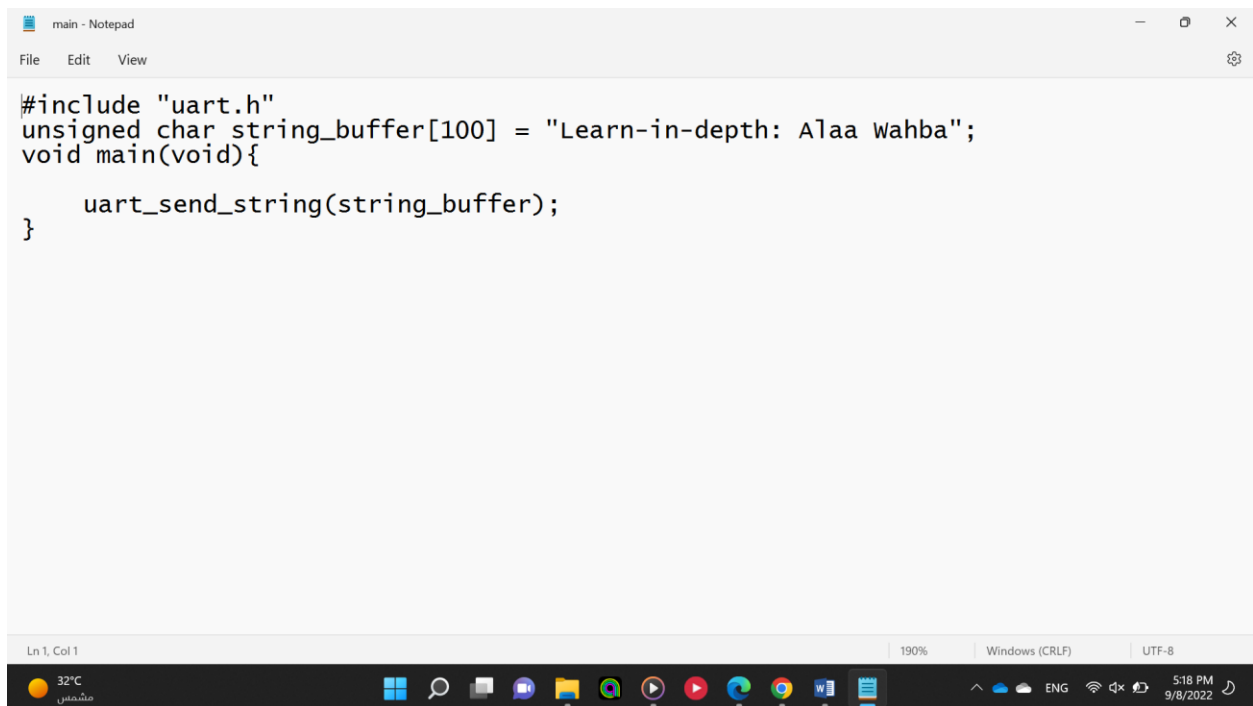


Main.c



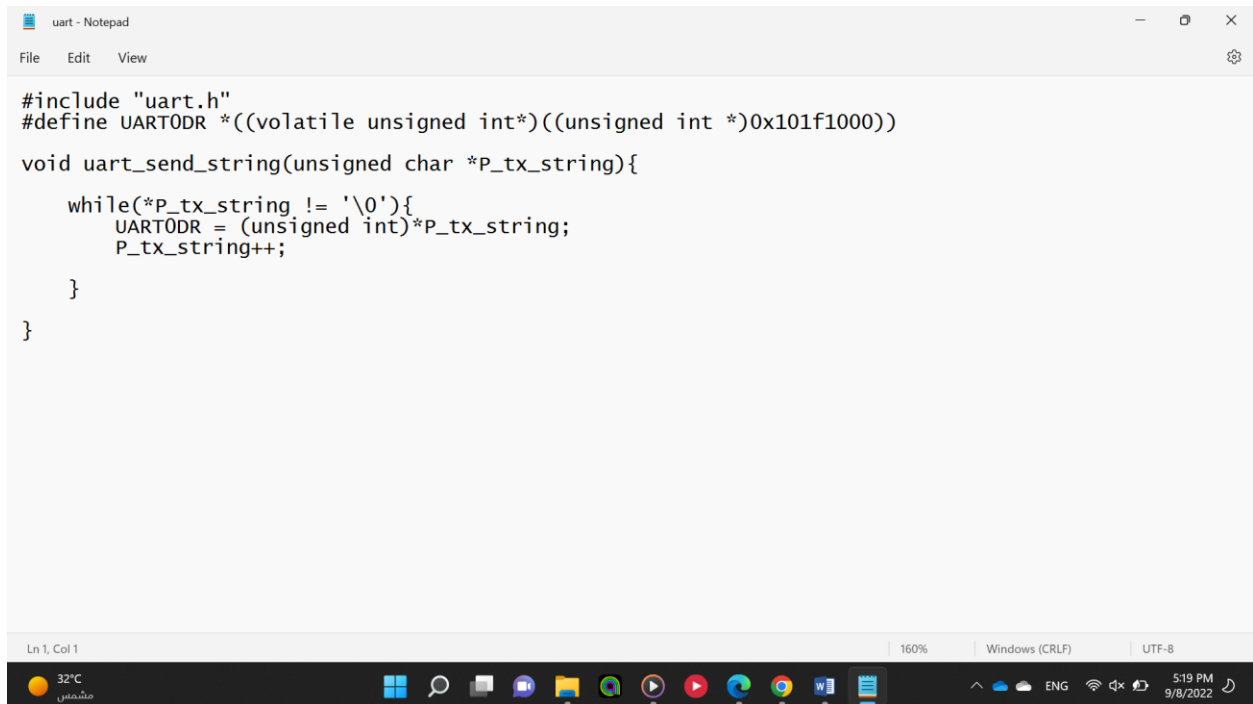
```
main - Notepad
File Edit View

#include "uart.h"
unsigned char string_buffer[100] = "Learn-in-depth: Alaa Wahba";
void main(void){

    uart_send_string(string_buffer);
}

Ln 1, Col 1 190% Windows (CRLF) UTF-8
32°C مشمس 5:18 PM 9/8/2022
```

Uart.c



```
uart - Notepad
File Edit View

#include "uart.h"
#define UART0DR *((volatile unsigned int*)((unsigned int *)0x101f1000))

void uart_send_string(unsigned char *P_tx_string){
    while(*P_tx_string != '\0'){
        UART0DR = (unsigned int)*P_tx_string;
        P_tx_string++;
    }
}

Ln 1, Col 1 160% Windows (CRLF) UTF-8
32°C مشمس
5:19 PM 9/8/2022
```

Uart.h

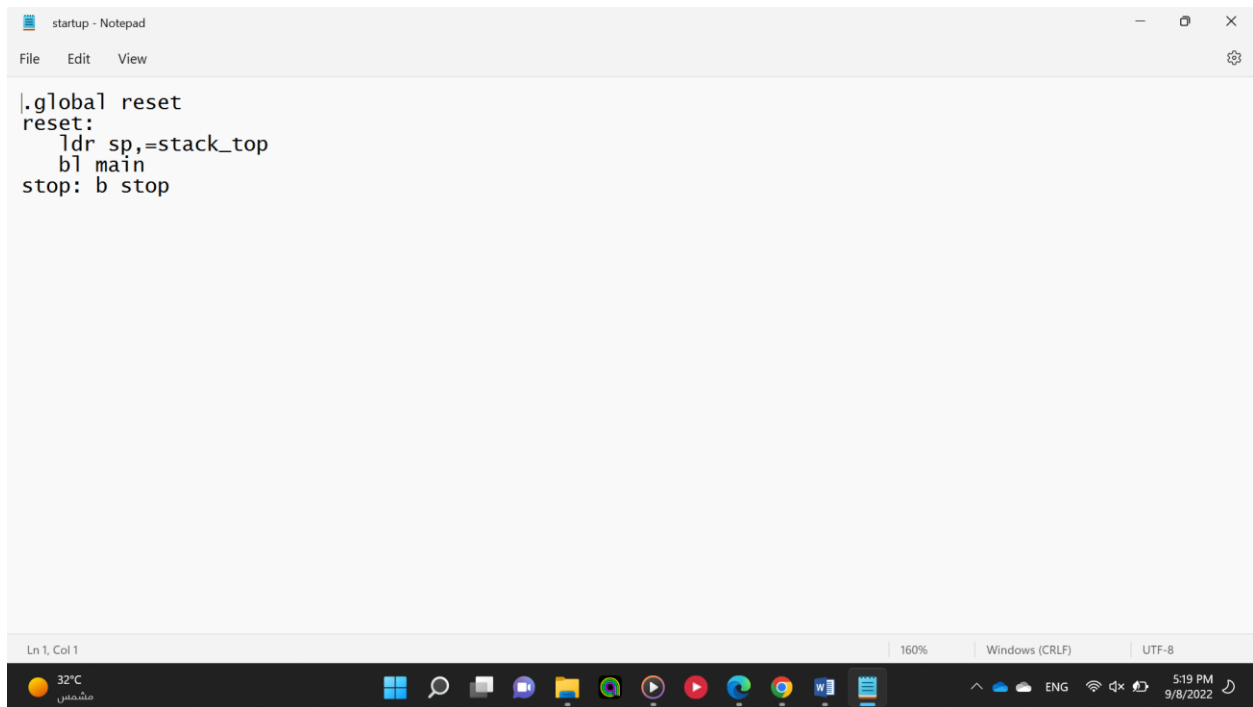


The image shows a Windows Notepad window titled 'uart - Notepad'. The window has a menu bar with 'File', 'Edit', and 'View'. The main text area contains the following C code snippet:

```
#ifndef _UART_H
#define _UART_H
void uart_send_string(unsigned char *P_tx_string);
#endif
```

At the bottom of the Notepad window, a status bar shows 'Ln 1, Col 1', '160%', 'Windows (CRLF)', and 'UTF-8'. Below the Notepad window is the Windows taskbar, which includes a weather widget showing '32°C' and 'مشمس' (Sunny), a search icon, and several application icons (File Explorer, Edge, Word, etc.). The system tray on the right shows network, volume, and power icons, along with the date and time: '5:19 PM 9/8/2022'.

Startup.s



```
startup - Notepad
File Edit View

|.global reset
reset:
    ldr sp,=stack_top
    bl main
stop: b stop

Ln 1, Col 1
160% Windows (CRLF) UTF-8
32°C 5:19 PM 9/8/2022
```

Linker_script.ld

```
linker_script - Notepad
File Edit View
ENTRY(reset)
MEMORY
{
    Mem(rwx) : ORIGIN = 0x00000000, LENGTH = 64M
}
SECTIONS
{
    . = 0x10000;
    .startup . :
    {
        startup.o(.text)
    }> Mem
    .text :
    {
        *(.text) *(.rodata)
    }> Mem
    .data :
    {
        *(.data)
    }> Mem
    .bss :
    {
        *(.bss) *(COMMON)
    }> Mem
    . = . + 0x1000;
    stack_top = .;
}
```

Ln 1, Col 1 130% Windows (CRLF) UTF-8

32°C مشمس 5:19 PM 9/8/2022

Main.o section

```
Alaa@Pizza MINGW32 /d/Lab1
$ 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        0000007f  00000000  00000000  000000b4  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000133  2**0
    CONTENTS, READONLY
```

Uart.o sections

```
Alaa@Pizza MINGW32 /d/Lab1
$ 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          00000054  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          00000000  00000000  00000000  00000088  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  00000088  2**0
    ALLOC
  3 .comment        0000007f  00000000  00000000  00000088  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  00000107  2**0
    CONTENTS, READONLY
```

Startup.o sections

```
Alaa@Pizza MINGW32 /d/Lab1
$ 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
```

Lab1.elf sections

```
Alaa@Pizza MINGW32 /d/Lab1
$ arm-none-eabi-objdump.exe -h lab1.elf

lab1.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           00000070  00010010  00010010  00010010  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .data           00000064  00010080  00010080  00010080  2**2
    CONTENTS, ALLOC, LOAD, DATA
  3 .ARM.attributes 0000002e  00000000  00000000  000100e4  2**0
    CONTENTS, READONLY
  4 .comment         0000007e  00000000  00000000  00010112  2**0
    CONTENTS, READONLY
```

Main.o & Uart.o symbols

```
$  
Alaa@Pizza MINGW32 /d/Lab1  
$ arm-none-eabi-nm.exe main.o  
00000000 T main  
00000000 D string_buffer  
          U uart_send_string  
  
Alaa@Pizza MINGW32 /d/Lab1  
$ arm-none-eabi-nm.exe uart.o  
00000000 T uart_send_string
```

Lab1.elf symbols

```
Alaa@Pizza MINGW32 /d/Lab1  
$ arm-none-eabi-nm.exe lab1.elf  
  
00010010 T main  
00010000 T reset  
000110e4 D stack_top  
00010008 t stop  
00010080 D string_buffer  
0001002c T uart_send_string
```