**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:** *“Мапування 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.

∑= \_\_\_\_\_ .

Львів-2021

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

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

**TЕОРЕТИЧНІ 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};

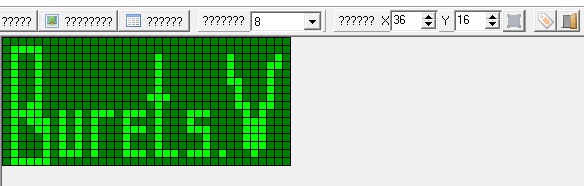


Рис.1 . Прізвище та ініціали

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};};

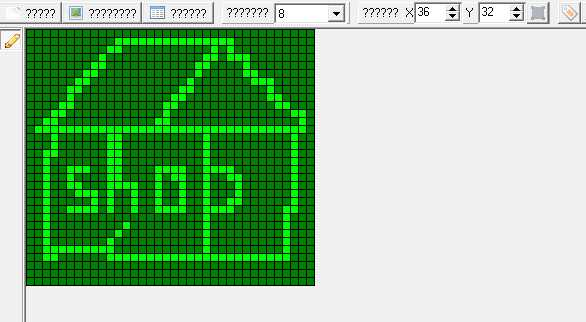


Рис.2 . Вивіска Shop

**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**

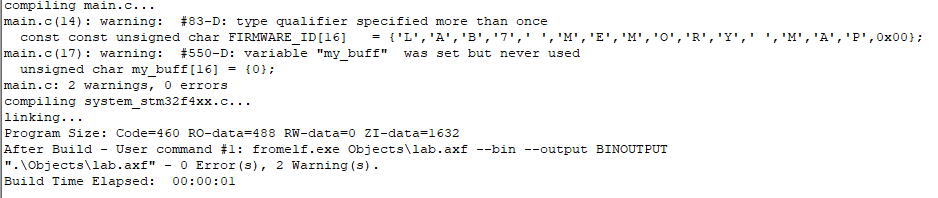


Рис. 3. Успішна компіляція проекту

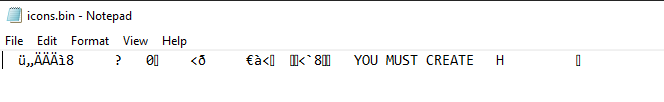


Рис. 4. Згенерований бінарний файл

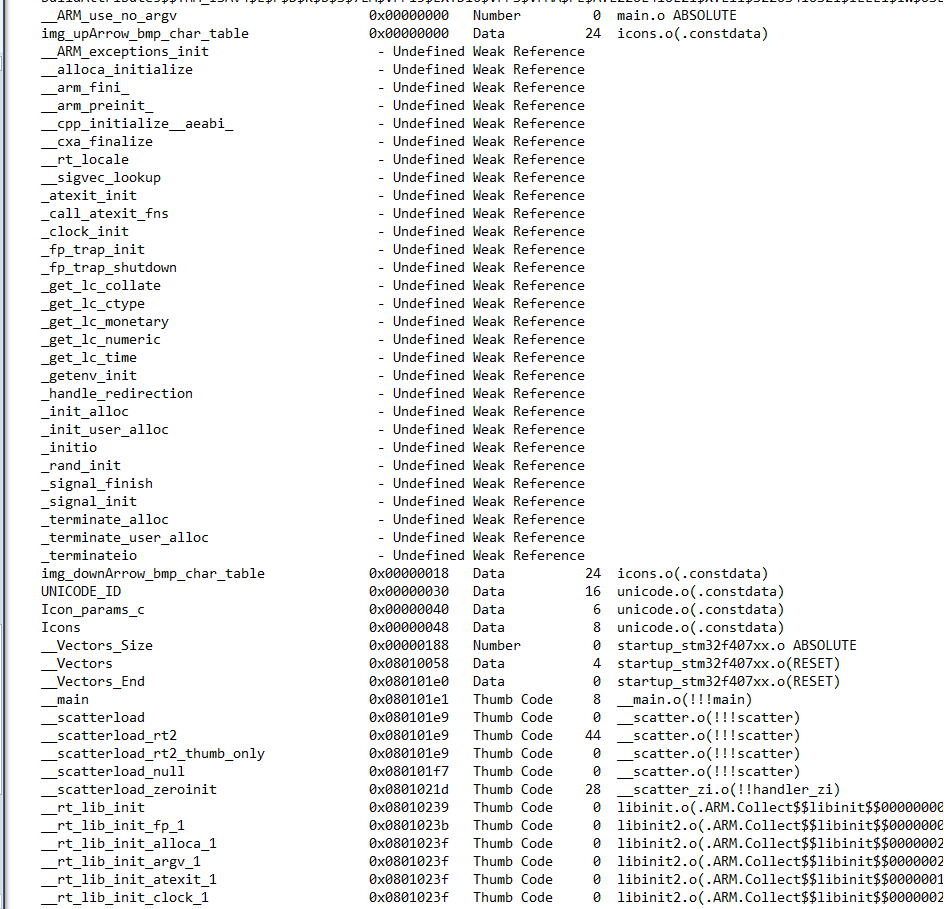


Рис. 5. Уривок із файлу мапування

**ВИСНОВКИ**

Протягом виконання лабораторної роботи я дізнався що таке лінкування, навчився в певній мірі використовувати його можливості. Також я створив файл матування 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.