

LAB TWO (Third embedded C Lecture)

1- Debugging

Using gdb to debug .elf without debugging information

```
salah@DESKTOP-OJRSJCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1
$ arm-none-eabi-gdb.exe learn-in-depth.elf
GNU gdb (GDB) 7.5.1
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=arm-none-eabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec10\lab1\lea
rn-in-depth.elf...(no debugging symbols found)...done.
(gdb)
```

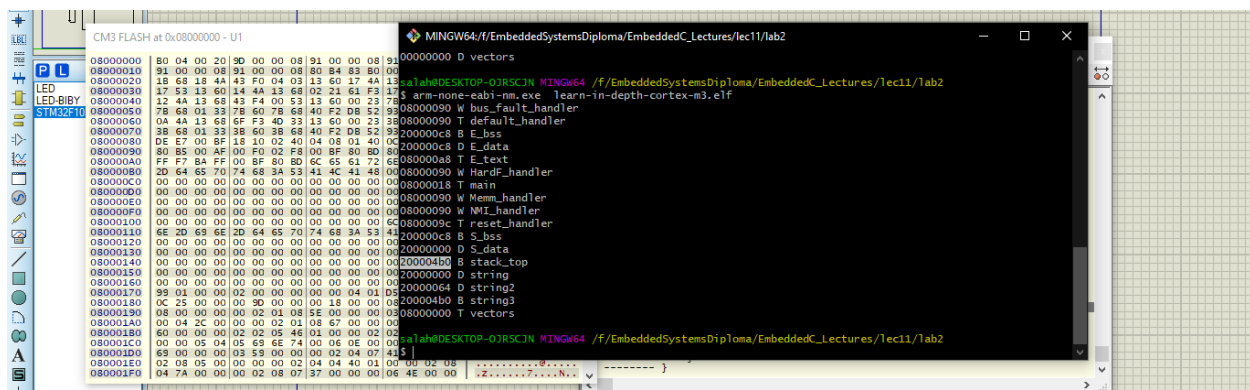
```
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
0x00000000 in ?? ()
(gdb) |
```

Using gdb to debug .elf with debugging information

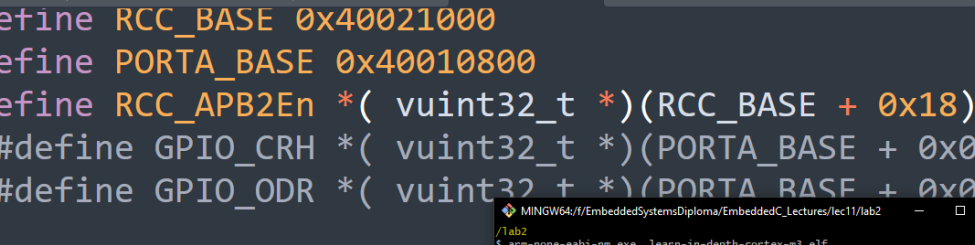
```
MINGW64/f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1
salah@DESKTOP-OJRSJCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1
$ arm-none-eabi-gdb.exe learn-in-depth.elf
GNU gdb (GDB) 7.5.1
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=arm-none-eabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec10\lab1\learn-in-depth.elf...done.
(gdb)
```

```
MINGW64/f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1
salah@DESKTOP-OJRSJCJN MINGW64 /f/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec10/lab1
$ arm-none-eabi-gdb.exe learn-in-depth.elf
GNU gdb (GDB) 7.5.1
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=arm-none-eabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec10\lab1\learn-in-depth.elf...done.
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
reset () at startup.s:3
3       ldr sp,=stack_address
(gdb) |
```

Adding vector section(vector table) on startup file using assembly code



Using the weak and alias attributes but without overriding



```
F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec11\lab2\main.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

volatile int *p = (volatile int *)0x121232312; startup.c main.c
2 #define RCC_BASE 0x40021000
3 #define PORTA_BASE 0x40010800
4 #define RCC_APB2En *(vuint32_t*)(RCC_BASE + 0x18)
5 // #define GPIO_CRH *(vuint32_t*)(PORTA_BASE + 0x04)
6 // #define GPIO_ODR *(vuint32_t*)(PORTA_BASE + 0x0C)
7
8 /*extern void NMI_handler()
9 {
10
11 }*/
12 typedef union
13 {
14     uint32_t ALL_Pins;
15     struct
```

MINGW64: /F/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec11/lab2

/lab2

arm-none-eabi-nm.exe learn-in-depth-cortex-m3.elf

08000098 W bus_fault_handler

08000098 T default_handler

20000000 B _bss

20000000 D _data

08000128 T _text

08000098 W HardF_handler

08000018 T main

08000098 W Mem_handler

08000098 W NMI_handler

20000000 D preg_CRH

20000004 D preg_ODR

08000044 T reset_handler

20000000 B _bss

20000000 D _data

200004b8 B stack_top

20000008 D string

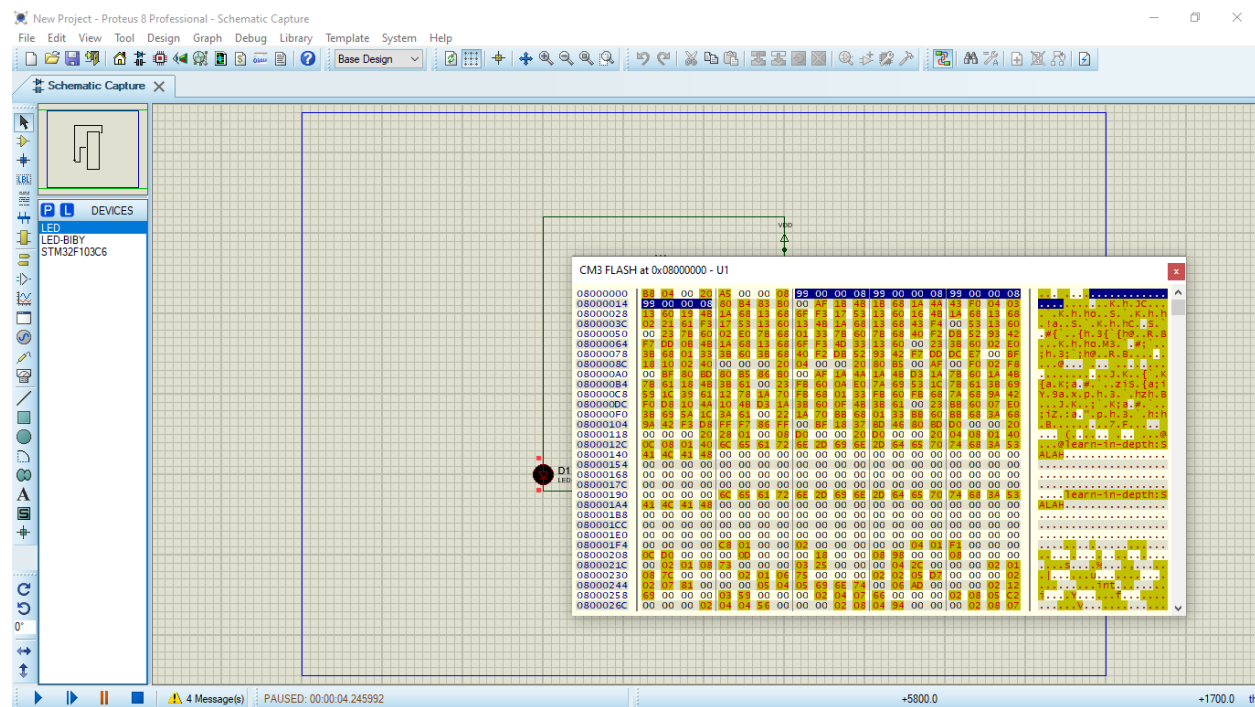
2000000c D string2

200004b8 D string3

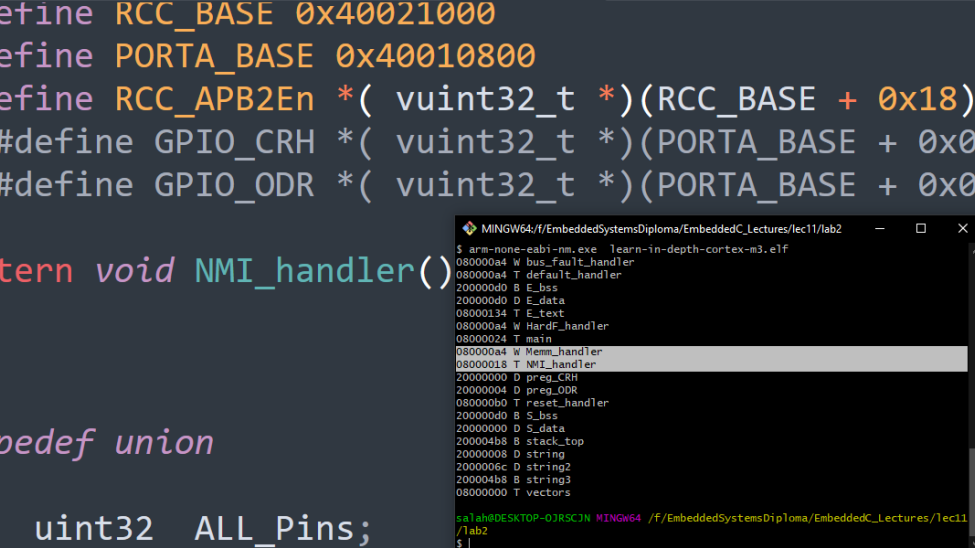
08000000 T vectors

salah@DESKTOP-DJRS3CN MINGW64 /F/EmbeddedSystemsDiploma/EmbeddedC_Lectures/lec11/lab2

All handler functions point to the same address on flash memory



Using the weak and alias attributes but with overriding

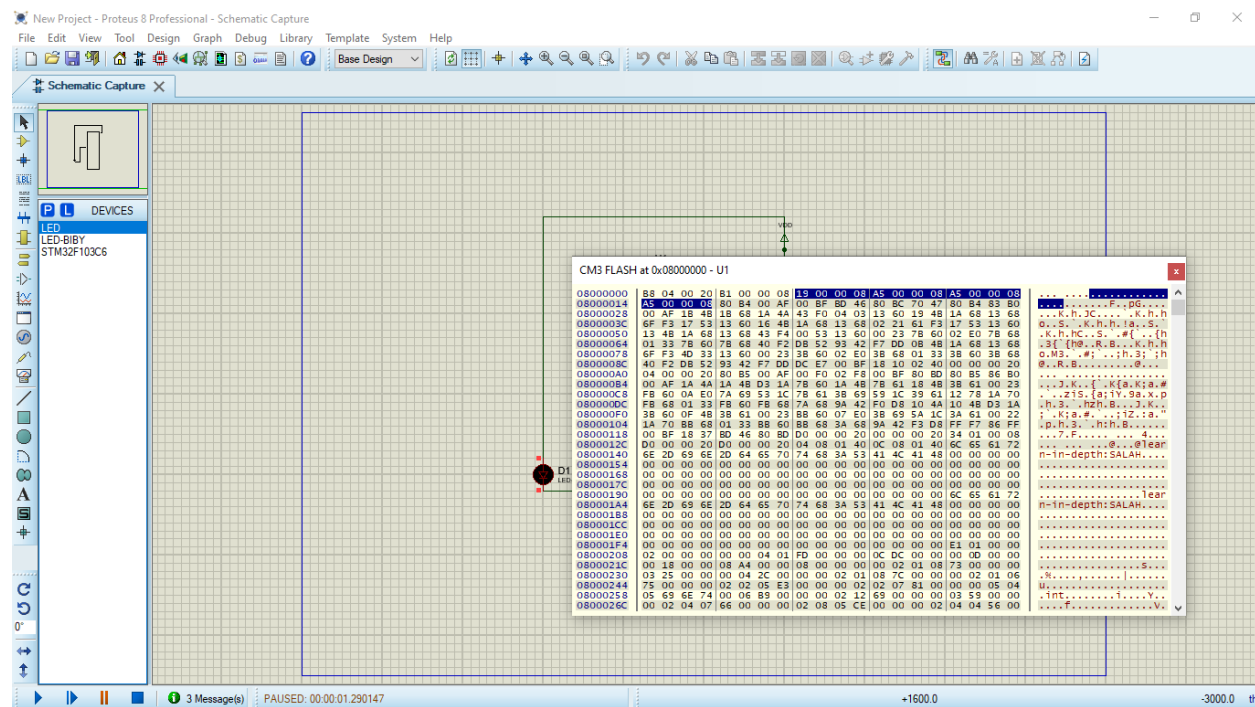


```
F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec1\lab2\main.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

volatile int *p = (volatile int *)0x121232312; startup.c main.c
2 #define RCC_BASE 0x40021000
3 #define PORTA_BASE 0x40010800
4 #define RCC_APB2En *(vuint32_t*)(RCC_BASE + 0x18)
5 // #define GPIO_CRH *(vuint32_t*)(PORTA_BASE + 0x04)
6 // #define GPIO_ODR *(vuint32_t*)(PORTA_BASE + 0x0C)
7
8 extern void NMI_handler()
9 {
10
11 }
12 typedef union
13 {
14     uint32_t ALL_Pins;
15     struct
```

```
MINGW64/F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec1\lab2
$ arm-none-eabi-nm.exe learn-in-depth-cortex-m3.elf
08000004 T bus_fault_handler
08000004 T default_handler
200000d0 B E_bss
200000d0 D E_data
08000134 T E_text
08000004 W HardF_handler
08000024 T main
08000004 W Memm_handler
08000018 T NMI_handler
20000000 D preg_CRH
20000004 D preg_ODR
080000b0 T reset_handler
200000d0 B S_bss
20000000 D S_data
200004b8 B stack_top
20000008 D string
2000006c D string2
200004b8 B string3
08000000 T vectors
c:\lab2\SKTOP-07R3CJN MINGW64 /F:\EmbeddedSystemsDiploma\EmbeddedC_Lectures\lec1\lab2
$ |
```

All handler functions point to the same address on flash memory except NMI



[illegible]

The screenshot displays the STM32CubeIDE environment. The 'Schematic Capture' window is active, showing a memory dump for CM3 RAM at address 0x20000000. The dump consists of a grid of hexadecimal values, mostly zeros, with some non-zero values at the end. The 'CM3 Source Code' window is also open, showing the 'startup.c' file. The line '*(uint8*)pointer_to_ram+=*(uint8*)pointer_to_flash++;' is highlighted in blue. The background shows a schematic diagram of the STM32 microcontroller.

Run the program on the proteus simulator

New Project - Proteus 8 Professional - Schematic Capture

File Edit View Tool Design Graph Debug Library Template System Help

CM3 Variables - U1

| Name | Address | Value |
|----------|--------------|--------------|
| preg_ODR | 20000004 | 0x4001080C |
| preg_CRH | 4001080C | 0x00 0x2... |
| S_Pins | 4001080C | 0x00 0x2... |
| CRH | 4001080C | 1 |
| string | 20000008 | byte[100] |
| string2 | 2000000C | byte[100] |
| string3 | 20000488 | byte[100] |
| vectors | 08000000 | dword[6] |
| preg_CRH | 20000000 | 0x40010804 |
| preg_CRH | 40010804 | 0x44 0x44... |
| ALL... | 40010804 | 1143227460 |
| S_Pins | 40010804 | 0x44 0x44... |
| p1n13 | 40010804 | 0 |
| CRH | 40010804 | 2 |
| 1 | BP+12 = 0... | 1500 |
| 1 | BP+16 = 0... | 1500 |

CM3 Source Code - U1

```
main.c
-----
uint32 p1n13=1;
uint32 i;
uint32 CRH=4;
}S_Pins;
}MyReg;
volatile MyReg *preg_CRH=(volatile MyReg *) (PORTA_BASE + 0x04);
volatile MyReg *preg_ODR=(volatile MyReg *) (PORTA_BASE + 0x0C);
volatile unsigned char string[100]="learn-in-depth: SALAH";
volatile unsigned const char string2[100]="learn-in-depth: SALAH";
volatile unsigned char strings[100];

int main(void)
{
    RCC_APB2ENR |=1<<2;
    preg_CRH->S_Pins.CRH=0;
    preg_CRH->S_Pins.CRH=2;
    /*GPIO_CRH=0xFFFFFFF;
    GPIO_CRH=0x00200000;*/

    while(1)
    {
        preg_ODR->S_Pins.p1n13=1;
        for(int i=0;i<1500;i++);
        preg_ODR->S_Pins.p1n13=0;
        for(int i=0;i<1500;i++);
    }
}
```

3 Message(s) [U1_CM3CORE] Digital breakpoint at time 127.44ms (500.00ns elapsed) - Function S -1400.0 +2900.0 th

New Project - Proteus 8 Professional - Schematic Capture

File Edit View Tool Design Graph Debug Library Template System Help

CM3 Variables - U1

| Name | Address | Value |
|----------|--------------|--------------|
| preg_ODR | 20000004 | 0x4001080C |
| preg_CRH | 4001080C | 0x00 0x0... |
| S_Pins | 4001080C | 0x00 0x0... |
| CRH | 4001080C | 0 |
| string | 20000008 | byte[100] |
| string2 | 2000000C | byte[100] |
| string3 | 20000488 | byte[100] |
| vectors | 08000000 | dword[6] |
| preg_CRH | 20000000 | 0x40010804 |
| preg_CRH | 40010804 | 0x44 0x44... |
| ALL... | 40010804 | 1143227460 |
| S_Pins | 40010804 | 0 |
| p1n13 | 40010804 | 2 |
| CRH | 40010804 | 2 |
| 1 | BP+12 = 0... | 1500 |
| 1 | BP+16 = 0... | 1500 |

CM3 Source Code - U1

```
main.c
-----
uint32 p1n13=1;
uint32 i;
uint32 CRH=4;
}S_Pins;
}MyReg;
volatile MyReg *preg_CRH=(volatile MyReg *) (PORTA_BASE + 0x04);
volatile MyReg *preg_ODR=(volatile MyReg *) (PORTA_BASE + 0x0C);
volatile unsigned char string[100]="learn-in-depth: SALAH";
volatile unsigned const char string2[100]="learn-in-depth: SALAH";
volatile unsigned char strings[100];

int main(void)
{
    RCC_APB2ENR |=1<<2;
    preg_CRH->S_Pins.CRH=0;
    preg_CRH->S_Pins.CRH=2;
    /*GPIO_CRH=0xFFFFFFF;
    GPIO_CRH=0x00200000;*/

    while(1)
    {
        preg_ODR->S_Pins.p1n13=1;
        for(int i=0;i<1500;i++);
        preg_ODR->S_Pins.p1n13=0;
        for(int i=0;i<1500;i++);
    }
}
```

3 Message(s) [U1_CM3CORE] Digital breakpoint at time 130.25ms (1.6250us elapsed) - Function S -1400.0 +2900.0 th