**FACULTY OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**EE 402 – Senior Project II**

**2016 SPRING**

**FINAL REPORT**

**IMPEDANCE ANALYZER**

**İlyas Mert Düden**

**Jury Members:**

**Supervisor: Göksenin Yaralıoğlu**

**Jury member 1: Ahmet Tekin**

**Jury member 2: Murat Uysal**

# Project description and goals

Impedance Analyzer project aims that to measure impedance of a component in MATLAB with phase and magnitude plots. In order to do that GPIB connection and Instrument Toolbox connection are established.

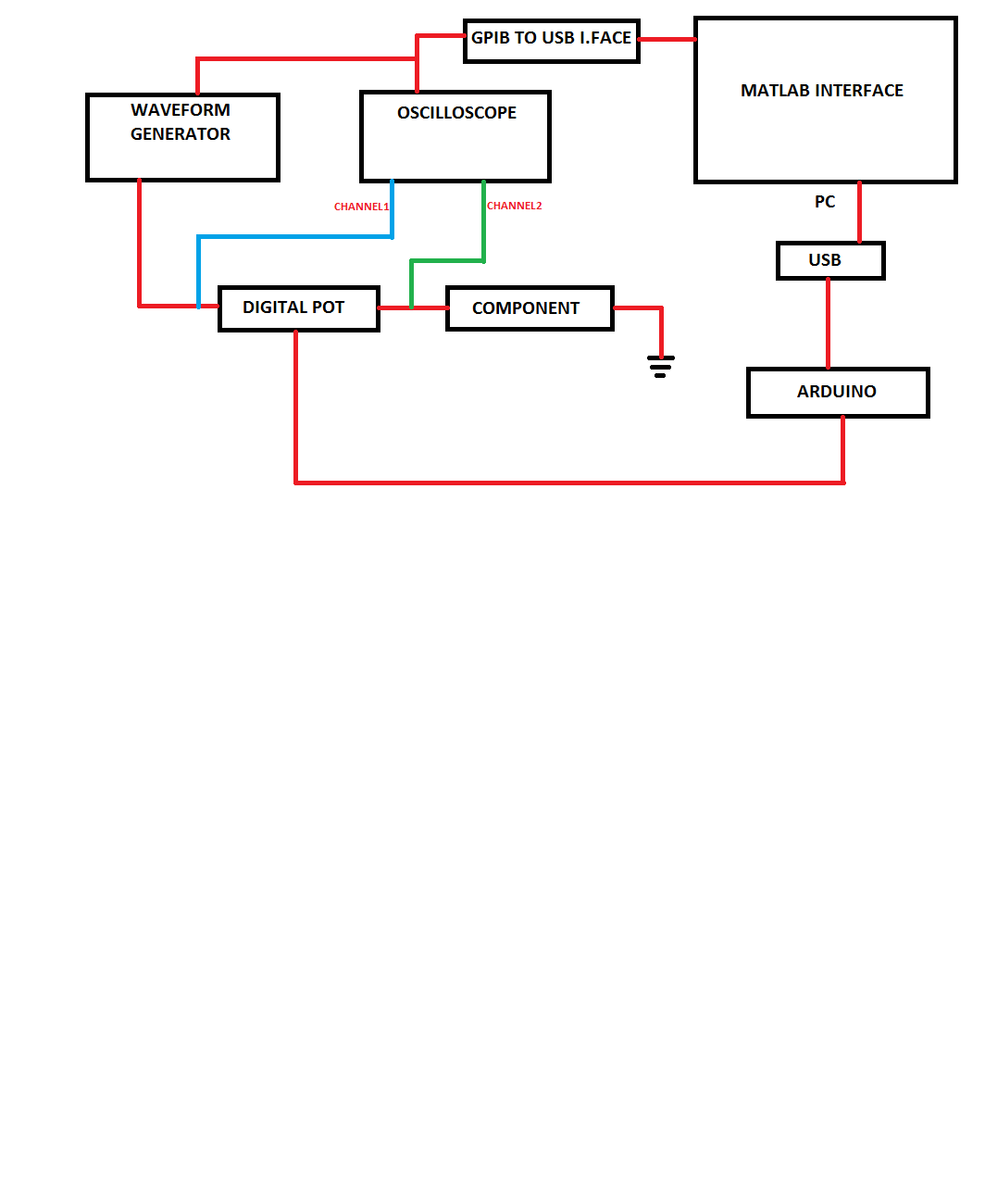
****

Figure 1: Block Diagram for the Impedance Analyzer

The list below shows deliverables for the project:

* Phase and Magnitude Plots of Different Components
* Digital Potentiometer Controlled By Arduino
* User Interface with Desired Inputs

# Work done

2.1) Electrical Background of the Project

Before implementing code blocks into MATLAB, there was a need for understanding impedance concept. Impedance is defined as the frequency domain ratio of the voltage to the current. In other words, it is voltage–current ratio for a single complex exponential at a particular frequency ω. In general, impedance will be a complex number, but this complex number has the same units as resistance, for which the SI unit is the ohm. For a sinusoidal current or voltage input, the polar form of the complex impedance relates the amplitude and phase of the voltage and current. In particular,

• The magnitude of the complex impedance is the ratio of the voltage amplitude to the

current amplitude.

• The phase of the complex impedance is the phase shift by which the current is ahead of the voltage.

2.2) Communicating with Instruments and SCPI

This project is consist of impedance measurement of components in a matlab GUI program over GPIB connection.

GPIB or its expansion General Purpose Interface Bus is one of the most popular and old connection standard created by Hewlett-Packard Company. Other name is IEEE 488 that is given by The Institute of Electrical and Electronic Engineers. GPIB is commonly used for communication with test equipment (instruments). (http://www.radio-electronics.com/info/t\_and\_m/gpib/ieee488-basics-tutorial.php & Poole, 2011)

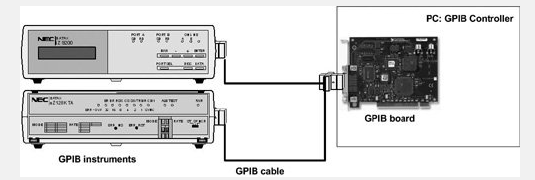


Figure 2: GPIB Connection Schema

GPIB connection in MATLAB is provided by [Instrument Control Toolbox](http://www.mathworks.com/products/instrument/) that connects instruments such as waveform generator, oscilloscope, network analyzer or power supply. This toolbox supports TCP/IP, UDP, I2C, SPI, and Bluetooth® serial protocols.

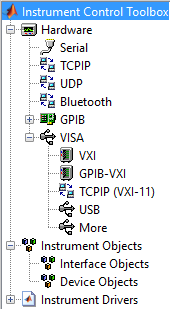


Figure 3: Instrument Control Toolbox

As Figure 3 shows, there are several connection type in the Toolbox. Once the instrument connects Matlab, Matlab creates an object to evaluate its data and write commands on it. It is up to user to write your own code to connect in a Matlab Script or use the code Matlab created as session log which keeps instrument commands as a matlab code. By saving this code to a MATLAB file, you can execute the same commands programmatically.

The Standard Commands for Programmable Instrumentation is created to develop common a kind of program language that can be used in all instruments. The SCPI standard is developed on IEEE-488.2 communication standard. Although SCPI is a programming language itself, it can be embedded to MATLAB, C, C+, C#, LabVIEW or Python compilers. This means an instrument can be controlled by using only SCPI commands.

2.3) Impedance Calculation Algorithm

Since impedance is complex value and gathering complex data from oscilloscope is not available, we need to create a complex number ourselves.



Component impedance illustrated in block diagram is formulated as follows:

*Component Impedance = Voltage at Channel 2 (Green Probe) / Current*

The steps below explain producing complex number impedance in MATLAB algorithm with the data comes from oscilloscope:

Step 1: Gather voltage value at Channel 1 and Channel 2, and also phase between these voltages. Assume that voltage 1 has no phase.

* Voltage amplitude at Channel 1 = VAMP1
* Voltage amplitude at Channel 2 = VAMP2
* Phase between Voltages = Phase
* Voltage at Channel 1 = V1
* Voltage at Channel 1 = V2

Step 2: Form voltage at Channel 2.

V2 = VAMP2 \* ( cos(phase) + jsin(phase))

Step 3: Calculate current by voltage drop on digital potentiometer

Current = (V2-V1) / R

Step 4: Calculate Impedance

Z = V2 / I

2.4) MATLAB Interface

The interface below asks several inputs (start frequency, stop frequency, data points, voltage amplitude and resistor value) from the user to execute linear sweep and data gathering. Then the program collects data from the oscilloscope to plot impedance magnitude and phase.

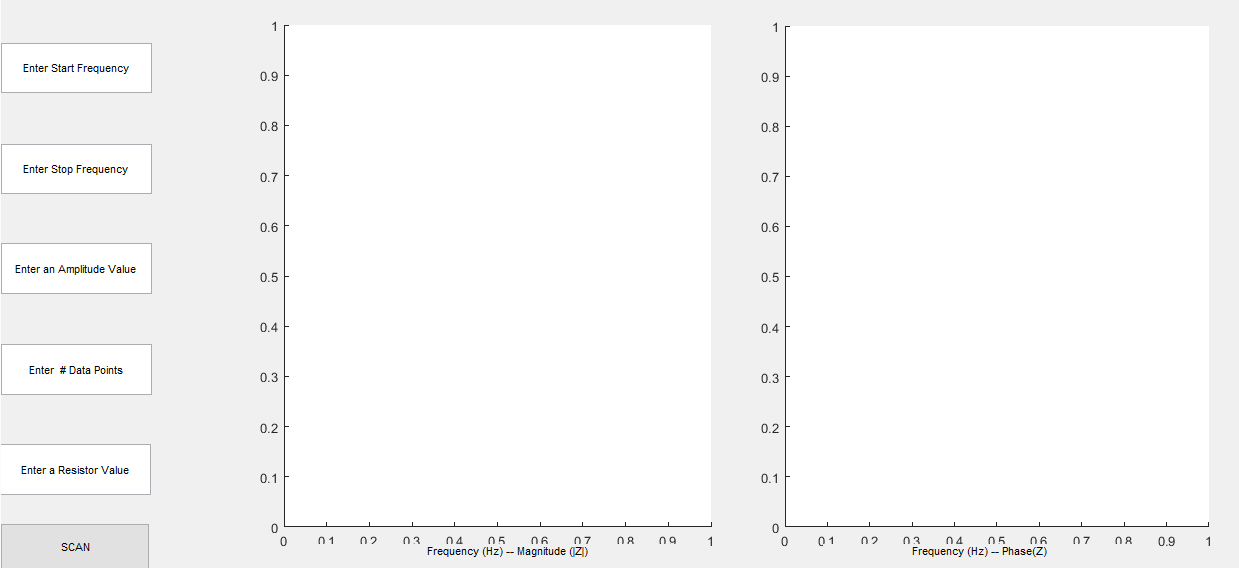
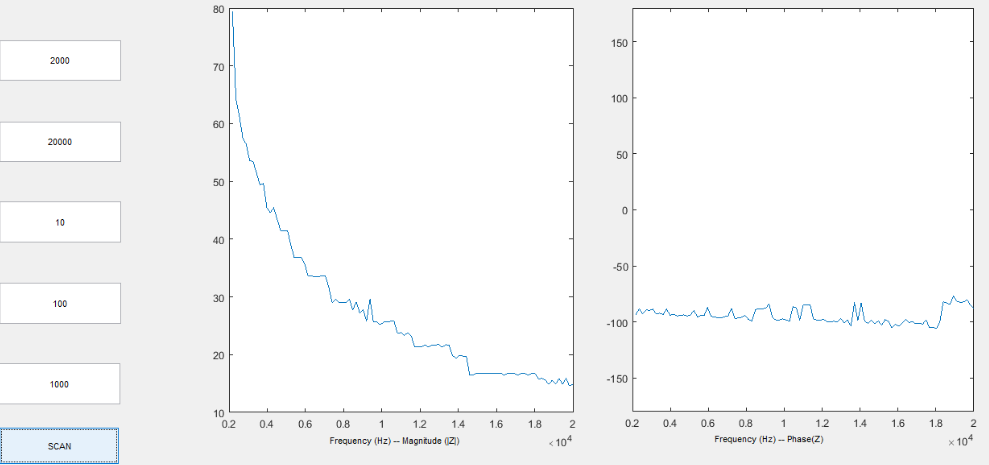


Figure 4: MATLAB GUI

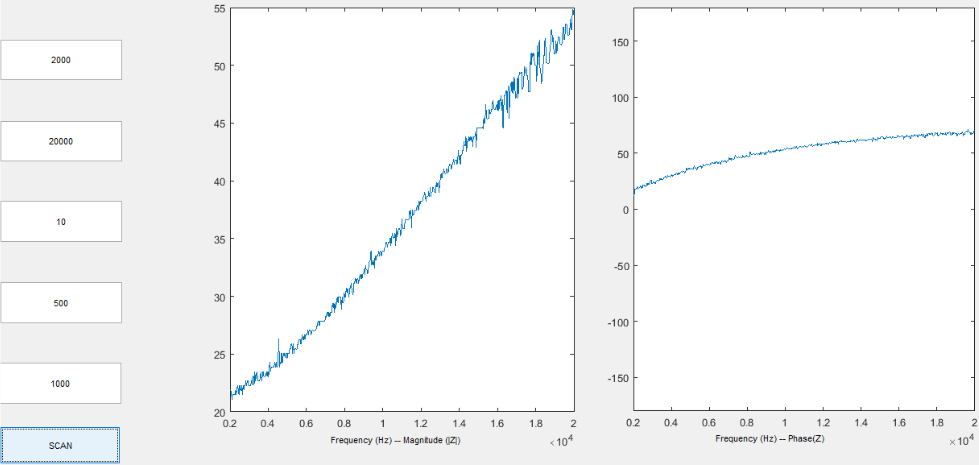
2.5) Results

The system were tested with different components and several results are illustrated below.

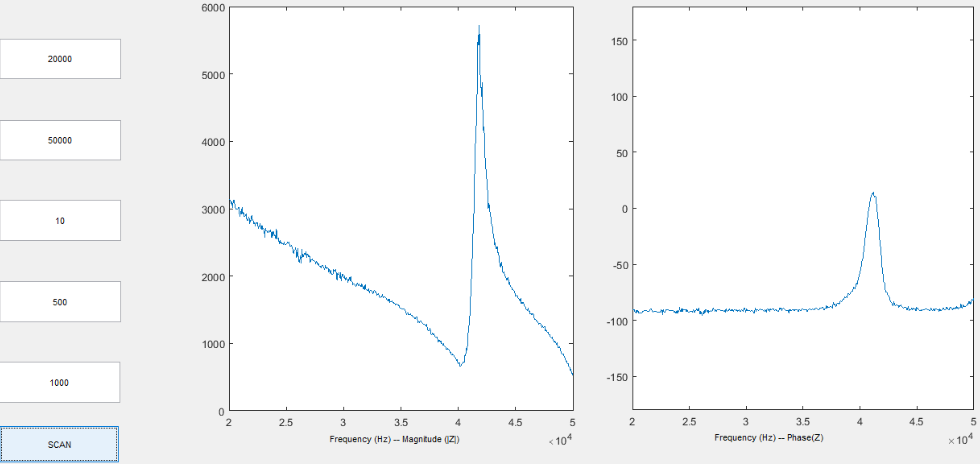
1) Capacitor (4.7 uF)



2) Inductor (460uH)



3) Buzzer



2.6) Digital Potentiometer

AD5290 (10k version) digital potentiometer was programmed in Matlab thanks to Arduino package add-on. The package allows to program Arduino without any C programming. Like scripts that are wrote in Instrument Control Toolbox, Arduino is defined in Matlab as an object.

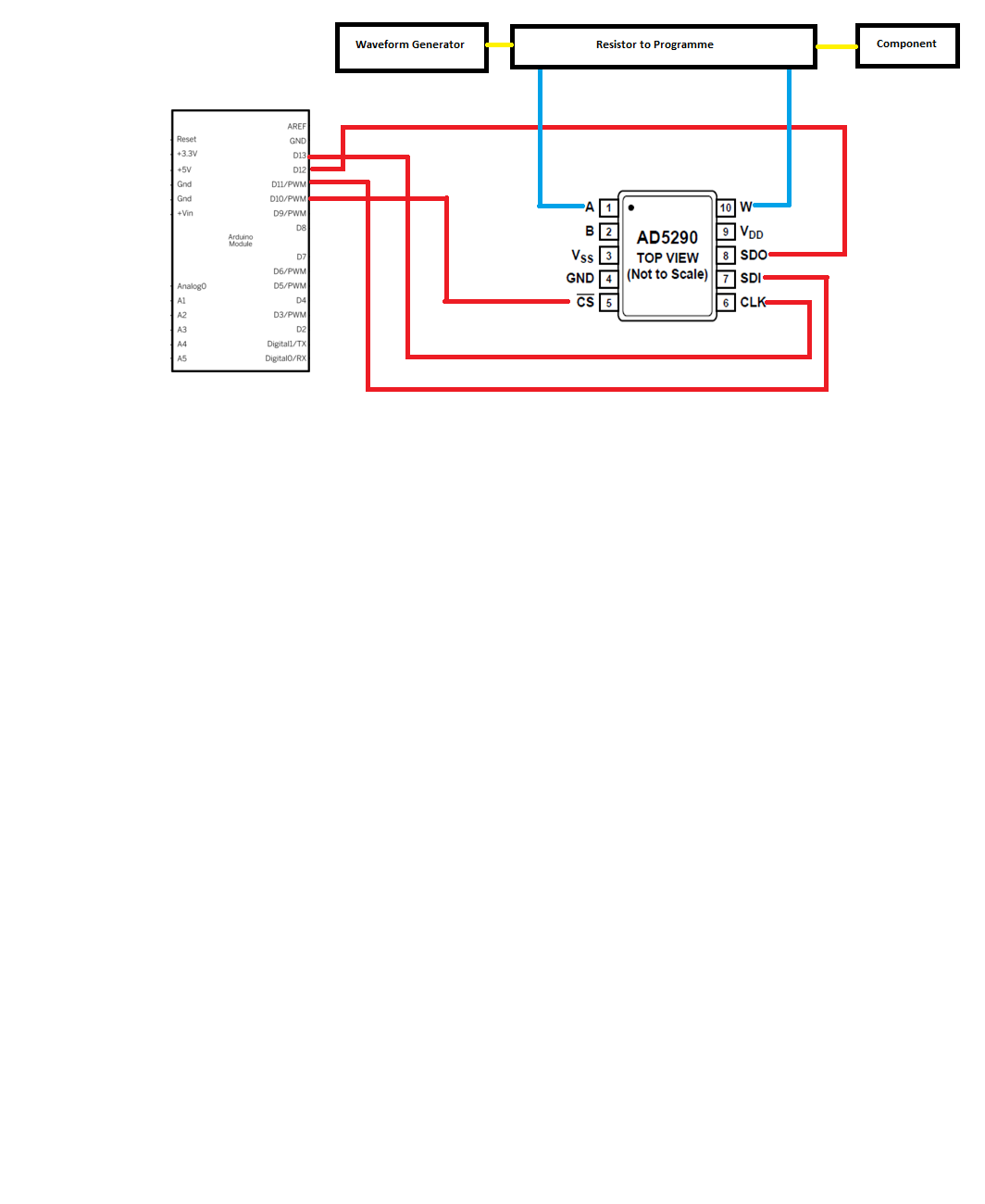
****

Figure 2: Pin Connection for Digital Potentiometer

The nominal resistance between Terminal A and Terminal B, RAB, is available in 10 kΩ with ±30% tolerance and has 256 tap points accessed by the wiper terminal. In this project, I will use RWA value as illustrated in the figure 6. The equation for this resistor value is;

****

Since the RW (wiper) value equals the 50 ohm, minimum value can be assigned for RAW is 150 ohm.

2.7) Time Planning

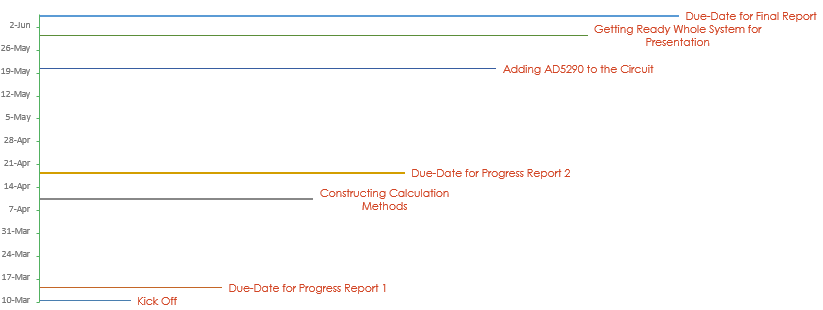


Figure 3: Time Planning

Briefly, I spent less time that I thought on Digital Potentiometer coding and connection. The most challenging part was to adjust sweep time and data acquiring time synchronously.

# References

Coates, E. (2007). AC Theory.

Device, A. (2005). AD5290 DataSheet.

http://www.radio-electronics.com/info/t\_and\_m/gpib/ieee488-basics-tutorial.php, & Poole, I. (2011). *GPIB / IEEE 488 Tutorial*. Retrieved from Radio Electronics: http://www.radio-electronics.com/info/t\_and\_m/gpib/ieee488-basics-tutorial.php

Keysight. (2015, 11 20). *33522B Waveform Generator, 30 MHz, 2-Channel with Arb.* Retrieved from Keysight Technologies: http://www.keysight.com/en/pd-2155053-pn-33522B/waveform-generator-30-mhz-2-channel-with-arb?nid=-33134.1026944.00&cc=TR&lc=eng

Keysight. (2015, 06 09). *Programmer's Guide for InfiniiVision 2000 X-Series Oscilloscopes.* Retrieved from Keysight Technologies: http://www.keysight.com/main/redirector.jspx?action=ref&cname=EDITORIAL&ckey=2015143&lc=eng&cc=TR&nfr=-33575.970737.00