***Efficient Embedded Course***

**LAB 2**

**LAB EXERCISE:**

**C AS IMPLEMENTED IN ASSEMBLY**

**Issue 1.0**

Contents

[1 Introduction 1](#_Toc87594022)

[1.1 Lab overview 1](#_Toc87594023)

[2 Learning Outcomes 1](#_Toc87594024)

[3 Requirements 1](#_Toc87594025)

[4 Procedure 2](#_Toc87594026)

[4.1 Assembly Code Listing 2](#_Toc87594027)

[4.1.1 Examine the function INIT\_LIST 2](#_Toc87594028)

[4.1.2 Examine the function FIND\_IN\_LIST 2](#_Toc87594029)

[4.2 Map File 2](#_Toc87594030)

[4.2.1 Default Program 2](#_Toc87594031)

[4.2.2 Specifying C Data as Read-Only to save RAM 3](#_Toc87594032)

# Introduction

## Lab overview

In this lab exercise, you will compile a C program and examine the assembly language program output of the compiler and the map file output of the linker.

# Learning Outcomes

* Modify and compile a C program and observe the assembly listing and the map file.

# Requirements

In this lab, we will be using the following hardware and software:

* **KEIL µVision5 MDK IDE**
  + Please check the Getting Started with KEIL guide on how to download and install it.
* **STM32F407G-DISC1**
  + For more information, click [here](https://www.st.com/en/evaluation-tools/stm32f4discovery.html).

# Procedure

1. Open the CinAsmLab project.
2. Verify that the Compiler will create assembly code listings (Options->Listing->C Compiler Listing) with maximum optimization (Options->C/C++->Optimization Level 3 (-O3).
3. Compile the program and examine the listing and map files to answer the following questions.

## Assembly Code Listing

### Examine the function INIT\_LIST

1. Which context (if any) is saved after entering the function?
2. Which instruction restores context and returns from the subroutine?
3. Which register is used as the loop counter variable i?
4. Which assembly instructions perform the operation i\*2000? Explain how they work.
5. Which assembly instructions access operand offset[i]? Explain how they work.

### Examine the function FIND\_IN\_LIST

1. Which context (if any) is saved after entering the function ?
2. Which instruction restores context and returns from the subroutine? Why is this different from init\_list?
3. Which register holds the argument key?
4. Which instructions perform the loop repeat test? Explain how they work.

## Map File

### Default Program

1. Examine the Image Symbol Table (Global Symbols subsection) and complete the table below.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Starting Address** | **Size** |
| $Super$$main |  |  |
| list\_init |  |  |
| list\_find |  |  |
| list |  |  |
| offset |  |  |

1. Examine the “Memory Map of the image”. List the section name, size, type, attributes for each section from object main.o:
2. What is the size of the STACK? Does it match the value declared in the memory layout file (scatter file or startup.s file)?
3. Examine the “Image Component Sizes” section of the map file. Complete the table to show how much memory space is needed for main.o.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Code | data | RO Data | RW Data | ZI Data | Debug |
| non-const offset |  |  |  |  |  |  |

### Specifying C Data as Read-Only to save RAM

Change your C source code so that the offset array is declared as “const” and rebuild the project.

1. Examine the updated Image Symbol Table (Global Symbols subsection) and complete the table below. Identify what has changed and why.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Starting Address** | **Size** |
| $Super$$main |  |  |
| list\_init |  |  |
| list\_find |  |  |
| list |  |  |
| offset |  |  |

1. What changes do you see in main.txt?
2. Examine the updated “Memory Map of the image” List the section name, size, type, attributes for each section from object main.o:
3. Copy the values you determined in question 16 above into the first row. Then examine the “Image Component Sizes” section of the map file and complete the second row to show how much memory space is needed for main.o. What has changed, and why?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Code | data | RO Data | RW Data | ZI Data | Debug |
| non-const offset |  |  |  |  |  |  |
| const offset |  |  |  |  |  |  |