

# Development of Software Platform for UTM

## Software Requirements Specification

1.0

26/1/18

Prathamesh Naik  
Sai Kumar Reddy  
Vishal Maurya  
Rahul Kumar  
Balmangaraj

Prepared for  
CS 258 Software Engineering  
Spring 2018

## Table of Contents

<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1 PURPOSE.....	1
1.2 SCOPE.....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	2
1.4 REFERENCES.....	3
1.5 OVERVIEW.....	3
<b>2. GENERAL DESCRIPTION.....</b>	<b>4</b>
2.1 PRODUCT PERSPECTIVE.....	4
2.2 PRODUCT FUNCTIONS.....	4
2.3 USER CHARACTERISTICS.....	5
2.4 GENERAL CONSTRAINTS.....	5
2.5 ASSUMPTIONS AND DEPENDENCIES.....	6
2.6 USER DOCUMENTATION.....	6
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>7</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	7
3.1.1 User Interfaces.....	7
3.1.2 Hardware Interfaces.....	8
3.1.3 Software Interfaces.....	10
3.1.4 Communications Interfaces.....	10
3.2 FUNCTIONAL REQUIREMENTS.....	11
3.2.1 To control the Universal Testing Machine.....	11
3.2.2 Showing real time values of varying mechanical quantities.....	11
3.2.3 Plotting graphs in real-time: load vs deformation and stress vs strain.....	12
3.2.4 Exporting data to Excel file.....	12
3.2.5 Printing graphs.....	13
3.2.6 "Smoothing" of raw data from amplifier connected to load cell sensor.....	13
3.3 NON-FUNCTIONAL REQUIREMENTS.....	14
3.3.1 Performance.....	14
3.3.2 Reliability.....	14
3.3.3 Availability.....	14
3.3.4 Security.....	14
3.3.5 Maintainability.....	14
3.3.6 Portability.....	14
3.5 DESIGN CONSTRAINTS.....	14
3.5.1 Operating system.....	14
3.5.2 Software platform.....	14
3.5.3 Microcontroller.....	14
3.5.4 Hardware components.....	14

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the software system/platform for Universal Testing Machine (UTM). It will explain the purpose, functionalities and features of the system, the interfaces of the system, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

## 1.2 Scope

This software system will be a software platform for a custom made Universal Testing Machine. This system will be designed to maximize the researcher's productivity by providing tools to assist in automating the material testing process. Specifically, it'll be used to find out the mechanical properties of polyimide sheets with and without SMA (shape memory alloy) coating. The system will meet the user's needs while remaining easy to understand and use.

More specifically, this system is designed to

1. Allow a user to control the Universal Testing Machine. Users will be able to specify speed and no. of cycles as well as control the motion of motor (which involves pausing it, restarting it and controlling direction) through GUI.
2. Get data output from the UTM, process it and run calculations on it, then show the useful information to the user, in the form of Graph or Excel file.

This software will facilitate communication between Universal Testing Machine (the electrical components include Arduino, Load Cell and Gear Motor) and the PC with which it is connected through USB cable. The main objective of this software is to get values of stress on the polyimide sheet for different values of strain when it is elongated in the universal testing machine. This process will be repeated for a user specified no. of cycles.

The output will be shown to the user in the form of graph or excel file. Based on the data received from UTM, certain mechanical properties of UTM would also be calculated and shown to the user.

This software will also show real-time data output while the UTM is running.

### 1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
<b>UTM</b>	A universal testing machine (UTM), also known as a universal tester, materials testing machine or materials test frame, is used to test the tensile strength and compressive strength of materials. The "universal" part of the name reflects that it can perform many standard tensile and compression tests on materials, components, and structures (in other words, that it is versatile). In this project, we'll be working on a custom UTM which was made to test properties of polyimide sheets with and without SMA coating.
<b>SMA</b>	A shape-memory alloy (SMA) is an alloy that "remembers" its original shape and that when deformed returns to its pre-deformed shape when heated.
<b>GUI</b>	The graphical user interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.
<b>Strain</b>	Strain is the relative change in shape or size of an object due to externally-applied forces.
<b>Stress</b>	Stress is the associated force (per unit area) associated with a strain.
<b>Arduino</b>	Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) is used to write and upload computer code to the physical board. In this project we will be working on Arduino Uno which is one the variants of Arduino microcontrollers.
<b>Load cell</b>	A load cell is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured.
<b>Amplifier</b>	An amplifier is an electronic device that increases the voltage, current, or power of a signal. It'll be used for amplifying the output voltage of load cell to make it noticeable to the Arduino.
<b>Gear motor</b>	A geared DC Motor has a gear assembly attached to the motor. It's speed can be controlled.
<b>Motor controller/driver</b>	A Motor Controller is a device that acts as intermediary between a machine's microcontroller, batteries and motors.

## 1.4 References

Most of the definitions were adapted from Wikipedia entries of respective terms.

This SRS is based on following templates / examples:

- 1.The template Abhishek Srivastava Sir sent us.
- 2.IEEE Recommended Practice for Software Requirements template
3. SRS example from Michigan State University.

## 1.5 Overview

The rest of the SRS document is divided into four main parts: -

### 1. General Description

This section gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next section.

### 2. Specific Requirements

This section describes in technical terms the details of the functionality of the product. There are 4 subsections:

#### i. External Interface Requirements

A brief overview of the hardware, software and interface requirements is provided.

#### ii. Functional Requirements

This section describes specific features related to the basic functions of this software.

#### iii. Non-Functional Requirements

This section describes the characteristics of the system. That includes performance, reliability, portability etc.

## 2. General Description

### 2.1 Product Perspective

This software will interface a PC with a custom Universal Testing Machine through Arduino. There are two parts to it: the code that will run on PC and code that will run on Arduino. This software will be designed in an open source platform, processing. The client interacts with the software's GUI where he/she inputs the data. The software will control the motor in UTM as well as take sensor outputs from Arduino, run calculations on it to convert it to stress and strain values and then will show the real-time output to user. Based on the output the user can study the mechanical properties of the polyimide sheet.

### 2.2 Product Functions

(Note: Technical details are given in section 3.)

- **To control the UTM**

Users will control the UTM as per their needs through GUI of this software platform. This includes starting and stopping the motor, setting custom speed and no. of cycles, as well as changing direction of the motor.

- **To show real time output relating to mechanical properties of polyimide sheet being tested**

This software will show the value of load and the stress developed in the material as the strain changes due to deformation caused by motion of the motor. Output will also be shown visually in the form of a graph.

- **To export the results after testing is finished**

After testing has been finished, users will be able to get an excel file with all the captured and processed data.

## **2.3 User Characteristics**

There is only one class of users who will use this software: researchers studying mechanical properties of polyimide sheets. All the features and functions of this software pertain to them.

They will be knowledgeable about all the relevant mechanical properties that the software will show them. They will also decide on the testing procedure.

The GUI will be made self-explanatory so that effective testing may be carried out as per their requirements.

The software will be made such that knowledge of electrical components involved including the control circuit and the microcontroller will not be required in order to use it.

## **2.4 General Constraints**

The general constraints on this application software include: -

### **1. Fast Response Time:**

The software should be able to keep up with Arduino output and should also be able to give instructions to Arduino quickly.

### **2. User Friendly Interface:**

The GUI should be user friendly and self-explanatory enough so that researchers should be able to use test polyimide sheets as per their requirement.

### **3. Easy installation and software portability**

The software should run on Microsoft windows platform and any researcher wanting to use the UTM should be able to install the software on their PC with relative ease.

### **4. Hardware constraints**

The software should interface well with the custom UTM. This means it should communicate with the Arduino which will control the motor using a motor controller, as well as get the sensor outputs such as force values from the load cell and amplifier. The Arduino code is part of the software platform.

## **2.5 Assumptions and Dependencies**

This software will be created using the "Processing" platform (an open-source computer programming language and integrated development environment).

It will have multiple dependencies such as external libraries which have not been completely determined yet.

It'll be assumed that the UTM is in proper physical state and available with user.

Further, it is dependent on multiple electrical components such as the Arduino, Load cell, Amplifier, Motor, Motor Controller, Power supply. It'll be assumed that all of them are working properly and that the circuit connections are properly made.

Users are assumed to use the Microsoft Windows operating system. Further it is assumed that a USB cable can be used to connect their PC and the UTM (more specifically, the Arduino part).

## **2.6 User Documentation**

A user manuals and a tutorial will be delivered along with the software.



### **3. Specific Requirements**

#### **3.1 External Interface Requirements**

##### **3.1.1 User Interfaces**

User Interface will consist of a window which will display:

###### **3.1.1.1 Real-time values of varying mechanical quantities.**

This will include stress, strain, load and deformation of the polyimide sheet.

###### **3.1.1.2 A Graph between Tensile Stress and Strain**

###### **3.1.1.3 A Graph between Load And Deformation**

Both of them should update in real-time.

It will also consist of the following options:

###### **3.1.1.4 Various input options for all required details.**

This will include speed of the motor, length, thickness and other relevant properties of the polyimide sheet, no. of cycles to test for etc.

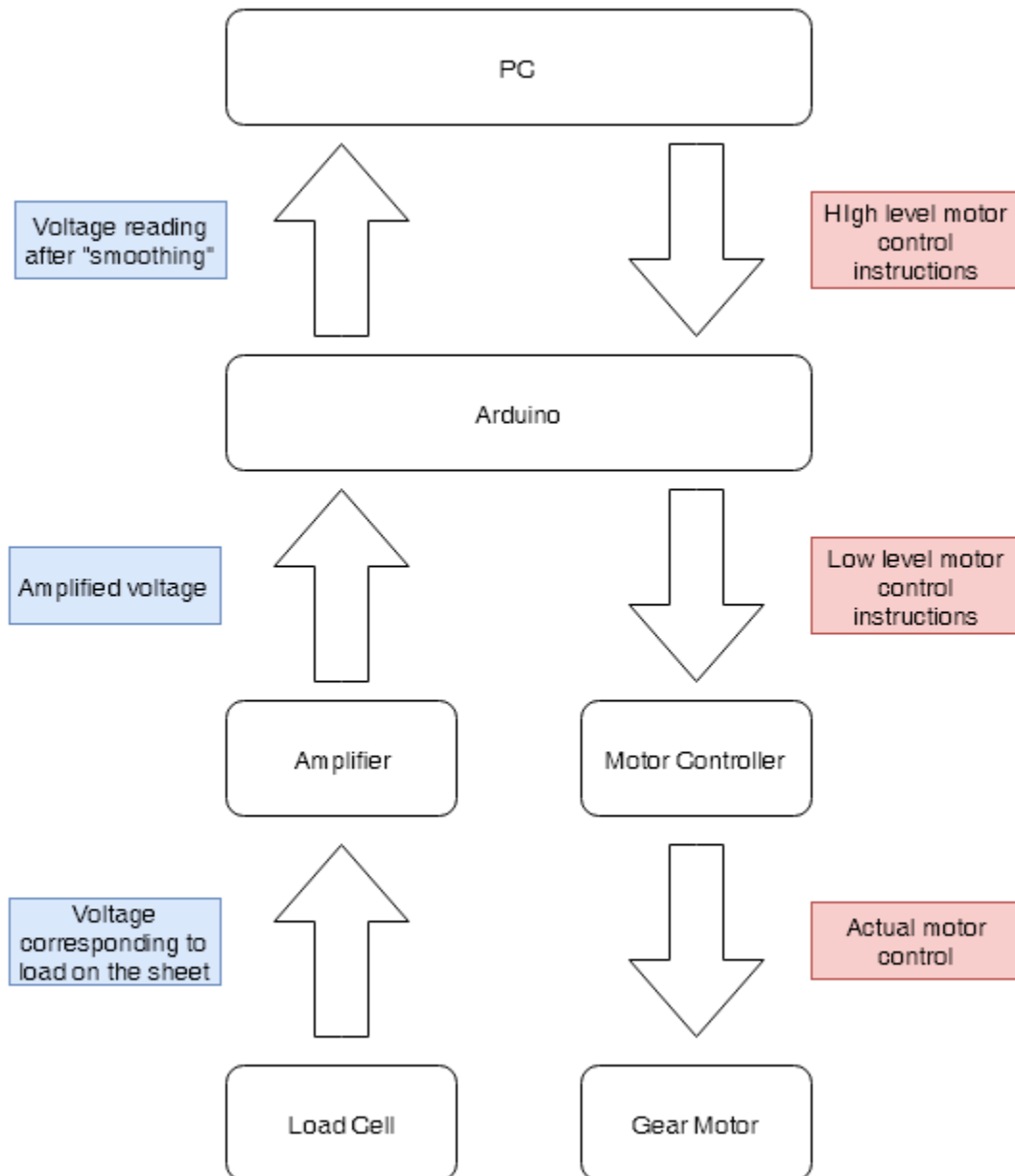
###### **3.1.1.5 Buttons to control the motor**

There will be a start button, a pause button and a change direction button in case the user wants manual control of the motor.

**3.1.1.6 Option to export the data to an excel file once the testing has finished.**

###### **3.1.1.7 Option to print the graph.**

### 3.1.2 Hardware Interfaces



While the Universal Testing Machine is working, the communication between various hardware components will be frequent (multiple messages per second).

### **3.1.2.1 PC <=> Arduino**

The software on PC will interface with an Arduino Uno microcontroller using a USB cable.

This will involve serial communication, i.e., sending data one bit at a time, sequentially. The software on PC will send control instructions to Arduino which the code on Arduino (which is also part of the software platform) will receive. These control instructions will predominantly involve instruction relating to control the motor.

The Arduino part of the software platform (i.e. the code running on Arduino) will collect information from sensor (Load cell) at a certain frequency and send it to the software running on PC.

This process will happen approximately thrice per second.

### **3.1.2.2 Arduino => Motor Controller => Motor**

Based on instruction received from the software on PC, the code on Arduino will appropriately control the motor using the motor controller. This will involve communication method specific to the motor controller. The motor controller has not been completely decided yet, so this section of the SRS will be updated later.

The motor controller will then control the motor behavior accordingly.

This process will happen multiple times per second. The exact frequency would depend on type of testing.

### **3.1.2.3 Arduino <= Amplifier <= Load cell**

The load cell will generate a voltage proportion to the force applied on the polyimide sheet. However, this voltage is very small so it will be amplified before being fed into the Arduino. The Arduino has an analog input pin which can read voltages from 0 to 5 V, it'll be used to read the voltage from the amplifier. The Arduino code will then do a process known as "smoothing" (which reduces the influence of random noise) before transmitting these values to PC.

This process will happen approximately thrice per second.

### **3.1.3 Software Interfaces**

#### **3.1.3.1 Processing 3.3.6**

Processing is a Java based programming language and Integrated Development environment(IDE) which will be used for the development of software platform for Universal Testing Machine(UTM).

Processing will be used to create the GUI as well as to interface with Arduino.

In the process of development, many Processing libraries could be used. This section of SRS will be kept updated each time a new library is used or if a library has been removed.

#### **3.1.3.2 Arduino Integrated Development environment (IDE)**

It will be used to write code and upload it to the Arduino Uno board. The Arduino programming language is based on c/c++.

### **3.1.4 Communications Interfaces**

No communication interface is required by this product.

## 3.2 Functional Requirements

### 3.2.1 To control the Universal Testing Machine

#### 3.2.1.1 Introduction

This will involve controlling the motor. Then the elongation and compression of polyimide sheet can be controlled.

#### 3.2.1.2 Inputs

Users will have to specify inputs such as custom speed input and no. of cycles or manually use options such as start, pause and change direction.

#### 3.2.1.3 Processing

Based on User input, the PC part of the software will send appropriate instructions to Arduino part of the software. The code on Arduino will interpret these instructions and transmit corresponding control commands to motor controller. Finally, the motor controller will control and drive the motor.

#### 3.2.1.5 Error Handling

It does not seem possible to detect errors in behavior of motor at runtime, since the motor only takes instructions and does not respond back. Thus extensive testing would have to be done to ensure motor is behaving appropriately to all situations and commands.

### 3.2.2 Showing real time values of varying mechanical quantities

#### 3.2.2.1 Introduction

These quantities include stress, strain, load and deformation of the polyimide sheet. Due to the motor, the polyimide sheet will be elongated and compressed (deformation). This strain on the sheet will have a corresponding stress which is what we're interested in.

This stress can be calculated knowing the load and area of sheet. The load on the sheet will be measured by load cell. the load cell will generate a voltage proportion to the force applied on the polyimide sheet. However, this voltage is very small so it will be amplified before being fed into the Arduino.

#### 3.2.2.2 Inputs

Users will start testing by first specifying motor control options as given in 3.2.1. The Arduino has an analog input pin which can read voltages from 0 to 5 V, it'll be used to read the voltage from the amplifier connected to the load cell.

#### 3.2.2.3 Processing

The Arduino code will then do a process known as "smoothing" (which reduces the influence of random noise) before transmitting these values to PC. The PC software will interpolate the force on object based on the voltage reading by Arduino. Then the PC software will calculate stress from force and area using a formula. The strain will be

calculated from the position of motor, which itself will be calculated using time elapsed, speed of motor and period or frequency of cycles.

#### *3.2.2.4 Outputs*

The real time stress and strain values will be shown to the user, along with the load and deformation values.

#### *3.2.2.5 Error Handling*

If there is any sudden decrease in stress, that might indicate the polyimide sheet has broken. In that case the motor will be temporarily stopped and User will be notified to decide on the next action.

### **3.2.3 Plotting graphs in real-time: load vs deformation and stress vs strain**

#### *3.2.3.1 Introduction*

Based on the values obtained each instant (as described in 3.2.2), a new vertex will be added to the graph and it will be updated.

#### *3.2.3.2 Inputs*

The inputs will be the real-time values of various mechanical quantities, which will be calculated as described in 3.2.2.

#### *3.2.3.3 Processing*

Each instant a new vertex will be added to graph (based on the corresponding values of the mechanical quantities) and it'll connected to the previous one through a curve.

#### *3.2.3.4 Outputs*

Two continuously updating graphs will be shown to the user.

### **3.2.4 Exporting data to Excel file**

#### *3.2.4.1 Introduction*

After testing has been finished, users will be able to get an excel file with all the captured and processed data.

#### *3.2.4.2 Inputs*

Users will click on a button for exporting data.

#### *3.2.4.3 Processing*

Data will be written to an Excel file.

#### *3.2.4.4 Outputs*

After it is ready, users will be notified that the Excel file has been created.

### **3.2.5 Printing graphs**

#### *3.2.5.1 Introduction*

After testing has been finished, users will be able to print the graphs obtained.

#### *3.2.5.2 Inputs*

Users will click on a button for printing graphs.

#### *3.2.5.3 Processing*

Data will be written to a file such as a PDF or a JPEG. (To be determined). Then a print command would be sent to printer.

#### *3.2.5.4 Outputs*

Printer will print the graph. The file containing the graph will also be available for users.

### **3.2.6 "Smoothing" of raw data from amplifier connected to load cell sensor.**

#### *3.2.6.1 Introduction*

The voltage read by Arduino from amplifier connected to load cell has a lot of random noise. To fix this problem, Arduino code will do a process called "Smoothing" before sending voltage reading to PC software.

#### *3.2.6.2 Inputs*

The Arduino has an analog input pin which can read voltages from 0 to 5 V, it'll be used to read the voltage from the amplifier connected to the load cell.

#### *3.2.6.3 Processing*

Smoothing involves taking average of multiple data points. So for example if a voltage reading is required every 100 ms, Arduino will be coded to actually take 10 readings separated by 10 ms and then average them. This process results in most of the random noise in sensor data cancelling out.

#### *3.2.6.4 Outputs*

The Arduino code will send the processed data (voltage readings) to the PC software.

### **3.3 Non-Functional Requirements**

#### **3.3.1 Performance**

Performance of the software is not a big concern since no heavy processing is done. However, the motor and load cell, amplifier and motor driver should have appropriate resolution and response time.

#### **3.3.2 Reliability**

Since the UTM will be used for research, reliably getting correct results is important. While testing can make sure the software is bug free, it will be important to periodically check that hardware components such as the load cell sensor are working properly and giving correct output.

#### **3.3.3 Availability**

The software will be used by users with PCs having a wide variety of hardware configuration. Thus the software should work on PCs with the windows operating system and any hardware configuration.

#### **3.3.4 Security**

Since physical access to UTM will be required for using the software, security is not an issue.

#### **3.3.5 Maintainability**

The hardware components in the UTM can be changed later for any reason. Thus the software should be such that minimal amount of code (only the code interfacing with the hardware) needs to be changed if that happens.

#### **3.3.6 Portability**

Any user with a windows PC should be able to install this software and connect to the UTM.

### **3.4 Design Constraints**

#### **3.4.1 Operating system: Windows**

#### **3.4.2 Software should be made using the “Processing” platform (see 3.1.3.1)**

#### **3.4.3 Arduino Uno as the microcontroller. The coding for it would be done using the Arduino IDE. (see 3.1.3.2)**

#### **3.4.4 Arduino Uno will be required to interface with specific motor controller, motor, amplifier and load cell. The models are not yet completely fixed and this constraint will be updated later.**