

---

---

# USER HANDBOOK

## PROJECT 2 : 6502 Debugger

---

---

EDITED BY

FRAZER BAYLEY  
HALEY WHITMAN  
ABDULAZIZ AL-HEIDOUS  
ALISON LEGGE  
JEREMY BRENNAN

PROJECT 2 - TEAM 3

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The 6502 Debugger . . . . .	3
1.2	How to Use this Handbook . . . . .	3
<b>2</b>	<b>Getting Started with the 6502 Debugger</b>	<b>4</b>
2.1	Download and Install . . . . .	4
2.2	Starting the Application . . . . .	4
<b>3</b>	<b>6502 Debugger Programming Nuances</b>	<b>5</b>
3.1	Brief . . . . .	5
3.2	Instruction Memory . . . . .	5
3.3	Line Termination . . . . .	5
3.4	Labels . . . . .	5
3.5	Comments . . . . .	5
<b>4</b>	<b>Importing Assembly Code</b>	<b>6</b>
4.1	Overview . . . . .	6
4.2	Procedure . . . . .	6
<b>5</b>	<b>Assemble your Code</b>	<b>9</b>
5.1	Overview . . . . .	9
<b>6</b>	<b>Execute your Code</b>	<b>9</b>
6.1	Overview . . . . .	9
6.2	Procedure . . . . .	9
<b>7</b>	<b>Using the "Step" Function</b>	<b>10</b>
7.1	Overview . . . . .	10
7.2	Procedure . . . . .	10
<b>8</b>	<b>Using the "Step to" Function</b>	<b>11</b>
8.1	Overview . . . . .	11
8.2	Procedure . . . . .	11
<b>9</b>	<b>Save an Existing Program</b>	<b>13</b>
9.1	Overview . . . . .	13
<b>10</b>	<b>Programs Included</b>	<b>14</b>
10.1	Fibonacci . . . . .	14
10.2	Sieve . . . . .	14
10.3	Greatest Common Divisor . . . . .	14
10.4	Perfect Square . . . . .	14

10.5 Print CIS 422 . . . . . 14

Revision History

Revision	Date	Author(s)	Description
.1	11.03.17	Alison Legge	Created initial outline of document

# 1 Introduction

## 1.1 The 6502 Debugger

Thank you for using this 6502 Debugger. This 6502 Debugger is designed for a retro software developer who may have an interest in analyzing 6502 assembly code. It provides functionality such as the ability to assemble, execute, and step through code, and importing and exporting several different programs.

## 1.2 How to Use this Handbook

This handbook covers a range of scenarios you might encounter when using this Debugger. This handbook will walk you through each scenario step-by-step.

- **Read Every Step:**

Make sure you do not skip any steps that are listed in this handbook. Every step has been carefully thought out to make using the Debugger easier.

- **Use the Handbook in Conjunction with the 6502 Debugger:**

After your first read through of the handbook, have it open and ready as you first start to use the 6502 Debugger.

- **Experiencing Issues:**

If you are experiencing issues and are unable to solve it with the help of this handbook, please contact our support team.

## **2 Getting Started with the 6502 Debugger**

### **2.1 Download and Install**

The development team will have sent you the 6502-Debugger.jar file that launches the application. In case you did not receive the link or have lost the 6502-Debugger.jar file, contact the support team.

### **2.2 Starting the Application**

To launch the application, select and open the 6502-Debugger.jar file. This will immediately launch the Debugger application. If you have any problems, please contact the support team. Enjoy!

## **3 6502 Debugger Programming Nuances**

### **3.1 Brief**

There are a few specific requirements that come with working with this specific 6502 debugger that you as the user should be aware of when writing your 6502 Assembly Code. The following covers those specifics.

### **3.2 Instruction Memory**

This debugger has 16 kb of memory. This means that you as the programmer can only write as many instructions that fit into 16 kb of memory, else the program will not run to completion. This in practice means you have access to 0x0200 to 0x3FFF of the address space, since there needs to be room left for the zero page and high index RAM. Different 6502 implementations can do memory somewhat differently, except for the zero page and high index ram, this is the implementation that we chose that fits the requirements.

### **3.3 Line Termination**

This debugger recognizes individual lines of code through the use of new line characters. Make sure to always hit "Enter" at the end of every individual line of code. Else the debugger might not recognize it.

### **3.4 Labels**

Labels must be terminated with a colon for the debugger to recognize them as labels. For example "Label:" is the correct use of a label while "Label" is not.

### **3.5 Comments**

This debugger allows for comments in the program. Comments must be preceded by a semi colon to be recognized as a comment. Comments must either be separate from 6502 instructions or following 6502 instructions, never preceding 6502 instructions. For example "DEY ;this is a comment" or ";this is a comment" is valid, but ";this is a comment DEY" is invalid.

## 4 Importing Assembly Code

### 4.1 Overview

Importing assembly code is easy to do with this 6502 Debugger, just make sure you are using 6502 ISA assembly code.

### 4.2 Procedure

#### 1. Begin at the Main Page

Below is an image of the main page to help you verify you are where you need to be.

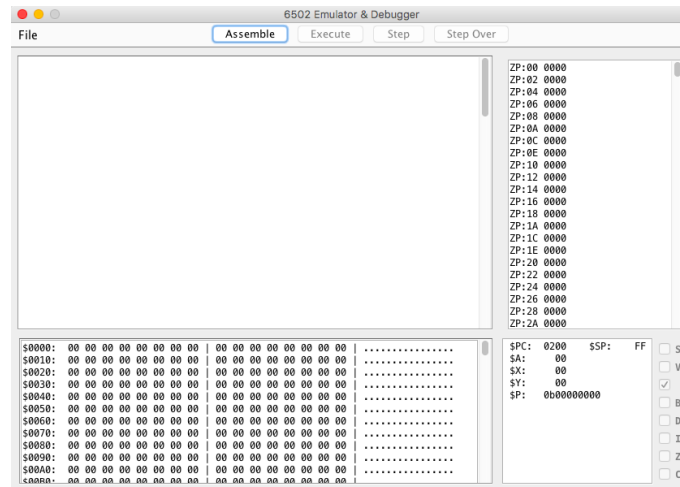


Figure 1: Main Page

2. **Click Open in the drop down File Menu** You will see the File menu on the top left-hand corner of the window. Click File so the drop down menu appears and click Open, a new window will appear.

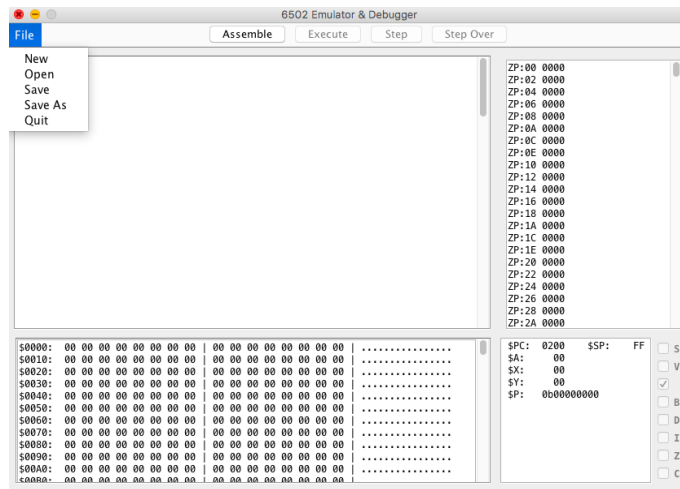


Figure 2: Drop down File Menu

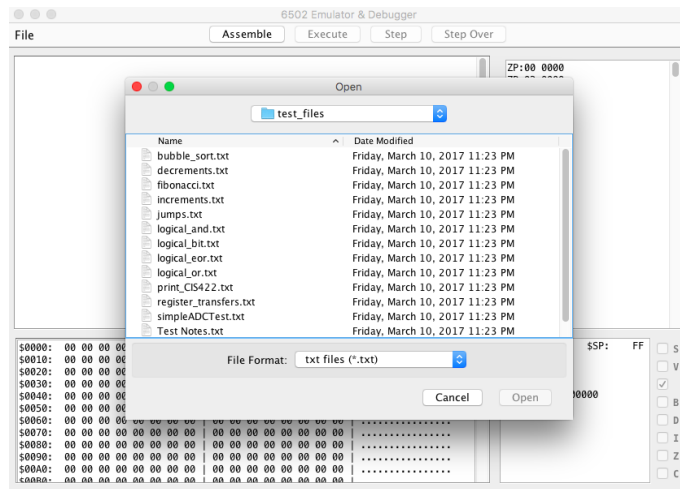


Figure 3: The Open Window



### 3. Find Text File to Import

Select the file that you would like to import in the Open window. Be sure that the file selected is a .txt file or else you will not be able to open the file. Select the file and press Open.

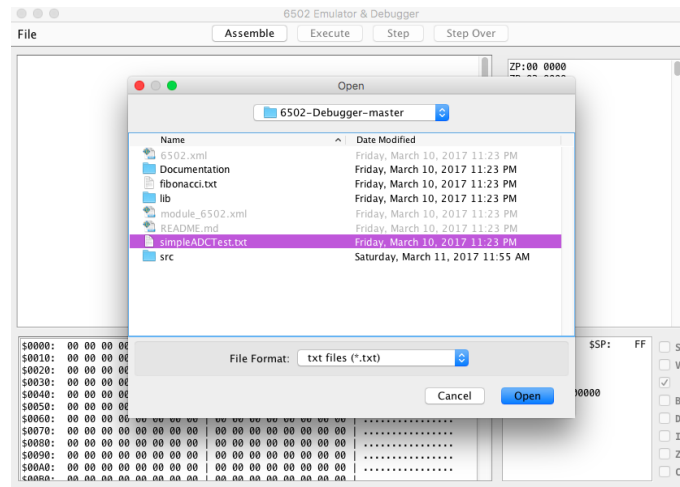


Figure 4: The Open Window

After this step, you will be left with the Main page of the 6502 Debugger with your code loaded in the main text window.

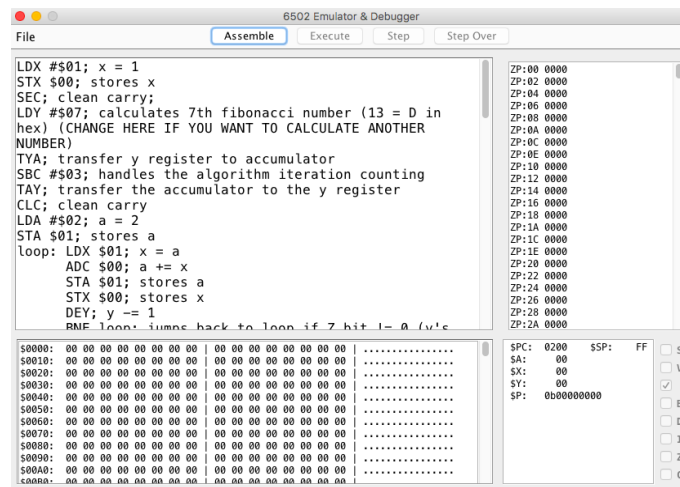


Figure 5: Main Page with code loaded in the main text window

## 5 Assemble your Code

### 5.1 Overview

Assembling your code can be done with a click of a button using this Debugger.

1. **Begin at the Main Page with Assembly Code loaded in the Main Window**

To assemble code, you must have code loaded in the main window. This can be done by importing the code from a .txt file, copy and pasting the code, or typing it directly into the text window.

2. **Click the Assemble Button**

The Assemble button is located in the centre of the task bar at the top of the window.

After this step, the first line of code should be highlighted green with the appropriate flags selected.

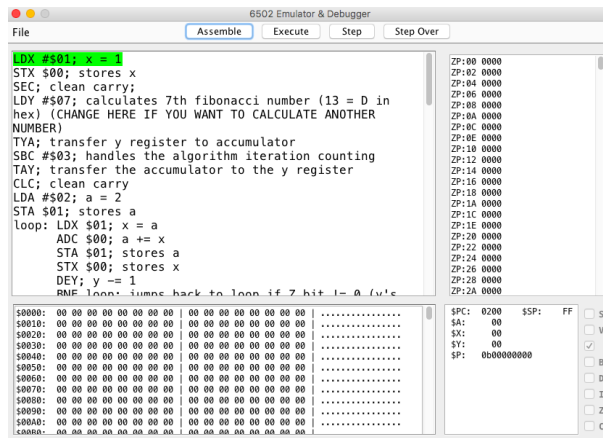


Figure 6: The First Line of Code is Highlighted Green

## 6 Execute your Code

### 6.1 Overview

Executing your loaded assembly code is easily done in the 6502 Debugger.

### 6.2 Procedure

1. **Begin by Assembling your Code** If needed, see "Assemble your Code" section for further instruction.
2. **Click the Execute Button** The Execute button is located in the centre of the task bar at the top of the window.

## 7 Using the "Step" Function

## 7.1 Overview

To execute your code line by line, you can use the Debugger's Step function by simply clicking.

## 7.2 Procedure

## 1. Begin by Assembling your Code

If needed, see "Assemble your Code" section for further instruction.

## 2. Click the Step Button

The Step button is located in the centre of the task bar at the top of the window.

After this step, the current line of code highlighted in green will execute and the highlighting will move to the next line.

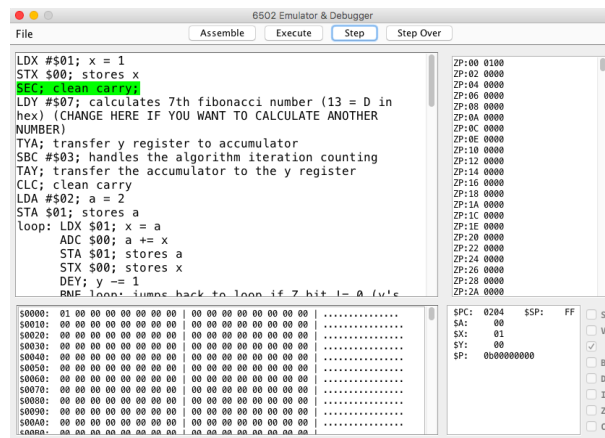


Figure 7: The Line Highlighted will Change

## 8 Using the "Step to" Function

### 8.1 Overview

The Step To function will allow you to skip to a function of your choosing.

### 8.2 Procedure

#### 1. Begin by Assembling your Code

If needed, see "Assemble your Code" section for further instruction.

#### 2. Click the Step To Button

The Step To button is located in the centre of the task bar at the top of the window. If your Assembly Code does not have any defined functions, it will display an error.

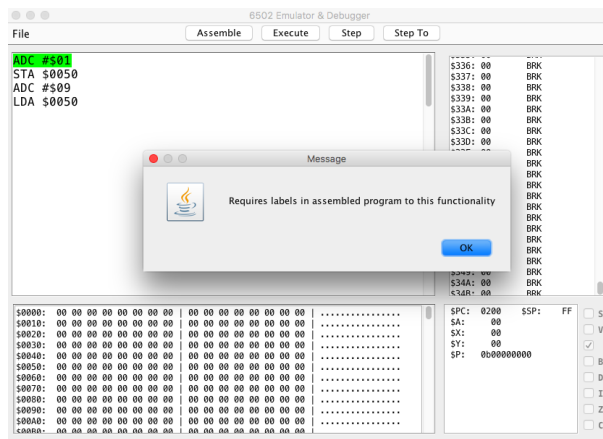


Figure 8: Unable to Perform the Step To Function

Otherwise, the Step To window will appear .

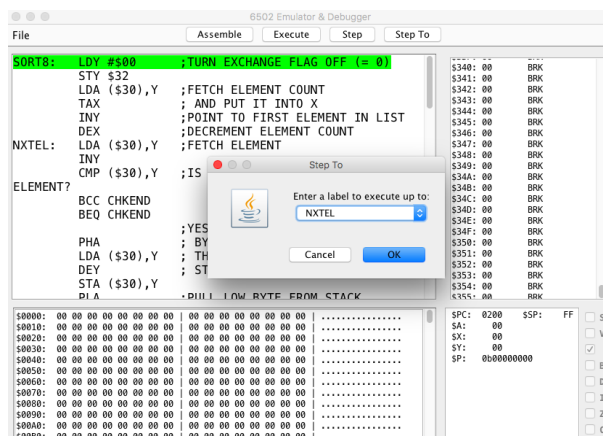


Figure 9: The Step To Window

### 3. Select the Function to Step To

If you have multiple functions, open the drop down menu in the Step To window, select your chosen function, and click OK.

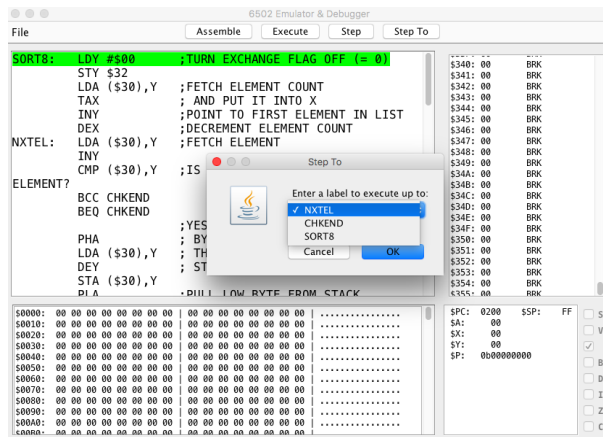


Figure 10: The Open Drop Down Menu in the Step To Window

After this step, your chosen function will be highlighted in green.

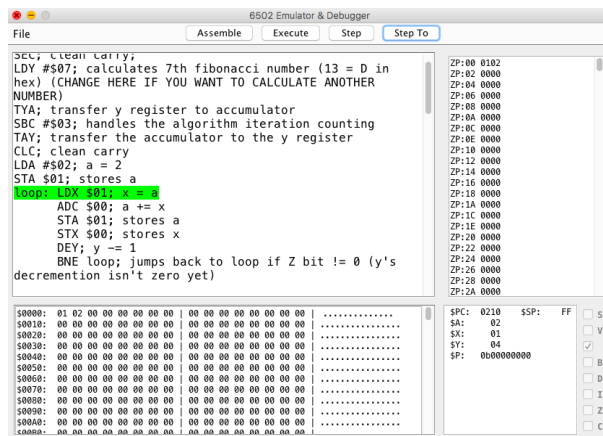


Figure 11: After Step To has been executed

## 9 Save an Existing Program

### 9.1 Overview

1. **Click Save in the drop down File Menu** You will see the File menu on the top left-hand corner of the window. Click File so the drop down menu appears and click Save, a new window will appear.

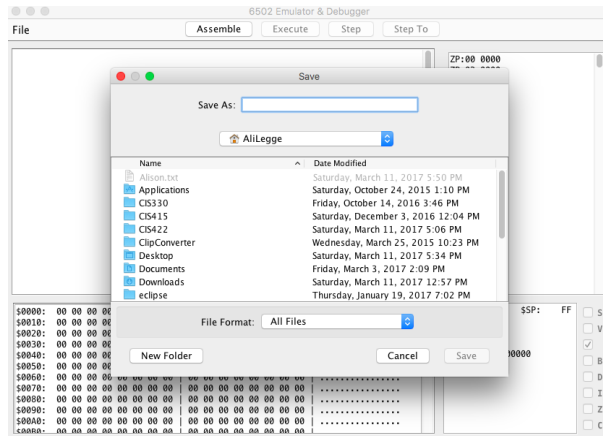


Figure 12: After the Save Window Opens

2. **Save the File Type the filename and click Save.**

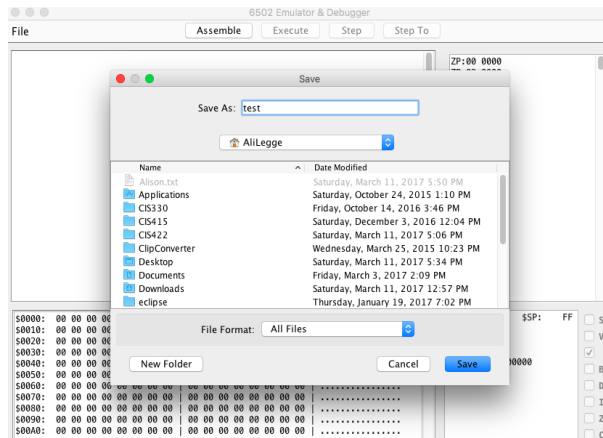


Figure 13: After the Name has been Typed

## **10 Programs Included**

ba

### **10.1 Fibonacci**

The file is named "fibonacci.txt". This program takes a hexadecimal number and calculates the corresponding fibonacci value, the value calculated is in hex and is stored in the Accumulator. The file indicates where to change the input so you can calculate another fibonacci value.

### **10.2 Sieve**

### **10.3 Greatest Common Divisor**

### **10.4 Perfect Square**

### **10.5 Print CIS 422**

The file is named "print\_CIS422.txt". This program performs hex printing of the phrase "C15 422" over and over again. You can see the hex print in the memory viewer below the main window. If you would like to change the amount of times "C15 422" is printed change the value that follows the very first instruction "LDY".