BIRZEIT UNIVERSITY

Electrical Engineering Department

Engineering Simulation Laboratory

ENEE4104

BZUENEE

Prepared By:

ENEE DEPARTMENT

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Table of Contents

Experiment 1 Negative Feedback, Modular Design and Sub-circuit in PSpice **3**

Experiment 2 Power Electronic Converters in Orcad **13**

Experiment 3 PCB design and implementation **21**

Experiment 4 Introduction to MicroC Program **32**

Experiment 5 Linear Systems Simulation in MatLab **40**

Experiment 6 Modeling a DC motor in Simulink **47**

Experiment 7 Amplitude Modulation and Demodulation in LabVIEW **57**

Experiment 8 Introduction to Data Acquisition Systems in LabVIEW **63**

Experiment 9 PWM Control Project **70**

Experiment 10 Filter Design Using MATLAB **72**

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment 1

Negative feedback, Modular design and sub-circuits using Orcad

Introduction

Negative Feedback

In most of electronic circuits design, negative feedback is used to improve the system performance and stability. In this experiment we'll simulate an open loop amplifier, then, add a negative feedback network, find the new characteristics, and compare the two amplifiers in terms of input impedance, output impedance, gain, and bandwidth.

Modular Design and Sub-Circuits in Orcad

Orcad simulation tool offers the ability to employ hierarchal design "Modular Design" i.e. any circuit design can be fragmented into smaller simpler circuits "Modules" each of which can be represented by a block with input and output pins. These blocks then used to build a more complicated system. Also, one can use Orcad to build his own library components.

In this experiment, we'll begin by implementing two blocks that are then used to implement a multi-stage amplifier. Then, the amplifier circuit will be converted into a sub-circuit "symbol" and added to Orcad library.

Objectives

- To learn the concept of modular design using Orcad.
- To make a sub-circuit "symbol" and add it to the Orcad Library.
- To compare the characteristics of an open-loop and a closed-loop amplifier.

Pre-Lab

Figure 1 and figure 2 show an open loop multistage amplifier and a closed loop multistage amplifier, respectively.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

You have to:

1. Simulate both circuits in Fig.1 and Fig.2 using Orcad.
2. Determine Avmid "Gain", Zin, Zout and Bandwidth of both circuits.
3. Repeat the simulation with the load values: 10kΩ to 100Ω.
4. Compare the two amplifiers' results with both load values.

Hint : you can either use the part "Port" to represent the VCC and VEE as shown in the following figures, or, you can easily connect the VCC and VEE sources directly.

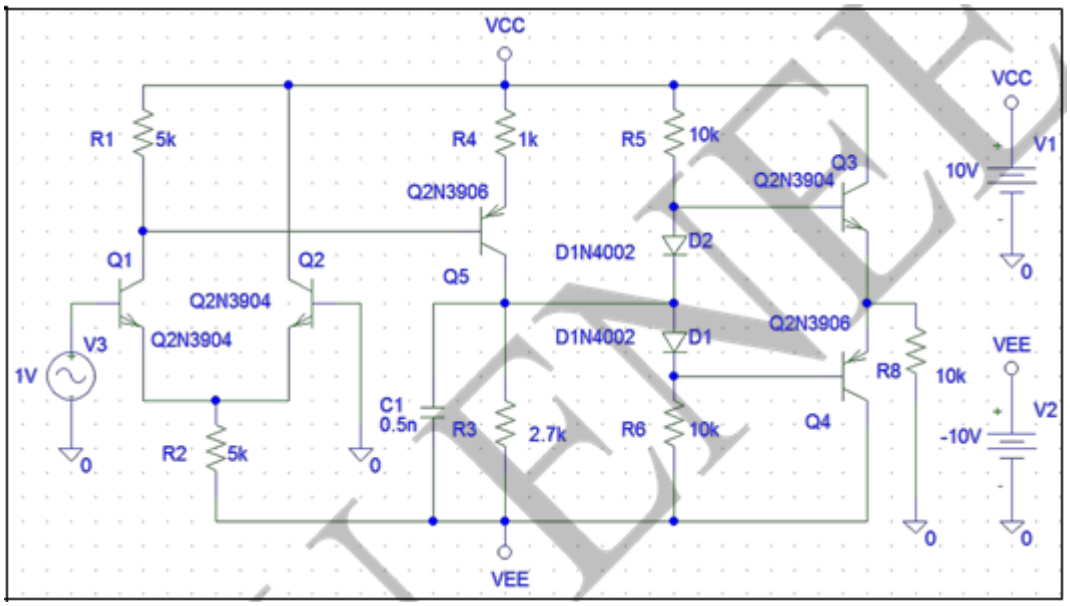


Figure 1: Open Loop Multistage Amplifier

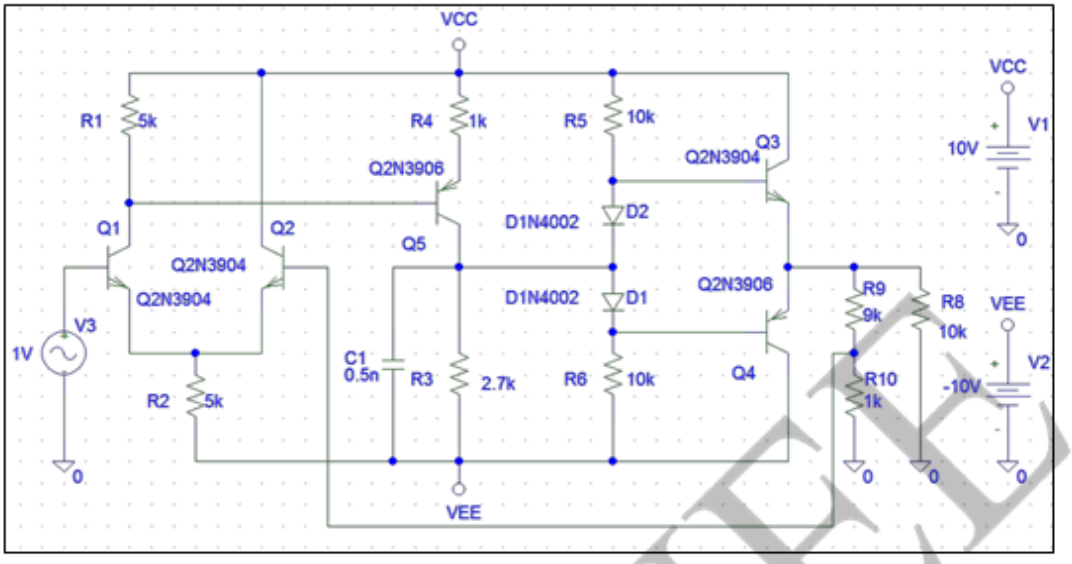
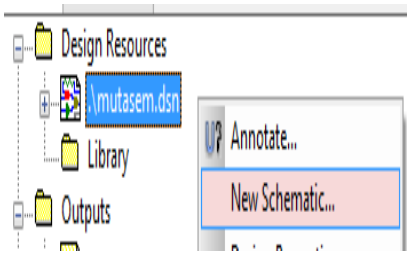


Figure 2 Closed Loop Multistage Amplifier

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Procedure

Modular Design:



I. As mentioned earlier, we are dividing the amplifier into smaller modules, in our case, two modules as shown in Fig.3 and 4. Draw the circuits shown Fig.3 and 4 on separate schematics and save them as "lowfirst.sch" and "lowsecond.sch", respectively.

Hint:

Right-click on **.dsn** in **Project Manager -> new schematic..**
Right-click on **new Schematic** Folder in **project Manager -> Rename** TO " lowfirst.sch "
Right-click on **root Schematic** Folder in **Project manager -> New Page**

Repeat the same steps for "lowsecond.sch",

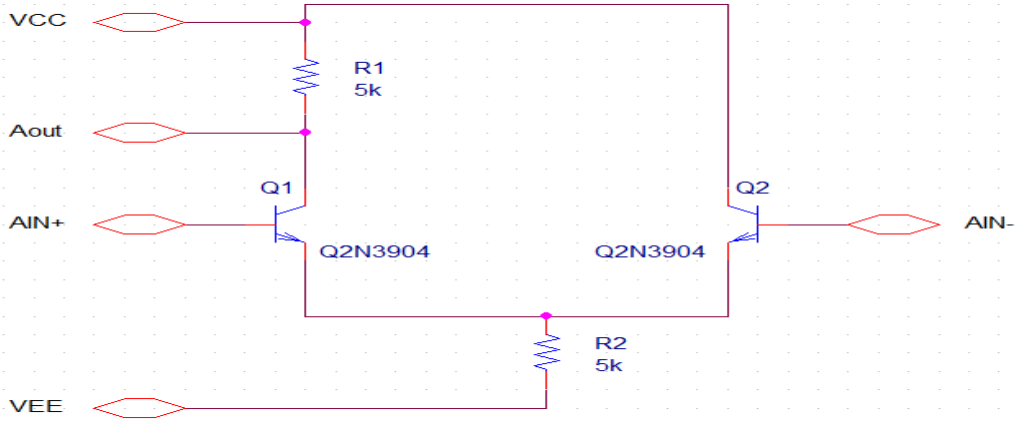


Figure 3: lowfirst schematic

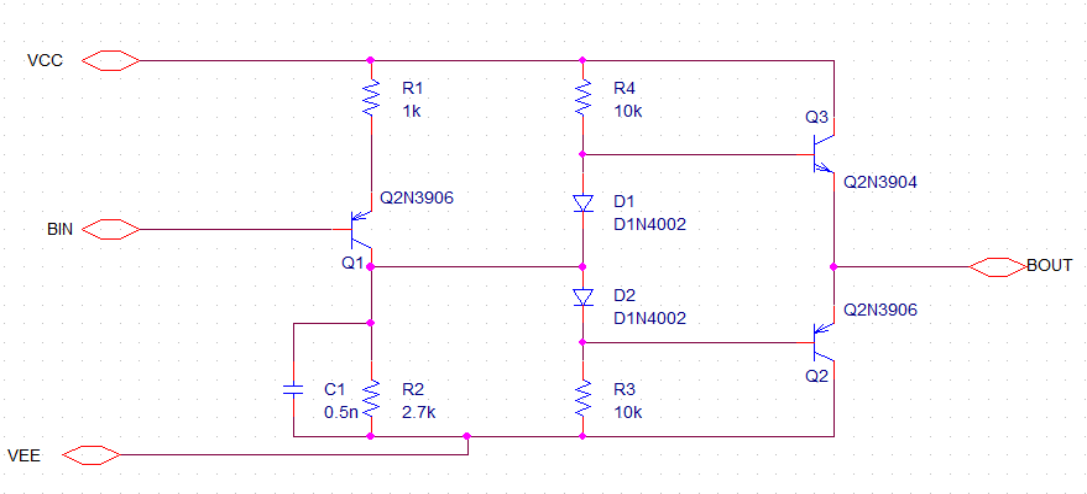


Figure 4: lowsecond schematic

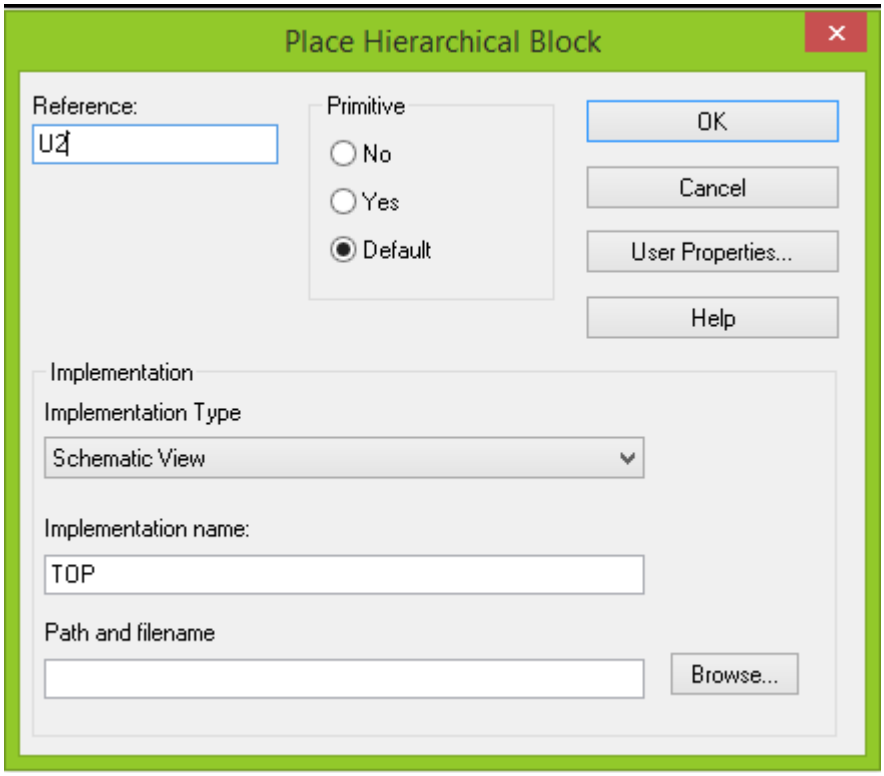
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

II. Open-Loop Amplifier

The circuit shown in Fig. 5 represents a BJT amplifier.

1. Draw the circuit in a new schematic page and save it as "top.sch".

Hint: *the block in the middle can be added by clicking* **place →Hierarchical Block** and setting the parameters as in the following dialog box:



You have to type the implementation name, because the schematic does not exist yet. Click **[OK]** and draw a rectangle on the schematic page. You will see the reference name **U2** and the implementation name **TOP** for the hierarchical block.

*To add pins select the rectangle and use the command "**place H pin**"* *and locate the pin on the rectangle after naming them.*

Hint:

Right-click on **.dsn** in **Project Manager -> new schematic**.
Right-click on **new Schematic Folder** in **project Manager -> Rename** TO TOPA
Right-click on **Schematic Folder** in **Project manager -> Make Root**
Right-click on **Root Schematic Folder** in **Project manager -> New Page**

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

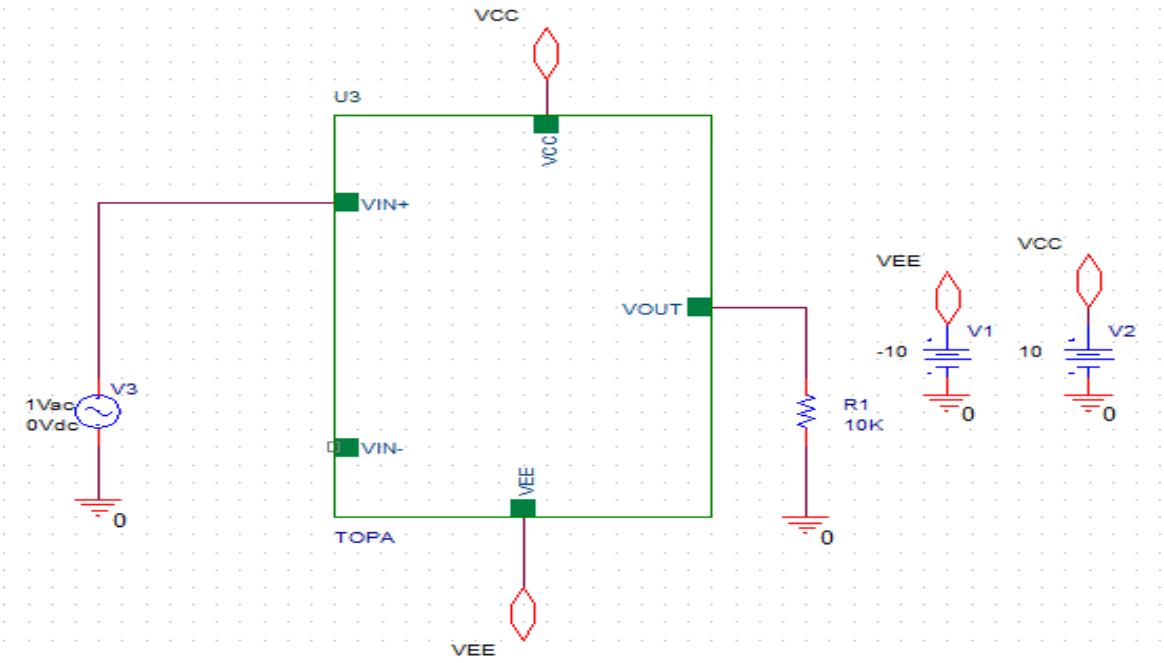
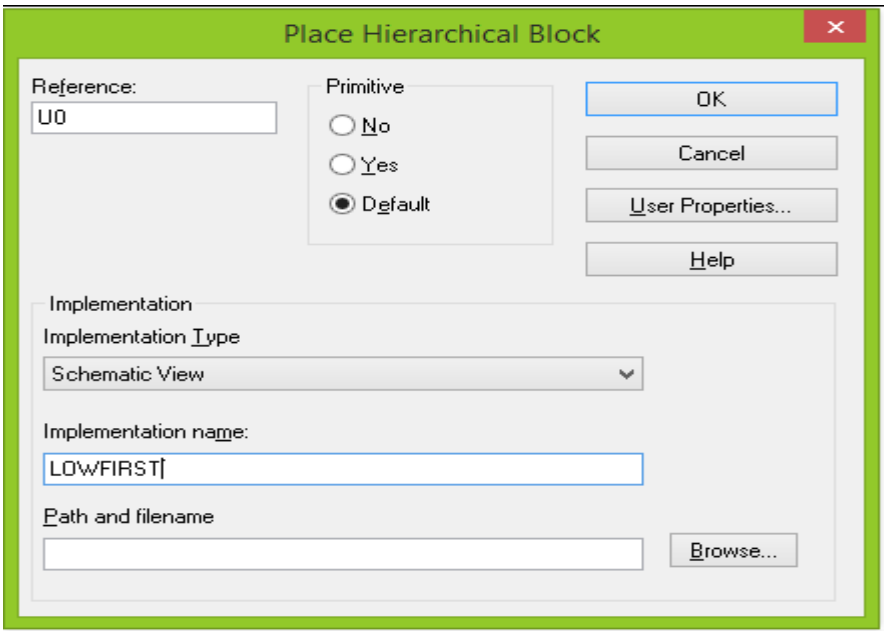


Figure 5: top schematic

2. Double clicking the block. A pop-up window will appear, asking you to type the name of the circuit to be added inside this block, name it "middle".

3. Now, draw the "firstlow" block inside the middle schematic *: the block in the left can be added by clicking* **place →Hierarchical Block.** Set the **Place Hierarchical Block** dialog box *as follow " make sure that you named the implementation name the same name that you make for the first schematic*



STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

4. Now, draw The first low block inside the middle schematic *: the block in the right can be added by clicking* **place →Hierarchical Block** Set the **Place Hierarchical Block** dialog box *before " make sure that you named the implementation name the same name that you make for the second schematic*"

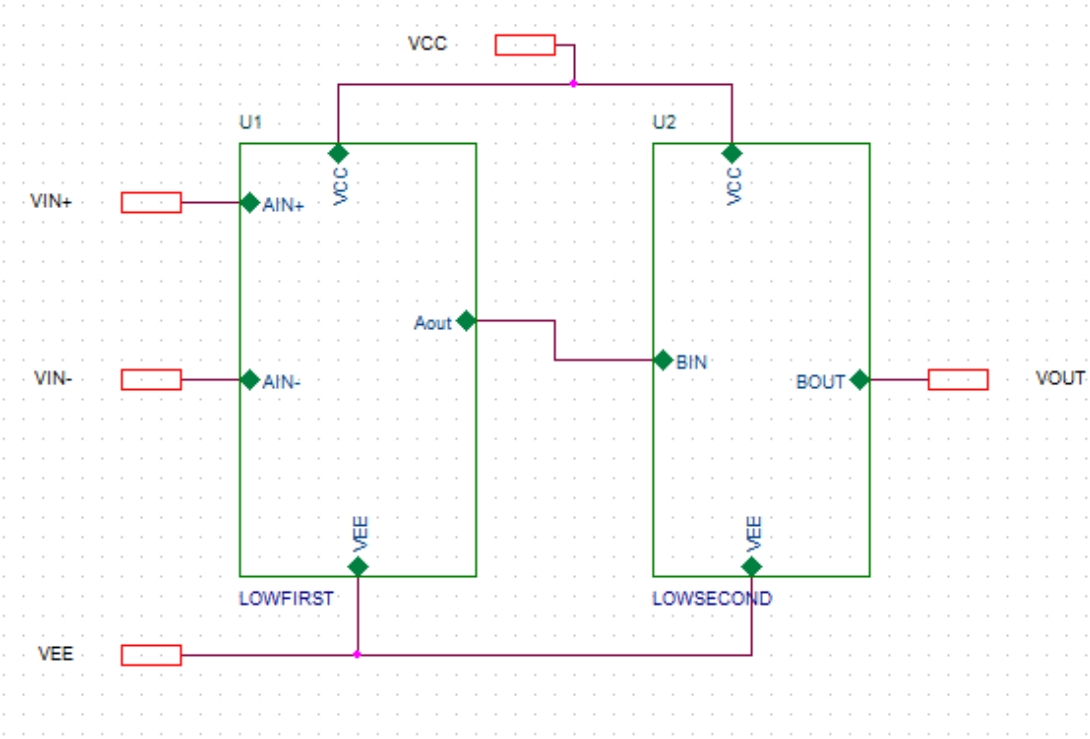


Figure 6 : MIDDLE SCHEMATIC

Now, go back to the "top" schematic which represents the amplifier. Use AC sweep analysis to obtain the characteristics of the open loop amplifier shown earlier in figure 5.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Making a sub-circuit using PSpice

We need to make our own amplifier and add it to the OrCAD library so that it can be accessed and used in any design.

1. Draw the circuit shown in figure 7 below, which is the multistage amplifier we want to convert into a symbol "sub-circuit".

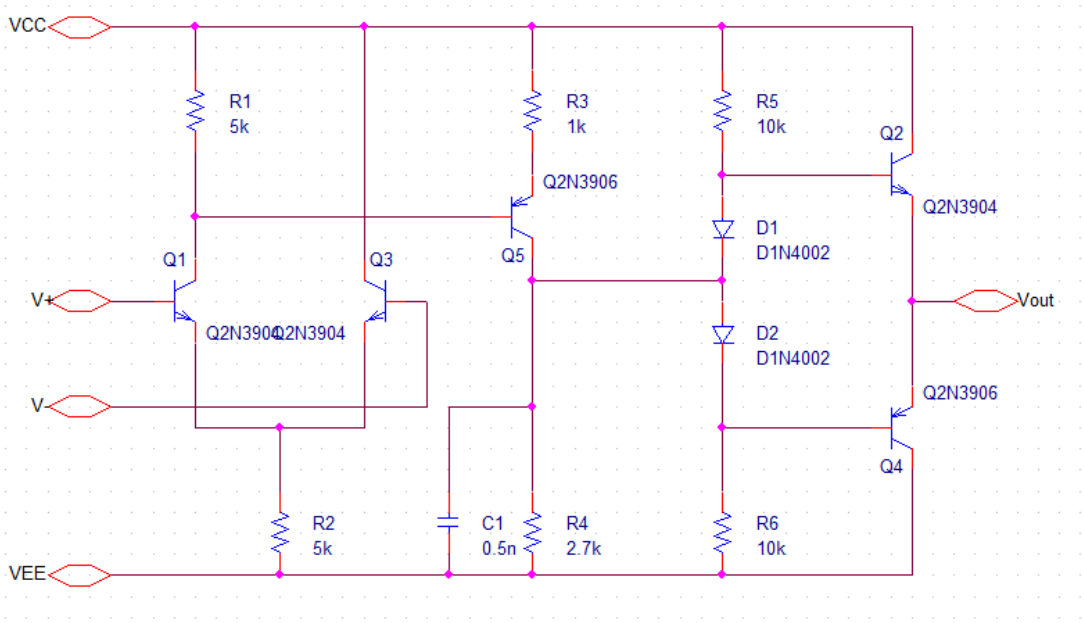
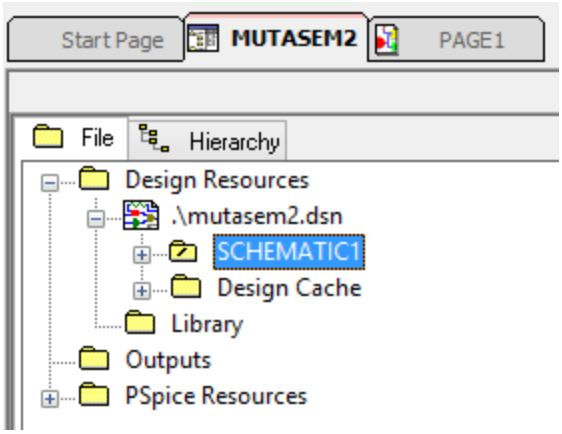


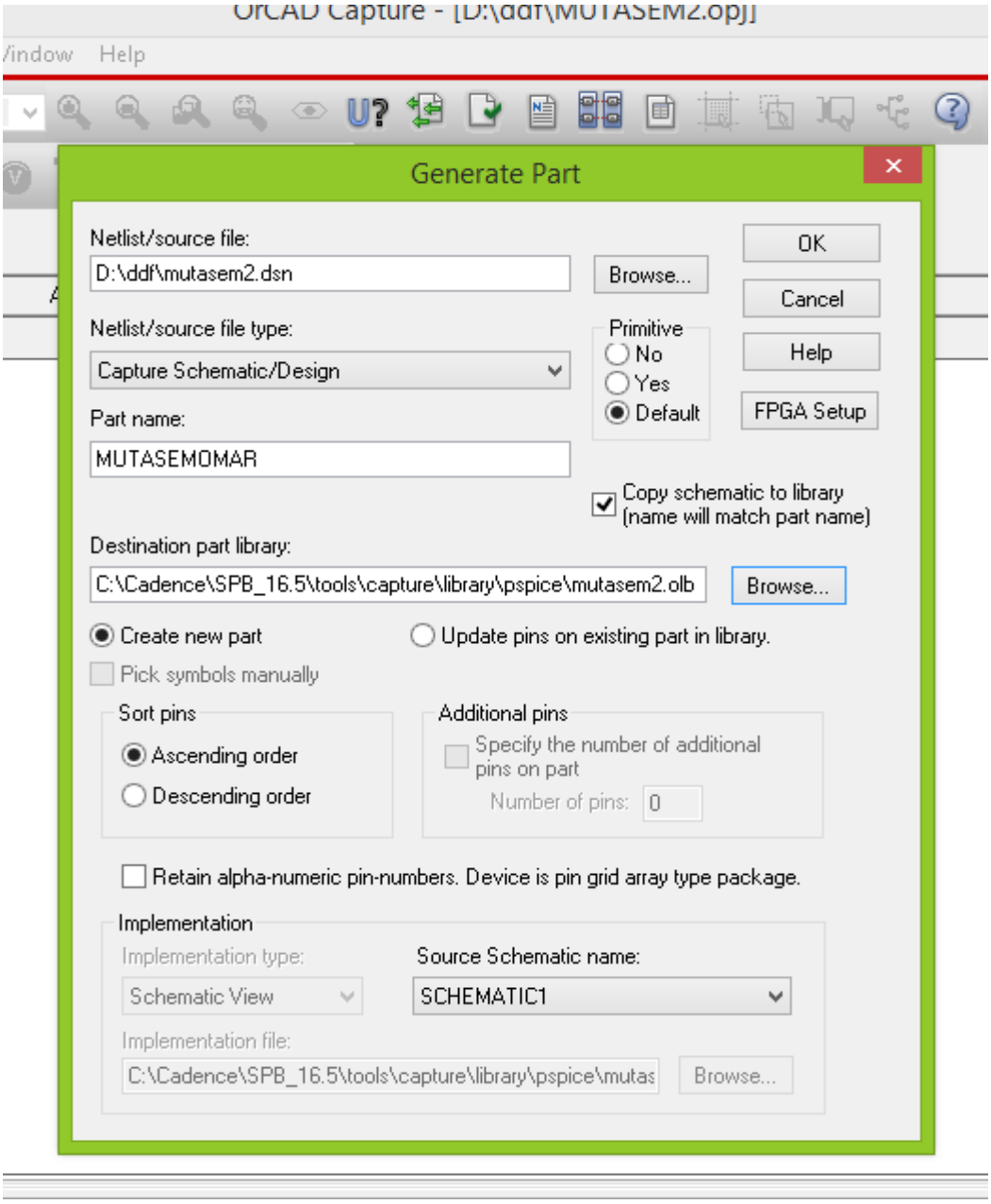
Figure 7: MULTISTAGE AMPLIFER

2. To convert the circuit into a sub-circuit, first , select the schematic you want to turn into a sub-circuit. NOT THE PAGE! THE SCHEMATIC!



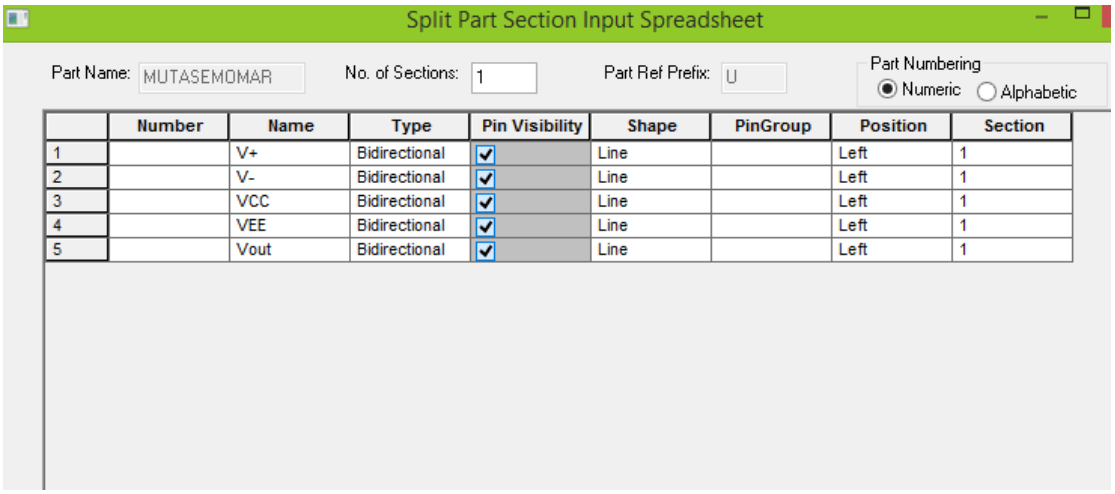
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

3. Then, while the schematic is highlighted, click Tools → Generate part



4. Make “Part Name” to YOUR NAME . Under “Netlist/source file”, make sure that the design you are working on is the one selected. Make sure “Create new part” is selected. The “Destination part library” is where you want to save your part CHOOSE THIS PATH : *"C:\Cadence\SPB_16.5\tools\capture\library\pspice"* ”. Source Schematic name is the name of the schematic your are turning into a sub-circuit. Once everything is to your liking, hit okay. You should get a screen that looks like the following:

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

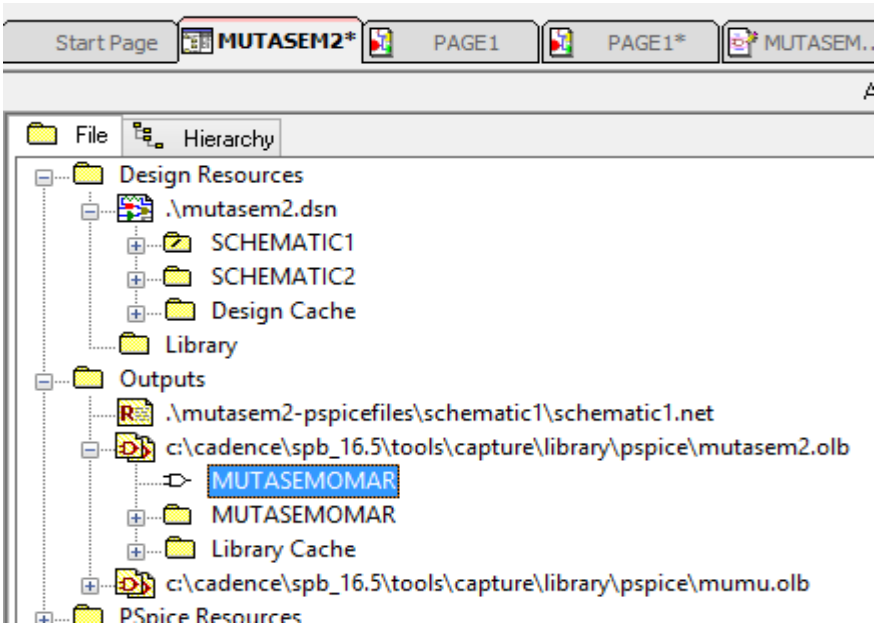


	Number	Name	Type	Pin Visibility	Shape	PinGroup	Position	Section
1		V+	Bidirectional	☑	Line		Left	1
2		V-	Bidirectional	☑	Line		Left	1
3		VCC	Bidirectional	☑	Line		Left	1
4		VEE	Bidirectional	☑	Line		Left	1
5		Vout	Bidirectional	☑	Line		Left	1

5. Under "Position", you can click each cell for a drop-down menu with left, right, top, and bottom. This is the position of the pin. Everything else is pretty self explanatory. You can add in pins if you want. You can change the name of pins. Make pins default to visible or invisible. Point is, the next thing you do is hit SAVE once you're pins are where you want them.

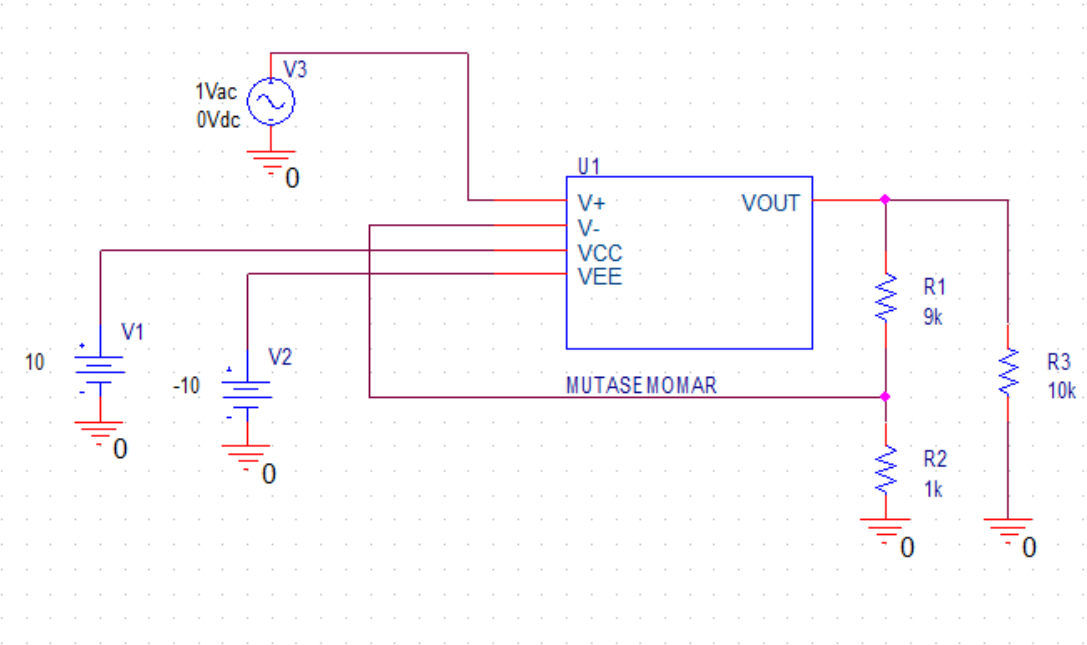
6. Now, you can easily open a new schematic and add your symbol from the library like any other part.

7. To edit the shape, name, pins' names of your new symbol, DOUBLE click on the part name which appeared on the output as shown, and then you can change positions , and whatever you want.



STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

8. Use your new component to simulate an amplifier with a negative feedback as shown in the figure below. And find its characteristics



STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment 2

Power Electronic Converters in Orcad

Introduction

Orcad[1] is a software package that can be used to simulate and design electrical circuits, analog and digital. The Orcad library contains most of the widely used electrical components, starting with analog components like resistors and capacitors up to IC's of different sizes and functions. Also, Orcad can be used to design Printed Circuit Boards "PCB's" as will be demonstrated in Experiment 4 of this lab.

Power Electronic Converters[2]

Power electronic converters are mostly used in the design of DC power supplies "switch mode power supplies" to convert a DC voltage level to another level that is higher or lower, this type of converters employs high speed power electronic switches like MOSFET's and IGBT's that can be operated at high frequencies where the output voltage can be controlled by controlling the duty cycle of these switches.

In this experiment you'll be simulating a variety of DC to DC converters in addition to an Inverter "DC to AC" and you'll be studying the effect of varying different parameters like the switching duty cycle, the output filter size and the load.

Step down Converters "Buck Converters"

In a step down converter the output voltage can be varied between 0V and V_{in} by controlling the duty cycle of the switch, the output voltage can be expressed in terms of the duty cycle (ρ) and the input voltage as follows,

$$V_{out} = \rho V_{in} \quad , \quad 0 < \rho < 1$$

Step Up Converters "Boost Converters"

A step up converter is used to boost the input voltage and deliver a higher DC output voltage; its output can be expressed as follows,

[1] You'll need a full version of Orcad to continue with this experiment and to be able to do the prelab part - the student version isn't enough- you need Orcad 10.3 or higher.
[2] You're asked to review the theoretical part by referring to your Power Electronics class notes.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

$$V_{out} = -\frac{V_{in}}{1-\rho} \quad , \quad 0 < \rho < 1$$

As noticed, its operation isn't stable at high values of the duty cycle since the output voltage can change rapidly for the slightest change in the duty cycle.

Step Up/Down Converters "Buck-Boost Converters"

A buck-boost converter combines the operation of both the buck and boost converters i.e. the same device can be used to either step up the voltage or step it down. The output voltage of this type of converters is given by the following equation,

$$V_{out} = -\frac{\rho V_{in}}{1-\rho} \quad , \quad 0 < \rho < 1$$

A closer look at the equation shows that the converter acts as a step down converter for $\rho < 0.5$ and a step up converter for $\rho > 0.5$.

Full Bridge Converters

A full bridge or H-bridge has many applications; it can be used as a DC-DC Converter or a DC to AC converter where its output depends on both, the switching duty cycle and the switching strategy. In this experiment the H-bridge will operate as in Inverter i.e. DC to AC converter with a sinusoidal pulse width modulation "SPWM", the output voltage of such an inverter is given by the formula,

$$V_{out} = MV_{DC}\sin(2\pi ft)$$

M: is the modulation index and defined as the ratio between the carrier signal amplitude and the reference signal amplitude

f: is the frequency of the carrier signal

$\boldsymbol{V_{DC}}$: is the input DC voltage

Objectives

- To become familiar with Orcad.
- To verify the operation of different power electronic converters, and see the effect of changing the load, the filter size and duty cycle.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Prelab

The aim of this prelab is to get you started with Orcad. The following example is a simple RLC circuit that you have to simulate and handout before class.

Starting Orcad and creating a new project

a. In your start menu go to Programs>> Orcad >> Capture
b. When the program starts go to File >> New >> Project
c. A pop-up window will appear, choose Analog or Mixed A/D and fill the name of your project, for example "Exp#2_Prelab", then choose the directory where you want your project to be saved, for example C:\University_files\Simulation_Lab\Prelabs, then press OK.
d. A new pop-up window will appear, choose Create a blank project then OK.

Drawing the circuit

e. Now your project is created and you're ready to draw your circuit which is shown in the figure below. Go to Place >> Part, a pop-up window will appear to you, containing all the part you need. In the libraries section select all libraries by selecting one of them then pressing "Ctrl+A", if it's empty i.e. no libraries are in the libraries section, go to Add Library and add all the libraries you find, you won't need all of them, but sooner or later you're going to need some of them.

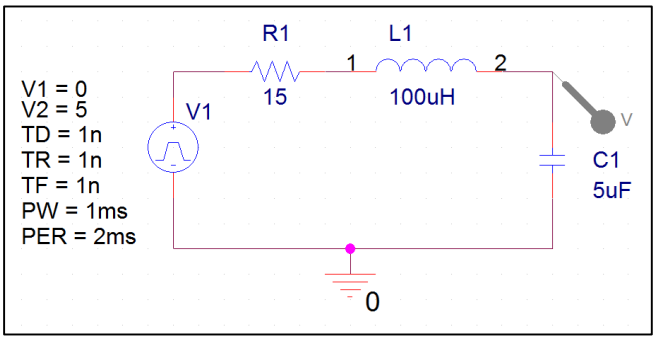


Figure 1

f. Now, you can find any part by typing its name or shortcut in the Part section above, for example, type "r" to add a resistor and choose r/analog , "c" to add a capacitor and choose c/analog, "l" to add an inductor and choose l/analog, you'll also need a "vpulse", type vpulse then choose vpulse/source.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

g. Now that you have placed all your parts, connect them by a wire, go to Place >> Wire.
h. Finally, you need to add a ground before you can simulate your circuit, go to Place >> Ground and choose 0/source.

Simulation

i. In order to simulate the circuit, you have to create a simulation profile first with the simulation parameters you need. In this experiment we are doing a transient analysis. Go to Pspice >> New Simulation Profile, then fill the name of your profile and press create. A pop-up window called simulation settings will appear, click the analysis tab, choose the analysis type "time domain-transient", fill the run time with 2ms, and the maximum step size with 0.01ms. Go to the probe window tab and check the Display Probe Window box and choose After simulation has completed then press OK.
j. Now place a voltage marker on the capacitor to show its voltage by going to Pspice >> Markers >> Voltage level. Now that your circuit and simulation profile are ready, go to Pspice and press Run. A new window will appear showing the voltage across the capacitor. Copy the simulation results to your prelab report.

Exercise

Now that you've learned to use Orcad, calculate the critically damping value of the resistor i.e. the value of the resistor which gives a critically damped output on the capacitor and show the calculation in your prelab report. Then, use your result to do a parametric analysis using 3 values for the resistor "1 Ohm, your result, 15 Ohms".

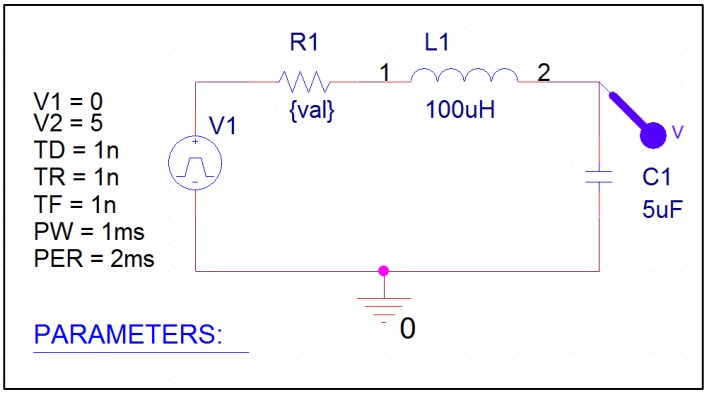


Figure 2

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

You need to rename your resistor to "{val}", then add the "PARAM/DesingCache" part, double click on the PARAMETERS part, click Add column and fill "val" in the name section, and 1 in the value section.

Go to Pspice >> Edit simulation profile, then, check the parametric sweep box and choose global parameter and fill the Parameter Name with "val", then, choose value list and fill in your values separated by a comma, then press OK and run the simulation.

The simulation output must contain three different graphs with different colors; copy the results to your prelab report.

Procedure

Step-down Converter

1. Draw the circuit shown in the figure below.

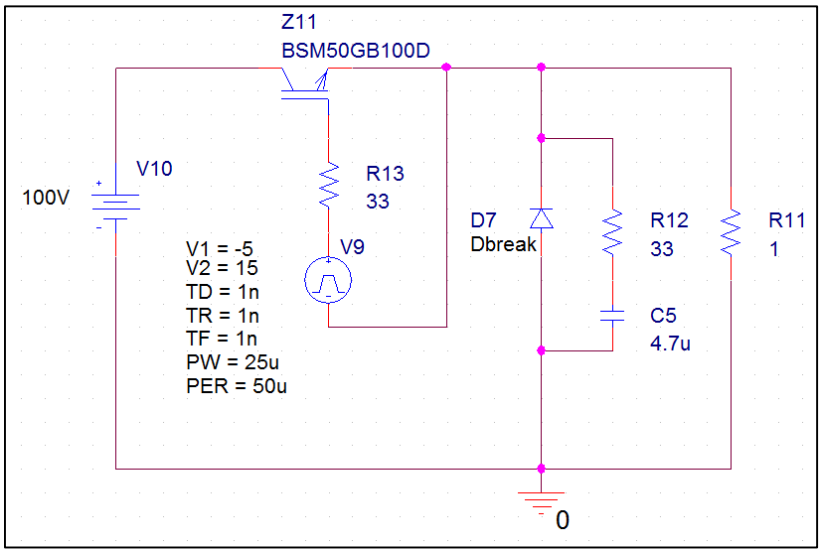


Figure 3

2. Create a simulation profile with a 5ms run time.
3. Show the output voltage and its average.
4. Show the IGBT current, the IGBT gate current and V_{ce} on different graphs[3]. You need to magnify your plot for few cycles.

[3] To add different plots, run your simulation, then, in the simulation output window go to Plot >> Add plot to window, click at the new blank window then go back to your schematic and place a marker on whatever you want to display. You can also use the alternative display button beside the help button to expand your graph.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

5. Repeat step 3 for different duty cycles (20% and 70%). use the FFT analysis to show the frequency spectrum of the output voltage for the 70% case only. "note the effect of changing the duty cycle on the output average value"
6. Add an LC low pass filter as shown in the figure below and repeat the simulation for 20%, 50% and 70% duty cycles. Show the output voltage and IGBT current for each duty cycle. use the FFT analysis to show the frequency spectrum of the output voltage for the 70% case only "note the effect of adding a filter"

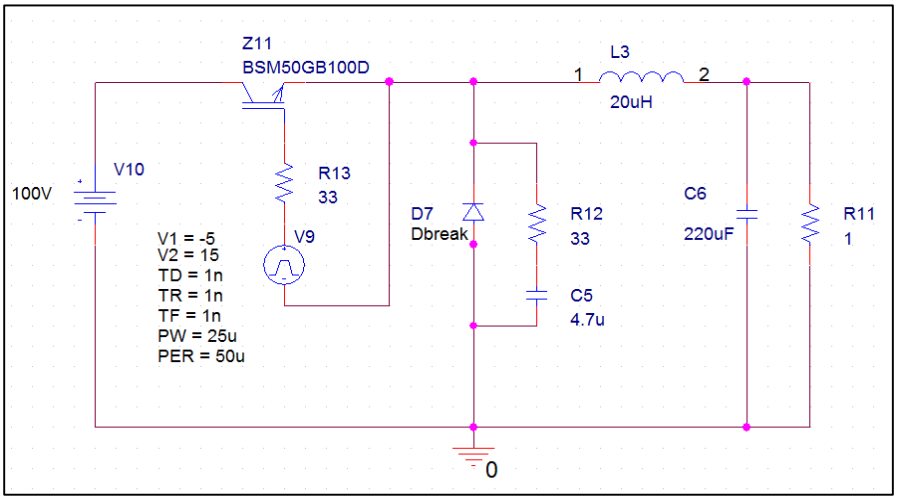


Figure 4

7. Change the filter capacitance to 22µF instead of 220µF and repeat the simulation for a 70% duty cycle. Show the output voltage. "note the effect of changing the filter size"
8. Return the filter to its original size then change the resistive load value to 20Ω instead of 1Ω. Show the output voltage for a duty cycle of 70%. "note the effect of changing the load value"

Buck-Boost Converter

1. Draw the circuit shown in the figure below.

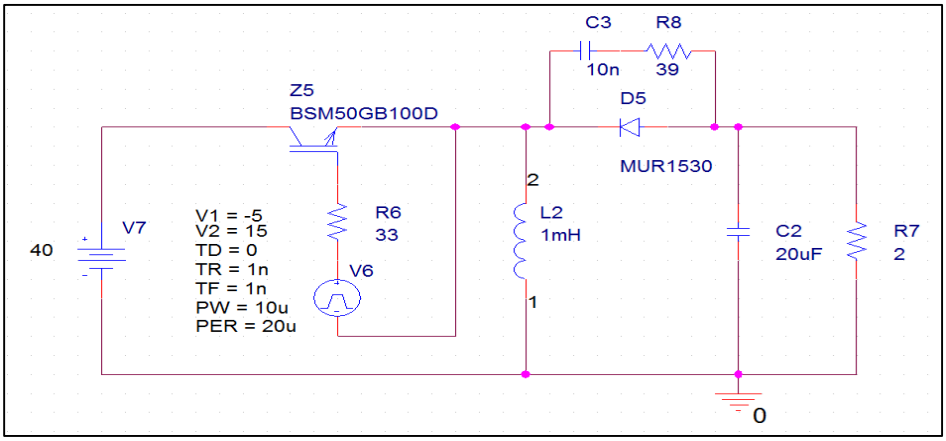


Figure 5

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

2. Create a simulation profile with 100ms run time.
3. Show the output voltage and its average, also, show the inductor current and IGBT current for a duty cycle of 20%, 50% and 70%. You need to magnify your plot for the IGBT and inductor currents to show few cycles. "note the effect of changing the duty cycle"
4. Show the ripple in the output voltage for a 70% duty cycle. Use the FFT analysis to show the frequency spectrum of the output voltage.
5. Change the output capacitance value to 1µF instead of 20µF and show the new ripple on the output voltage for a 70% duty cycle. Use the FFT analysis to show the frequency spectrum of the output voltage. "note the effect of changing the output capacitance value"

Full Bridge Converter

1. Draw the circuit shown in the figure below[4].

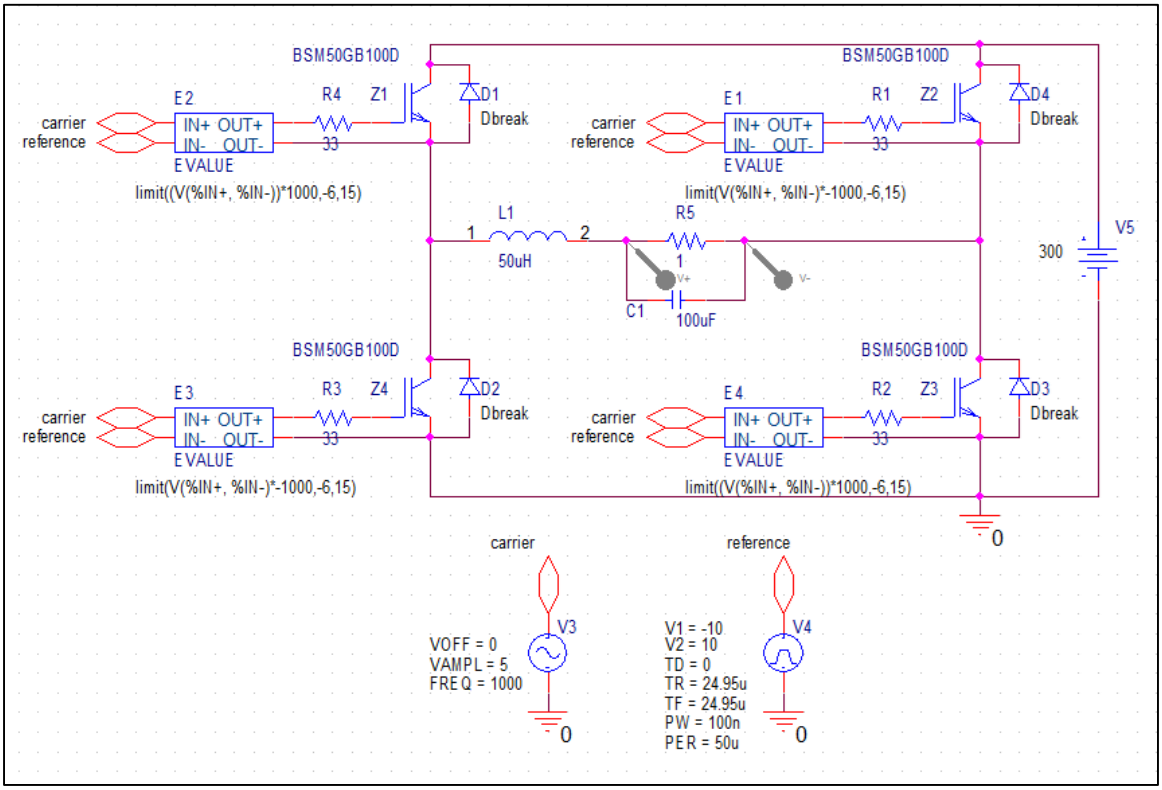


Figure 6

2. Create a simulation profile with 5ms run time.
3. Show the output voltage over the load resistor and show the output voltage before the filter on the same graph. Record the amplitude and frequency of the output voltage

[4] You can either use the PORTBOTH-L/CAPSYM port to connect the reference and carrier as shown in the figure, or, you can simply use a wire. To add the port, go to Place >> Hierarchical Port and choose PORTBOTH-L/CAPSYM, place as many as you need, then, you can simply name the port by double clicking the part and filling the name.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

4. Show the voltages across and currents through Z1 and Z4 on different plots. Magnify the graph to show few cycles only.
5. Show the gate signals for Z1 and Z4 on different plots. Magnify the graph to show few cycles only.
6. Change the frequency of the sinusoidal signal to 200Hz instead of 1 kHz, show the output voltage and record its frequency and amplitude. You should change the simulation run time to suit the new frequency. "note the effect of changing the carrier frequency"
7. Set the carrier frequency back to 1 kHz, and redo the simulation for a modulation index of 0.2 then 0.8. Show the output voltage and record its amplitude for each case. "note the effect of changing the modulation index"
8. Use the FFT analysis to show the frequency spectrum of the output voltage for a 1 kHz carrier signal and a 0.8 modulation index.
9. Replace the sinusoidal source with a 0V DC source and show the output voltage, record its frequency, amplitude and average. Use the FFT analysis to show the frequency spectrum of the output voltage. "note the effect of changing the switching strategy"
10. Repeat the previous part for a 4V DC source.

The EVALUE is a block that can be used to apply mathematical expressions on signals, for instance, it can be used to find the sum, the difference, the square root… of its input signals.

When added, the basic expression is V(%IN-, %IN+) , keeping it as it is, the output between OUT+ and OUT- will be the difference between the input signals, for example, if the voltage difference between IN+ and IN- is 5V , the output voltage between OUT+ and OUT- will be 5V and so on.

The expression can be edited by adding scalars, parameters and mathematical expressions. For example, the expression 5v*(V(%IN+,%IN-)) would result in an output voltage that is five times the difference between the input voltages.

In our experiment, we're using the EVALUE block to simulate a comparator using the LIMIT function which can be explained as follows

$$\text{LIMIT}(X, V1, V2) = \begin{cases} V1 & \text{for } X < V1 \\ V2 & \text{for } X > V2 \\ X & \text{otherwise} \end{cases}$$

A comparator output assumes one of two values, either high or low, so, to simulate a comparator, we multiplied the input by a large scale "1000" such that the input is –in most cases- higher or lower than the limits of the limit function resulting in one of two outputs V1 or V2 which are necessary to turn on or off our IGBT's.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment 3

PCB Design & Implementation

Introduction

A Printed Circuit Board (PCB) is used to serve two main purposes in the construction of electrical system cards; it is a place to mount the components and it provides the means of electrical connection between the components.

A PCB consists of two basic parts: a substrate (the board) and printed wires (the copper traces). The substrate provides a structure that physically holds the circuit components and printed wires in place and provides electrical insulation between conductive parts. A common type of substrate is FR4, which is a fiberglass– epoxy laminate. Substrates are also made from Teflon, ceramics, and special polymers.

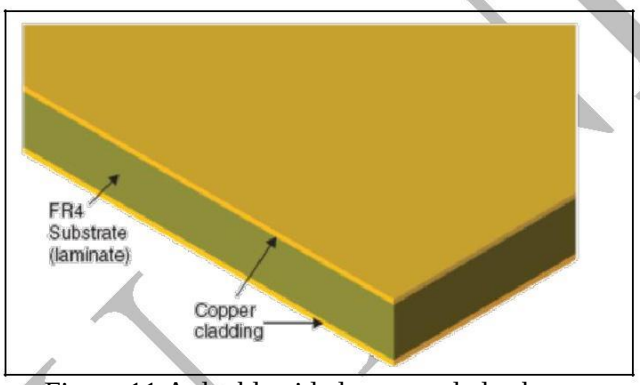


Figure 11 A double-sided copper clad substrate

Once the layout is designed, it can be implemented on the board by several technologies. The main processes in the board fabrication are:

- Removing the unwanted copper cladding such that the designed layout (tracks, pads, etc.) is implemented by the copper.
- Drilling the board to place the components.
- Solder the desired connections.

The copper cladding can be removed in two main ways:

- Mechanical milling: To mill the board, a computer numerical control (CNC) machine is programmed with the digital map of the board and grinds away the unwanted copper.
- Wet acid etching: it is more common when manufacturing large quantities of boards because many boards can be made simultaneously.

Selectively removing the copper with etching processes requires etching the unwanted copper while protecting the wanted copper from the etchant. This protection is provided by a

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

polymer coating (called photoresist) that is deposited onto the surface of the copper cladding as shown in Figure2. The photoresist is patterned into the shape of the desired printed circuit through a process called photolithography. The patterned resist protects selected areas of the copper from the etchant and exposes the copper to be etched.

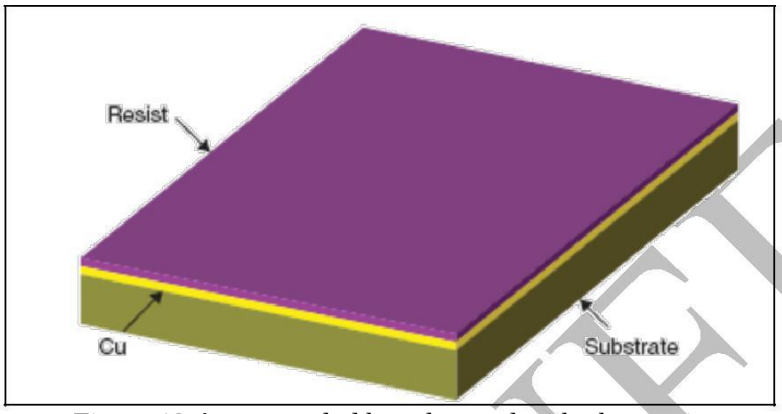


Figure 12 A copper clad board coated with photoresist

There are two steps to photolithography, patterning the photoresist - exposing the resist to light (typically ultraviolet (UV) light) and developing it (selective removal in a chemical bath). A mask is used to expose the desired part of the photoresist. A mask is a specialized black and white photographic film or glass photoplate on which a picture of the traces and pads is printed with a laser printer. The mask is placed on top of the photoresist as shown in Figure 3(b), and the assembly is exposed to the UV light. The dark areas block UV light and the white (transparent) areas allow the UV light to hit the photoresist, which imprints the circuit image into the photoresist. A separate mask is used for each layer of a circuit board. After the photoresist has been exposed with the mask and UV, it is washed in a chemical called the developer. The resist breaks down during exposure and is removed by the developer. Common developer is sodium hydroxide (NaOH). Once the resist has been exposed and developed, a circuit image made of the photoresist is left on the copper as shown in Figure 4(a).

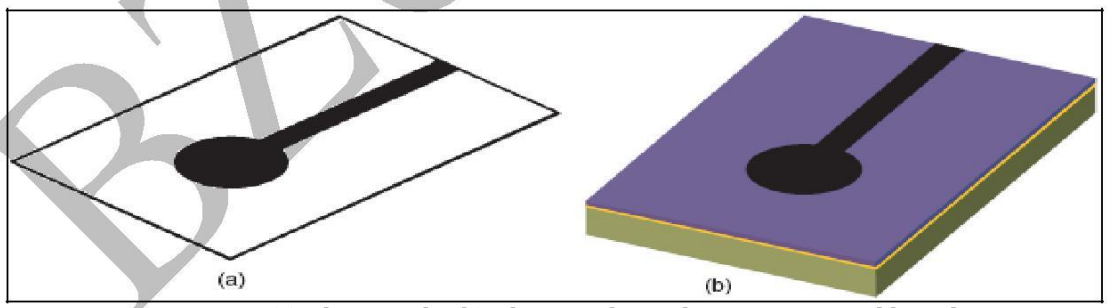


Figure 13 (a) Photomask. (b) Photomask on photoresist-coated board.

Next, the board is etched in an acid solution such as ferric chloride (FeCl3) or sodium persulfate (Na2S2O8). The etching solution does not significantly affect the photoresist but attacks the bare copper and removes it from the substrate, leaving behind the resist-coated copper as shown in Figure 4(b). Finally the photoresist is cleaned from the copper with a resist stripper, leaving behind the copper traces. Figure 4(c) shows the final patterned copper.

BZU ENEE
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

In this experiment, you will construct a Proteus Ares layout for a simple circuit, and to use the constructed layout to make a PCB in the lab using the acid etching method.

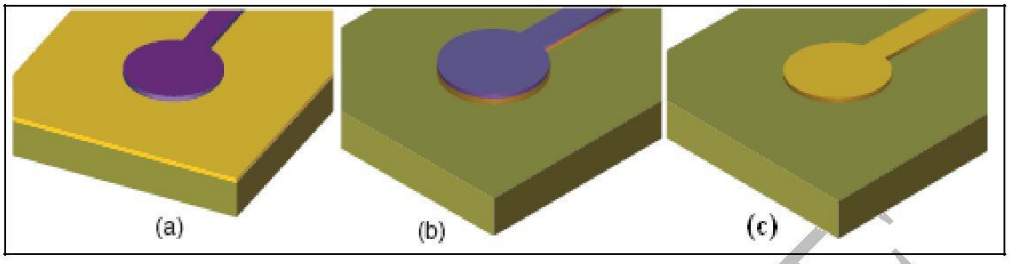


Figure 14 (a) Developed photoresist on copper. (b) Unwanted copper removed after etching. (c) Copper pad and trace after etching and resist stripping.

Objectives:

- To introduce the student to the PCB technology.
- To learn using Proteus software.
- To be familiar with the Gerber files.
- To implement the designed board using the acid etching method.

Procedure:

The main steps for making a PCB layout using Protues are as follows:

- Start the Isis software.
- Draw the circuit.
- Generate a netlist and move it to Ares.
- Place the parts.
- Route the board.
- Generate the files needed for the PCB manufacturing process.

Start Isis

Go to ***Start Menu >> Programs >> Proteus 7 Professional >> Isis 7 Professional***
An overview of the software is shown in figure 5.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

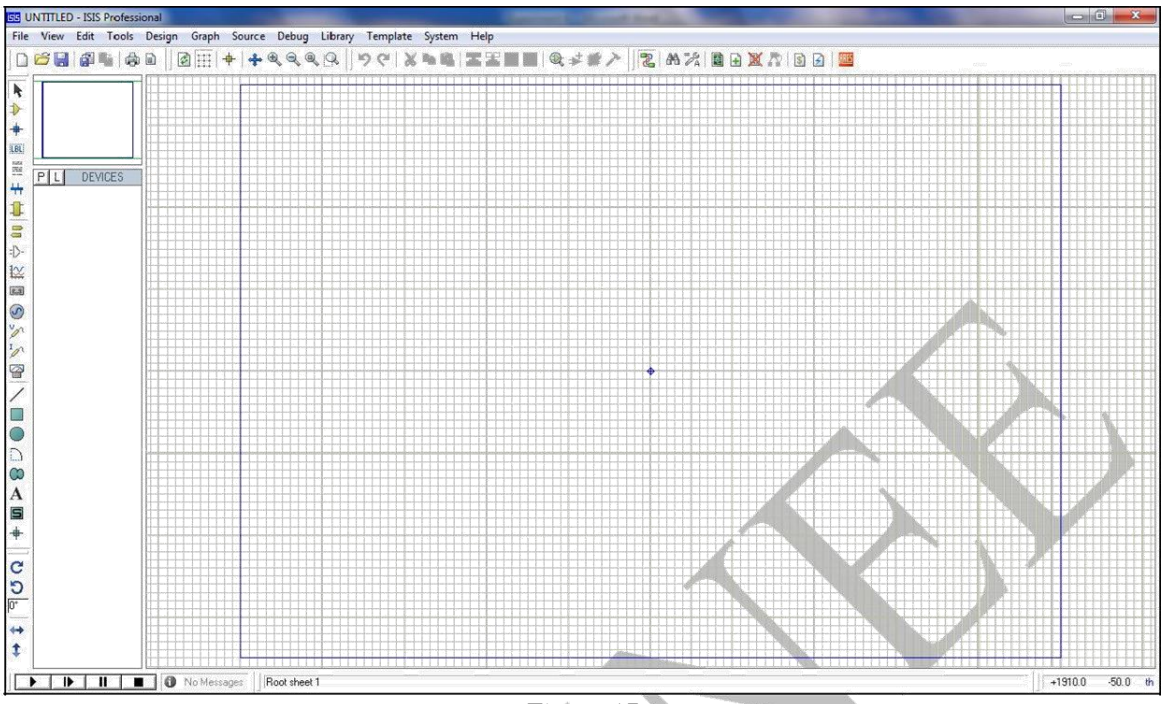


Figure 15

Draw the circuit

Draw the circuit shown in figure 6.

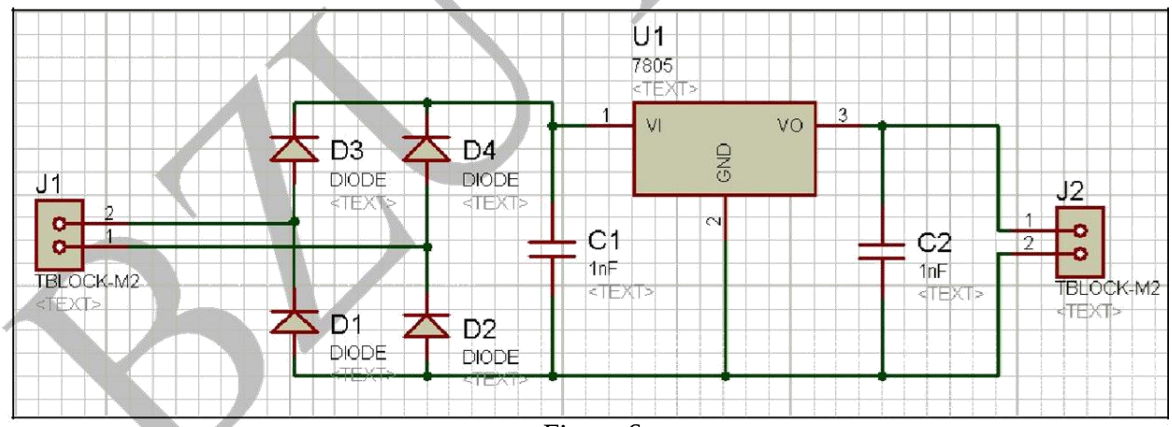


Figure 6

To add any of the components shown, Press the ***Component Mode*** as shown in figure 7, then press the ***P*** for ***Pick from Libraries***, and add the following set of parts:

- 2 Terminal Blocks
- 4 Diodes
- 2 Capacitors
- 7805 Regulator

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

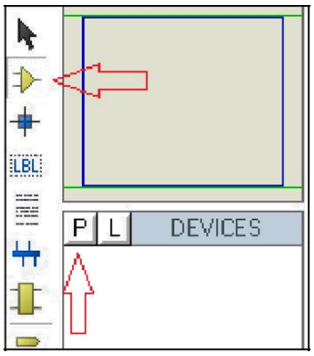


Figure 7

Make sure that any component you add has a PCB footprint as shown in the PCB View section below in figure 8.

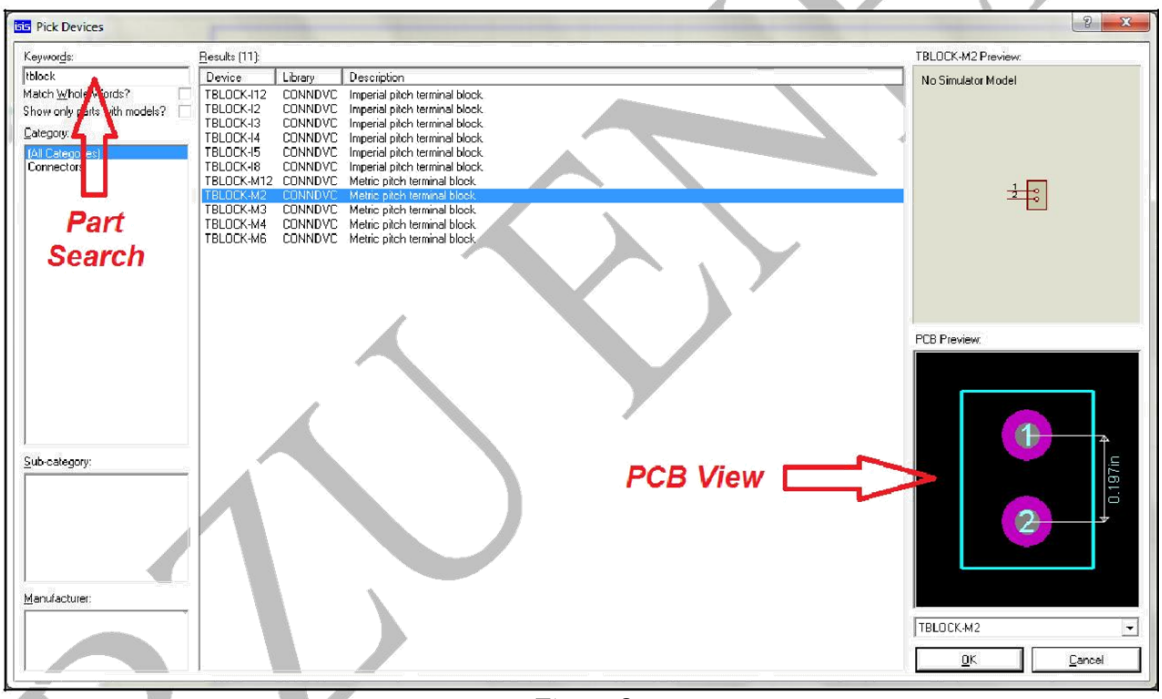


Figure 8

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

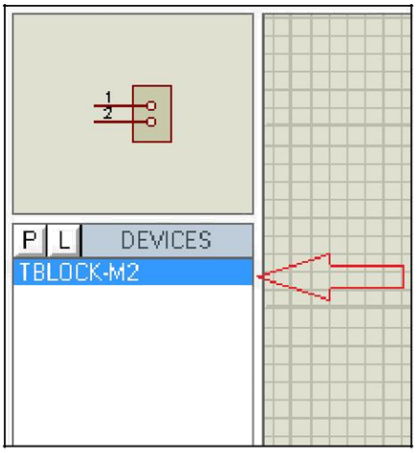


Figure 9

Generate a netlist and move it to Ares

Save your schematic in a folder of your name on the desktop then go to ***Tools >> Netlist to Ares.***

The Ares software will start as shown below.

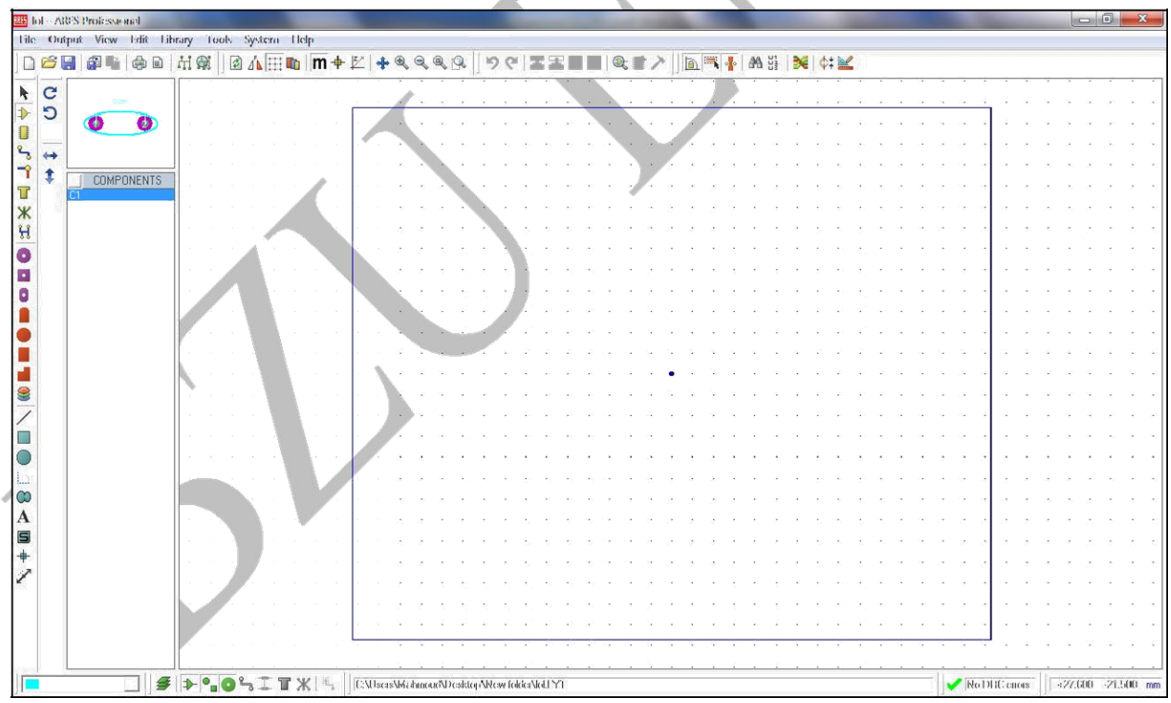


Figure 10 Ares Software overview

ZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Place the Parts

To place the parts press the ***component mode*** as shown in figure 11 and place the parts as shown in figure 12.

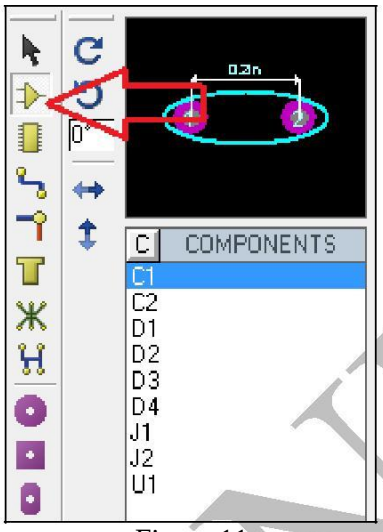


Figure 11

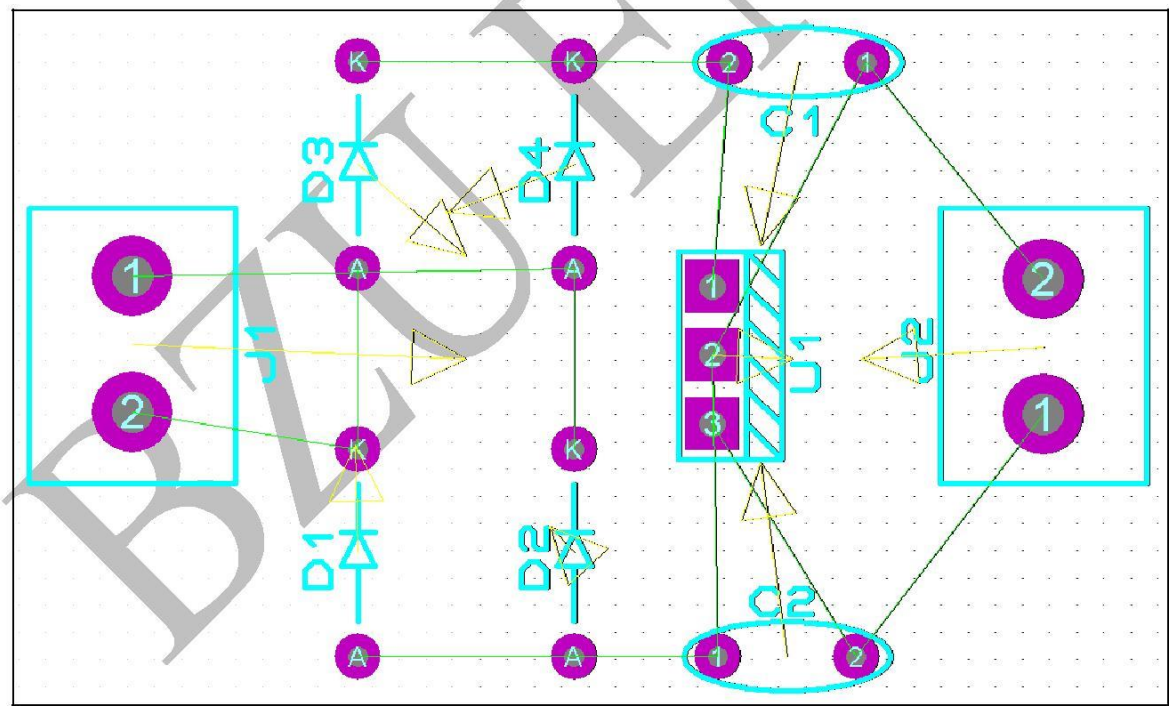


Figure 12

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Route the layout

Before routing the PCB, the routing layers must be determined, in our example, we'll be making a single layer PCB. To do so go to ***Tools >> Design Rule Manager >> Net Classes.***

Adjust the layers as shown in figure 13. Check the ***Net Class*** for any other layers and adjust them the same way done with the SIGNAL layer.

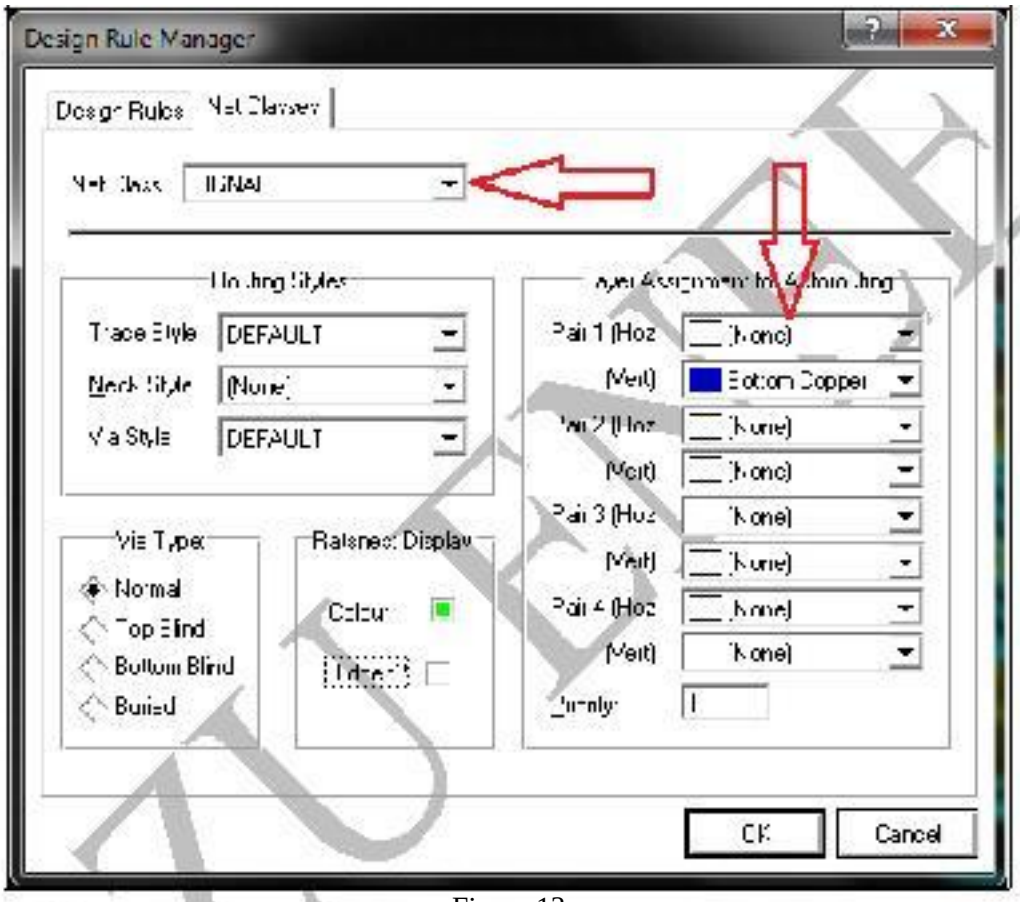


Figure 13

Go the ***Track Mode*** as shown in figure 14, then, press the ***E*** to edit the width of the tracks "the lines that connect the components" and increase the width to 75th[5].

To route the board go to ***Tools >> Auto Router >> Begin Routing.*** The routed board should look like the one in figure 16.

[5] 1th is 0.001 of an Inch.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

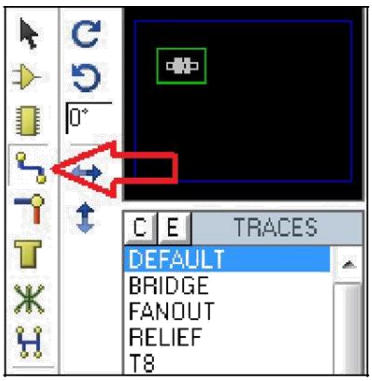


Figure 16

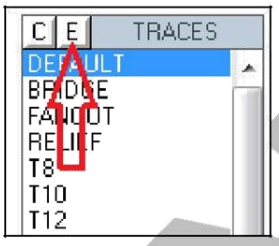


Figure 17

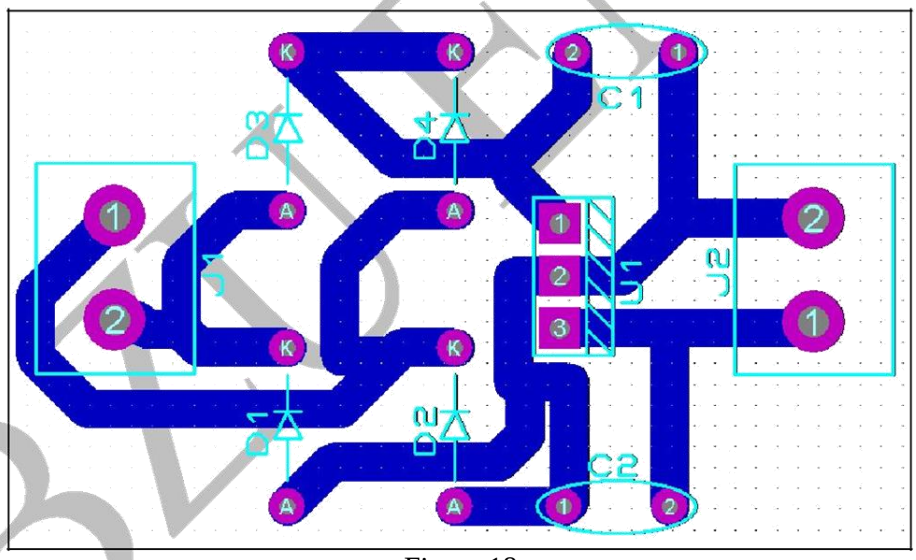


Figure 18

BZU ENEE

Editing Pins

To change the size of any of the pins, ***Right click on the desired pin >> Edit Pin*** then, change the ***Style*** to a greater size.

To change the size of all the pins with the same style, go to the ***pad mode*** as shown in figure 17 then select the specified style, then, press ***E*** to and edit the diameter of the pin as needed.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

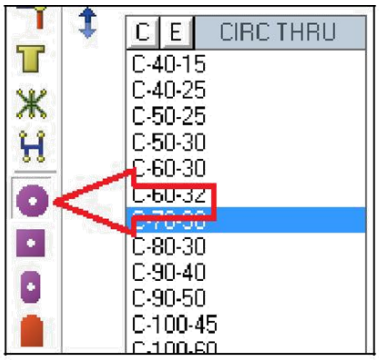


Figure 19

Gerber files

Gerber files are the main output of most PCB layout software since the Gerber format is a standard format that contains a map of the tracks "the copper lines that connect all the components", the pins "the drilling holes" and other specifications of the board.

Since Gerber files are a standard, you can simply use the Isis software to generate a Gerber file and hand it to the manufacturer to proceed with the PCB manufacturing process.

Before generating a Gerber file, the design needs to be checked, to do so, go to ***Output*** >> ***Pre-Production Check*** and check if there were any errors associated with the designed PCB layout. If no errors were found, go to ***Output*** >> ***Gerber/Excellon Output*** a screen as the one in figure 18 should appear, select the bottom copper only, then, press ***OK.***

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

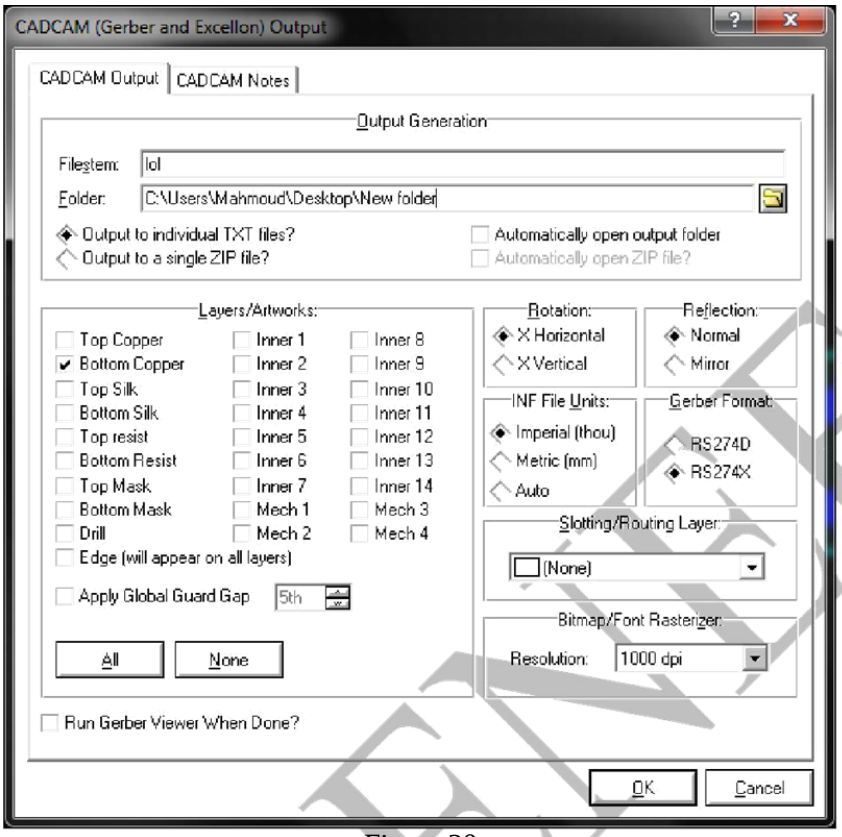


Figure 20

Other Output formats exist, like direct printing, PDF format and images. To output an image, go to ***Output*** >> ***Export Graphics*** >> ***Export Bitmap.*** The window in figure 19 should appear, select the bottom copper as shown in the figure below then press ***OK***.

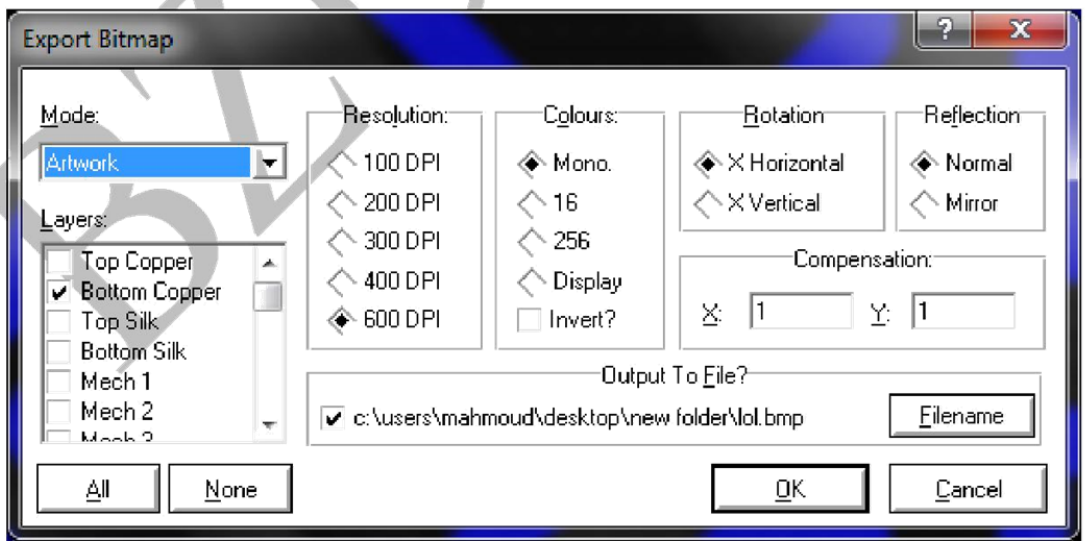


Figure 21

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment #4
Introduction to MicroC Program

Introduction

PIC Microcontrollers

Microchip PIC microcontrollers are a family of chips of different sizes, shapes and abilities, each PIC mainly consists of a processor, memory and I/O ports, where some PIC's may contain analog to digital conversion modules, serial communication modules, pulse width modulation modules and other features.

PIC microcontrollers are somehow easy to program and use compared to other controllers. The fact that they've a built in memory and I/O ports makes it easier for the user to use since there is no need to interface any memory or I/O ports externally like the case of the Intel 8086 chips. Like other controllers, the PIC has its own instruction set and can be programmed using an assembly language of its own or using higher level languages like the C language where special compilers are needed.

When choosing a PIC for your project you should take into account the number of ports needed, the memory size and what modules are needed (analog to digital converter, PWM module…). In our experiment we'll be using two different controllers for two different applications.

MicroC program[6]

MicroC Pro is a computer software that is used to develop applications that run on Microchip PIC microcontrollers, the software runs on windows operating systems. The MicroC Pro with its graphical interface can be considered as an editor that enables the user to write his code using C language, then, compiles it using the proper compiling tools and generates a hexadecimal file containing the machine language that can be downloaded and run on the microcontroller.

PIC Simulator IDE[7]

PIC Simulator IDE is another computer based software that simulates a variety of the Microchip PIC microcontrollers, using this software, one can test his code and verify its operation before downloading it on a real PIC. The simulator has many features and tools that can be added to the simulation like LCD modules, Oscilloscope channels and others.

Also you can use Protues "ISIS 7 Professional" to simulate your code and see the output in real circuits.

Objectives

-To become familiar with MicroC Pro and be able to develop PIC based programs.

---

[6] You'll need to install MicroC Pro for this experiment.

[7] You'll need to install PIC Simulator IDE v5.22 or higher to proceed with this experiment.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Procedure:

A. Interfacing a PIC microcontroller with an LCD

In this part we're creating a code that controls a 16X2 LCD using a 16F84A PIC.

Creating a project:

1. Start the MicroC Pro.
2. New project ➔ set name "your name" of the project.
3. Set its directory for example: "C:\Users\esmat\Desktop".
4. Set your device from the menu which is the 16F84A in our case.
5. Set the clock frequency to be 4MHz
6. Press next ➔ next ➔ next ➔ finish.
7. Type the code shown below:

```
// Lcd pinout settings
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D4 at RB0_bit;
// Pin direction
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB0_bit;
```

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

```
void main()

{

TRISA = 0x00; // set all pins of port A as output

TRISB = 0x00; // set all pins of port B as output

while(1)

{ lcd_init(); // initialize the lcd

lcd_out(1,1,"ENEE413 EXP#3"); // print this message

Lcd_Cmd(_LCD_CURSOR_OFF);

}
}
```

8.Save your work and Build it: go to **Build menu**→ choose **build**, so the **hex file** will be created in the same fold

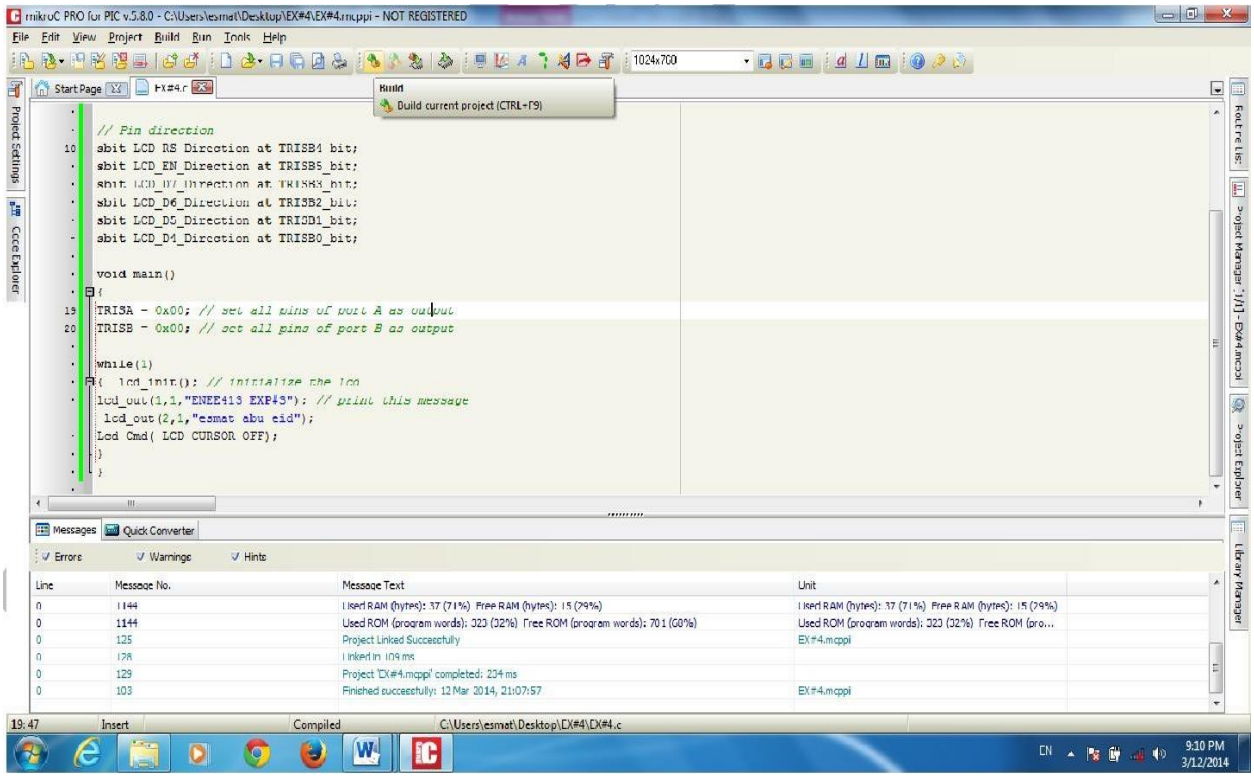


Figure (4.1)

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

9. Go to Protues ISIS 7 Professional program and draw the circuit shown in figure (4.2):

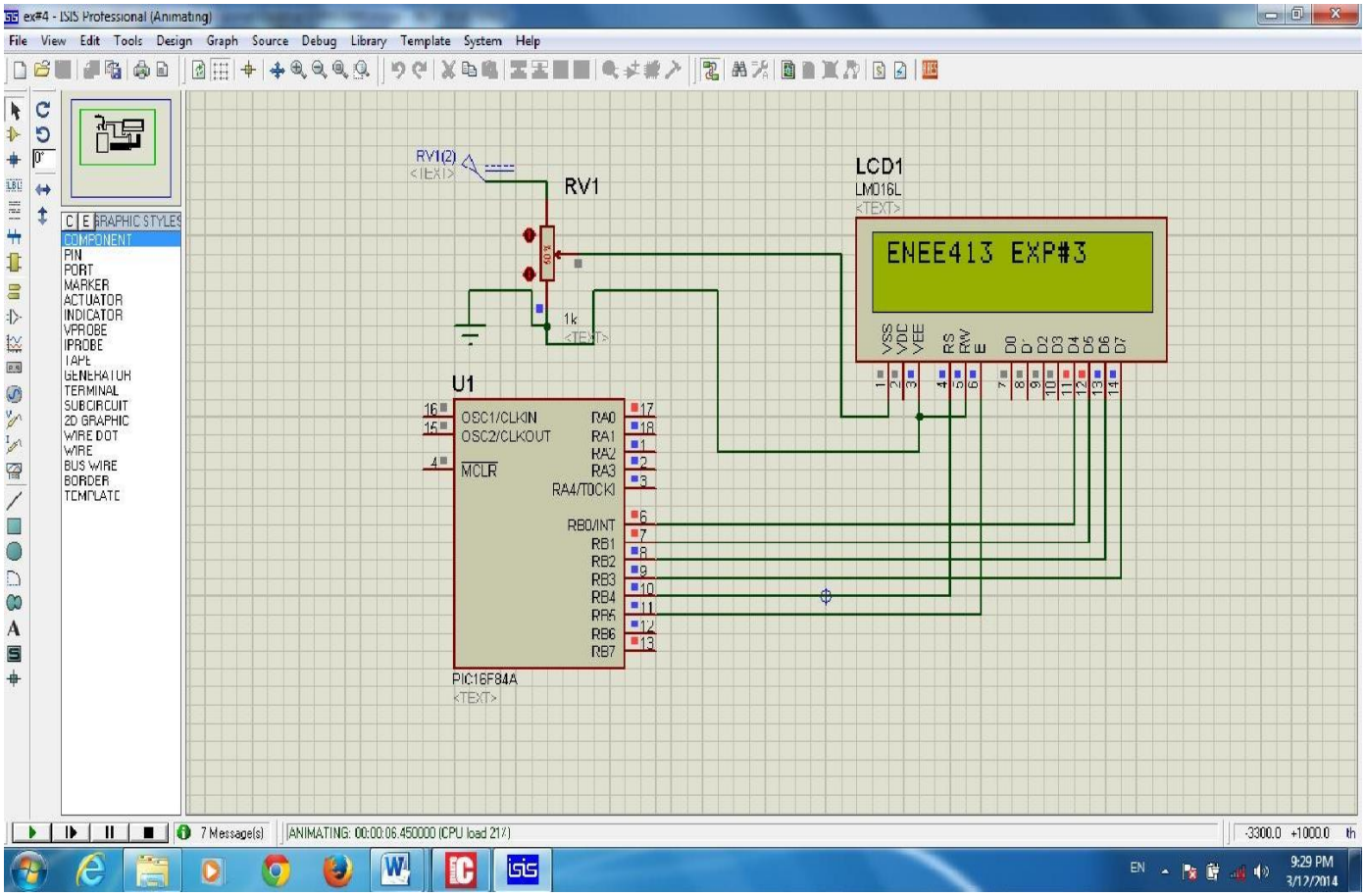


Figure (4.2)

10. Double click on the **PIC.**
11. In the **program file** field, click browse and choose your **Hex file**.

12. Change the **clock frequency** to 4MHz.
13. Go to play simulation as shown in Fig.(4.3) below and see the output:

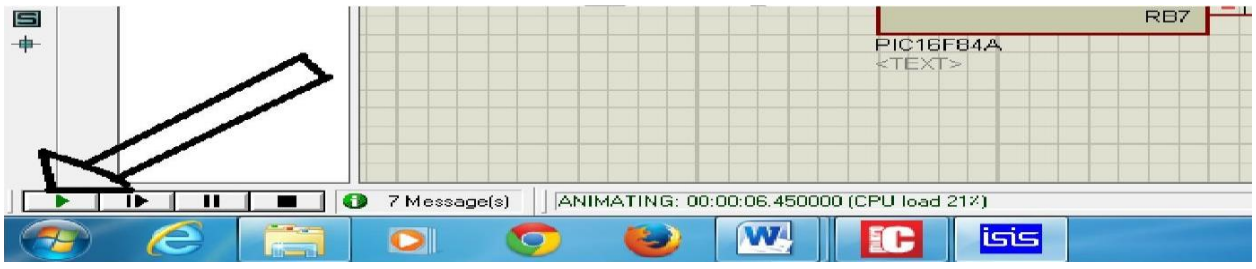


Figure (4.3)

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Class Exercise :

Change the previous code to display your first name and ID number simultaneously with each on separate lines i.e. your name on the upper line and the ID number on the lower line.

B. Using the A/D module

In this part we'll be using the 16F877A PIC microcontroller which has a built in analog to digital converter, the PIC will be reading the analog input, convert it into digital and outputs the digital value in binary representation on port B.

1. Create new MicroC project, and write the below code in it and build it:

```
unsigned int a;
void main() {
  TRISA = 0xFF;            // PORTA is input
  TRISC  = 0;              // PORTC is output
  TRISB  = 0;              // PORTB is output
  do {
    a = ADC_Read(0); // Get 10-bit results of AD conversion
    PORTB = a; // Send lower 8 bits to PORTB
    PORTC = a >> 8; // Send 2 most significant bits to RC1, RC0 }
  while(1);
}
```

2. Go to **protues ISIS 7 Professional** and draw the circuit shown in Fig.(4.4):

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

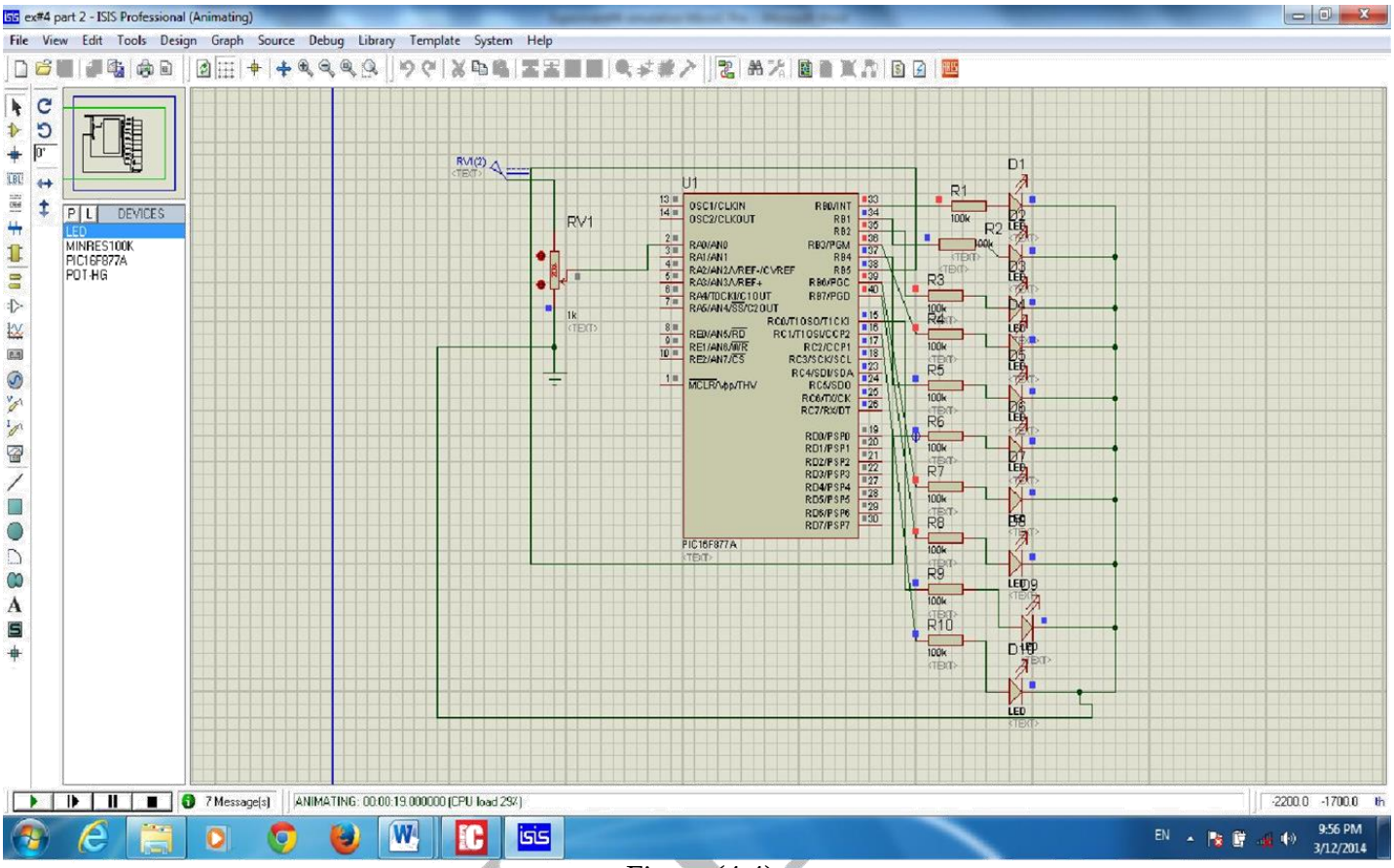


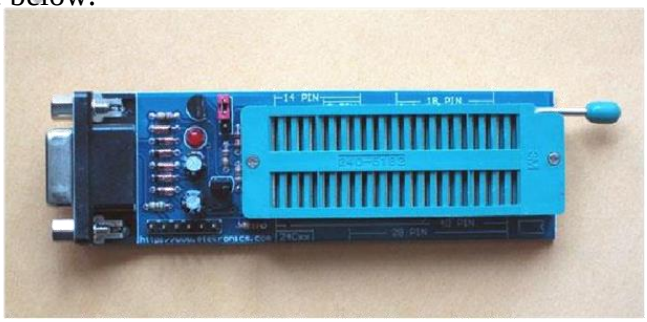
Figure (4.4)

3.Load your **Hex file** to the PIC and change its clock frequency. 4.Run the simulation and see the results.

Programming a PIC microcontroller using a serial port programmer

In the previous experiment you were taught how to develop a program for a PIC, but, how to download this program on a real PIC?

First of all, you'll need a programmer, mostly, a conventional JDM serial port programmer as the one shown in the figure below.



This type of programmers is common, easy to use, non-expensive and can be homemade! The only problem with this programmer is that it needs a PC or a Laptop with a serial port fitted on it,

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

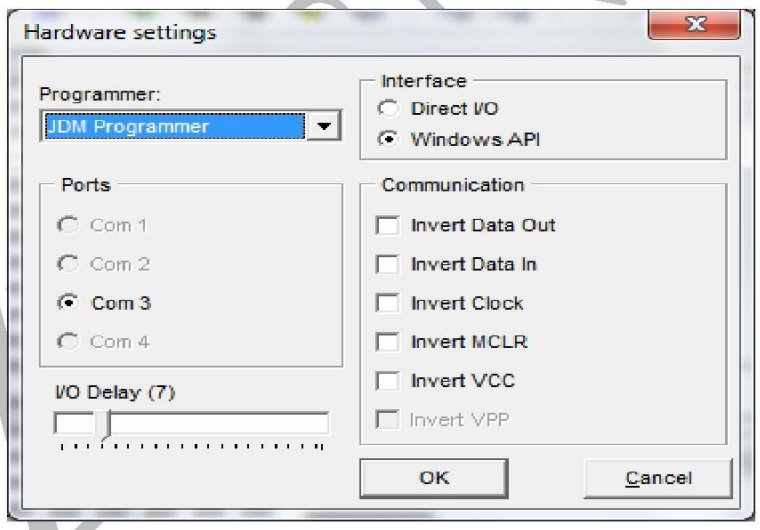
most PC's still have serial ports, but most laptops don't, so, if you're using this programmer, you'll probably need a PC to do the job.

Another type of PIC programmers are the USB programmers, they're also easy to use but the high price of such programmers makes them less common.

Once you have the programmer you'll need a computer software that suits your programmer, normally, USB programmers have their software on a CD that is packed with the programmer, as for the common serial programmers, they usually don't! Fortunately, a variety of programming software's can be downloaded for free. We recommend the use of "ICprog" which is a free programming software that can be easily downloaded from the web.

Programming Steps

1. Fit your PIC on the programmer; note the indicators on where the first pin must be. Most programmers have an arrow that indicates the right position for each PIC size.
2. Connect the programmer to the PC.
3. Start the programming software which is the ICprog in our case. When started for the first time, the following window will appear.



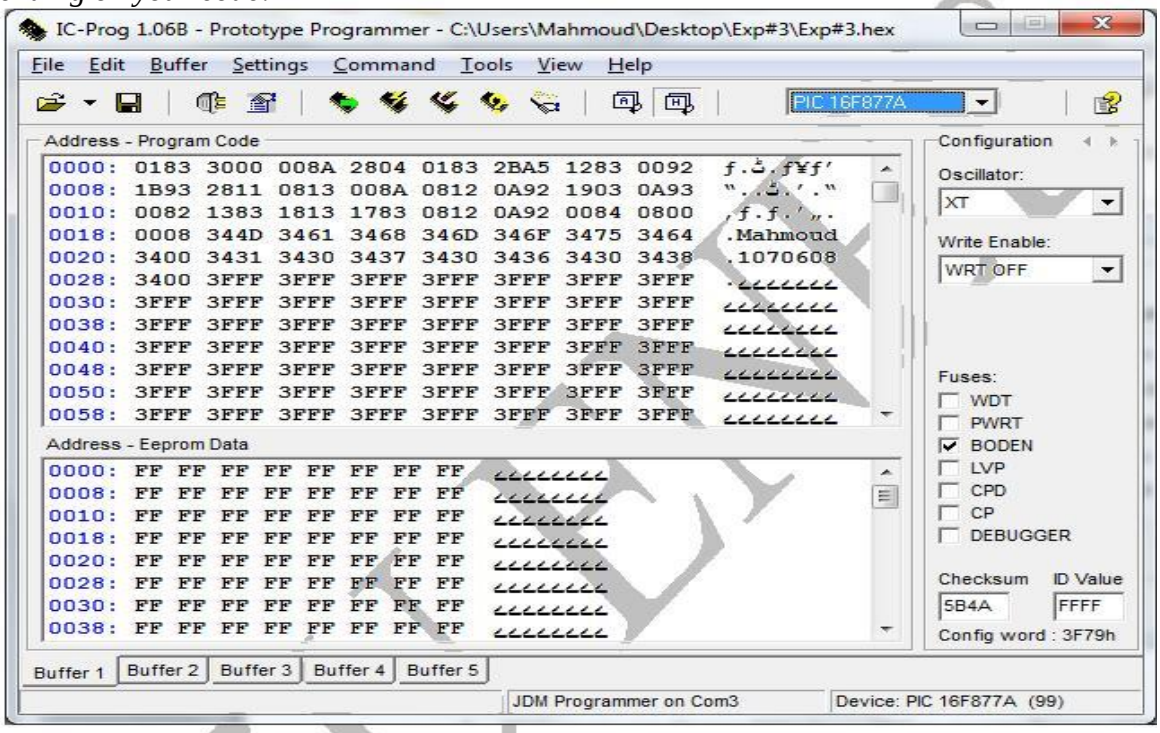
4. Make sure you choose "Windows API" as an interface and "JDM Programmer" as a programmer and choose Com 1 as a port –serial ports are normally Com 1- then press OK.
5. Now you must choose your PIC, go to Settings >> Device >> Microchip PIC and find your PIC.
6. Then, you need to load your program, go to File >> Open File and add your ".hex" file that you created using the MPLAB.
7. Finally, you'll need to set the settings of your microcontroller; the settings are shown in the configuration part to the right.
8. Choose the oscillator type to be RC if you're using an RC oscillator, choose LP if operating on low frequencies up to 200KHz, choose XT if you're using a 4MHz crystal oscillator and choose HS if you're using a crystal oscillator that is higher than 4MHz e.g. a 20MHz oscillator.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Normally, you'll be using a 4MHz crystal oscillator.

9. As for the fuses, only set the BODEN fuse, other fuses must be off.

These are the most common settings, different settings might be needed depending on your project. Do not tamper with Checksum or the ID value, these words are filled automatically and can differ depending on your code!



10. When you're finished with the configurations, go to Command >> Erase All, this should guarantee that the PIC memory is empty and ready for the new code. Once it's done, a pop window indicating the successful erasing process will appear.
11. Go to Command >> Blank check, this check will test whether your PIC is good to use or corrupt, once the check is done a pop up window saying "device is blank " will appear, this means your PIC is OK.
12. To download the program, go to Command >> Program All and wait till it finishes the programming.
13. To verify that the process went well, go to Command >> Verify and wait for the successful verification window to appear.

By this, your device should be ready to use, unplug the programmer and remove your device to use it in your project.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment 5
Linear Systems Simulation in MatLab

Objective:

The student should become acquainted with time and frequency domain analysis of linear systems using MatLab.

Prelab:

The purpose of the prelab is to introduce the student to some useful aspects of MatLab. In MatLab window, type the following commands for:

1. Matrix operations

```
>>% matrix operations
>>% to enter a matrix
>>A= [5 3; 5 5] ;
>>B= [-3 -7; 10 0] ;
>>A
>>B
>>% matrix addition
>>A+B % terminating with ';' will suppress the result
>>% matrix multiplication
>>A*B
>>% matrix
determinant >>det(A)
>>%inverse of a matrix
>>inv(A)
>>C=inv(A)
>>A*C
>>A/B
>> % eigenvalues of matrix
>>eig(A)
>>% transpose of a matrix
>>A'
>>% rank of a
matrix >>rank(A)
>>% create an identity
matrix >>I=eye(3)
```

2. Complex arithmetic

```
>>y=5j
>>z=10+4j
>>x=z/y
>> % obtain the magnitude and the phase of
x >>m=abs(x)
>>ph=angle(x)
>>% to convert to degree
```

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

>>deg=ph*180/pi

3. Vector operation

>>% create a 1x3 row
vector >>v1=[x y z]

>>% create a 3x1 column vector
>>v2=[x;y;z]

>>% multiply tow vectors
>>vectorprod=v1*v2
>>v2t=v2'

>>v3=v1.*v2t
>>v4=sqrt(v3)

4. Generating test point sets

>>t1=linspace(0,1,10)

>>t2=0:1:5
>>t3=logspace(-1,1,100)
>>val=cos(t3)
>>plot(t3,val);
>>semilogx(t3,val)

>> % plotting cos(pi*t/2+cos(pi*t/3);
>>t=-15:0.01:15;

>>y= cos(pi*t/2+cos(pi*t/3);
>>plot(t,y)
>>grid
>>xlabel('t(sec)');
>>ylabel('y(t)');

>>title('plot of y(t) = cos(pi*t/2) +
cos(pi*t/3)'); >>% plotting compound signal

>>t=-2:0.01:3;
>>y=t.*((t>=0) – (t>=2));
>>plot(t,y)

5. Working with polynomials

>>%coefficients of descending polynomial powers of s, such as 1s^3+0s^2+2s+5, are inserted into a defining coefficient array.

>>p1=[1 0 2 5]
>>r=roots(p1)
>>p2=[5 2 1]
>>p3=conv(p1,p2)
>>roots(p3)

>>%the laplace transform of a function will be a ratio of two polynomials in s
i.e. >>% F(s)=25s/10s^2+8s+4
>>n=[25 0];

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

```
>>d=[10 8 4];
>>roots(d)
```

6. Time and frequency response

```
>>num=[1 0 ];
>>den=[1 1 4];
>>impulse (num,den)
>>printsys(num,den,'s')
>>t=0:0.2:10;
>>impulse(num,den,t)
>>step(num,den,t)
>>y1=impulse(num,den,t);
y2=step(num,den,t); >>plot(t,y1,t,y2),grid
>>title('impulse resp=blue,step resp=green')
>>%step response of a third order system
>>%g(s)=12.5/(s+0.5)(s^2+s+25)
>>num=[12.5]
>>den=conv( [1 0.5],[1 1 25]);
>>step(num,den)
>>% plotting amplitude and phase response
>>% plotting using standard plotting and complex
number >>%capabilities for generating Bode plots
>>n=[20 -80];
>>d=[1 4 16];
>>w=logspace(-1,2,101)
>>Gain=freqs(n,d,w);
>>mag=abs(Gain)
>>db=20*log10(mag)
>>ph=angle(Gain)*180/pi
>>semilogx(w,db),grid
>>semilogx(w,ph),grid
>>% plotting using built-in Bode plot function.
>>num=[4]; >>den=[0.25
0.2 1]; >>w=logspace(-
1,2,200);
>>[mag,phase,w]=bode(num,den,w);
>>semilogx(w,20*log10(mag)),grid;
>>xlabel('freguency(rad/s)'),ylabel('Gain db')
>>% obtaining the transfer function in terms of wn and zeta using the ord2 command
>>t=[0:0.1:15];
>>wn=1;
>>zeta=0.2;
>>[num,den]=ord2(wn,zeta);
>>[y,x,t]=step(num,den,t);
>>zeta=0.4;
>>[num1,den1]=ord2(wn,zeta);
```


BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

```
>>[y1,x,t]=step(num1,den1,t);
>>zeta=0.6;
>>[num2,den2]=ord2(wn,zeta);
>>[y2,x,t]=step(num2,den2,t);
>>plot(t,y,t,y1,t,y2);
>>grid
```

7. Closed–loop control systems

```
>>numc=[0.5 0.1];
>>denc=[5 0];
>>nump=[15];
>>denp=[30 1];
>>[numo1,deno1]=series(numc,denc,nump,denp);
>>[numc1,denc1]=cloop(numo1,deno1);
>>printsys(numc1,denc1,'s');
>>step(numc1,denc1);
>>grid
```

8. Analyze the circuits:

Each student must analyze the circuits in figures (1, 2, 3, 4, 5) and each student must give us the solution at the start of the lab.

Procedures:

1.For the circuit shown in Figure (1), determine the Resistance matrix and the voltage vector.

2.Write a MatLab program for solving the mesh currents.

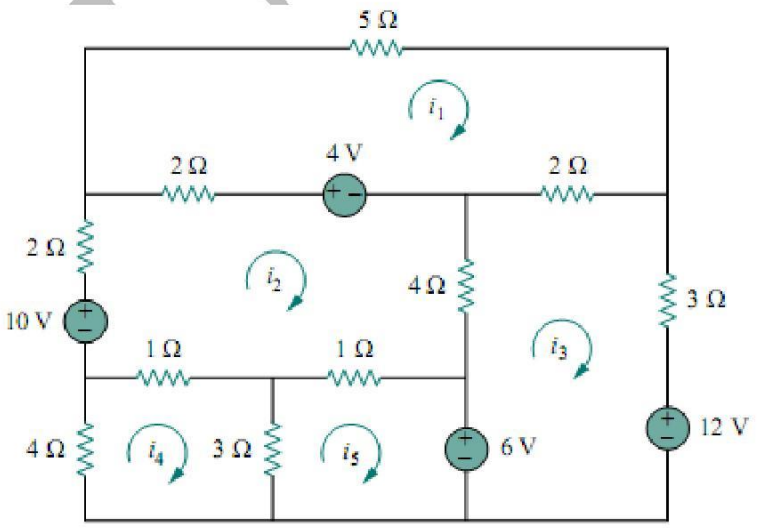


Figure (1)

3. For the transfer function:

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

$$F(s) = \frac{10^5(s+7)(s+13)}{s(s+25)(s+55)(s^2+7s+75)(s^2+7s+45)}$$

Write a MatLab program to plot the frequency response using:

a) Standard plotting and complex number capabilities,

b) Standard plotting and complex number capabilities for generating Bode plots.

4. Using the following commands (series, parallel, feedback). (K = 10);

a) Write a MatLab program to build up the control system shown in Figure (2).

b) Plot the step response of the system.

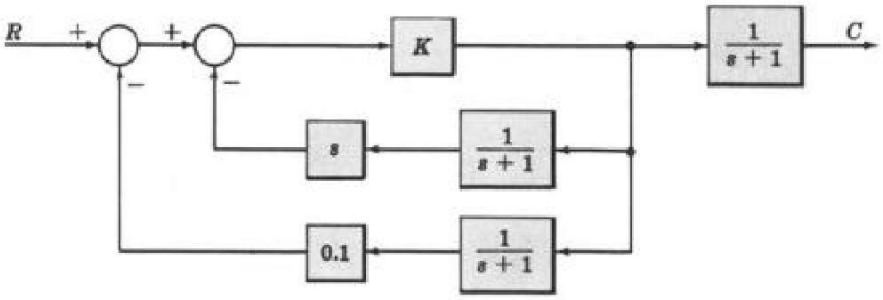


Figure (2)

5. Write a MatLab program to plot the function:

y(t)=t^2[u(t+1)-u(t)]+(t-1)[u(t)-u(t-1)]+(-1)[u(t-1)-u(t-2)]

for the period -3sec < t< 5sec

6. For the circuit shown in Figure (3)

a) Derive the transfer function.

b) Use MatLab to find the poles and the zeros of H(S), and plot the magnitude and the phase response.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

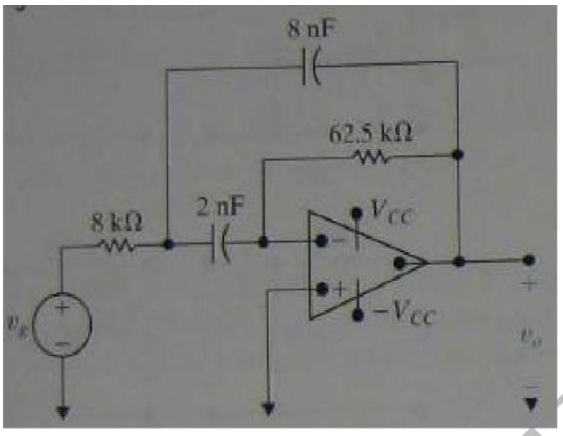


Figure (3)

7. For the circuit shown in Figure (4)

a) Derive the differential equation that describes the solution vc(t).

b) Using MatLab find the roots of the characteristic equation.

c) Find vc(t) for t>0.

d) Using MatLab plot vc(t) and il(t) for t>0.

e) Using MatLab verify the solution.

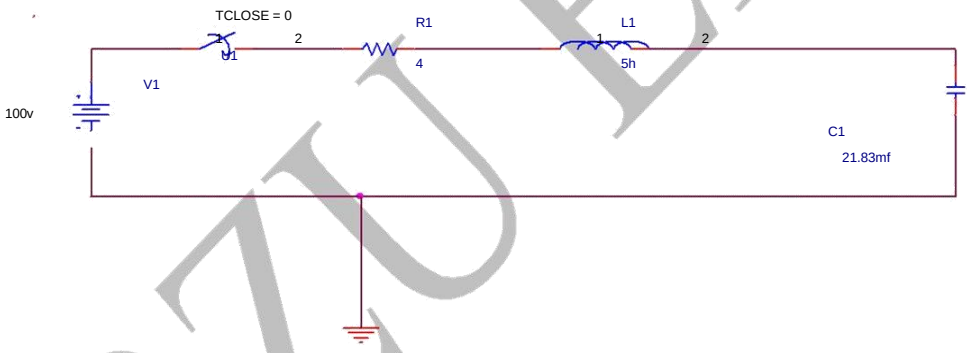


Figure (4)

8. For the circuit shown in Figure (5).

a) Derive the transfer function H(S) = VC(S)/VI(S)

b) Find the corner frequency ω_n for the H(S).

c) Compute the damping coefficient.

d) Using MatLab plot a Bode magnitude diagram.

e) From the Bode plot compute the amplitude in db at

$\omega=\omega_n/2$, ω_n, and ω_{max}

BZU ENEE
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

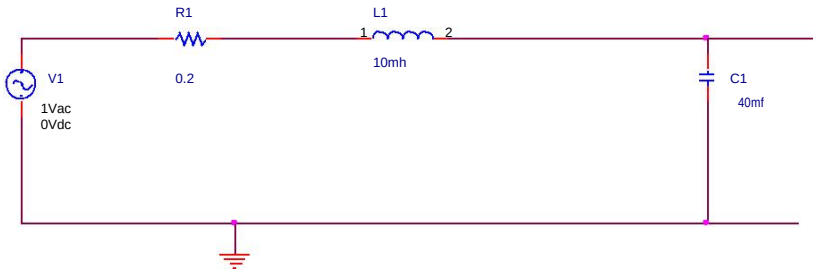


Figure (5)

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment # 6
Modeling a DC motor in Simulink

Introduction:

Simulink is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in a familiar mathematical notation.

Typical uses include math and computation algorithm development Data acquisition Modeling, simulation, and prototyping data analysis, exploration, and visualization data Scientific and engineering graphics Application development, including graphical user interface building.

Simulink is an interactive system whose basic data element is an array that does not require dimensioning. This allows solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of a time it would take to write a program in a scalar noninteractive language such as C or Fortran.

Simulink in university environment is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, Simulink is the tool of choice for high-productivity research, development, and analysis.

Objectives:

- To become familiar with Simulink program basic icons and options.
- To build a model of a DC motor in Simulink and test the effect of various parameters on motor performance.

Theory:

A DC motor can be modeled by the following circuit (Figure(1)), which includes the parameters representing the DC motor.

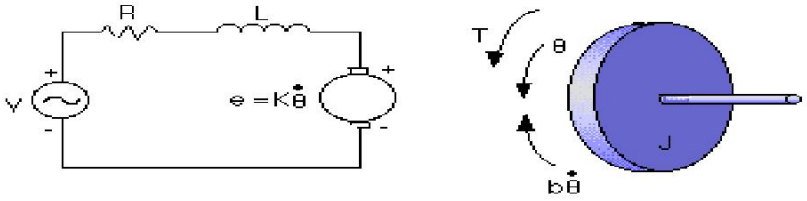


Figure (1): DC motor Equivalent Circuit

BZU EE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

These parameters are:

- moment of inertia of the rotor (J)
- damping ratio of the mechanical system (b)
- electromotive force constant (K=Ke=Kt)
- electric resistance (R)
- electric inductance (L)
- input (V): Source Voltage
- output (theta): position of shaft

The motor torque, **T**, is related to the armature current, **i**, by a constant factor **Kt**. The induced voltage, **e** is related to the rotational velocity by a constant Ke as follows:

The system will be modeled by summing the torques acting on the rotor inertia and integrating the acceleration to give the velocity. Also, Kirchhoff's laws will be applied to the armature circuit.

First, the integrals of the rotational acceleration and of the rate of change of armature current are modeled as:

Next, both Newton's law and Kirchhoff's law are modeled. These laws applied to the motor system give the following equations:

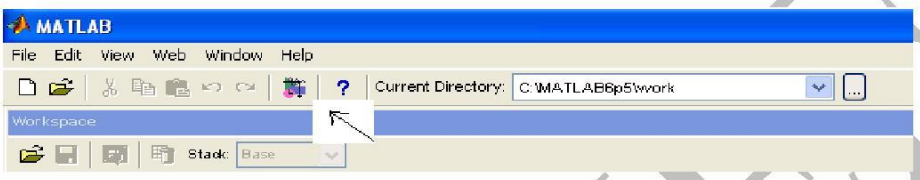
BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

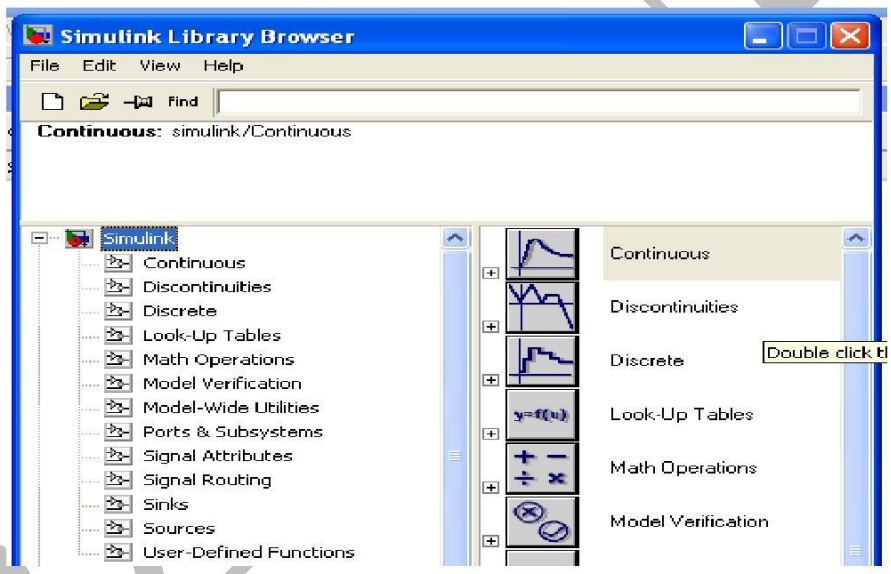
The angular acceleration is equal to 1/J multiplied by the sum of two terms (one pos., one neg.). Similarly, the derivative of current is equal to 1/L multiplied by the sum of three terms (one pos., two neg.).

Procedure:

- Open Simulink by clicking on this icon:



- The page below will be opened, then open a new model window using the icon

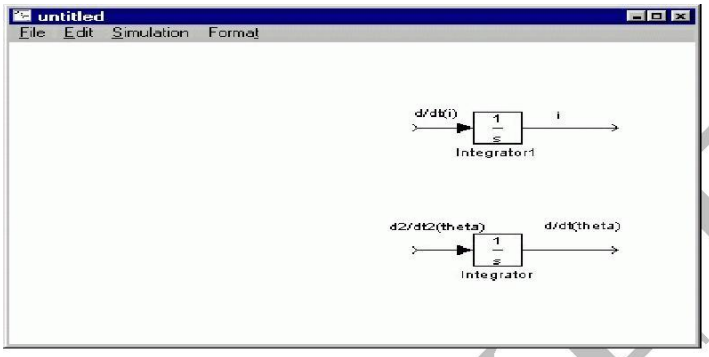


- From the Simulink library click on the continuous block

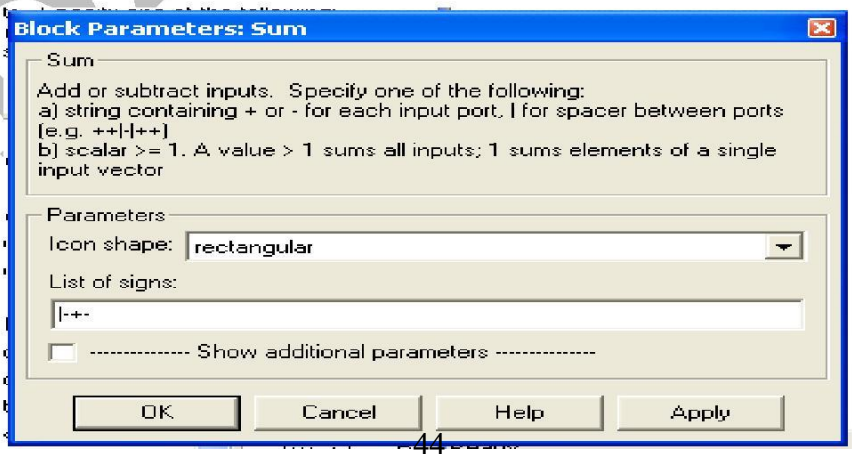
insert an Integrator block by dragging its icon to the new window and draw lines to and from its input and output terminals by holding down the mouse button and move the curser away. Notice that one or both sides of the blocks have angle brackets. The > symbol points to a block, it is an input port.
- Label the input line "d2/dt2(theta)" and the output line "d/dt(theta)" as shown below. To add such a label, double click in the empty space just above the line.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

- Insert another Integrator block above the previous one and draw lines to and from its input and output terminals.
- Label the input line "d/dt(i)" and the output line "i".



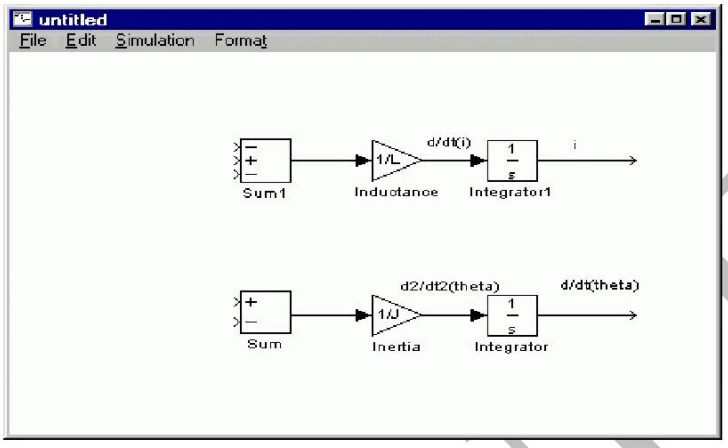
- Insert two Gain blocks, (from the math operations) one attached to each of the integrators.
- Edit the gain block corresponding to angular acceleration by double-clicking it and changing its value to "1/J".
- Change the label of this Gain block to "inertia" by clicking on the word "Gain" underneath the block.
- Similarly, edit the other Gain's value to "1/L" and it's label to Inductance.
- Insert two Sum blocks (from the math operations), one attached by a line to each of the Gain blocks.
- Edit the signs of the Sum block corresponding to rotation to "+-" since one term is positive and one is negative. This can be done with a double click on this block then change the list of signs.



BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

- Edit the signs of the other Sum block to "-+-" to represent the signs of the terms in Kirchhoff's equation.



Now, add in the torques, which are represented in Newton's equation. First, add in the damping torque.

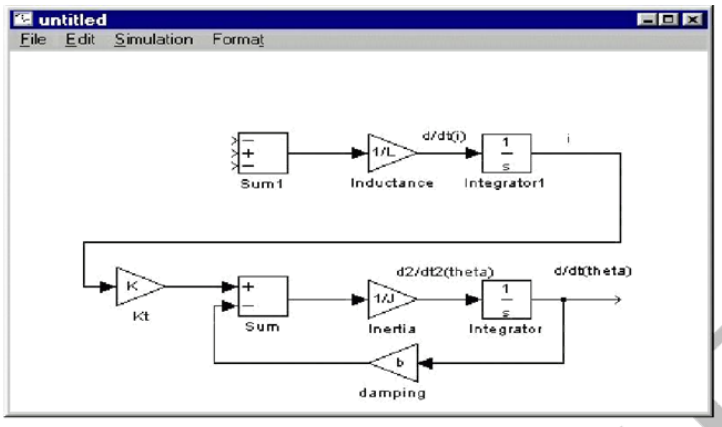
- Insert a gain block below the inertia block, select it by single-clicking on it, and select Flip from the Format menu to flip it left-to-right.
- Set the gain value to "b" and rename this block to "damping".
- Tap a line off the rotational integrator's output and connect it to the input of the damping gain block. Do this by single click on the integrator then hold Ctrl while clicking the gain block a single click too.
- Draw a line from the damping gain output to the negative input of the rotational Sum block.

Next, add in the torque from the armature.

- Insert a gain block attached to the positive input of the rotational Sum block with a line.
- Edit its value to "K" to represent the motor constant and Label it "Kt".
- Continue drawing the line leading from the current integrator and connect it to the Kt gain block.

BZU ENLE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

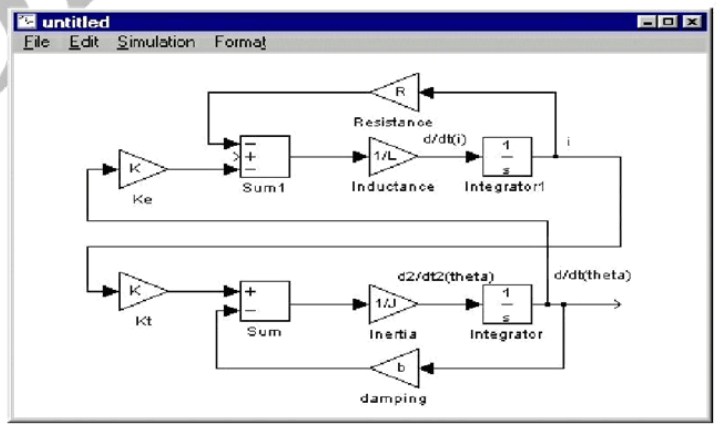


Now, add in the voltage terms which are represented in Kirchhoff's equation. First, add in the voltage drop across the coil resistance.

- Insert a gain block above the inductance block, and flip it left-to-right.
- Set the gain value to "R" and rename this block to "Resistance".
- Tap a line off the current integrator's output and connect it to the input of the resistance gain block.
- Draw a line from the resistance gain output to the upper negative input of the current equation Sum block.

Next, add in the induced voltage from the motor.

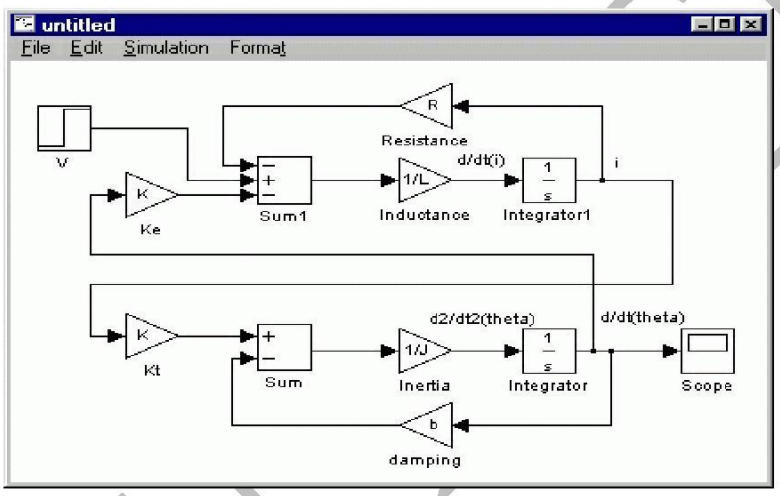
- Insert a gain block attached to the other negative input of the current Sum block with a line.
- Edit its value to "K" to represent the motor constant and Label it "Ke".
- Tap a line off the rotational integrator output and connect it to the Ke gain block.



BZU EE
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

The third voltage term in the Kirchhoff equation is the control input, V. Apply a step input.

- Insert a Step block (from the Sources block library) and connect it with a line to the positive input of the current Sum block.
- To view the output speed, insert a Scope (from the Sinks block library) connected to the output of the rotational integrator.
- To provide a appropriate unit step input at t=0, double-click the Step block and set the Step Time to "0".



Open-loop response:

To simulate this system, first, an appropriate simulation time must be set. Select Parameters from the Simulation menu and enter "4" in the Stop Time field. 4 seconds are long enough to view the open-loop response. The physical parameters must now be set. Run the following commands at the MatLab prompt:

$J=0.2*10^{-3}$ $kg.m^2$; b=0.001;

K=0.2;

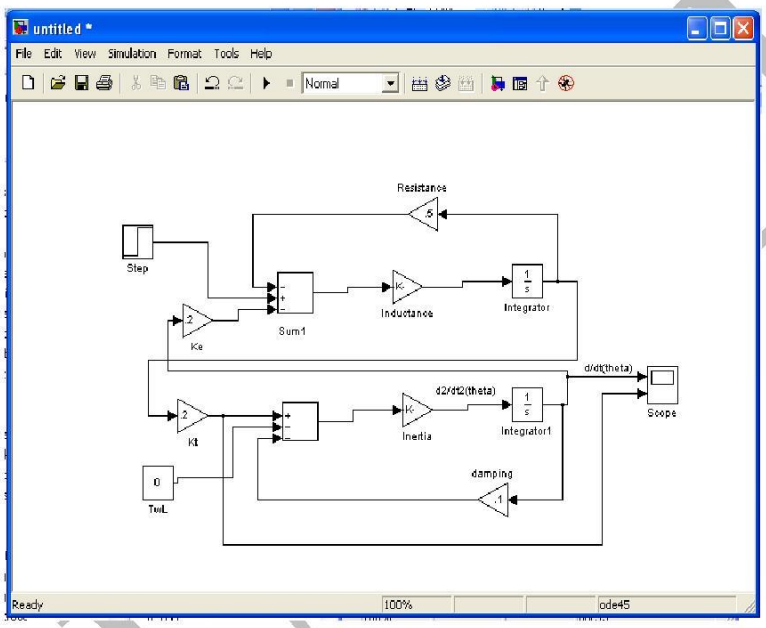
R=0.5Ω;

L=10mH;

V=150V;

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

- Run the simulation (Ctrl-t or Start on the Simulation menu). When the simulation is finished, double-click on the scope and hit its autoscale button. You should see the motor speed simulation with time.
- The last part to add to the simulation is the working load torque (TwL). Apply a constant input.
- Insert a constant block (from the sources block library) and connect it with a line to the negative input of the current Sum block as follows:



- Then save the model then run it and open the scope to see the output.
- After building the DC motor model study the effect of various parameters on Torque-speed curves.

- **Step (1):** Effect of load torque T :

Change the value of T to be: 0.2, 0.4, 0.6, 0.8, 1.0, and 1.2 Nm each time run the model and find the speed using the scope. From the values you have got, Plot the torque-load curve under changing T. Find the current for each value of T.

- **Step (2):** Effect armature resistance

For R = 2, 4, 6, and 8Ω repeat **Step (1)** for each value of resistance then tabulate your results. Then plot the output characteristics on same figure.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

TwL	R (Ω)	ω
0.2	0.5	
	2	
	4……8	
0.4	0.5	
	2	
	4……8	
0.6……	……..	…..

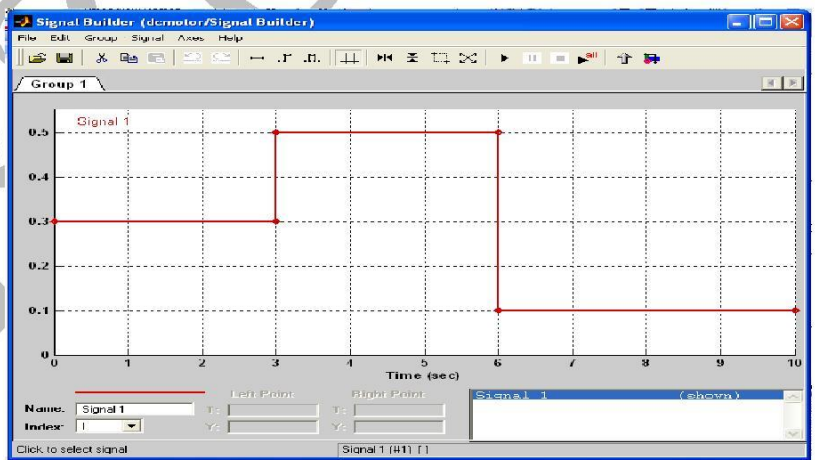
- **Step (3):** Effect of input voltage V:

Keeping R=0.5Ω, change the value of V to be: 50, 100, 150, 180, and 220V. Each time run the model for T= 0.2, 0.4, 0.8, 1.0, and 1.2. And find speed using the scope. Tabulate your results, plot the torque-load curve under changing V.

- **Step (4):**Effect of moment of inertia of the rotor (J) :

-In this part keep the damping ratio (b)=0.001, and change the load torque TwL in steps, to achieve that insert signal builder (From the sources library) and replace it with the constant which represented the TwL before.

-Double click on the signal builder; the window below will appear use curser to change the values of the curves till you get the one below:

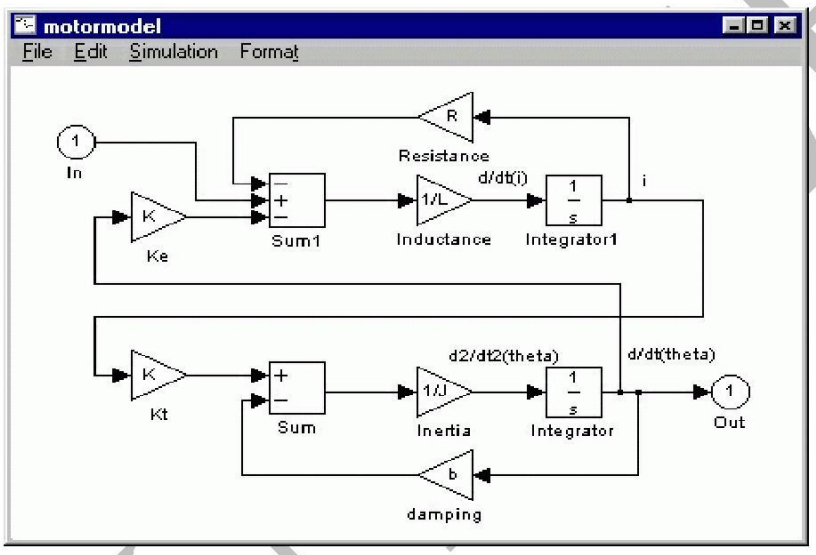


-After you finish Change the value of (J) to be: $0.05*10^{-3}$, $1.0*10^{-3}$, $5.0*10^{-3}$, $10*10^{-3}$ $kg.m^2$, each time run the model and find the speed using the scope.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Extracting a Model in MatLab:

A linear model of the system (in state space or transfer function form) can be extracted from a Simulink model into MatLab. This is done through the use of In and Out connection blocks and the MatLab function linmod. First, replace the Step Block and Scope Block with an In Connection Block and an Out Connection Block, respectively (these blocks can be found in the Connections block library). This defines the input and output of the system for the extraction process.



Save your file as "motormod.mdl" (select Save As from the File menu). MatLab will extract the linear model from the saved model file, not from the open model window. At the MatLab prompt, enter the following commands:

```
[A,B,C,D]=linmod('motormodel')
[num,den]=ss2tf(A,B,C,D)
```

You should see the output, providing both state-space and transfer function models of the system. Print the output, and comment!

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment # 7
Amplitude Modulation and Demodulation in LabVIEW

Introduction:

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help you troubleshoot code you write. In LabVIEW, you build a user interface, or front panel, with controls and indicators. Controls are knobs, push buttons, dials, and other input mechanisms. Indicators are graphs, LEDs, and other output displays. After you build the user interface, you add code using VIs and structures to control the front panel objects. The block diagram contains this code.

Theory:

Modulation is a process by which some parameter of a carrier signal is varied in accordance with a message signal. The message signal is called a modulating signal. When the amplitude of the carrier is varied in accordance with the message signal we have amplitude modulation. Thus, an AM signal can be:

$$s(t) = A\left[1 + k_a\, m(t)\right]\cos(2\pi f_c t)$$

Where A is a constant, k_a is modulation index. The carrier signal is generally a high frequency sinusoidal signal used to "carry" the information on the *envelope* of the message. The result is a double-side band signal, centered on the carrier frequency, with twice the bandwidth of the original signal.

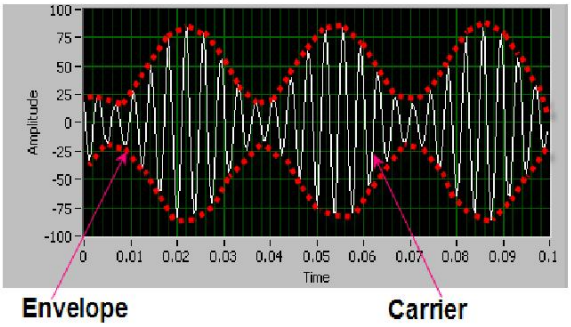


Figure (1-a): Time domain of an AM signal.

BZU ENEE
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

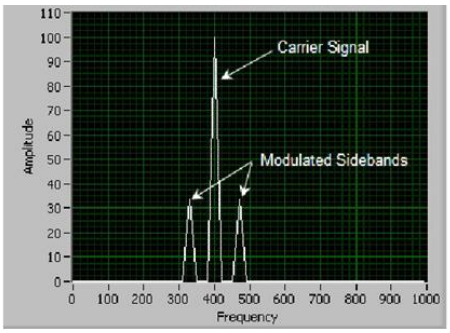


Figure (1-b): Frequency domain of an AM signal

Procedure

AM Modulation:

The following steps describe how to assemble a VI that implements amplitude modulation equations shown above. When this VI is completed, you will be able to set the amplitude and frequency of both the carrier and data signals as well as see the time and frequency domain representation of the signals. Inspect Figure (2-a) and Figure (2-b) which represent the front panel and block diagram of amplitude modulator VI.

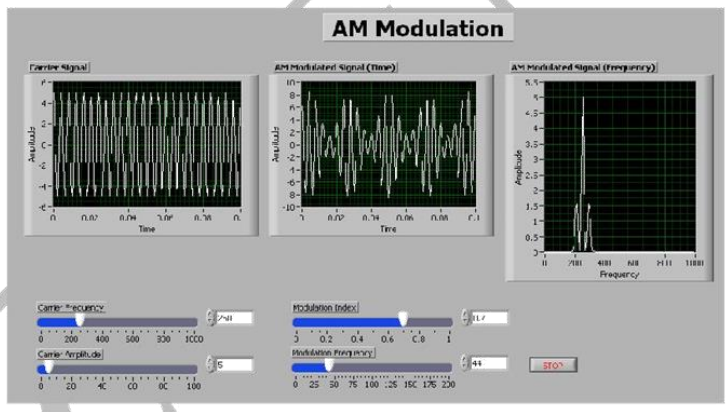


Figure (2-a): AM Modulation VI Front Panel

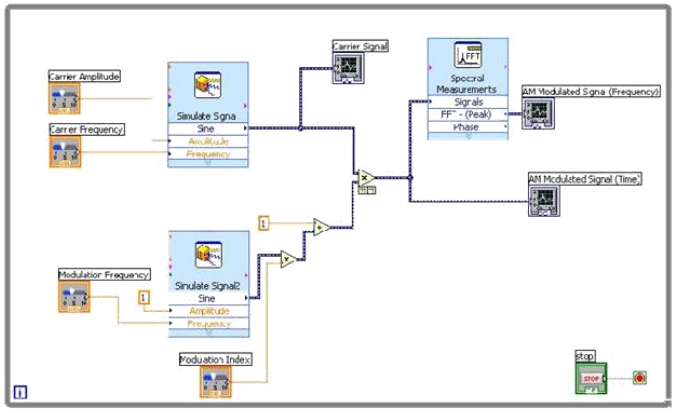
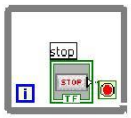


Figure (2-b): AM Modulation VI Block Diagram

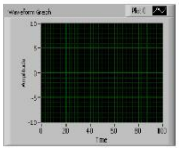
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

The block diagram consists of a while loop which contains various controls and graphs to display and control the AM signal component information.

In the following steps, we will build the AM modulation VI:



1. On the block diagram, choose from the Functions palette>Express>Exec Control>while loop. Expand the while loop block to proper size. Note that a stop button will appear on the front panel which stops the execution of the whole VI when pressed. (*To obtain the Functions palette, right click on any blank space in the block diagram window*). All controls and indicators that appears on the block diagram window should be placed in the while loop.



2. On the front panel, let us build the VI block diagram shown in Figure (2-a) above. First, place three "Waveform Graph" VI and rename them as required (Control palette>Modern>Graph>Waveform Graph).
3. Then, place four "Horizontal Pointer Slide" VI rename and resize them as required (Control palette>Modern>Numeric> Horizontals Pointer Slide). Right click on each

Pointer Slide, choose properties. Then a window will appear, in the Appearance tab write label required and check the option "show the digital display(s)). And in scale tab set the min. and max. fields as required for each *Pointer Slide*.

4. Up to this point, the front panel should looks like Figure (2-a).
5. On block diagram, drag and drop all the blocks into the will loop you will get an initial view as shown in Figure (3).

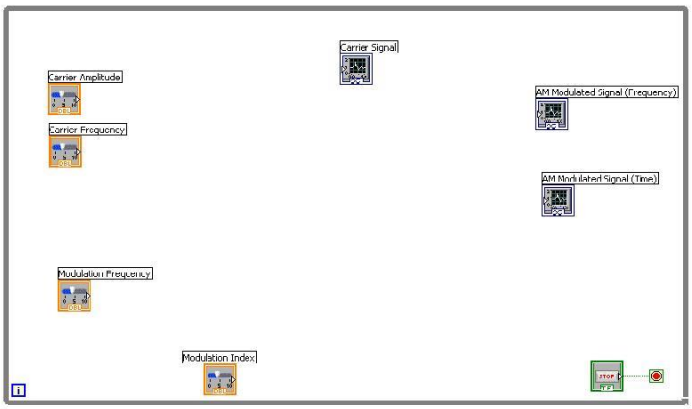


Figure (3)

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

6. Place "Simulate Sig" VI from Functions>Express>Input. A window box will open to configure the function. Select the *signal type* sin (or set phase to 90 for the cos). Set the samples per seconds to 2000 Hz. Once you have finished the window box should looks like the Figure (4) shown below.

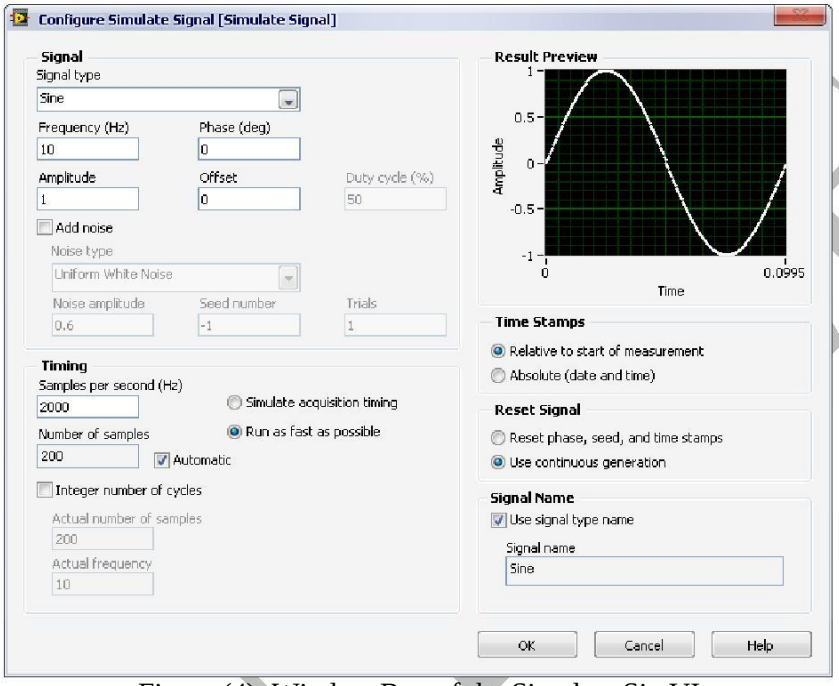


Figure (4): Window Box of the Simulate Sig VI

7. Select "OK". Copy and paste the Simulate Signal VI block to make another block. For the first simulate signal VI, wire the Carrier Amplitude into amplitude input and Carrier Frequency into frequency input. For the second Simulate Signal VI, wire the output of the modulation Frequency slider control into the frequency input. Wire a constant value of 1 into the amplitude input by right clicking on the connector and selecting "Create>Constant".
8. Place a "Multiply" VI on the block diagram. Wire the sine wave output of the second

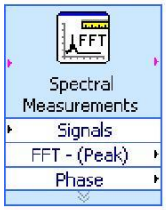
Simulate Signal VI into the multiply function. Wire the output of the Modulation Index slider control into the other input of the multiply function. Also, wire the output of the first Simulate Signal VI into the Carrier Signal graph as shown in Figure (2-b) above.

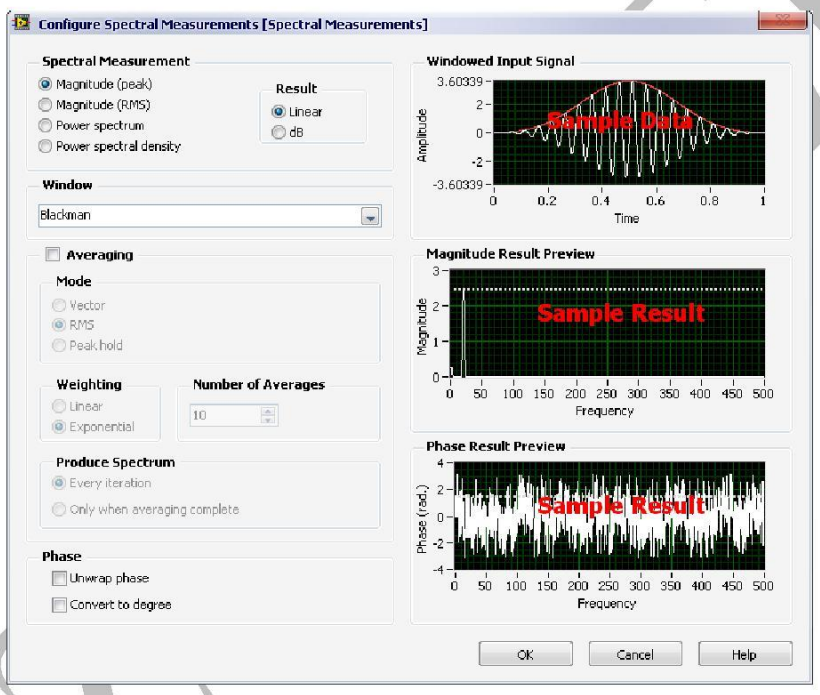
9. Place an "Add" VI on the block diagram and wire the output of the multiply function from the previous step into the function. Also wire a constant value of 1 into the Add function.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

10. Place a second “Multiply” VI on the block diagram. Wire the output of the Add function into the first input of multiply function. Wire the output of the first Simulate Signal VI (carrier Signal) into the second input of the Multiply function.

11. Place a “Spectral” VI from Functions>Express>Signal Analysis on the block diagram. A window box will open to configure the function. Select the spectral measurement to be magnitude (peak) and Linear. Set the Window to be “for example, Hanning” (do not enable averaging). Once you have finished the window box should looks like the figure shown below.





12. Select “OK”. Wire the output of the multiply function from the previous step into the *signals* input connector. Also wire the output of the multiply function to the AM Modulated Signal (Time) graph. Finally wire the output of the Spectral Measurements VI to the AM Modulated Signal (Frequency) graph.

13. Up to this point, your block diagram should be complete and looks like Figure (2-b).

    Press the run icon to execute your VI. Vary the values for the carrier and modulation amplitude and frequency to see the effect it has on the signal.

AM Demodulation:

You should use the basic technique of demodulation AM signal: you need a rectification device (diode), low pass filter and a DC removal. Fortunately, LabVIEW contains all the required

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

blocks to perform these operations. They are: Absolute Value VI, Filter VI (express palette) and AC-DC estimator VI. Implement the demodulation of AM signal on the same VI you previously built as shown in Figure (5).

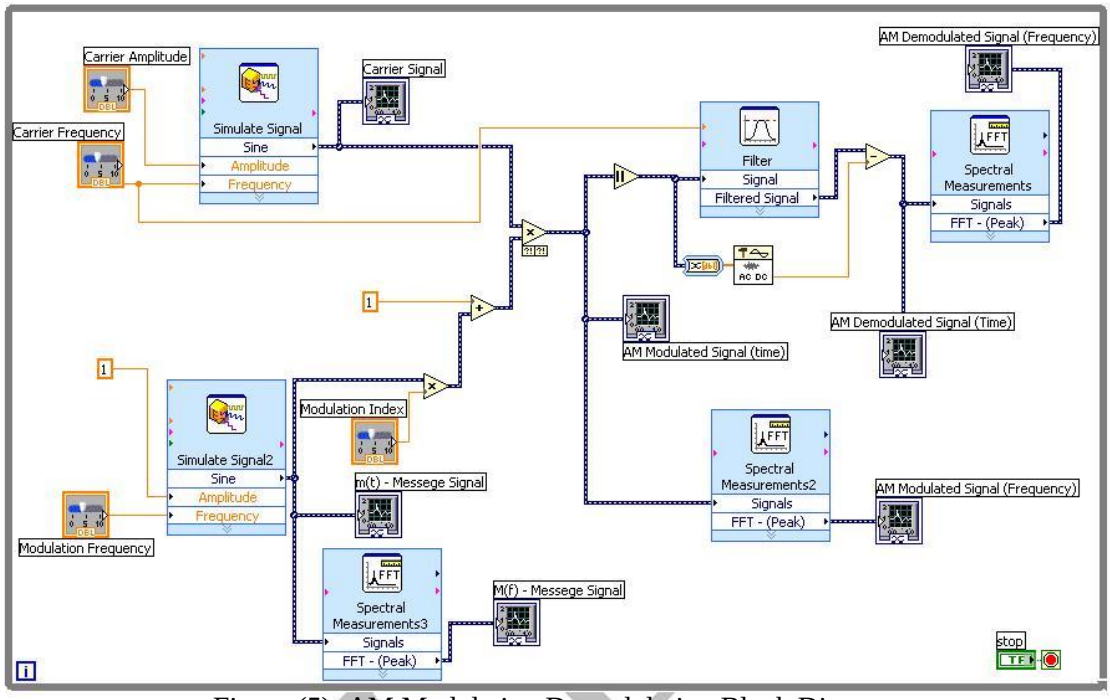


Figure (5): AM Modulation Demodulation Block Diagram

Exercise: Calculator

Build a Labview VI file that functions as a calculator. The front panel should contain: two controls for two numbers, an indicator to display the result of operation (add, subtract, multiply, divide) and a slide control to specify the operation to be performed. Note: you need a case structure to implement the four cases of operations.

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment # 8
Introduction to Data Acquisition Systems in LabVIEW

Prerequisite

C programming language, LabVIEW basics, building projects under visual C++ compiler, basics of Digital-to-Analog and Analog-to-Digital conversions.

Introduction

In this lab, you will learn about the basics of data acquisition. Initially you explore how the DAQ card in your computer is configured using MAX software. Then, you will write different data acquisition programs with help from the DAQ Assistant and features found in LabVIEW 8.0. These programs will also help to illustrate the difference between software and hardware timing. Finally, you will write data acquisition programs using ANSI C language.

Objectives

- Become familiar with the data acquisition hardware in your computer.
- Learn how LabVIEW acquires data from the DAQ hardware.
- Differentiate between finite acquisition; continuous acquisition; and on demand acquisition.
- Use ANSI C language to configure and run DAQ.

Equipments

- PC running Windows.
- LabVIEW 8.0 software & NI-DAQmx driver software.
- Visual C++ compiler (.NET 2003).
- NI USB-6009 DAQ card.
- Function generator.
- DC power supply.
- Digital Oscilloscope.
- Digital Multimeter.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Background

Measurement & Automation Explorer

Measurement & Automation Explorer, or MAX, is a software interface that gives you access to all National Instruments devices connected to your system.. MAX is used primarily to configure and test National Instruments hardware. The functionality of MAX is divided into four categories:

- Data Neighborhood
- Devices and Interfaces
- Scales
- Software

1) Data Neighborhood

Data Neighborhood contains all of the DAQmx tasks that are currently configured. DAQmx tasks are the objects that communicate with the DAQ hardware in LabVIEW. They can be configured in LabVIEW as well as MAX. Data Neighborhood also provides tools for testing and reconfiguring these tasks, as well as a utility for creating a new task.

DAQmx Tasks

LabVIEW 8.0 and NI-DAQmx center around the formation of tasks. A task can be created in either LabVIEW or MAX and contains the relevant information to a data acquisition such as channel, scale, and timing. By using a task the same type of data can be collected in many different programs without configuring the acquisition every time. In addition, a Task Control allows the user to use many different tasks in the same LabVIEW VI enabling them to quickly acquire different sets of data using the same program.

Creating a new DAQmx Task

When creating a new DAQmx task, the user is given the options of analog input, analog output, counter input, counter output, and digital I/O. After choosing the measurement category, the user is prompted to select the specific type of measurement from a list. The user then selects the measurement device and physical channel. After that the user gives the task a name to remember. Task names are consistent between MAX and LabVIEW and their descriptive names are easier to remember than arbitrary numbers.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

When the task is given a name the creation of that task is complete. The task configuration window then appears so that the user can specify the parameters of the task. Options such as scale, timing and input range are all displayed and can be edited. The task configuration window will be dealt with more extensively later in this experiment.

2) Devices and Interfaces

Devices and Interfaces display the currently installed and detected National Instruments hardware. Devices and Interfaces also include utilities for both user testing and device self tests. The three utilities that are specific to DAQ devices are Properties, Self Test, and Test Panels.

Self Test

The Self Test utility in MAX will inform the user if the computer is able to communicate with the device. This is the lowest level of testing and trouble shooting.

Test Panels

The Test Panel is a utility for testing the analog input, analog output, digital I/O, and counter functionality of the DAQ device. The Test Panel is useful for troubleshooting because it allows you to test the functionality of the device directly from NI-DAQ. If the device does not work in the Test Panel, it will not work in LabVIEW.

3) Scales

Scales shows you all the currently configured custom scales and provides utilities for testing and reconfiguring those custom scales. Scales also provides access to the DAQ Custom Scales Wizard, which allows you to create new custom scales.

4) Software

Software shows all the currently installed National Instruments software. The icon for each software package is also a shortcut that you can use to launch the software. The Software category also includes a Software Update Wizard. The purpose of the Software Update Wizard is to check if the National Instruments software is the latest version. If the software is not the latest version, the Software Update Agent opens the Web page on ni.com to download the latest version of the software.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Procedure

Testing DAQ

1. Plug your DAQ card into the USB port. The PC will automatically detect your hardware, and displays a window to ask you what to do. Just close this window.
2. Open MAX software, and use the browser to go to **My System»Devices and Interfaces»NI-DAQmx Devices»NI USB 6009:"Dev1"**
3. Right-click on **NI USB 6009:"Dev1"** and choose **Self-Test**. A message will be displayed to inform you if the computer is able to communicate with the device.
4. Again, right-click on **NI USB 6009:"Dev1"** and choose **Test Panels.** A window is opened for testing the analog input, analog output, digital I/O, and counters of the DAQ.
5. Adjust the power supply to give 2V at its output, and connect it to **AI0+** and **AI0-** pins on the DAQ (Analog Input Channel 0: Differential mode).
6. On the **Test Panels**, choose **Analog Input** tap, and set the following configuration:

Channel Name	**:**	**Dev1/ai0**	**Max Input Limit:**	**5**
Acquisition Mode	**:**	**On Demand**	**Min Input Limit :**	**0**
Input Configuration:		**Differential**		

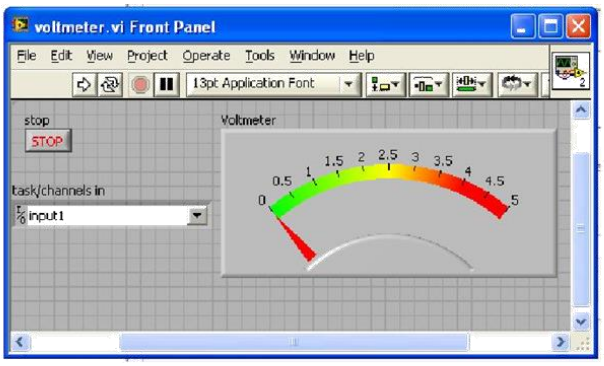
7. Click Start, and see if the displayed value equals the input voltage.
8. Connect the Digital Multimeter to read the output of Analog Output channel 1(AO1)
9. On the **Test Panels**, choose **Analog Output** tap. Select **ao1** channel, and set the DC voltage to give 3 volts, the click **Update Channel**. Is the displayed voltage on the DVM the same as you set in the **Test Panels**.

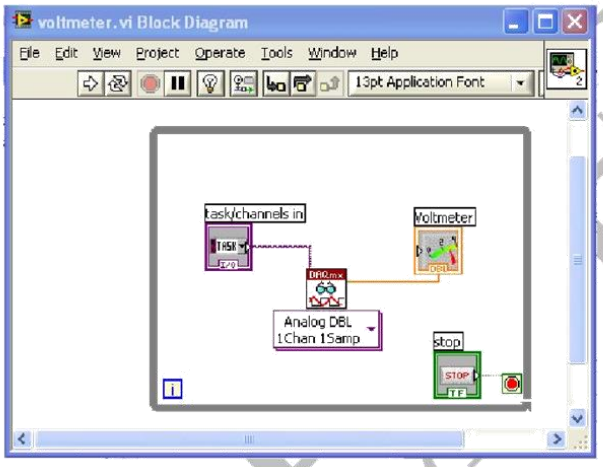
Voltmeter VI

1. Open LabVIEW and create new VI.
2. Build the following front panel, and block diagram.

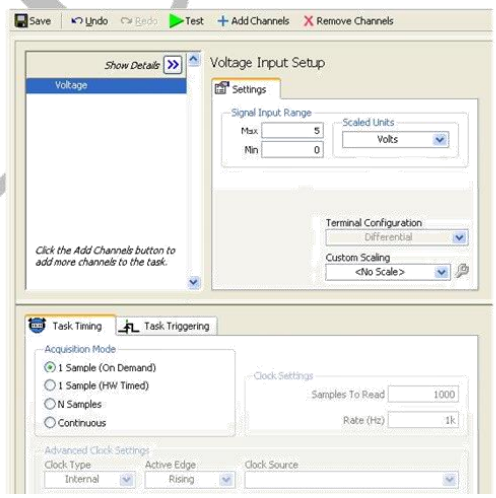
BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh





3. Create a new task by clicking on the combo box (task/channels in) and select **Browse**. Then click **Create New»MAX Task** and choose **Analog Input»Voltage** then click **Next**.
4. Select **ai0** channel and click **Next**. Name the task "input" and click **Finish**.
5. Configure the task as shown in the following figure.



6. Connect the output of the DC power supply to channel AI0 and to DVM, then run the voltmeter VI. Vary the DC voltage in the range 0 – 5V and compare the reading from the DVM and your voltmeter VI.

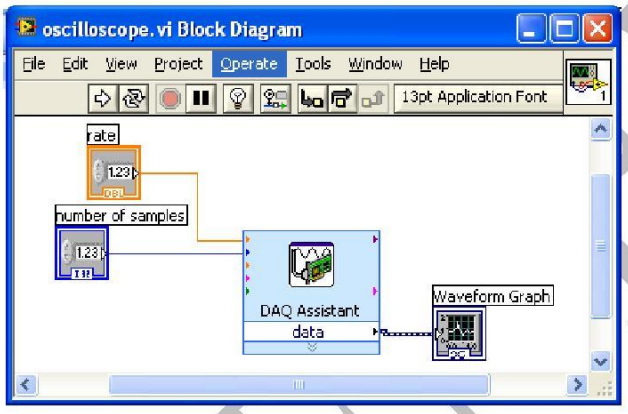
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Exercise:

Modify the voltmeter VI by adding two LEDs such that the first LED becomes green if the input voltage exceeds 2V, and the second LED becomes red if the input voltage exceeds 3V.

Reading Continuous Signals

1. Create the following VI, and name it **oscilloscope.vi** .



2. When you insert the **DAQ Assist**, a Wizard will be opened. Select **Analog Input»Voltage**, and then choose channel **ai0**.
3. Change the **Signal Input Range** to 0 – 5V, and the **Terminal Configuration** to Differential, and the **Acquisition Mode** to N Samples. Also set the **Samples To Read** to 1000, and the **Rate** to 1KHz.
4. Connect the function generator output to analog input channel 0 (Differential) of the DAQ, and adjust its output to give sinusoidal signal of 2Vpk-pk and 20Hz.
5. Run your VI and compare between the displayed signal and the input signal. Change the signal type and frequency and see the changes in the displayed signal.
6. Change the **DAQ Assist** settings to the Continuous Acquisition Mode. You will be asked to put the **DAQ Assist** in a loop, click yes. Run your VI and examine the output by changing the signal type and frequency of the input signal.

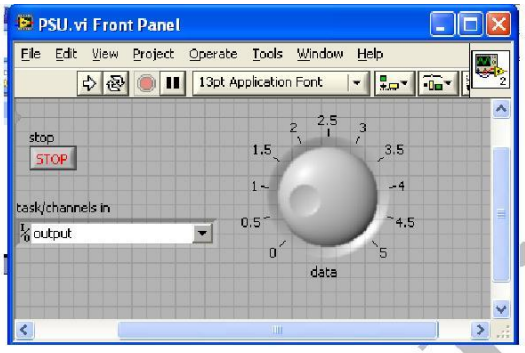
Exercise:

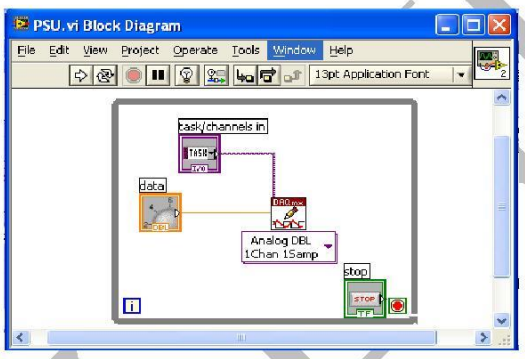
Modify **oscilloscope.vi** to display the power spectrum of the input signal. You can use the **Spectral Measurements** block found under **Express»Signal Analysis**.

BZU ECE
STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Generating Voltage Output Using LabVIEW

1. Create new VI, and name it **PSU.vi**
2. Build the following front panel, and block diagram.





3. Create a new Task and choose **Analog Output»Voltage**, then select **ao1**. Name the task **output** and set the **Generation Mode** to be **"1 Sample (On Demand)"**.
4. Connect the output channel 1 to the Digital Multimeter and run your VI, then test the DAQ output.

Exercise:

Modify **PSU.vi** to generate a sinusoidal signal of Amplitude 2V and frequency of 20Hz. Do not forget not to generate voltage out of the range 0 – 5V. Do not use **DAQ Assist** in this VI, you must use the **DAQmx Write** block with the same task generated in the previous part.

Hint: you can use the **Expression Node** found under **Mathematics»Numeric** in the **Function Pallete**. Also do not forget to use delay in the loop to control sampling rate.

BZU FREE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment # 9
PWM Control Project

Objective:

In this experiment the student should implement an electronic circuit that employs PWM to control a DC motor.

Procedure:

1) Download all the data sheets related to the component that will be used in the circuit shown below in Figure (1).
2) Analyze and simulate the circuit
3) Familiarize yourself with pin connections of each component.
4) Build the circuits shown below on a breadboard.
5) Follow any instructions given to you by your instructor
6) Check the function of your circuit,
7) Show your instructor that the circuit is functioning properly.

Note that: a) a voltage regulator should be included in your design.

b) The gate resistor of the MOSFET is 33 Ohm

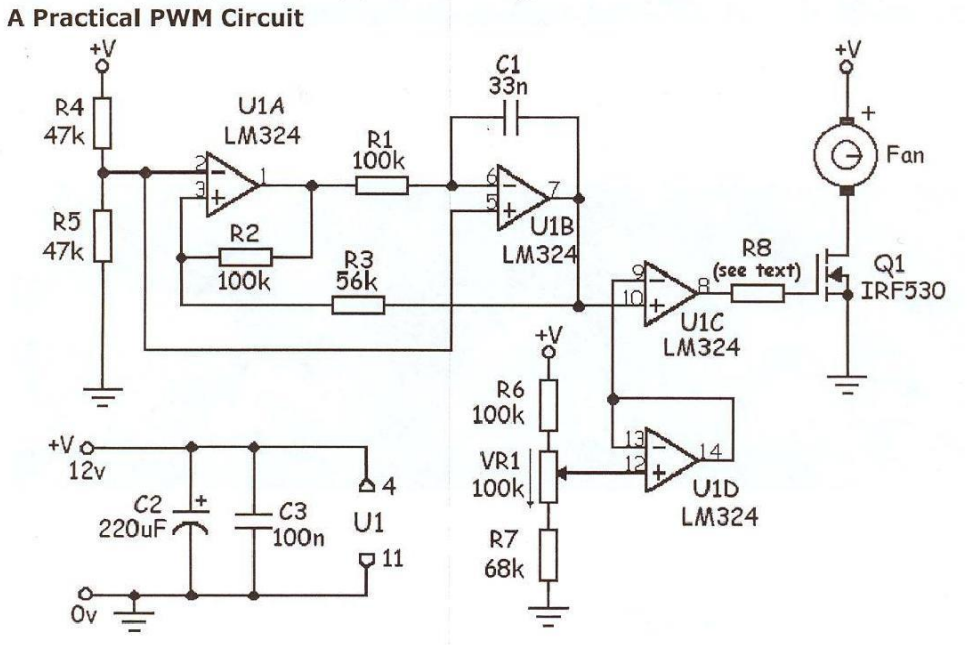


Figure (1)

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Repeat a similar procedure for the circuit shown in **Figure (2)**,

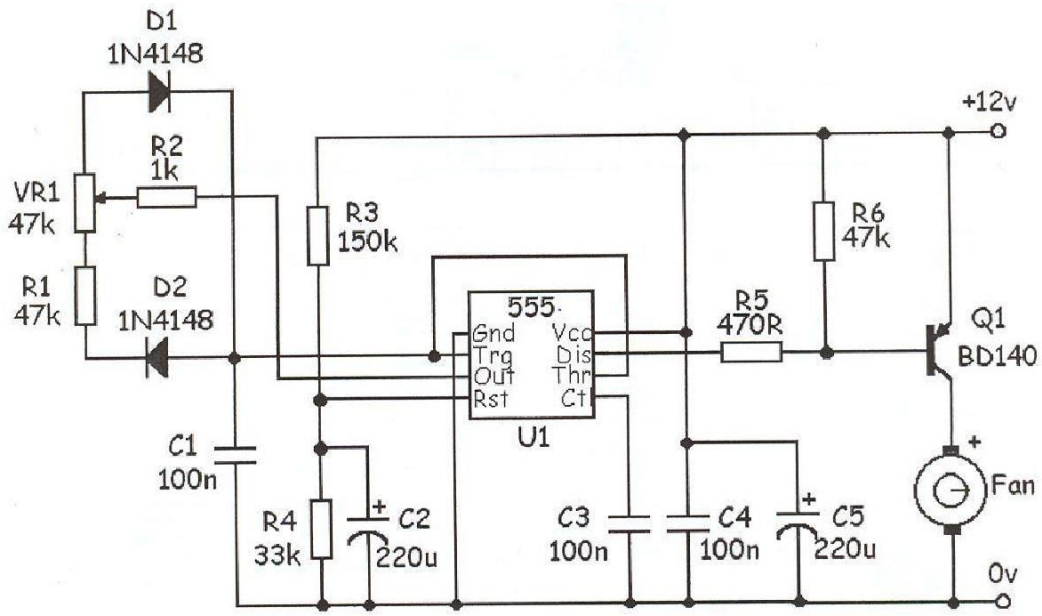


Figure (2)

BZU

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Experiment # 10
Filter Design Using MATLAB

Objective

This experiment is intended to provide an introduction to filter design using MATLAB. You will use the SPTool to view a signal, find its spectrum and design a digital filter that will be applied to any desired signal.

Learning Outcomes

After completing this session the student should be able to:

- Create signals consisting of sampled sinusoids of different amplitudes and frequencies
- Add White Noise to the signal(s)
- Be able to visualize the signal(s) using SPTool
- Generate and interpret the frequency spectrum of a signal(s)
- Design Digital Filters
- Apply the filters to signals
- Find the filters parameters.

Procedure

Data creation:

1) Create a MATLAB script that generates a discrete-time signal (x) which is the sum of three sine signals of frequencies 100, 250 and 400Hz and with RMS amplitudes of 1, 2 and 3 respectively. Use the following parameters in your code:
   fs = 1000 samples/sec
   n = [0 1 2 3 … 1023]
   Your code must be in terms of (fs) and (n).
2) Create a signal (y) which is the signal (x) with AWGN noise and signal-to-noise ratio of 20 dB. Use the MATLAB function awgn().
3) Plot the signal (y) showing the axis labels. Attach the plot to your report.

Using SPTool

It is possible to analyze the signal using command lines, but we will use one of the many MATLAB tools, in this case SPTool.

Type sptool at the matlab prompt, a window should appear.
This tool will enable you to

- Visualize signals (signals column)
- Design Digital Filters (filters column)
- Look at the signals in the frequency domain. (Spectra column)

Importing signals

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh

Click on file (top left-hand side of SPTool window) Choose import. A window should appear. Select the signal (y) and the sampling frequency fs. When you are finished, click on OK. Click view under the signals column to view your signal. show the output.

Use of the Spectrum Viewer

It is worth checking the spectrum of the signal to see if the three components are present.

Make sure **sig1** is highlighted. Choose the Create- option from the spectra column of the main menu.

There are many options, but the default is sufficient for this lab

Click on Apply, (bottom left hand corner). Your spectrum should be displayed. Note

- The position of the three peaks,
- The amplitude of the three peaks

What is the position and amplitude of the three peaks?

	1^{st} peak	2^{nd} peak	3^{rd} peak
Amplitude	0.41(-7.76dB)	0.85(-1.39dB)	1.26(2dB)
Frequency	100	250	400

Filter Creation.

In order to separate the three signals it is necessary to create three filters: a low-pass, a high-pass and a band-pass filter. There are a number of methods for filter design. In this lab we will concentrate on the elliptic design method as that will produce filters with a small number of coefficients.

- Click on new under the column of filters to create a new filter and close the window.
- Change the filter name to lowpass (click edit -> name ->filt1)
- Click on **Edit** in the SPTool window and change sampling frequency of lowpass to fs.
- Click on **edit design** and the filter designer window will appear.

There are a number of options. Choose Elliptic and Low-pass from the menu and type in the large white window

- **Fpas: 150**
- **Fstop: 200**
- **Apass: 1**
- **Astop: 60**

Click apply under the filters column to apply the filter to sig1 and create sig2. do the results agree with theory?

Show how to find the filter parameters (use the MATLAB help)

Now design 2 more filters a band-pass and high-pass to recover the two other signals.

BZU ENEE

STUDENTS-HUB.com Uploaded By: Mohammad Awawdeh