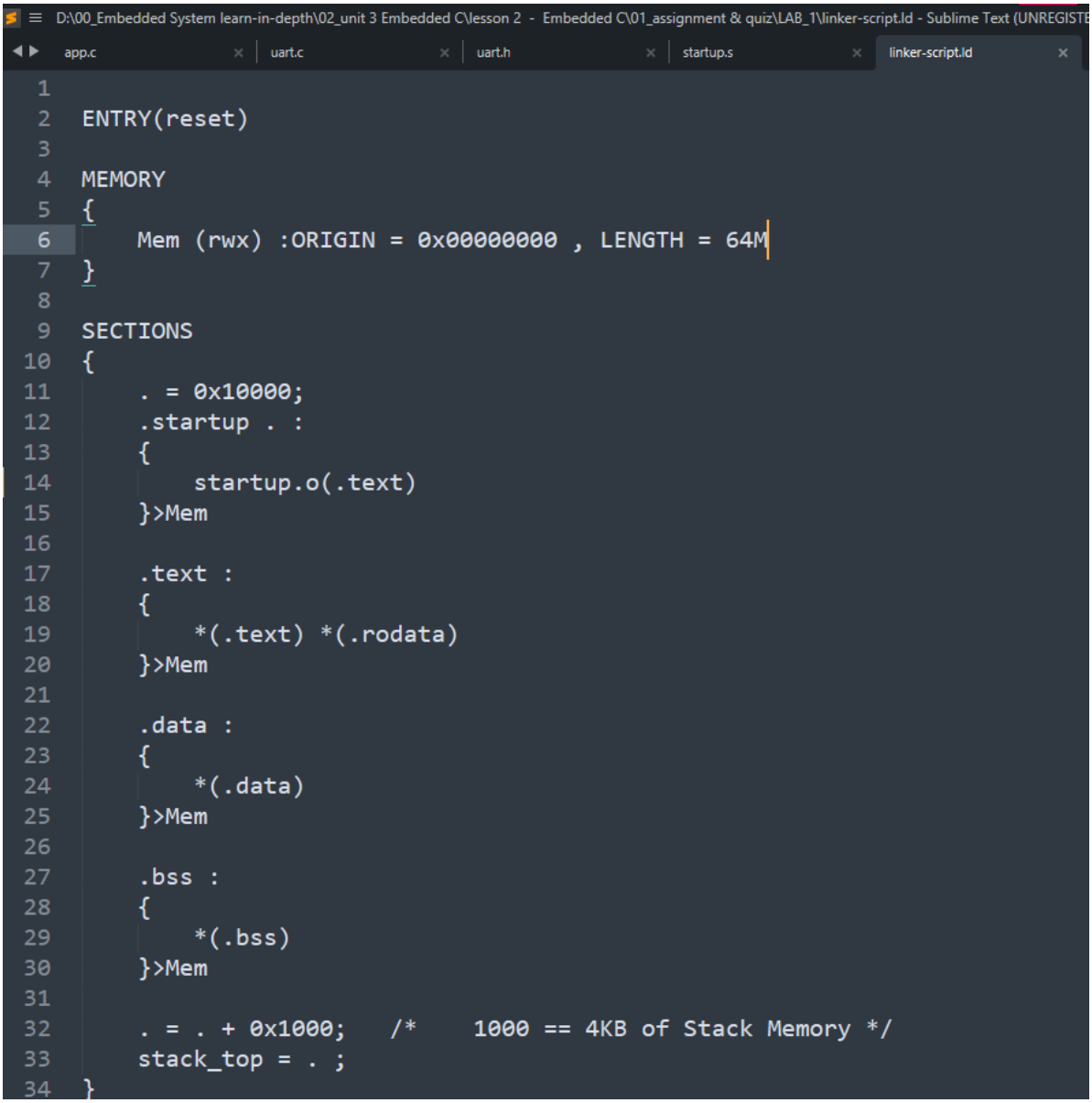
1.Uart.h



1.3 startup.s



1.5 linker_script.ld



2 get obj_file form App.c Uart .c included Uart.h

2.1 App.o

```
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . app.c -o app.o
```

2.2 Uart.o

```
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . uart.c -o uart.o
```

2.3 startup.o

```
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s startup.s -o startup.o
```

3 To show sections for object_file

3.1 app.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unif
edded C/01_assignment & quiz/LAB_1
$ 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 .rodata       00000064  00000000  00000000  000000b0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000012  00000000  00000000  00000114  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  00000126  2**0
                  CONTENTS, READONLY
```

3.2 uart.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_uni
edded C/01_assignment & quiz/LAB_1
$ 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 .comment      00000012  00000000  00000000  00000084  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000096  2**0
                  CONTENTS, READONLY
```

3.2 startup.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_u
edded C/01_assignment & quiz/LAB_1
$ 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
```

4 To show symbol table

4.1 app.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/OO_Embedded
edded C/01_assignment & quiz/LAB_1
$ arm-none-eabi-nm.exe app.o
00000000 T main
00000000 D string_buffer
00000000 R string_buffer2
         U Uart_Send_String
```

4.2 uart.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/OO_Er
edded C/01_assignment & quiz/LAB_1
$ arm-none-eabi-nm.exe uart.o
00000000 T Uart_Send_String
```

4.3 startup.o

```
moham@DESKTOP-4ID1J68 MINGW64 /d/OO_Embedded S
edded C/01_assignment & quiz/LAB_1
$ arm-none-eabi-nm.exe startup.o
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

5 use linker_script to get executable_file (App.elf) and map_file

```
arm-none-eabi-ld.exe -T linker-script.ld -Map=Map_file.map startup.o app.o uart.o
-o learn_in_depth.elf
```

6 To show sections for App.elf

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 2  -  Embedded C/01_assignment & quiz/LAB_1
$ arm-none-eabi-objdump.exe -h learn_in_depth.elf

learn_in_depth.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000000dc  00000000  00000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000064  000000dc  000000dc  000080dc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .ARM.attributes 0000002e  00000000  00000000  00008140  2**0
                  CONTENTS, READONLY
  3 .comment      00000011  00000000  00000000  0000816e  2**0
                  CONTENTS, READONLY
```

7 To show symbol table for App.elf

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 2  -  Embedded C/01_assignment & quiz/LAB_1
$ arm-none-eabi-nm.exe learn_in_depth.elf
00000010 T main
00000000 T reset
00001140 D stack_top
00000008 t stop
000000dc D string_buffer
00000078 T string_buffer2
00000028 T Uart_Send_String
```

8 get binary file to use in burn

```
arm-none-eabi-objcopy.exe -O binary learn_in_depth.elf learn_in_depth.bin
```

9 burn binary file on board using qemu

```
qemu-system-arm.exe -M versatilepb -m 128M -nographic -kernel learn_in_depth.bin
```

```
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded
$ qemu-system-arm.exe -M versatilepb -m 128M -nographic -kernel learn_in_depth.bin
learn-in-depth:MohamedBelal
```