8/20/2023

Assignment Lesson 2

Report of compilation, linking and simulation

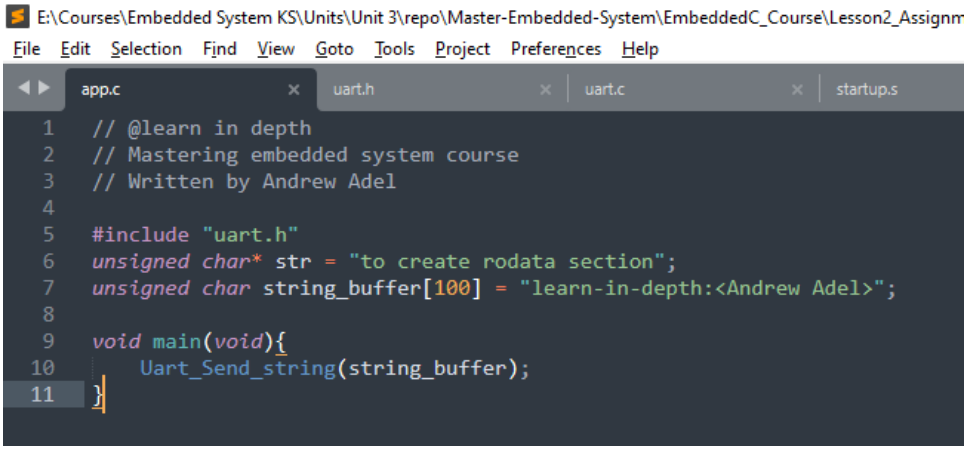
Name: Andrew Adel Hosny Goued

LEARN IN DEPTH DIPLOMA

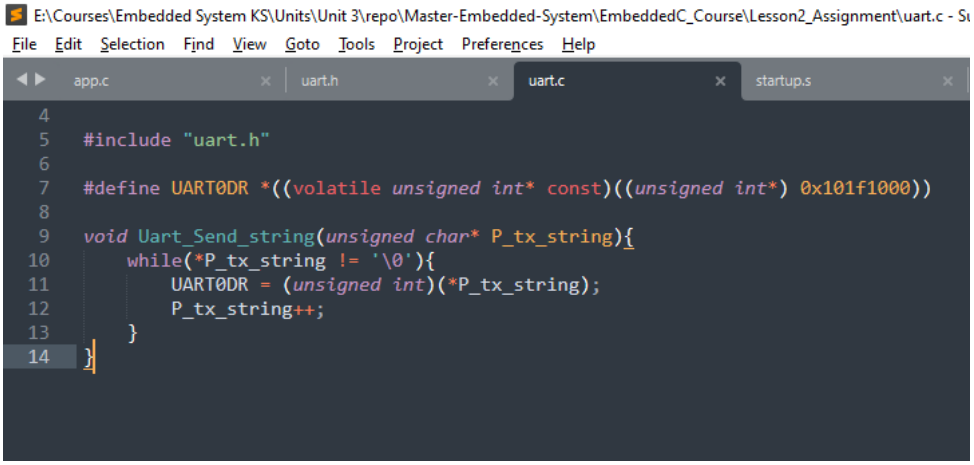
1) Contents

2) Files

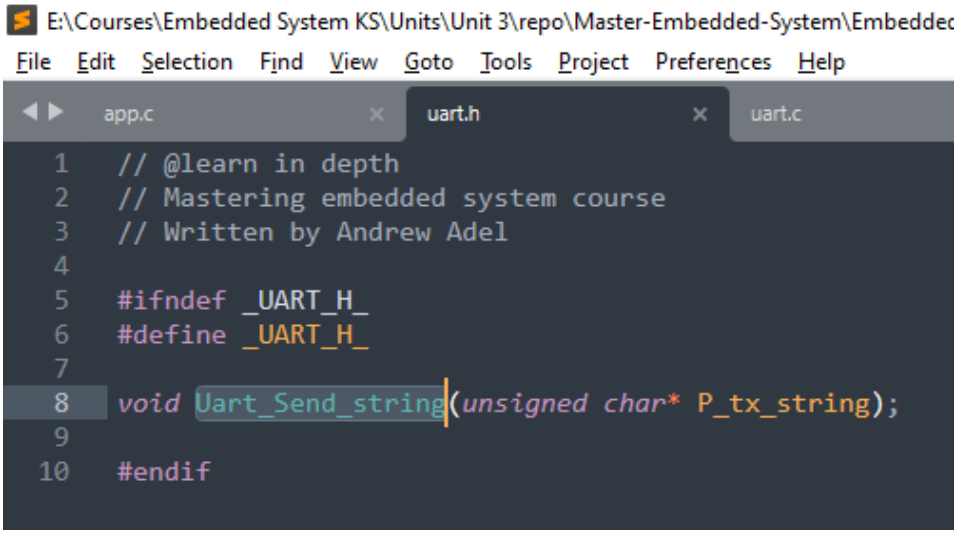
I- App.c



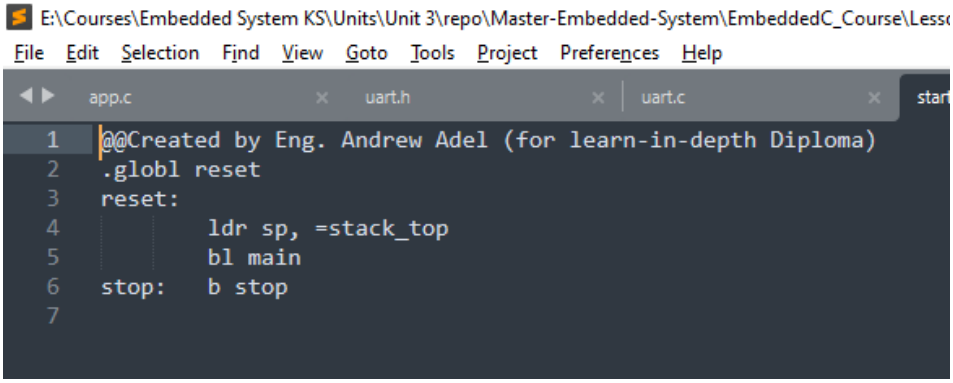
II- Uart.c



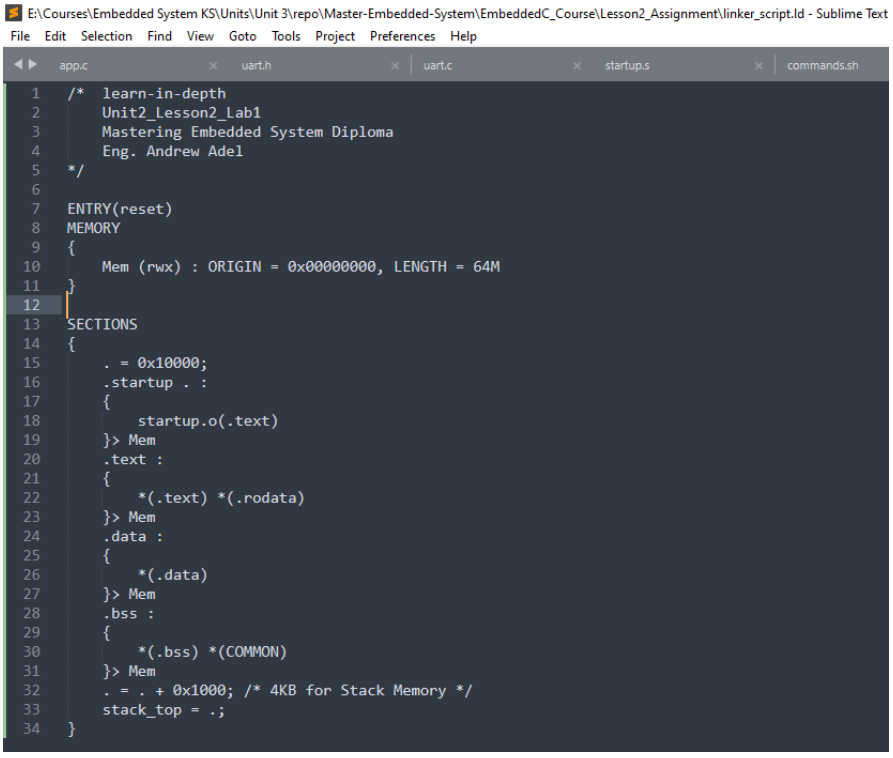
III- Uart.h



IV- Startup.c



V- Linker_script.ld



3) Compile files

I- Commands for compile

```
arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s app.c -o app.o
arm-none-eabi-gcc.exe -c -I . -mcpu=arm926ej-s uart.c -o uart.o
arm-none-eabi-as.exe  -mcpu=arm926ej-s -g startup.s -o startup.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         00000068  00000000  00000000  0000004c  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, DATA
  2 .bss          00000000  00000000  00000000  000000b4  2**0
                  ALLOC
  3 .rodata       0000001c  00000000  00000000  000000b4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000012  00000000  00000000  000000d0  2**0
                  CONTENTS, READONLY
  5 .ARM.attributes 00000032  00000000  00000000  000000e2  2**0
                  CONTENTS, READONLY
```

b- 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
```

c- 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 .debug_line   0000003a  00000000  00000000  00000066  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_info   0000009d  00000000  00000000  000000a0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_abbrev 00000014  00000000  00000000  0000013d  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00000158  2**3
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

III- Show symbol table for object files

```
show symbol table for all object files
--------------------------------------
app.o:
00000000 T main
00000000 D str
00000004 D string_buffer
         U Uart_Send_string
uart.o:
00000000 T Uart_Send_string
startup.o:
         U main
00000000 T reset
         U stack_top
00000008 t stop
```

4) Use linker_script to get executable_file and mab file

I- Command

```
arm-none-eabi-ld.exe -T linker_script.ld startup.o app.o uart.o -o learn-in-depth.elf -Map=Map_file.map
```

II- Show sections for learn-in-depth.elf

```
learn-in-depth.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .startup      00000010  00010000  00010000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text         00000084  00010010  00010010  00008010  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data         00000068  00010094  00010094  00008094  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000080fc  2**0
                  CONTENTS, READONLY
  4 .comment      00000011  00000000  00000000  0000812a  2**0
                  CONTENTS, READONLY
  5 .debug_line   0000003a  00000000  00000000  0000813b  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_info   0000009d  00000000  00000000  00008175  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_abbrev 00000014  00000000  00000000  00008212  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000020  00000000  00000000  00008228  2**3
                  CONTENTS, READONLY, DEBUGGING
```

III- Show symbol table of learn-in-depth.elf

```
analyze the executable file:
00010010 T main
00010000 T reset
000110fc D stack_top
00010008 t stop
00010094 D str
00010098 D string_buffer
00010028 T Uart_Send_string
```

5) Binary file to use in burn

I- Command for get binary file

```
arm-none-eabi-objcopy -O binary learn-in-depth.elf learn-in-depth.bin
```

II- Burn binary file on board using qemu

a. Command

```
qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
```

b. Running