

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

ІКНІ
Кафедра ПЗ

ЗВІТ

До лабораторної роботи № 7

З дисципліни: *“ Основи програмування вбудованих систем ”*

На тему: *“Манування пам'яті та робота зі scatter-файлом”*

Лектор:

доц. каф. ПЗ
Марусенкова Т.А.

Виконав:

ст. гр. ПЗ-32
Бурець В.В.

Прийняв:

доц. каф. ПЗ
Крук О.Г.

« ____ » _____ 2021 р.

Σ = ____ .

Львів-2021

Тема роботи: Мапування пам'яті та робота зі scatter-файлом.

Мета роботи: Навчитися використовувати можливості лінкувальника та розташовувати дані за потрібними адресами.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Як відкрити scatter-файл?

Щоб доступитися до scatter-файлу, необхідно на вкладці Linker діалогу “Options for Target ...” скинути прапорець “Use Memory Layout from Target Layout”, після чого натиснути кнопку “Edit” і закрити діалог

ХІД РОБОТИ

Вміст файлу Icons.c

```
const unsigned char img_upArrow_bmp_char_table[] =
{ 0x00, 0xFE, 0x02, 0x02, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xFC, 0x80, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x3C, 0xE0, 0x00, 0x00, 0x00, 0xF8, 0x0C, 0x00,
  0x00, 0xFF, 0x82, 0x87, 0x8F, 0xFC, 0x00, 0x7E, 0x40, 0x7E, 0x00, 0x7E,
  0x02, 0x02, 0x00, 0x7E, 0x4A, 0x4A, 0x00, 0x7F, 0x40, 0x40, 0x00, 0x4E,
  0x4A, 0x7A, 0x00, 0x40, 0x00, 0x01, 0x3F, 0x7C, 0x3F, 0x01, 0x00, 0x00};
```

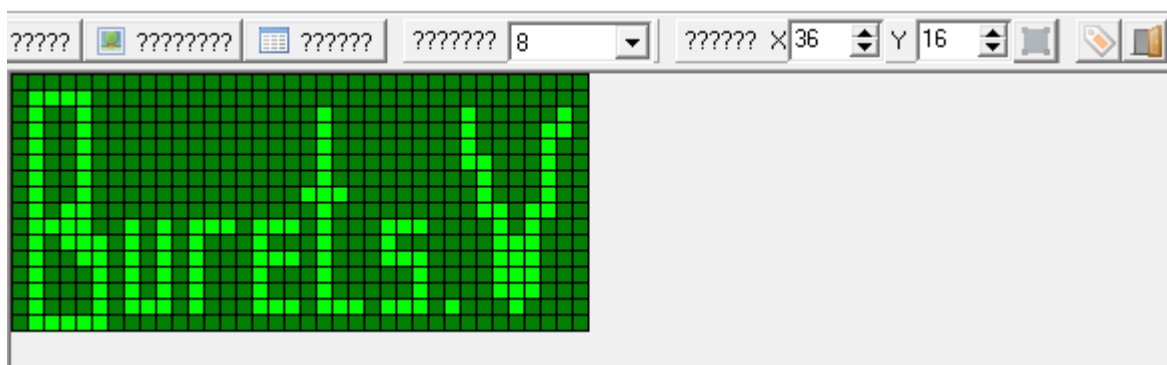


Рис.1 . Прізвище та ініціали

```
const unsigned char img_downArrow_bmp_char_table[] =
{ static const char lcd_image_mas[144] =
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x18, 0x04, 0x06,
  0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x82, 0xC2, 0x32, 0x1A, 0x0E,
  0x02, 0x0E, 0x08, 0x38, 0x60, 0x40, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x10, 0x98, 0xFC, 0x16, 0x13, 0x10, 0x10, 0x10, 0x10, 0xF0, 0xF0,
  0x10, 0x10, 0x10, 0x10, 0x18, 0x1E, 0x13, 0x11, 0x10, 0x10, 0xF0, 0x10,
  0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x11, 0x13, 0x12, 0xF4, 0x1C, 0x00,
  0x00, 0x00, 0xFF, 0x00, 0x00, 0x4E, 0x5A, 0x52, 0x70, 0x00, 0x7F, 0x0F,
  0x08, 0x78, 0x00, 0x00, 0x7E, 0x42, 0x46, 0x7E, 0x00, 0x00, 0xFF, 0x42,
  0x42, 0x66, 0x3C, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x7F, 0x00, 0x00,
  0x00, 0x00, 0x1F, 0x18, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x1C, 0x12,
  0x11, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x1F, 0x10,
  0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x1F, 0x00, 0x00, 0x00}};
```

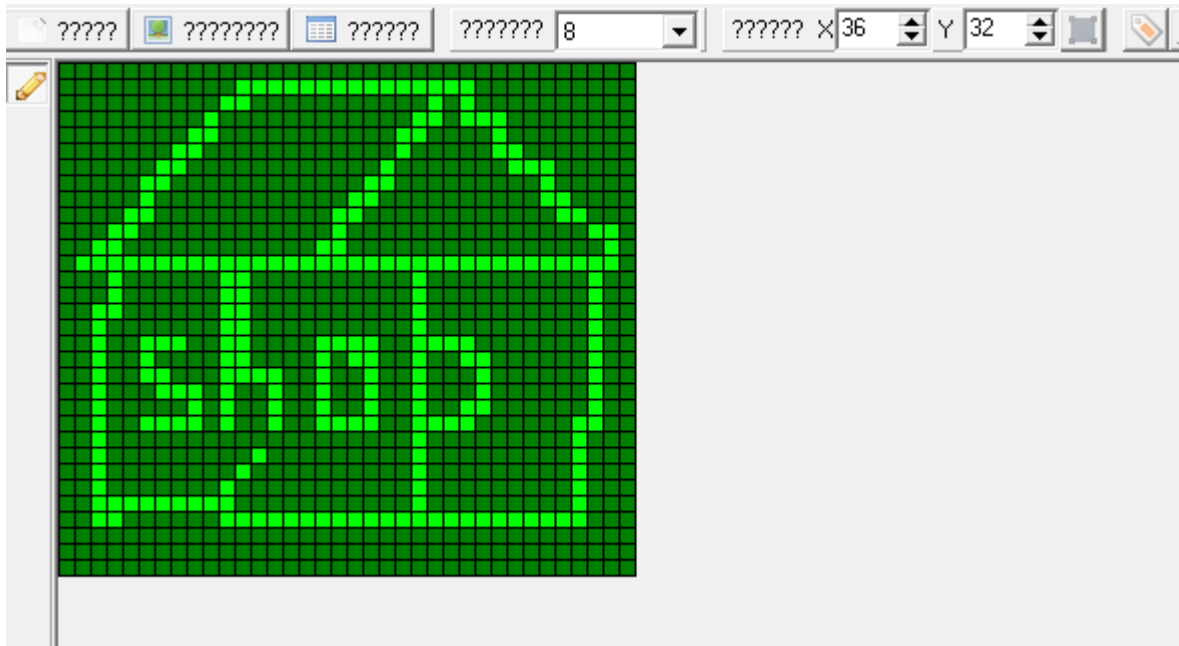


Рис.2 . Вивіска Shop

Main.c

```

#include "stm32f4xx.h"          // Deviceheader
#include "Unicode.h"

#define ADDRESS                  0x08010000

#ifdef COMPILE_UNICODE_GRAPHIC
extern ICON_PARAMS_T Icon_params;
#else
ICON_PARAMS_T Icon_params;
#endif

extern const unsigned char UNICODE_ID[16];

//Розміщення змінних у RAM
const const unsigned char FIRMWARE_ID[16] = {'L','A','B','7',' ','M','E','M','O','R','Y',' ','M','A','P',0x00};
int main ()
{
    unsigned char my_buff[16] = {0};

    my_buff[0] = UNICODE_ID[0];
    while (1);
}

```

IconsRes.c enum - структура яка буде використовуватись для формування таблиці зміщення на кожну піктограму:

```
enum
```

```
{
    BM_Ref_upArrow_bmp,
    BM_Ref_downArrow_bmp,
    BM_Ref_MAXVALUE
};
```

iconsDef.h

```
extern const unsigned char img_upArrow_bmp_char_table[];
extern const unsigned char img_downArrow_bmp_char_table[];
```

IconsList.h

```
img_upArrow_bmp_char_table,
img_downArrow_bmp_char_table,
```

Unicode.c

```
#include "Unicode.h"
#include "IconsDef.h"
#include "IconsRes.h"
```

```
#ifndef COMPILE_UNICODE_GRAPHIC
const unsigned char UNICODE_ID[16] = {'Y','O','U',' ','M','U','S','T',' ','C','R','E','A','T','E',' '};
```

```
const ICON_PARAMS_T Icon_params_c = {BM_Ref_MAXVALUE, Icons};
```

```
const ICON_T Icons[BM_Ref_MAXVALUE] =
{
#include "IconsList.h"
};
#endif
```

Unicode.h

```
typedef const unsigned char *ICON_T;
#pragma pack(push, 1)
typedef struct
{
    short NIcons;
    ICON_T const *pIcons;
} ICON_PARAMS_T;
#pragma pack(pop)
extern const ICON_T Icons[];
```

scatter-файл

```
; *****
; *** Scatter-Loading Description File generated by uVision ***
; *****

// задає адреси, за якими буде розміщено те, що визначено всередині цієї конструкції
LR_IROM1 0x08010058 0x00100000 { ; load region size_region
ER_IROM1 0x08010058 0x00100000 { ; load address = execution address
    *.o (RESET, +First) // всі об'єктні файли, причому секція з міткою RESET повинна розміщатися першою
    *(InRoot$$Sections)
    .ANY (+RO)
```

```

.ANY (+XO)
}
RW_IRAM1 0x20000000 0x00020000 { ; RW data
//ANY (+RW +ZI) означає, що змінні (не константні), можуть розміщатися в довільному порядку, «на
//розсуд компілятора
.ANY (+RW +ZI)
}
}
//для розміщення піктограми у зовнішній пам'яті
LR_ROM1 0x00000000 0x00100000 {
    icons.bin 0x00000000 0x00100000 {
        ; loadaddress != executionaddress (about 1MB)
        Unicode.o (+RO)
        Icons.o (+RO)
    }
}
}

```

РЕЗУЛЬТАТИ

```

compiling main.c...
main.c(14): warning: #83-D: type qualifier specified more than once
    const const unsigned char FIRMWARE_ID[16] = {'L','A','B','7',' ','M','E','M','O','R','Y',' ','M','A','P',0x00};
main.c(17): warning: #550-D: variable "my_buff" was set but never used
    unsigned char my_buff[16] = {0};
main.c: 2 warnings, 0 errors
compiling system_stm32f4xx.c...
linking...
Program Size: Code=460 RO-data=488 RW-data=0 ZI-data=1632
After Build - User command #1: fromelf.exe Objects\lab.axf --bin --output BINOUTPUT
"..\Objects\lab.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed: 00:00:01

```

Рис. 3. Успішна компіляція проекту

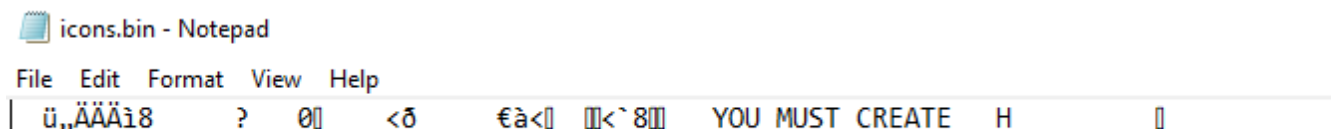


Рис. 4. Згенерований бінарний файл

__ARM_use_no_argv	0x00000000	Number	0	main.o ABSOLUTE
img_upArrow_bmp_char_table	0x00000000	Data	24	icons.o(.constdata)
__ARM_exceptions_init	- Undefined	Weak Reference		
__alloca_initialize	- Undefined	Weak Reference		
__arm_fini	- Undefined	Weak Reference		
__arm_preinit	- Undefined	Weak Reference		
__cpp_initialize__aeabi__	- Undefined	Weak Reference		
__cxa_finalize	- Undefined	Weak Reference		
__rt_locale	- Undefined	Weak Reference		
__sigvec_lookup	- Undefined	Weak Reference		
__atexit_init	- Undefined	Weak Reference		
__call_atexit_fns	- Undefined	Weak Reference		
__clock_init	- Undefined	Weak Reference		
__fp_trap_init	- Undefined	Weak Reference		
__fp_trap_shutdown	- Undefined	Weak Reference		
__get_lc_collate	- Undefined	Weak Reference		
__get_lc_ctype	- Undefined	Weak Reference		
__get_lc_monetary	- Undefined	Weak Reference		
__get_lc_numeric	- Undefined	Weak Reference		
__get_lc_time	- Undefined	Weak Reference		
__getenv_init	- Undefined	Weak Reference		
__handle_redirection	- Undefined	Weak Reference		
__init_alloc	- Undefined	Weak Reference		
__init_user_alloc	- Undefined	Weak Reference		
__initio	- Undefined	Weak Reference		
__rand_init	- Undefined	Weak Reference		
__signal_finish	- Undefined	Weak Reference		
__signal_init	- Undefined	Weak Reference		
__terminate_alloc	- Undefined	Weak Reference		
__terminate_user_alloc	- Undefined	Weak Reference		
__terminateio	- Undefined	Weak Reference		
img_downArrow_bmp_char_table	0x00000018	Data	24	icons.o(.constdata)
UNICODE_ID	0x00000030	Data	16	unicode.o(.constdata)
Icon_params_c	0x00000040	Data	6	unicode.o(.constdata)
Icons	0x00000048	Data	8	unicode.o(.constdata)
__Vectors_Size	0x00000188	Number	0	startup_stm32f407xx.o ABSOLUTE
__Vectors	0x08010058	Data	4	startup_stm32f407xx.o(RESET)
__Vectors_End	0x080101e0	Data	0	startup_stm32f407xx.o(RESET)
__main	0x080101e1	Thumb Code	8	__main.o(!!!main)
__scatterload	0x080101e9	Thumb Code	0	__scatter.o(!!!scatter)
__scatterload_rt2	0x080101e9	Thumb Code	44	__scatter.o(!!!scatter)
__scatterload_rt2_thumb_only	0x080101e9	Thumb Code	0	__scatter.o(!!!scatter)
__scatterload_null	0x080101f7	Thumb Code	0	__scatter.o(!!!scatter)
__scatterload_zeroinit	0x0801021d	Thumb Code	28	__scatter_zi.o(!!!handler_zi)
__rt_lib_init	0x08010239	Thumb Code	0	libinit.o(.ARM.Collect\$\$libinit\$\$00000000
__rt_lib_init_fp_1	0x0801023b	Thumb Code	0	libinit2.o(.ARM.Collect\$\$libinit\$\$00000000
__rt_lib_init_alloca_1	0x0801023f	Thumb Code	0	libinit2.o(.ARM.Collect\$\$libinit\$\$00000002
__rt_lib_init_argv_1	0x0801023f	Thumb Code	0	libinit2.o(.ARM.Collect\$\$libinit\$\$00000002
__rt_lib_init_atexit_1	0x0801023f	Thumb Code	0	libinit2.o(.ARM.Collect\$\$libinit\$\$00000001
__rt_lib_init_clock_1	0x0801023f	Thumb Code	0	libinit2.o(.ARM.Collect\$\$libinit\$\$00000002

Рис. 5. Уривок із файлу мапування

ВИСНОВКИ

Протягом виконання лабораторної роботи я дізнався що таке лінкування, навчився в певній мірі використовувати його можливості. Також я створив файл мапування та згенерував бінарний файл. Під час компіляції я стикнувся з помилкою лінкування «L6047U: The size of this image (65624 bytes) exceeds the maximum allowed for this version of the linker». Причина помилки складається в тому, що для створення .bin файлу потрібно більше пам'яті ніж виділено у scatter файлі. Оскільки ділянка виділена під .bin файл починається у адресі 0x00000000 а ділянка головної програми починається з адреси 0x08000000 для .bin файлу просто не вистачає пам'яті. Одним з рішень може бути "переміщення" однієї з ділянок на іншу адресу, що призведе до збільшення потрібної нам пам'яті.