./

GENESIS

**Learning Report on**

**MODEL BASED SYSTEM ENGINEERING**

**(MBSE)**

(MBSE)



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
|  | 17-2-2021 | Sneha Anand |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

TABLE OF CONTENTS

[Contents 3](#_Toc64406838)

[ACTIVITY 1 : CREATE A MAKEFILE 5](#_Toc64406839)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**[**1.1 MAKEFILE 5**](#_Toc64406840)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**[**1.2 CREATING A MAKEFILE 5**](#_Toc64406841)

[ACTIVITY 2 : CREATE A STARTUP FILE 6](#_Toc64406842)

**2.1 STARTUP FILE………………………………………………………………….**

[ACTIVITY 3 : CREATE A LINKER SCRIPT](#_Toc64406843)

**3.1 LINKER SCRIPT ………………………………………………………………….**

[ACTIVITY 4 : DEBUGGING TECHNIQUES](#_Toc64406844)

[ACTIVITY 5 : AUTOMOTIVE FEATURE EXECUTION ( MINI PROJECT)](#_Toc64406844)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**[**5.1 DOOR LOCK SYSTEM**](#_Toc64406840)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**[**5.2 ANTI COLLISION SYSTEM**](#_Toc64406841)

**5.3**

**LIST OF FIGURES**

[**Figure 1: CODE SNIPPET OF AUTOMATION SCRIPT** Error! Bookmark not defined.](#_Toc59446097)

[**Figure 2: PLOT INDICATING THE USE OF PHONE/ DEVICE USING GYROSCOPE READING FROM \_\_\_\_\_\_ANDRO-SENSOR DATA** Error! Bookmark not defined.](#_Toc59446098)

[**Figure 3: Longitudinal Torsion** Error! Bookmark not defined.](#_Toc59446099)

[**Figure 4: Auto-adjusting Feature of steering** Error! Bookmark not defined.](#_Toc59446100)

[**Figure 5:** [**Body Control Module (BCM) in Automotive**](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.intellias.com%2Fbody-control-module-bcm-in-automotive%2F&psig=AOvVaw21Ubfks1YCJBAQ1RDxQsu5&ust=1613578037492000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCPiRq-Pk7u4CFQAAAAAdAAAAABAD)  **6**](#_Toc59446101)

**FIGURE 6: THE INTERIOR LIGHT SYTEM 10**

**FIGURE 7: THE SUBSYSTEM FOR INTERIOR LIGHT 10**

[**Figure 8: Internal structure of subsystem** Error! Bookmark not defined.](#_Toc59446104)

# ACTIVITY 1

**CREATE A MAKEFILE**

### 1.1 MAKEFILE

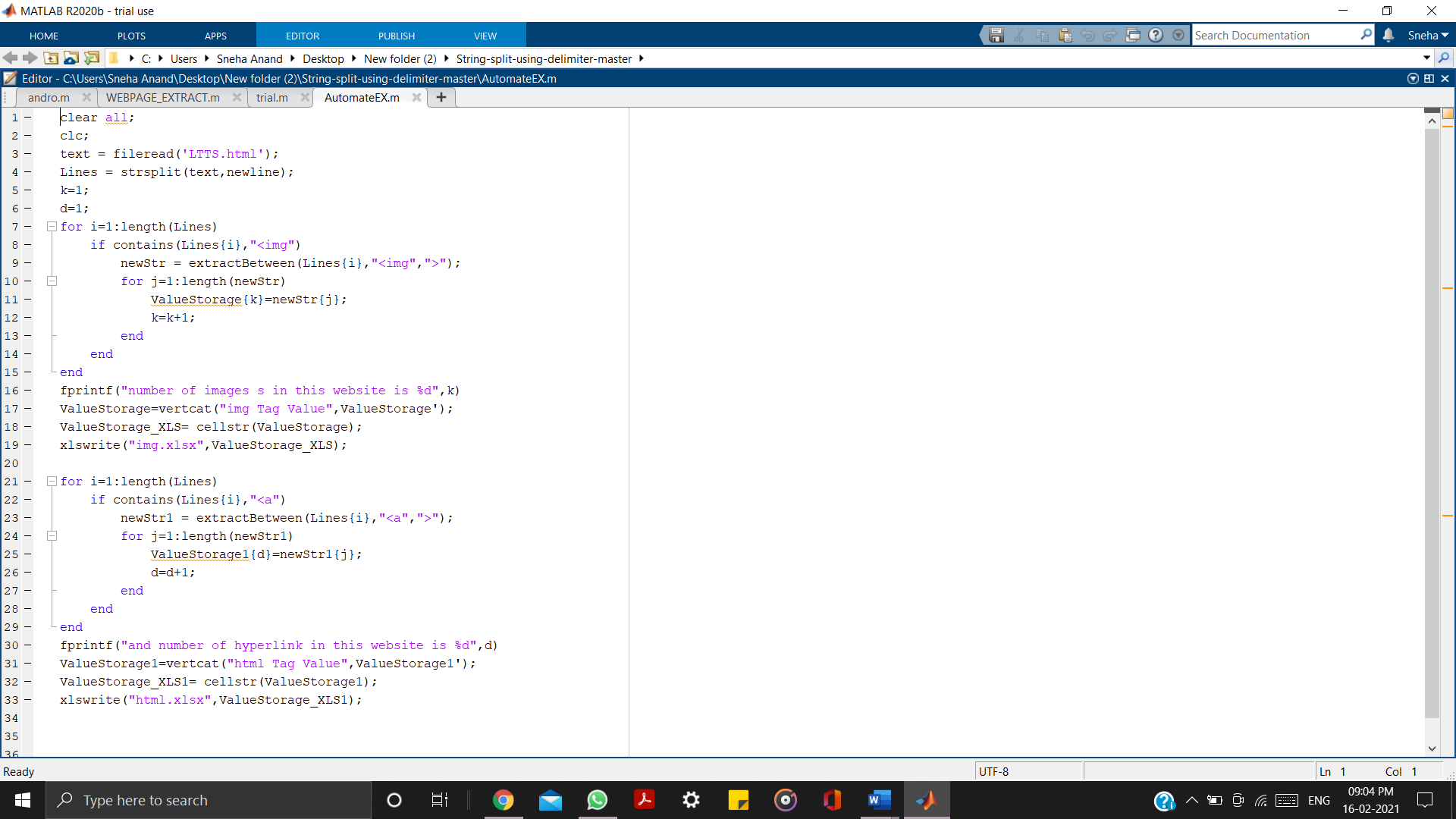
Make is Unix utility that is designed to start execution of a makefile. A makefile is a special file, containing shell commands, that you create and name makefile (or Makefile depending upon the system). While in the directory containing this makefile, you will type *make* and the commands in the makefile will be executed. If you create more than one makefile, be certain you are in the correct directory before typing make.

Make keeps track of the last time files (normally object files) were updated and only updates those files which are required (ones containing changes) to keep the sourcefile up-to-date. If you have a large program with many source and/or header files, when you change a file on which others depend, you must recompile all the dependent files. Without a makefile, this is an extremely time-consuming task.

As a makefile is a list of shell commands, it must be written for the shell which will process the makefile. A makefile that works well in one shell may not execute properly in another shell.

The makefile contains a list of *rules*. These rules tell the system what commands you want to be executed. Most times, these rules are commands to compile(or recompile) a series of files. The rules, which must begin in column 1, are in two parts. The first line is called a *dependency* line and the subsequent line(s) are called *actions* or *commands*. The action line(s) must be indented with a tab.

### 1.2 CREATING A MAKEFILE :



[**Link to MATLAB Code:**](https://github.com/99003525/MBSE/blob/main/MATLAB/AutomateEX.m)

**Links to Excel file**s:

1. [Webpage used](https://github.com/99003525/MBSE/blob/main/MATLAB/html.xlsx)
2. [HTML tag data](https://github.com/99003525/MBSE/blob/main/MATLAB/html.xlsx)
3. [Images data](https://github.com/99003525/MBSE/blob/main/MATLAB/img.xlsx)

Figure 1: Code snippet of Makefile

# 

# ACTIVITY 2

**CREATE A STARTUP FILE**

**2.1 START UP FILE :**

A startup file is a piece of code written in assembly or C language that executes before the main() function of our embedded application. It performs various initialization steps by setting up the hardware of the microcontroller so that the user application can run. Therefore, a startup file always runs before the main() code of our embedded application.

### 1.2 CREATING A START UP FILE :

Following are the main functions of a startup file:

1. Disable all interrupts
2. Copying initialized global, global static, and local static variable data from flash to .data section RAM memory of a microcontroller
3. Copying uninitialized global, global static, and local static variable data from flash to .bss section of RAM memory and initialize .bss section of RAM to zero.
4. Allocate space for the stack and initialize the stack pointer
5. It also contains an array of function pointers ( interrupt vector table) that point to various interrupt vector routines such as interrupts and exceptions. The startup file also contains definitions of these interrupt or exception routines such as reset handler, NMI handler, bus fault handler, etc.
6. Enable interrupts
7. Calls the main function

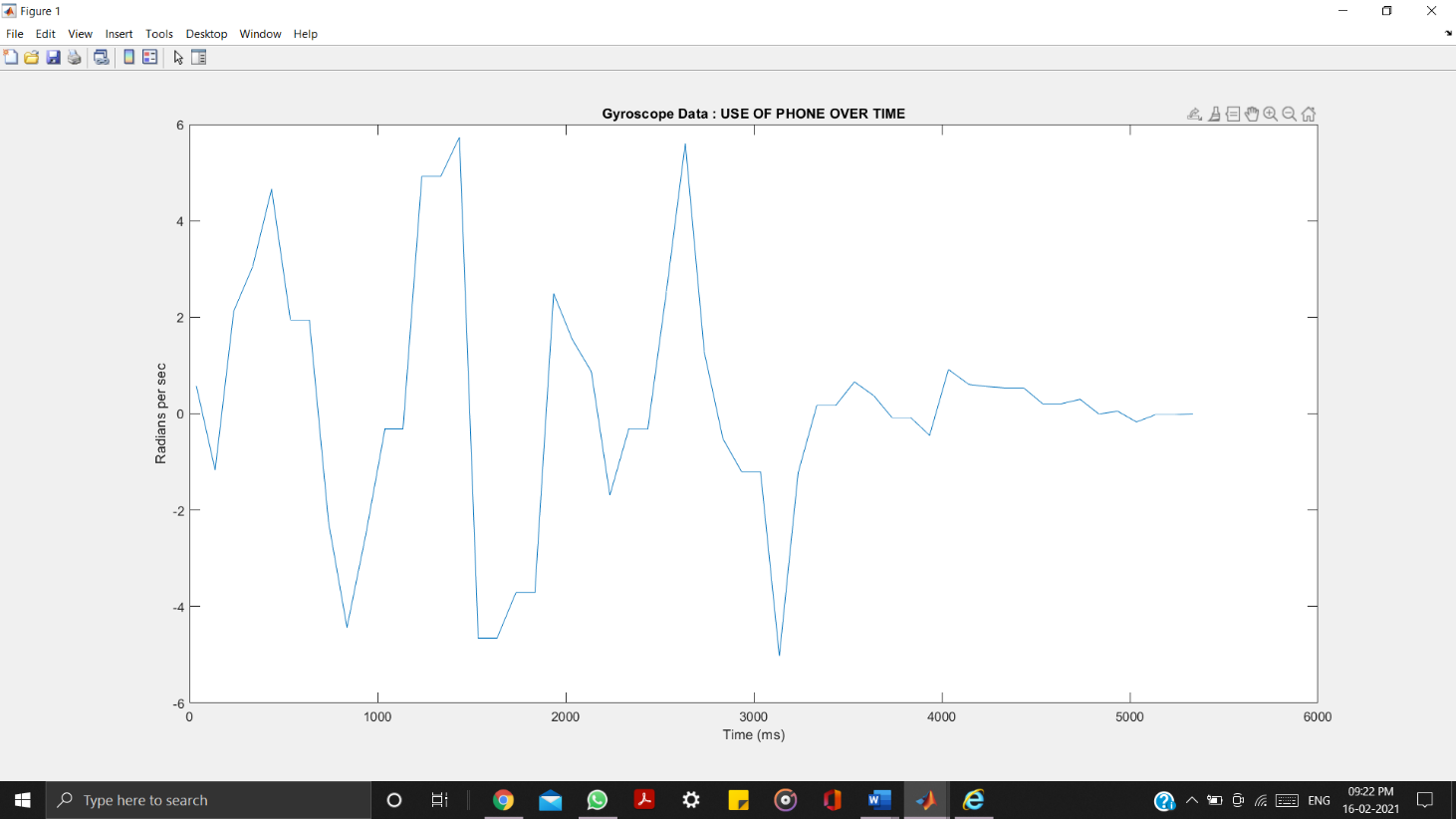
****

Figure 2: Snippet of StartUp file

[**Link to MATLAB Code**](https://github.com/99003525/MBSE/blob/main/MATLAB/Algorithm_matlab.m)

[**Link to Excel file**](https://github.com/99003525/MBSE/blob/main/MATLAB/sample.csv)

# ACTIVITY 3

**CREATE A LINKER SCRIPT**

* 1. **LINKER SCRIPT :**

The Linker Script is a text file made up of a series of Linker directives which tell the Linker where the available memory is and how it should be used. Thus, they reflect exactly the memory resources and memory map of the target microcontroller.

* 1. **CREATING A LINKER SCRIPT :**

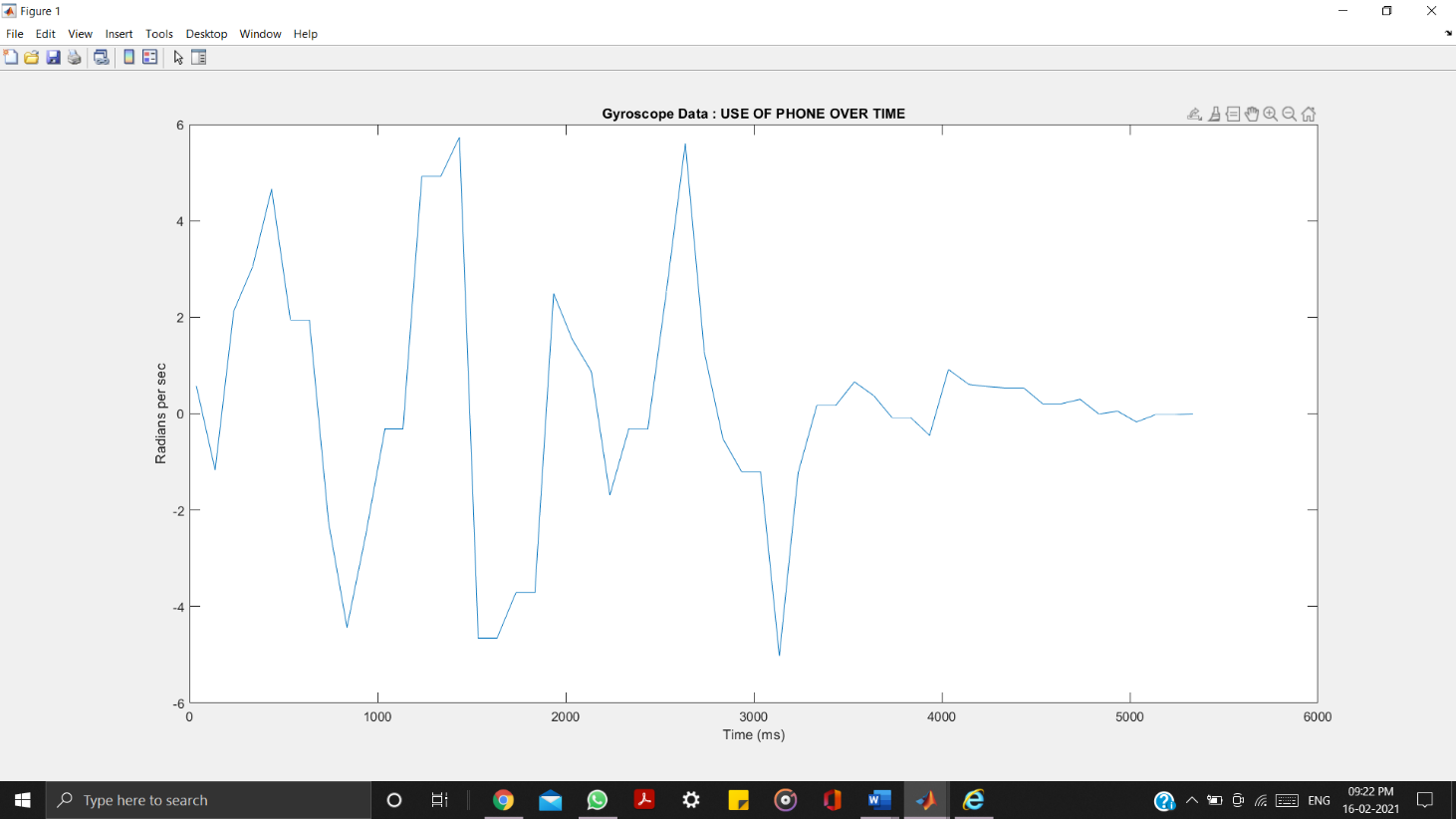
****

Figure 2: Snippet of LINKER SCRIPT file

[**Link to MATLAB Code**](https://github.com/99003525/MBSE/blob/main/MATLAB/Algorithm_matlab.m)

[**Link to Excel file**](https://github.com/99003525/MBSE/blob/main/MATLAB/sample.csv)

# ACTIVITY 4

**DEBUGGING TECHINQUES**

**4.1 DEBUGGING :**

Debugging is the process of detecting and removing of existing and potential errors (also called as ‘bugs’) in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects. When various subsystems or modules are tightly coupled, debugging becomes harder as any change in one module may cause more bugs to appear in another

**ACTIVITY 5**

**AUTOMOTIVE FEATURE EXECUTION**

**5.1 DOOR LOCK**

**Power door locks** (also known as electric door locks or **central locking**) allow the driver or front passenger to simultaneously lock or unlock all the doors of an automobile or truck, by pressing a button or flipping a switch

**5.1 ANTI-COLLISION SYSTEM**

Automated anti-collision system by detecting obstacles for automobile industry is one the emerging technologies nowadays. An automated vehicle anti-collision system is an automobile safety system which prevents collision among cars and objects automatically.

**5.3 ???????????????**