

Embedded c Lesson 2 Lab 1

Application

send a string using UART protocol in versatilepb microcontrol chip based on arm926ej-s micro-processor using qemu.

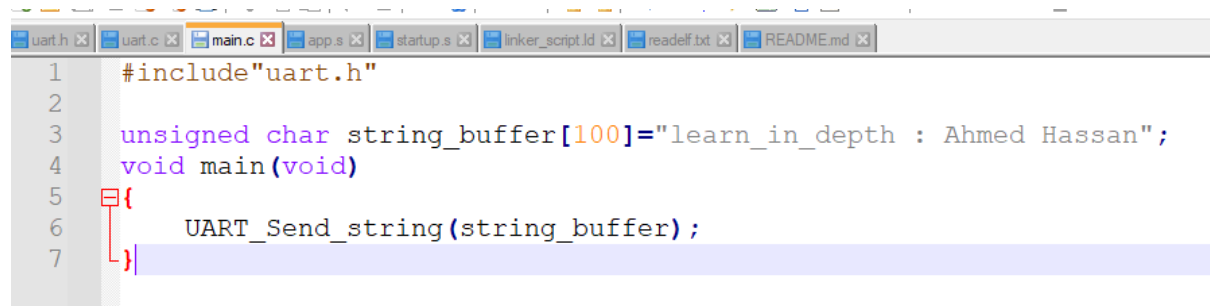
This application will be done from scratch including startup code and linker script.

This app will compile in arm cross tool chain arm-none-eabi.

This lapp will be simulated in qemu.

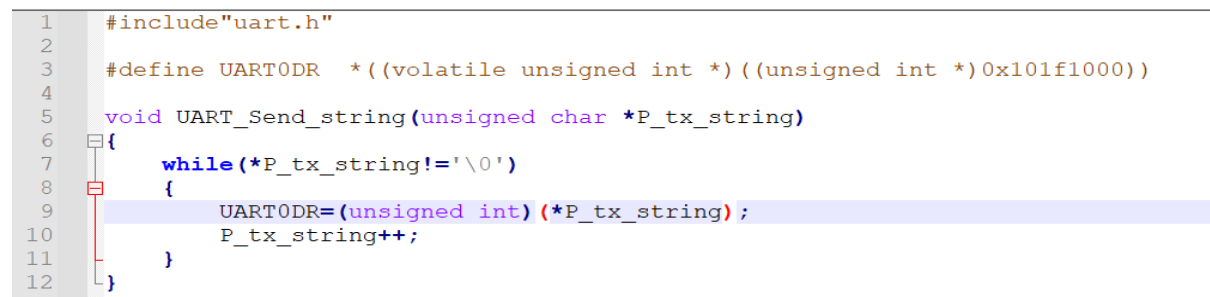
Firstly :c code of main.c for application , uart.c and uart.h

1- main.c



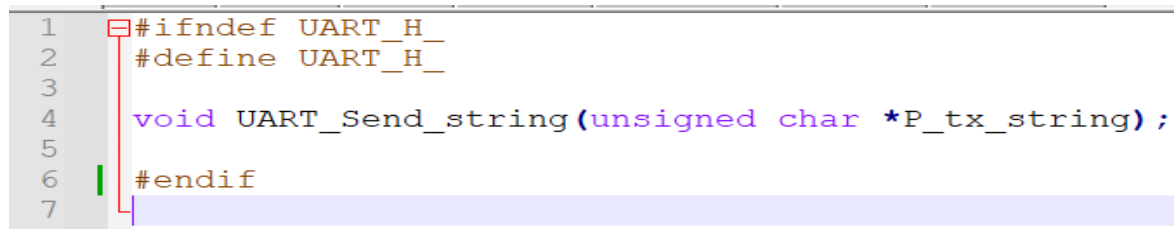
```
1 #include "uart.h"
2
3 unsigned char string_buffer[100]="learn_in_depth : Ahmed Hassan";
4 void main(void)
5 {
6     UART_Send_string(string_buffer);
7 }
```

2- uart.c



```
1 #include "uart.h"
2
3 #define UART0DR *((volatile unsigned int *) ((unsigned int *) 0x101f1000))
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 }
```

3- uart.h



```
1 #ifndef UART_H_
2 #define UART_H_
3
4 void UART_Send_string(unsigned char *P_tx_string);
5
6 #endif
7
```

4-startup code

```
1  @@ Eng. Ahmed Hassan (Learn_in_Depth.com)
2
3  .globl reset
4
5  reset:
6      ldr sp, = stack_top
7      bl main
8  stop:  b stop
9
```

5-linker script:

```
1  /*LEARN_IN_DEPTH
2  UNIT2_LESSON2_LAP1
3  ENG: AHMED HASSAN
4  */
5
6  ENTRY(reset)
7
8  MEMORY
9  {
10     Mem (rwx) : ORIGIN = 0x00000000, LENGTH = 64M
11 }
12
13 SECTIONS
14 {
15     . = 0x10000;
16     .startup . :
17     {
18         startup.o(.text)
19     }> Mem
20     .text :
21     {
22         *(.text) *(.rodata)
23     }> Mem
24     .data :
25     {
26         *(.data)
27     }> Mem
28     .bss :
29     {
30         *(.bss) *(.COMMON)
31     }> Mem
32     . = . + 0x1000; /*4kB of stack memory*/
33     stack_top = . ;
34 }
```

6-get main.o , uart.o startup.o:

```
MINGW64:/d/New folder/lap 1
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . main.c -o main.o

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-gcc.exe -c -mcpu=arm926ej-s -I . uart.c -o uart.o

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-as.exe -mcpu=arm926ej-s startup.s -o startup.o

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
```

7-sections in main.o

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-objdump.exe -h main.o

main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          0000001c  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data          00000064  00000000  00000000  00000050  2**2
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  000000b4  2**0
    ALLOC
  3 .comment       0000004a  00000000  00000000  000000b4  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 0000002c  00000000  00000000  000000fe  2**0
    CONTENTS, READONLY
```

8-sections in uart.o

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 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          00000058  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          00000000  00000000  00000000  0000008c  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  0000008c  2**0
    ALLOC
  3 .comment       0000004a  00000000  00000000  0000008c  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 0000002c  00000000  00000000  000000d6  2**0
    CONTENTS, READONLY
```

9- sections in startup.o

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 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
```

10- symbol table in main.c ,uart.c and startup.c

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-nm.exe main.o
00000000 T main
00000000 D string_buffer
          U UART_Send_string

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-nm.exe uart.o
00000000 T UART_Send_string

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-nm.exe startup.o
          U main
00000000 T reset
          U stack_top
00000008 t stop
```

11- link all object file in one obj file : "learn_in_depth.elf"

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-ld.exe -T linker_script.ld -Map=output.map startup.o main.o uart.o -o learn-in-depth.elf

Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$
```

12- section in " learn_in_depth.elf "

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 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 .startup       00000010  00010000  00010000  00010000  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .text          00000074  00010010  00010010  00010010  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data          00000064  00010084  00010084  00010084  2**2
    CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000100e8  2**0
    CONTENTS, READONLY
  4 .comment        00000049  00000000  00000000  00010116  2**0
    CONTENTS, READONLY
```

14-git bin code from " learn_in_depth.elf " to run it in qemu

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-objcopy.exe -O binary learn-in-depth.elf learn-in-depth.bin
```

15-symbol table in " learn_in_depth.elf "

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ arm-none-eabi-nm.exe learn-in-depth.elf
00010010 T main
00010000 T reset
000110e8 D stack_top
00010008 t stop
00010084 D string_buffer
0001002c T UART_Send_string
```

16- Run " learn_in_depth.bin" in qemu to show output in uart0:

```
Egypt_Laptop@DESKTOP-P909616 MINGW64 /d/New folder/lap 1
$ qemu-system-arm -M versatilepb -m 128M -nographic -kernel learn-in-depth.bin
learn_in_depth : Ahmed Hassan
```