System Integration & Programming of EHD 3D printer

A Thesis

Submitted to the Faculty

of Drexel University

by

Vipul Kumar Sharma

in partial fulfillment of the

requirements for the degree

of

Master of Science in Mechanical Engineering

June 2017

DREXEL UNIVERSITY

DEDICATION

I dedicate my dissertation work to my family and friends, who has supported me morally and economically to reach this instance in my education. A special feeling of gratitude to my parents, whose words of encouragement have instilled in me the necessity of higher education. My brother, Atul has never left my side and is very special. I love you guys very much and appreciate your efforts. I also dedicate this dissertation to my friends, Jaideep & Udit who always supported me throughout the process. I will always appreciate all they have done, for helping me develop my technology skills. I dedicate this work and give special thanks to my Professor, Dr. Jack G. Zhou and my wonderful research assistant Dajing Gao for being there for me throughout the entire program. Both of you have been my best inspiration. I promise to keep moving forward and try finding ways to improve personally and professionally, thank you for believing in me.

ACKNOWLEDGEMENT

First, I would like to thank my advisor, Dr. Jack G. Zhou, who has always helped and instructed me throughout my thesis with a lot of knowledge and patience. The door to Prof. Zhou office was always open whenever I ran into a trouble or had some questions about my research or writing. He steadily allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it. Although it was a slow start and his continued patience that I developed the dedication needed to perform technological related research studies for my Master's degree. This also could have not been possible without the support of my dearest friends, who have helped and guided me into the work completed in this thesis.

I would also like to acknowledge Dajing Gao as the second reader of this thesis, and I am gratefully indebted to his for his valuable comments on this thesis and for giving me the opportunity to continue my graduate studies at Drexel University and support throughout my degree.

Lastly, I must express my very profound gratitude to my parents and to my friends for providing me with unfading support and ceaseless encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

“We are what our thoughts have made us; so take care about what you think. Words are secondary. Thoughts live; they travel far.”

- Swami Vivekananda

Table of Contents

LIST OF FIGURES

ABSTRACT

System Integration & Programming
of EHD 3D printer

Vipul Kumar Sharma
Dr. Jack G. Zhou, Ph.D

This project is the base for Electro Hydro Dynamics (EHD) & Air Focusing Flow (AFF) 3D printer. Electro Hydo Dynamics (EHD) is the study of dynamics of electrically charged fluids and motions of ionized particles or molecules and their interactions with electric fields and the surrounding fluid. In order to study and work on EHD, this is the first phase. Before going to 3D printing or Manufacturing of the components, it's better to test the machine and evaluate its performance in a virtual environment. First, we took food printer and converted into a simple 3D printer. Then we took a high precision Semprex KL-series table to work on the EHD. With our project outlined, we had to choose a design, source the parts, build the printer, and get the printer to work with NI's software and hardware which has not yet been done before. The nozzle and base both are not stationary. The base lies on the XY-axis while the nozzle is fixed on the Z-axis. The LabVIEW has to be designed so that it can integrate MACH3 as well as Slicer with an integration of temperature and pressure control. For this project we are using NI LabVIEW 2016, which is the latest version.

CHAPTER 1. INTRODUCTION

3D printing, also known as additive manufacturing (AM), refers to processes used to synthesize a three-dimensional object in which successive layers of material are formed under computer control to create an object. Objects can be of almost any shape or geometry and are produced using digital model data from a 3D model or another electronic data source such as an Additive Manufacturing File (AMF) file. The term 3D printing has its origin sense, 3D printing in reference to a process that deposits a binder material onto a powder bed with inkjet printer heads layer by layer. More recently, the term is being used in popular vernacular to encompass a wider variety of additive manufacturing techniques. This project had several parts to it: choice of the design, sourcing of the parts, building the printer, and getting the software and hardware to function correctly. Main focus of the project is the EHD and AFF part. The EHD printing system consists four subsystems: a precision three-axis motion system, a dispensing system, a heating system and voltage supply. It works by electric field applied on a pendent drop at end of a nozzle and then Mobile ions in a polarizable liquid to gather at surface. AFF System contains pneumatic compressor which is capable of providing 15-20 psi to chamber which results in fine droplets.

Both the concepts work almost in the same way. They both form Taylor cone which is used for printing. When a small amount of electrically conductive liquid is exposed to an electric field, the shape of liquid starts to deform from the shape caused by surface tension alone. As the voltage is increased the effect of the electric field becomes more eminent. As exerting a similar amount of force on the droplet as the surface tension does, a cone shape begins to form with convex sides and a rounded tip.

1.1 EHD Concept

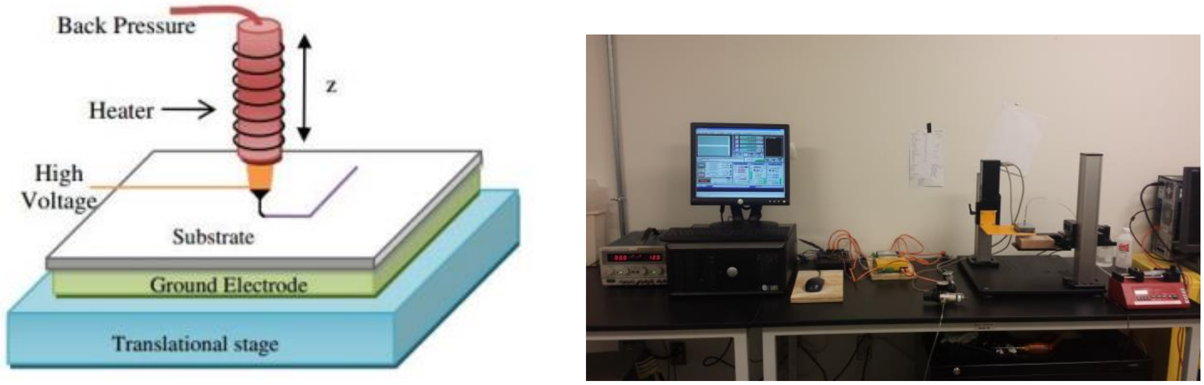


Figure 1.1:- Development and Modeling of Melt Electro hydrodynamic-Jet Printing of Phase-Change Inks for High-Resolution Additive Manufacturing. Journal of Manufacturing Science and Engineering.

Figure 1 shows the schematic setup of the EHD printing system. The precision three-axis system has three linear motion stages with an accuracy and precision of ACME lead screw is 2mm/rev and three motion stages were used to control the movement of the substrate and the print head with internal rotary encoders with error as little as 1 micron per 2500 steps for position feedback.

EHD-jet printing in the micro-dripping or pulsating mode provides drop-on-demand capabilities that have great capacity in micro-scale 3D printing when using wax-based support and build materials. The fine droplets can be deposited and solidified on demand to form layer-by-layer 3D structures. The printing results is determined by the ink properties, along with process conditions, mostly applied voltage. From Figure 2, clearly increasing the voltage will decrease droplet size and increase ejection/printing frequency, which can be easily explained by the mechanism of EHD printing. An increasing voltage will increase surface charge density and charge migration speed at the Taylor cone. This results smaller droplets to obtain large enough electrostatic force to overcome surface tension to be ejected from the cone tip with less charge accumulation time.

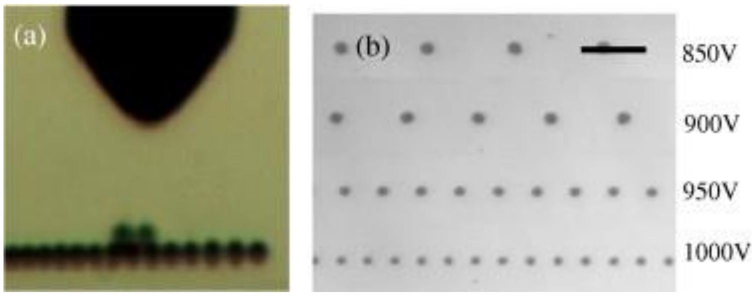


Figure 1.2:- (REF)Chuan Wei, Jingyan Dong. (a) Pulsating mode of EHD printing of wax. (b) Droplet size and printing frequency at positive voltages.

Figure 3(a) shows Finite Element Analysis (FEA) based model to predict droplet dimension at different system configurations and process conditions. It can be seen that a hemispherical meniscus forming at the nozzle tip. The EHD printing process depends on two competing forces, surface tension and the electrostatic Columbic force. The droplet with the fixed diameter has the largest surface tension force, and will be ejected if the electrostatic force acting on the droplet surface is larger than the surface tension force. The electrostatic forces on the droplet can be calculated at different printing conditions in the FEA. The electrostatic force *Fe* can be expressed as a function of *V*, droplet diameter *Dd*, & Dn
$F_e = cV^{k1}D_d^{k2}D_N^{k3}$

The surface tension force *Fs* can be expressed as

Fs = *σπdD* (*σ:* surface tension coefficient).

Then the droplet diameter can be calculator by equaling above two functions.

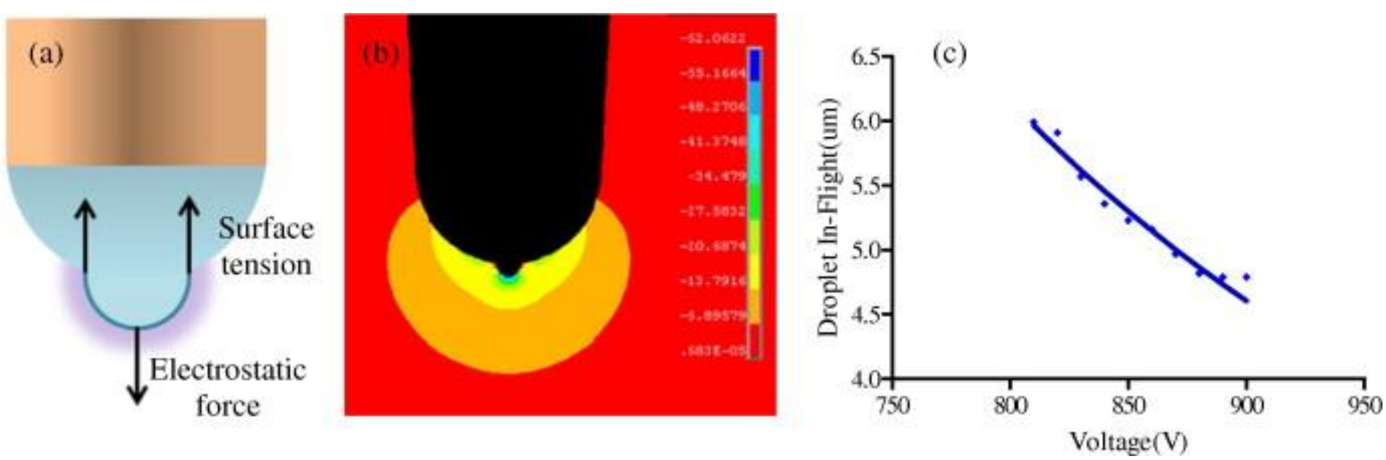


Figure 1.3:- (REF)Chuan Wei, Jingyan Dong. Schematic for FEA of the electrostatic force and surface tension force on the droplet. (b) Electrical field distribution during droplet ejection. (c) Result of droplets dimension from FEA (line) and experimentally measured (data points).

1.2 Air Focusing Flow

Air Flow focusing results from combining hydrodynamic forces with a specific geometry. AFF device (Figure 4a) consists of a pressure chamber which pressurizes with a continuous focusing fluid supply that enters through (1). Inside, a focused fluid (2) is injected through a tube whose extremity opens up in front of a small orifice linking the chamber with the exterior ambient. The focusing fluid stream (1) molds the fluid meniscus (3) into a cusp giving rise to a microjet exiting the chamber through the orifice; the jet diameter is much smaller than the exit orifice diameter, thus precluding any contact. Capillary instability breaks up the stationary jet into uniform droplets. [4] The feed tube may be composed of two or more concentric needles and different liquids can be directly injected, thus leading to multilayer microcapsules with multiple shells of controllable thickness Figure 4b.

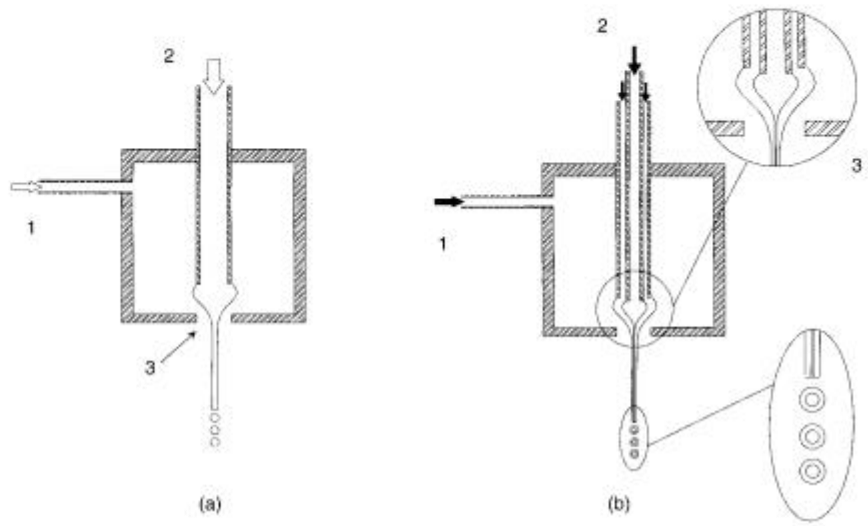


Figure 1.4:- (REF)Wiley-VCH Verlag GmbH &Co. Flow-focusing atomizer. a) Simple jet: 1) focusing fluid, 2) focused fluid, 3) meniscus; b) compound atomizer with two concentric needles: 1) focusing fluid, 2) focused fluids: core fluid and shell fluid, 3) compound meniscus.

A low-Reynolds planar device /microfluidic (where the Reynolds number (Re) expresses the ratio of inertial forces to viscous forces) based on the AFF principle has already been successfully utilized[6] to yield highly monodisperse objects with a no spherical shape, where the break-up mechanism is controlled by the surface tension and the viscous forces, while the geometry of the outlet channel determines the eventual shape of the particles. In contrast to such systems, the AFF applications described above enjoys superior productivity owing to their high- and moderate-Reynolds operation. This causes a break-up pattern where inertia, surface tension, and geometry play the leading roles: a exceptional degree of size-control is achieved, ranging up to the lowest microscales. High-Reynolds applications are able to “docile” inertia, a potentially unruly component of the flow pattern, by the judicious use of design; thus they prove particularly suitable for miniaturization, given their sensitive and strong response to geometric fine-tuning. The operation of such devices is strikingly energy-efficient, owing to the restrained role of viscosity. In short, AFF ensures an extremely fast creation of up to millions of droplets per second as the jet breaks up, [5] a rate much higher than that obtained by other microdripping techniques.

CHAPTER 2. LabVIEW INTRODUCTION

LabVIEW programs/software system are called virtual instruments (VIs). Each VI has mainly three components: a block diagram, a front panel and a connector panel. The endmost is used to represent the VI in the block diagrams of other, calling VIs. The front panel is made using controls and indicators. Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a system-design platform and development environment for a visual programming language from NI. The graphical language is named "G"; not to be misinterpreted with G-code. NI LabVIEW is the ultimate system design software used by engineers and scientists to produce design, prototype, and deploy embedded control and monitoring applications. It combines hundreds of prewritten libraries, tight assimilation with off-the-shelf hardware, and a variety of programming approaches including graphical development, .m file scripts, and connectivity to existing ANSI C and HDL code. In case of designing medical devices or complex robots, reduce time to market and the overall cost of embedded control and monitoring with LabVIEW. With the powerful NI LabVIEW system design environment, we can build any measurement or control system in dramatically less time. Unlike general-purpose tools, LabVIEW firmly integrates any hardware with extensive analysis and signal processing libraries, offers custom graphical user interfaces, and allows us to deploy these systems to a platform that uses the newest and most advanced technology.

2.1 The Value of a Platform

LabVIEW is the keystone of NI’s system design platform. Investing in a platform approach gives the ability to effectively scale the application to meet the changing requirements of the labview.

LabVIEW increases efficiency by abstracting low-level complexity and integrating all of the technology we need into a single, unified development environment, unlike any other text-based alternative. Programming in a unified environment means we don’t have to invest time in building expertise in a variety of tools to fulfill the goal. Instead, we can be confident that elements of the system will fit together seamlessly.

2.2 Theory

The LabVIEW Environment

LabVIEW programs are called Virtual Instruments, or VIs, because their appearance and operation emulate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a complete set of tools for acquiring analyzing, displaying, and storing data, as well as tools to help we troubleshoot the code.

The Getting Started window will appear when we launch LabVIEW. Use this window to generate new projects and open existing files. We also can access resources to expand the capacity of LabVIEW and information to help we learn about LabVIEW.

The Getting Started window vanishes when we open an existing file or create a new file and reappears when we close all open front panels and block diagrams. We also can display the window from the front panel or block diagram by selecting **View**» **Getting Started Window**.

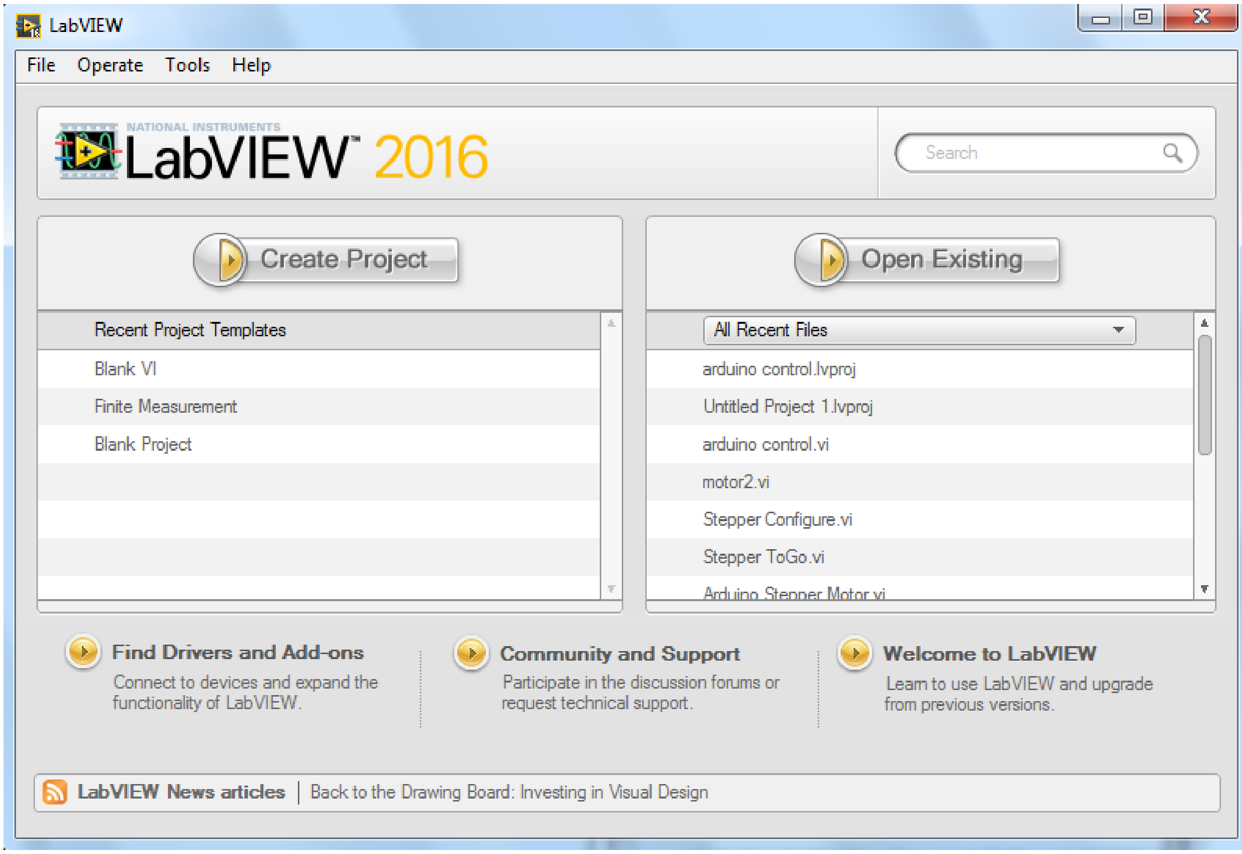


Figure 2.1: - Getting Started window

In order to generate a new VI, select “Blank VI” or in order to create a new LabVIEW project, select “Empty project”.

2.3 Front Panel and Block Diagram

In LabVIEW, we build a user interface, or front panel, with controls and indicators. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. We build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

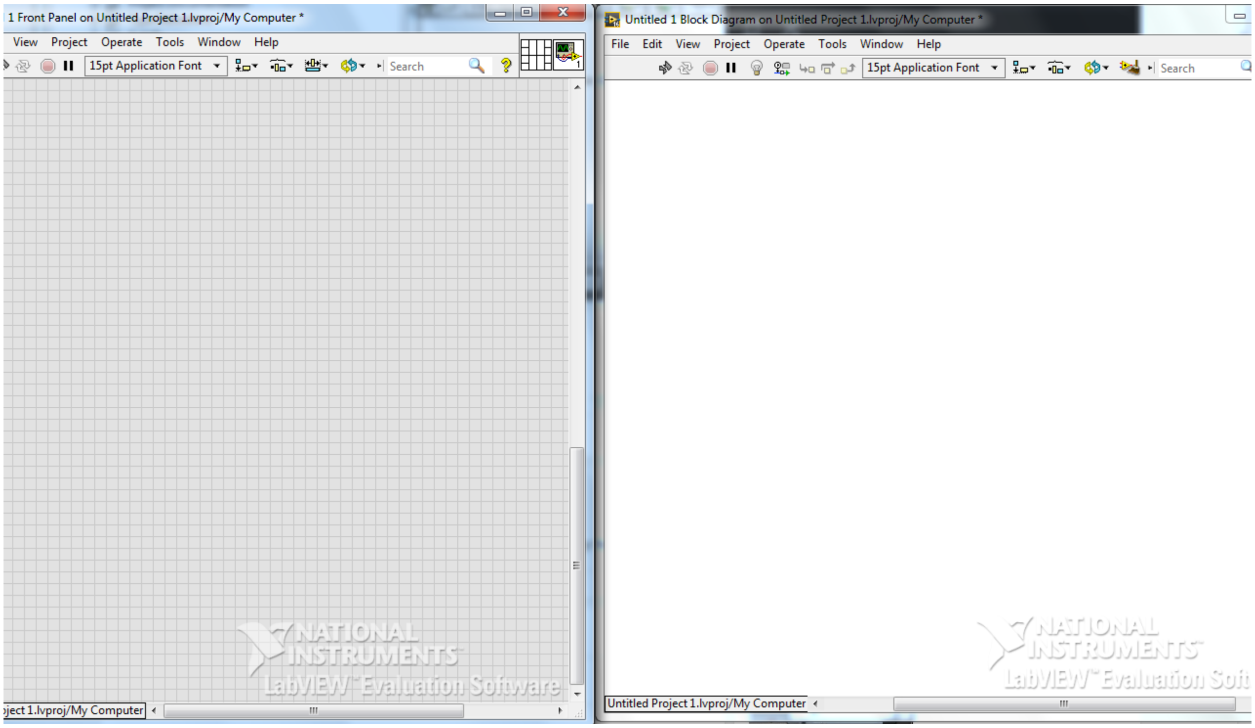


Figure 2.2: - front panel and block diagram

After we build the user interface, we add code using VIs and structures to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart. After we build the front panel, we add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals, on the block diagram. Block diagram objects include terminals, subVIs, functions, constants, structures, and wires, which transfer data among other block diagram objects.

2.4 Controls Palette

The Controls and Functions palettes contain sub palettes of objects we can use to create a VI. When we click a sub palette icon, the entire palette changes to the sub palette we selected. To use an object on the palettes, click the object and place it on the front panel or block diagram. The Controls palette is available only on the front panel. The Controls palette contains the controls and indicators we use to build the front panel.

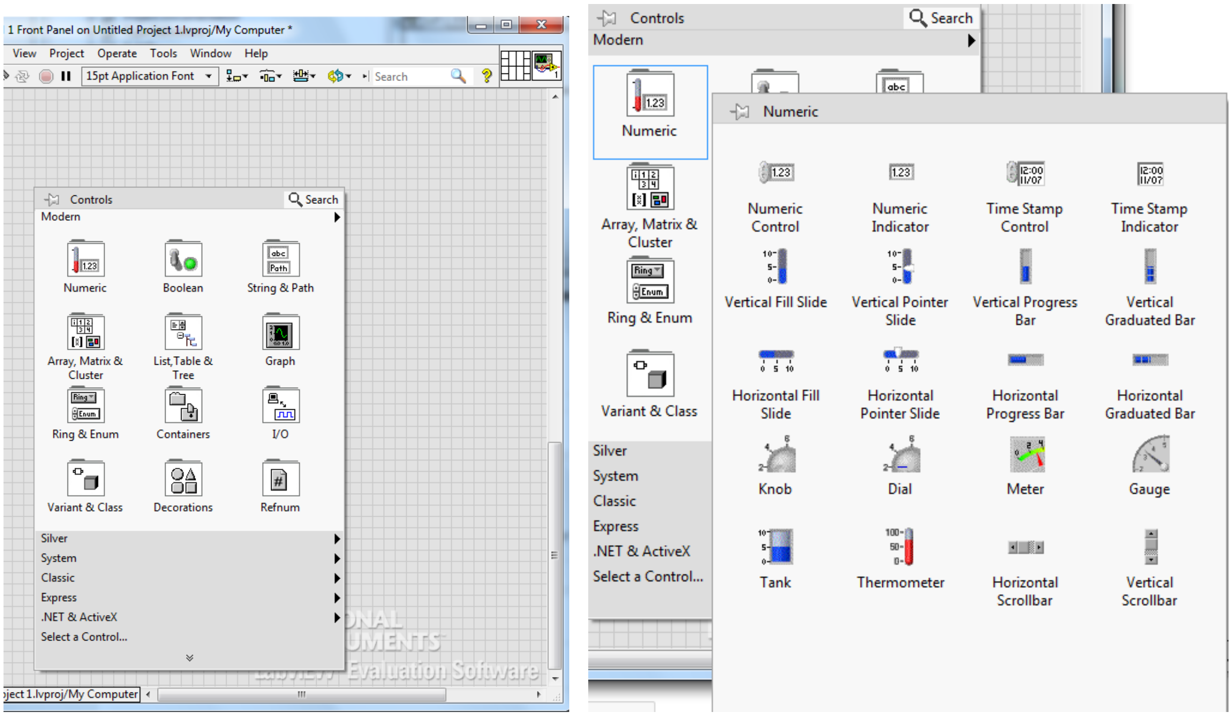


Figure 2.3: - front panel command window

The Functions palette is available only on the block diagram. The Functions palette contains the VIs and functions we use to build the block.

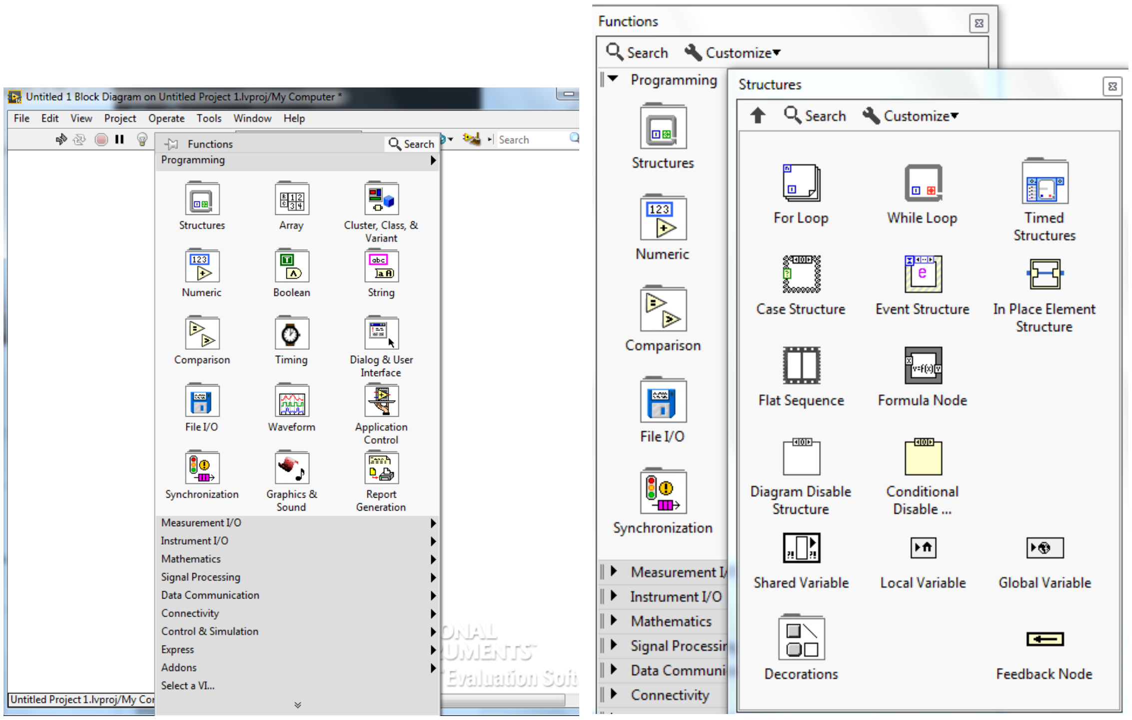


Figure 2.4: - block diagram command window

2.5 Tools Palette

We can create, modify, and debug VIs using the tools located on the floating Tools palette. The Tools palette is available on both the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. The cursor corresponds to the icon of the tool selected in the Tools palette. Use the tools to operate and modify front panel and block diagram objects.

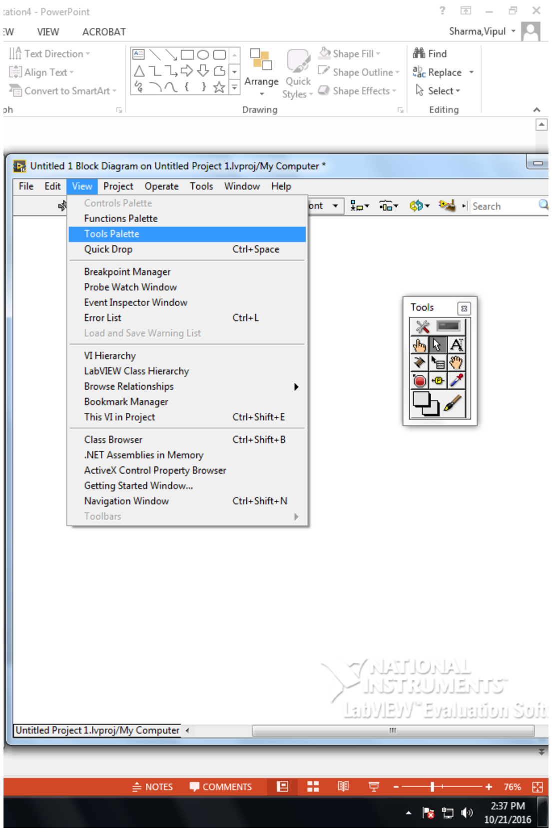


Figure 2.5: - tools palette command window

Use the **Operating tool**, shown at left, to change the values of a control or select the text within a control. The Operating tool changes to the icon shown at left when it moves over a text control, such as a numeric or string.

Use the **Positioning tool**, shown at left, to select, move, or resize objects. The Positioning tool changes to resizing handles when it moves over the edge of a resizable object.

Use the **Labeling tool**, shown at left, to edit text and create free labels.

Use the **Wiring tool**, shown at left, to wire objects together on the block diagram.

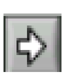
2.6 Wiring

In order to create the logical flow between the object on the Block Diagram, we need to use the Wiring tool in order to connect the different objects together.

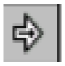
Available Keyboard Shortcuts when dealing with Wiring:

Wiring	
Ctrl-B	Removes all broken wires.
Esc, right-click, or click terminal	While wiring, cancels a wire you started.
Single-click wire	Selects one segment.
Double-click wire	Selects a branch.
Triple-click wire	Selects entire wire.
A	While wiring, disables automatic wire routing temporarily.
Double-click	While wiring, tacks down wire without connecting it.
spacebar	While wiring, switches the direction of a wire between horizontal and vertical.
spacebar	While moving objects, toggles automatic wiring.
Ctrl-click input on function with two inputs	Switches the two input wires.
Shift-click	While wiring, undoes last point where you set a wire.

2.7 Toolbar

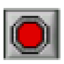
 Click the Run button to run a VI. LabVIEW compiles the VI, if necessary. We can run a VI if the Run button appears as a solid white arrow. The solid white arrow, shown above, also indicates we can use the VI as a subVI if we create a connector pane for the VI.

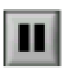
 While the VI runs, the Run button appears as shown at left if the VI is a top-level VI, meaning it has no callers and therefore is not a sub VI.

 If the VI that is running is a subVI, the Run button appears as shown at left.

 The Run button appears broken, shown at left, when the VI we are creating or editing contains errors. If the Run button still appears broken after we finish wiring the block diagram, the VI is broken and cannot run. Click this button to display the Error list window, which lists all errors and warnings.

 Click the Run Continuously button, shown at left, to run the VI until we abort or pause execution. We also can click the button again to disable continuous running.

 While the VI runs, the Abort Execution button, shown at left, appears. Click this button to stop the VI immediately if there is no other way to stop the VI. If more than one running top-level VI uses the VI, the button is dimmed.

 Click the Pause button, shown at left, to pause a running VI. When we click the Pause button, LabVIEW highlights on the block diagram the location where we paused execution, and the Pause button appears red. Click the button again to continue running the VI.

2.8 Execution

In addition to the Toolbar buttons above the following Keyboard Shortcuts are available when dealing with Execution:

Execution	
Ctrl-R	Runs the VI.
Ctrl-.†	Stops the VI.
Ctrl-M	Changes to run or edit mode.
Ctrl-Run button	Recompiles the current VI.
Ctrl-Shift-Run button	Recompiles all VIs in memory.
Ctrl-↓†	Moves key focus inside an array or cluster.
Ctrl-↑†	Moves key focus outside an array or cluster.
Tab†	Navigates the controls or indicators according to tabbing order.
Shift-Tab†	Navigates backward through the controls or indicators.
† While the VI is running	

Running a VI executes the solution of the VI. Click the Run button or press the <Ctrl-R> keys to run a VI. The Run button changes to a darkened arrow to indicate the VI is running. We can stop a VI immediately by clicking the Abort Execution button. However, aborting a VI that uses external resources might leave the resources in an unknown state. Design the VIs we create with a stop button to avoid this problem. A stop button stops a VI after the VI completes its current iteration.

2.9 Express VIs

Use Express VIs located on the Functions palette for common measurement tasks. When we place an Express VI on the block diagram, the dialog box we use to configure that Express VI appears by default. Set the options in this configuration dialog box to specify how the Express VI behaves. We also can double-click an Express VI or right-click an Express VI and select Properties from the shortcut menu to display the configuration dialog box. If we wire data to an Express VI and run it, the Express VI displays real data in the configuration dialog box. If we close and reopen the Express VI, the VI displays sample data in the configuration dialog box until we run the VI again.

CHAPTER 3. DATAFLOW PROGRAMMING

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when all its inputs are available. When a node completes execution, it supplies data to its output terminals and passes the output data to the next node in the dataflow path.

Examples:

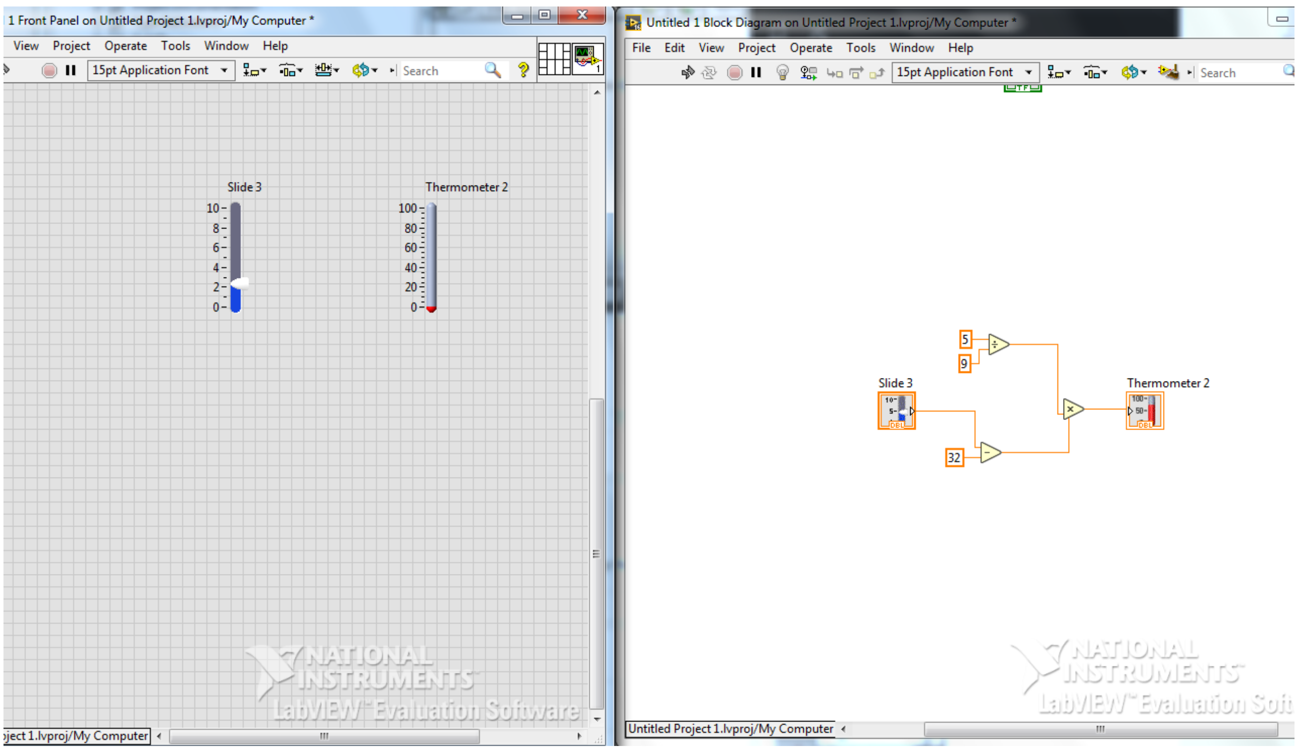


Figure 3.1: - temperature example

This is a simple example of a thermometer which is being used for the conversion of degree Celsius to degree Fahrenheit. Using the simple formula

$$\boldsymbol{T_{(°C)} = (T_{(°F)} - 32) \times 5/9}$$

We can use simple numerical functions in the block diagram to get the result.

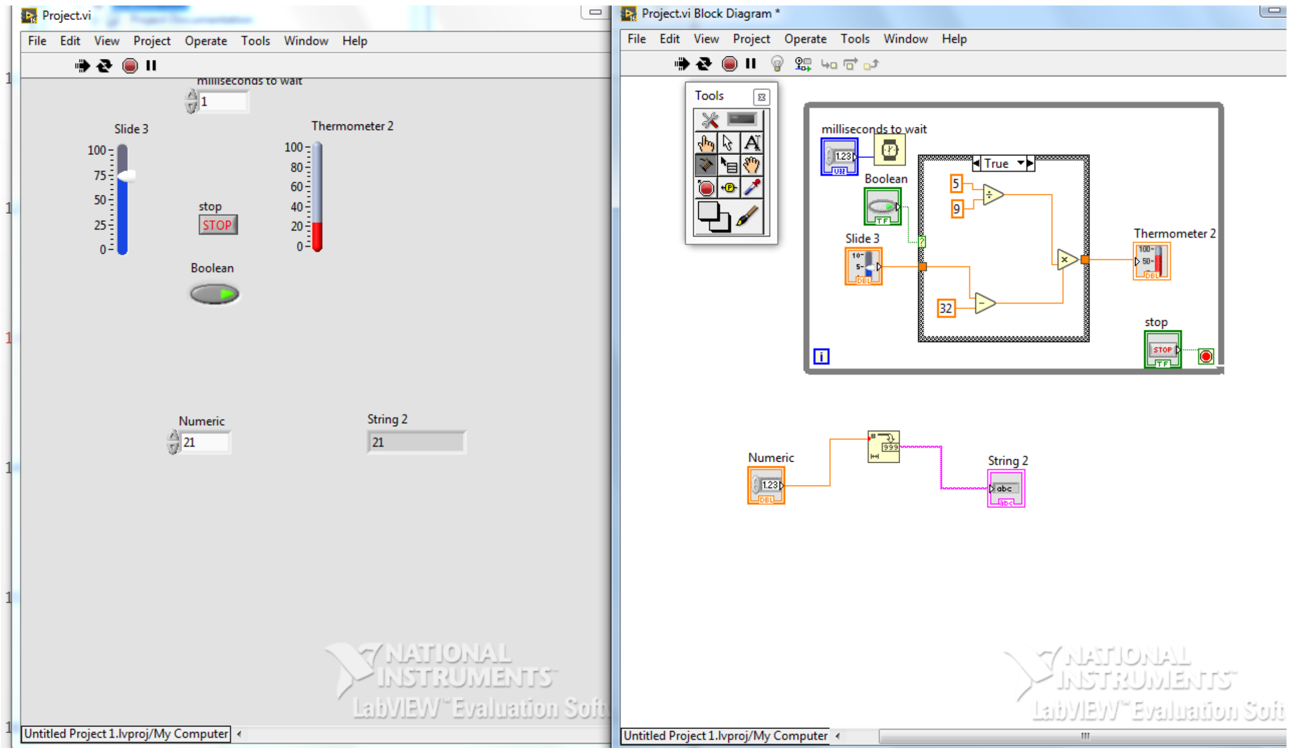


Figure 3.2: - working temperature example

This is the same example of a thermometer which is being used for the conversion of degree Celsius to degree Fahrenheit and the only difference is that we used **CASE STRUCTURE** and **WHILE LOOP**.

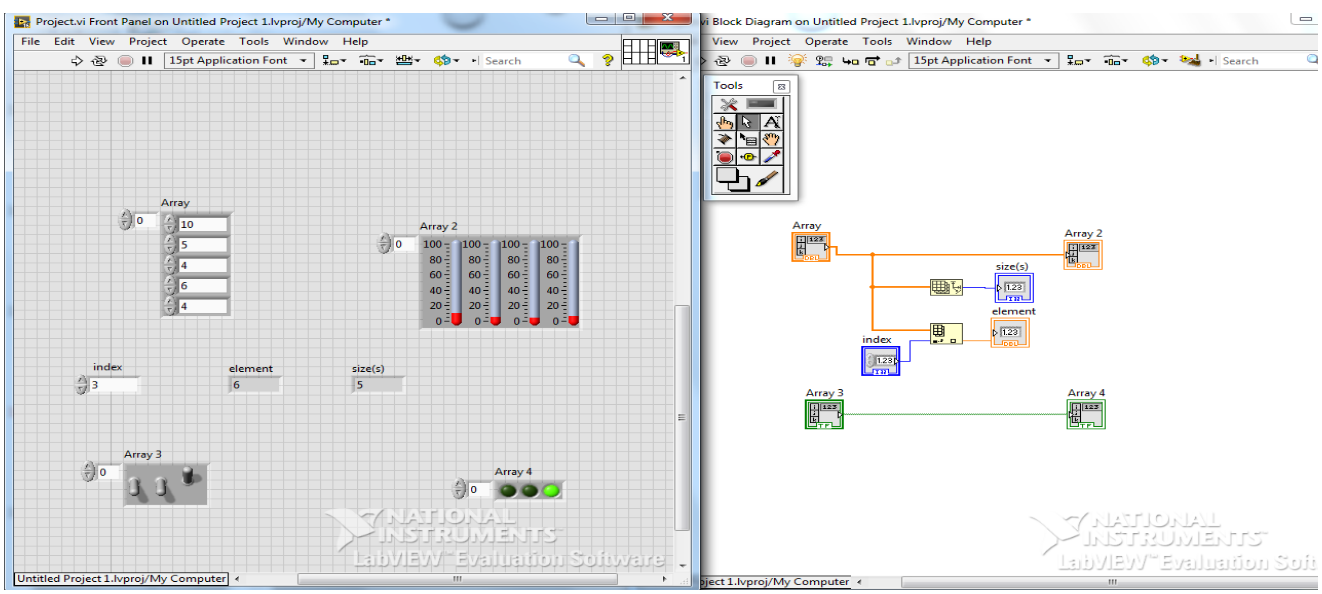


Figure 3.3: - case structure

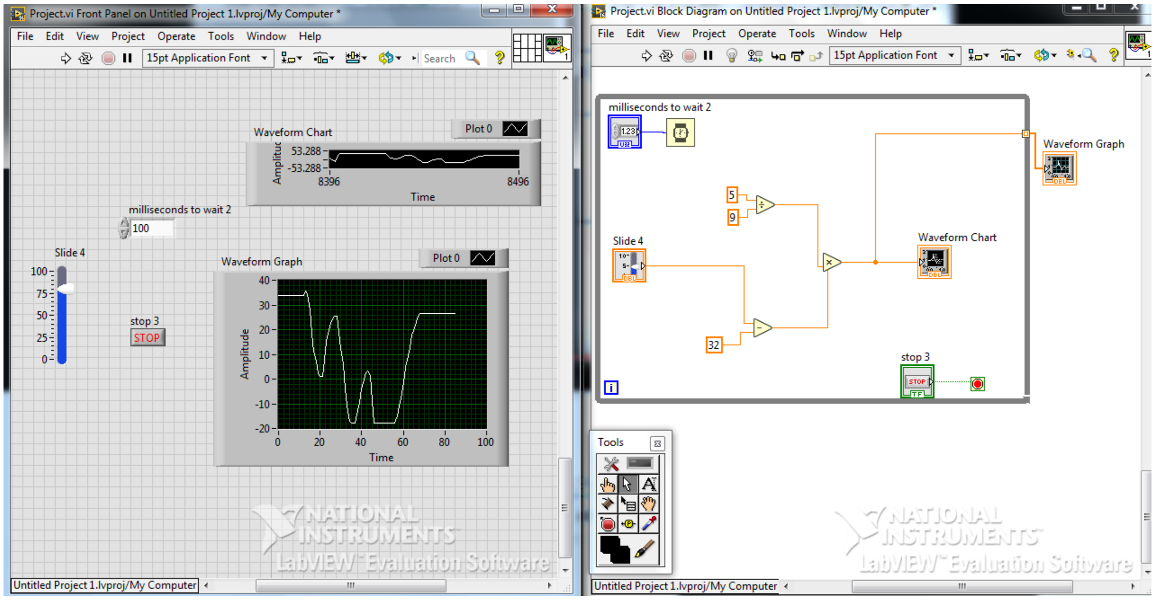


Figure 3.4: - waveform chart

This is an example of charts and table. It shows how the graphs are being used in the LabVIEW. The program varies with the time and it gives the amplitude vs time graph.

3.1 Signal Processing

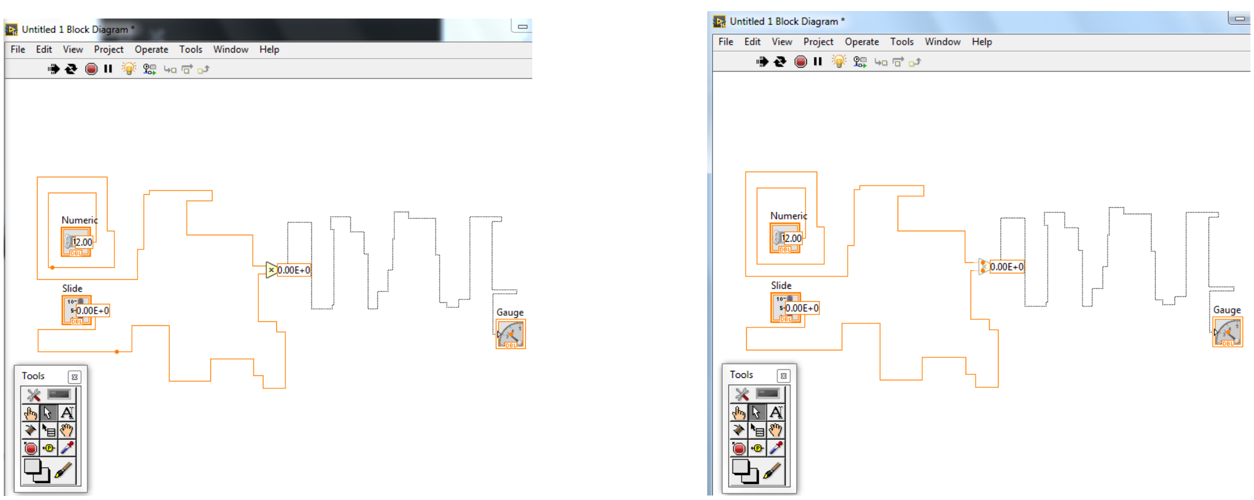


Figure 3.5: - signal processing

Time domain analysis is beneficial when observing data such as temperature. However, some applications require analyzing the frequency components of signals. This allows us to see which frequencies make up a sound signal, similar to an audio equalizer or the vibration frequencies of a motor while running. Use this module to identify and filter out frequencies in NI LabVIEW software.

3.2 Sub VIs

There are three basic concepts of creating and using Sub VIs in LabVIEW.

- •Create New SubVI from Scratch
- •Create SubVI from existing code
- •Using Sub VI

When we place a VI on the block diagram, LabVIEW considers the VI to be a subVI. When we double-click a subVI, its front panel and block diagram appear, rather than a dialog box in which we can configure options. The front panel includes controls and indicators. The block diagram includes wires, front panel icons, functions, possibly subVIs, and other LabVIEW objects. The upper right corner of the front panel and block diagram displays.

Available Keyboard Shortcuts when dealing with Sub Vis:

SubVIs	
Double-click subVI	Displays subVI front panel.
Ctrl-double-click subVI	Displays subVI block diagram and front panel.
Drag VI icon to block diagram	Places that VI as a subVI on the block diagram.
Shift-drag VI icon to block diagram	Places that VI as a subVI on the block diagram with constants wired for controls that have non-default values.
Ctrl-right-click block diagram and select VI from palette	Opens the front panel of that VI.

Create New Sub VI from Scratch

Select “Blank VI” in the “Getting Started” window when opening LabVIEW, or when LabVIEW is already opened select File→ New V” or use the short-cut Ctrl + N.

Create Sub VI from existing code

The procedure is as follows:

1. Select the part of the code we want to turn into a SubVI.
2. From the Edit menu, select “Create SubVI
3. LabVIEW will automatically create a SubVI for the selected code.
4. Clean up automatically created wires, etc.
5. Create a suitable icon for the SubVI.

Using Sub VIs

We may open a SubVI from the File menu, select a SubVI from the Functions palette or use drag and drop in different ways, e.g., we may drag a VI from the File Explorer in Windows directly into an existing VI we have already opened in LabVIEW.

3.3 Load .STL File

To make a program which can load .STL file we need to create certain steps:

- Go to Graphics and Sound
- 3D Picture Control=> Object => Create Object
- Again go to Graphics and Sound
- 3D Picture Control=> File Loading => Load STL Geometry
- Now use Application Control => Invoke node

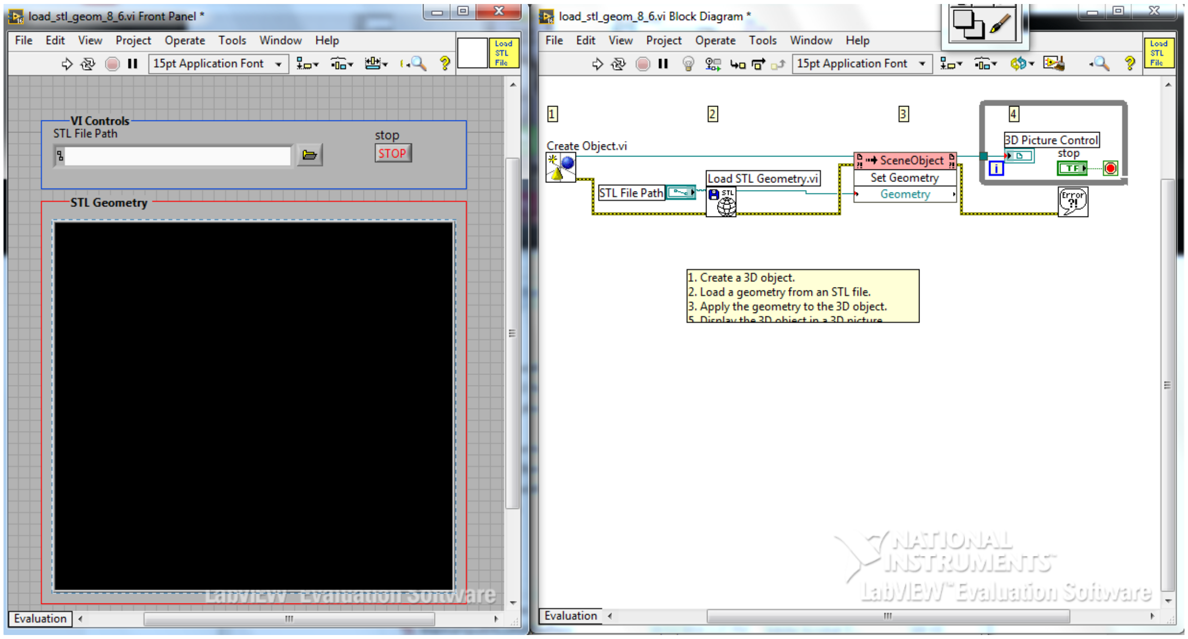


Figure 3.6: - STL file program

CHAPTER 4. LABVIEW AND ARDUINO

First of all install **VI PACKAGE MANAGER**. This is basically used for LabVIEW. Here we can find any VIs related with the software. The downloaded VI automatically integrates itself with LabVIEW**.**

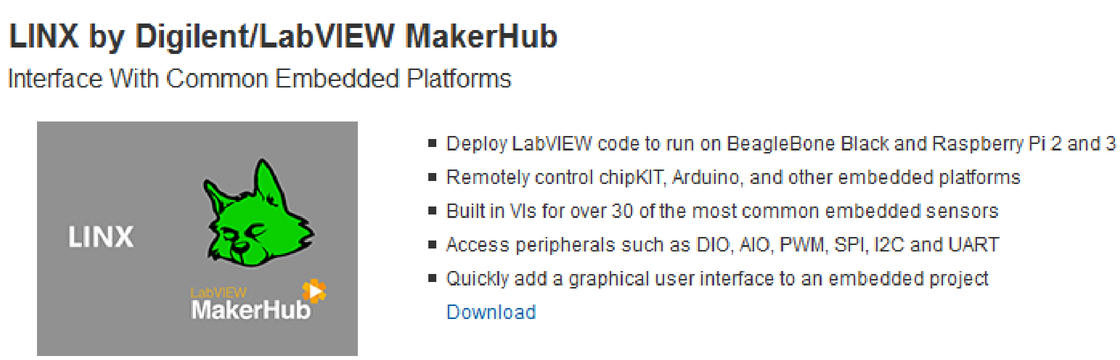


Figure 4.1: - Interface with common embedded platforms

4.1 Requirements for Labview

This 3D printer uses Arduino Mega 2560 board as a controller so there has to be compatibility between Arduino and LabVIEW. There are certain requirements for the LabVIEW to get integrated with the 3D printer. These are

- LabVIEW 2009+ or compatible
- NI-VISA Drivers
- NI-DAQmx 9.3.5 or compatible
- LabVIEW interface for Arduino VI package
- Arduino drivers

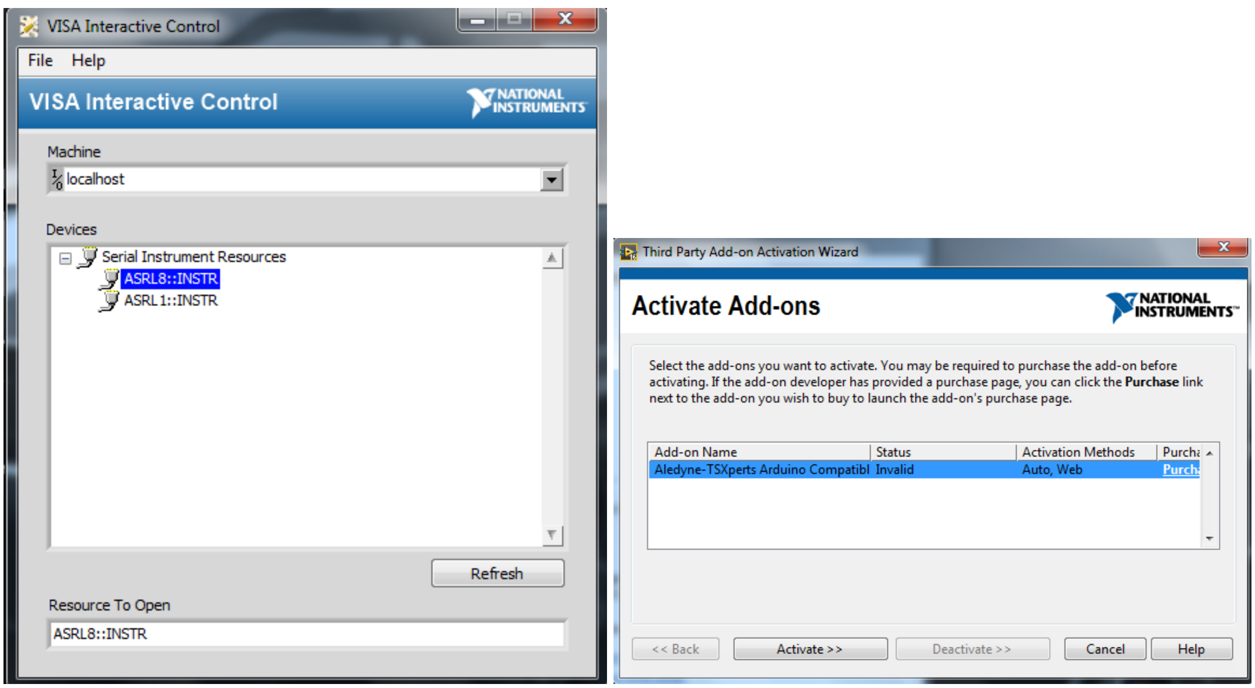


Figure 4.2: - VISA interactive control and compiler

4.2 Arduino Study

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

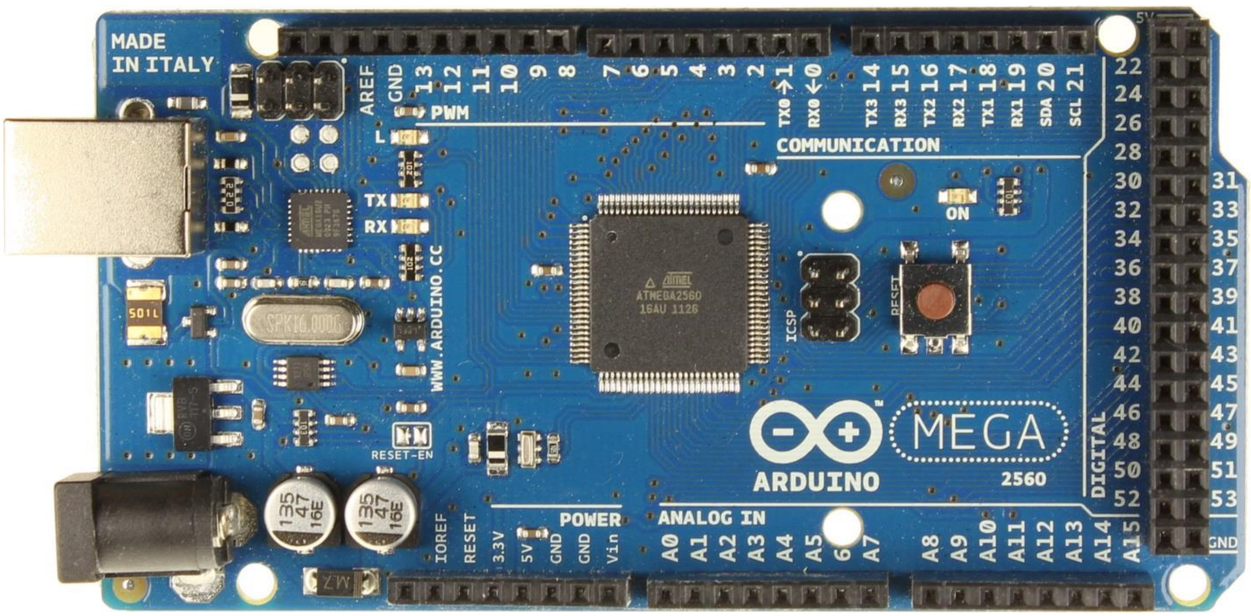


Figure 4.3: - Arduino Mega 2560

The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila. It is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing, MaxMSP).

Technical specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

The Mega 2560 board can be programmed with the Arduino Software (IDE). The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows us to upload new code to it without the use of an external hardware programmer. The Mega 2560 has a resettable polyfuse that protects the computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed. The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

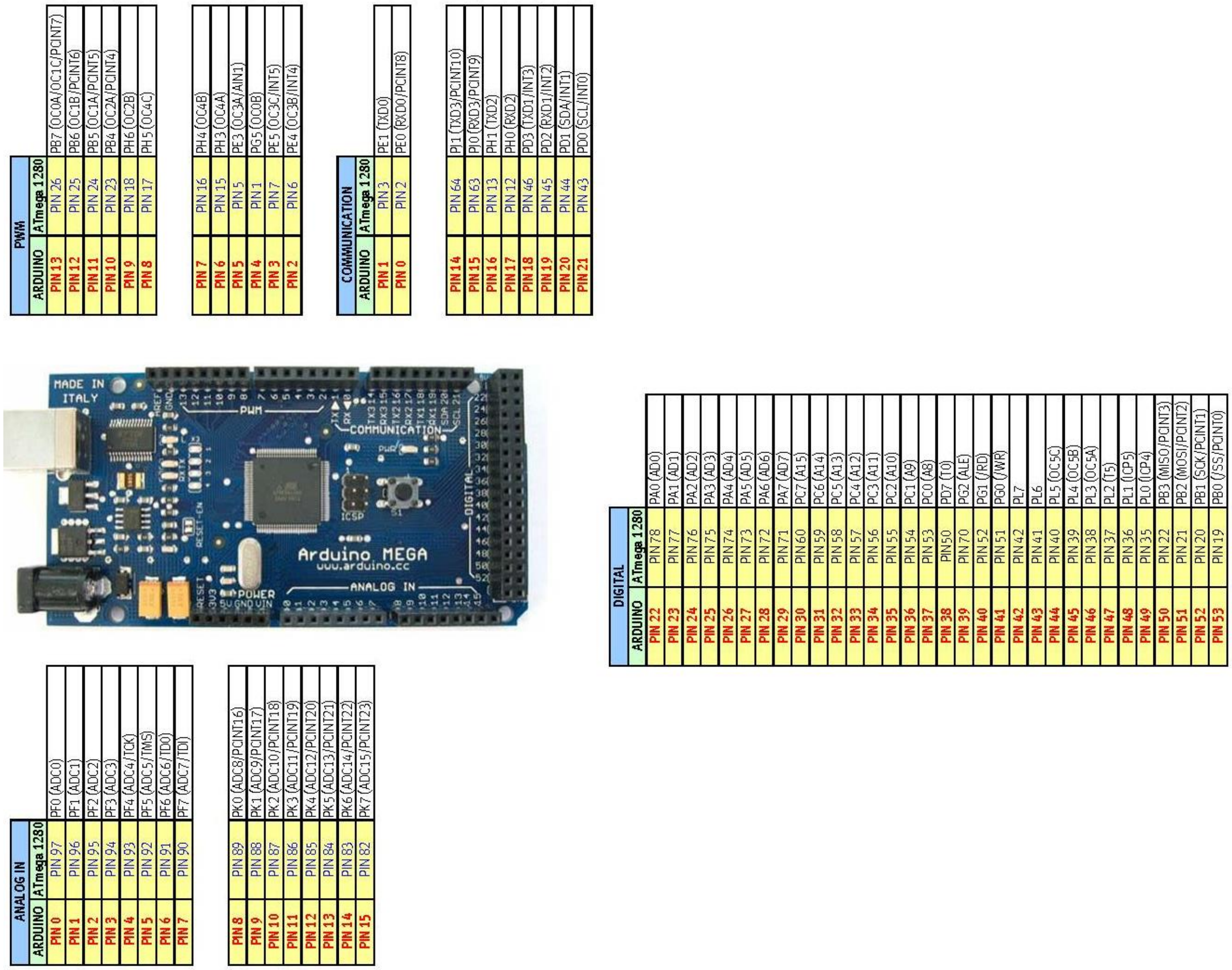


Figure 4.4: - (REF 3)Arduino Mega 2560 pin information

Once the printer had been fully assembled, all components had to be wired and tested to be sure they all functioned appropriately and had full functionality. The common open source 3D printing design is to use an Arduino Mega 2560 in combination with a RAMPS (RepRap Arduino Mega Pololu Shield) that is affixed to the top of the Arduino. There are also four stepper drivers that sit on top of the RAMPS to control the movement of each axis and the extruder.

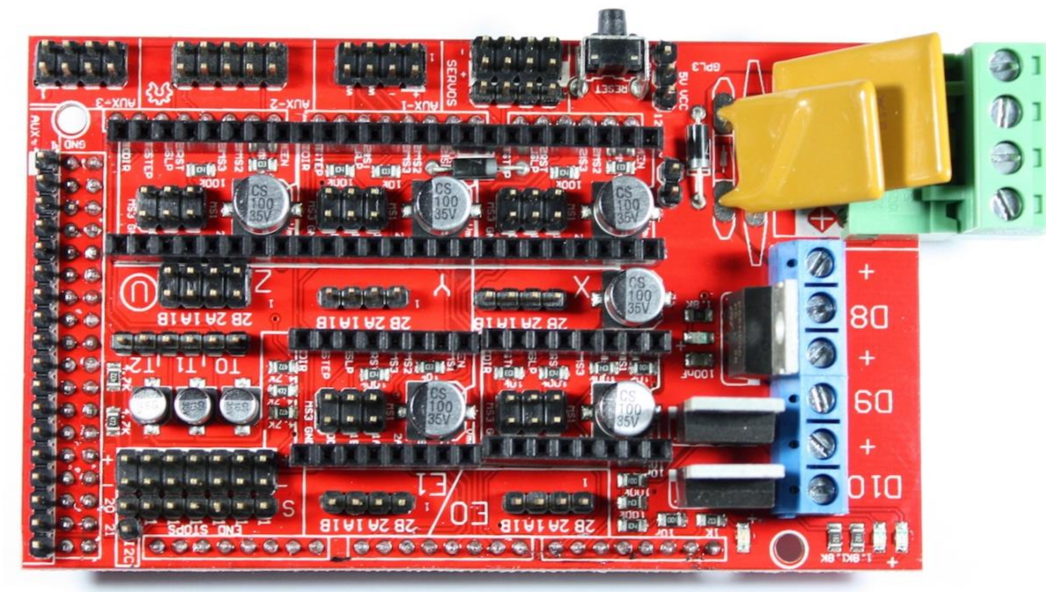


Figure 4.5: - RAMPS for arduino

4.3 Load Arduino Interface

- Open VI PACKAGE MANAGER and search for LabVIEW interface for Arduino VI package.

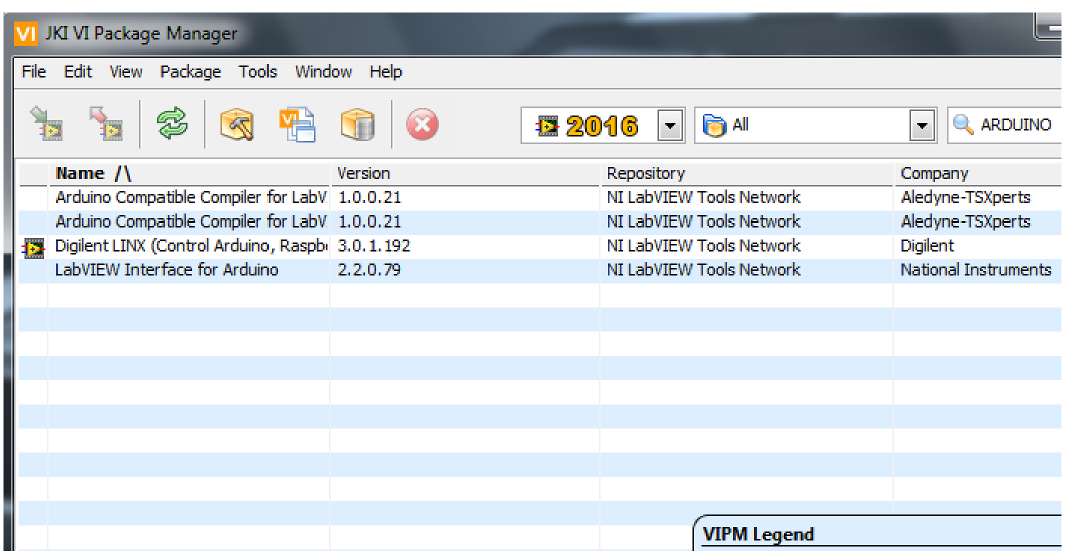


Figure 4.6: - Arduino interface download

Now we will get an ARDUINO tab under the Functions which consists of all the commands required for functioning.

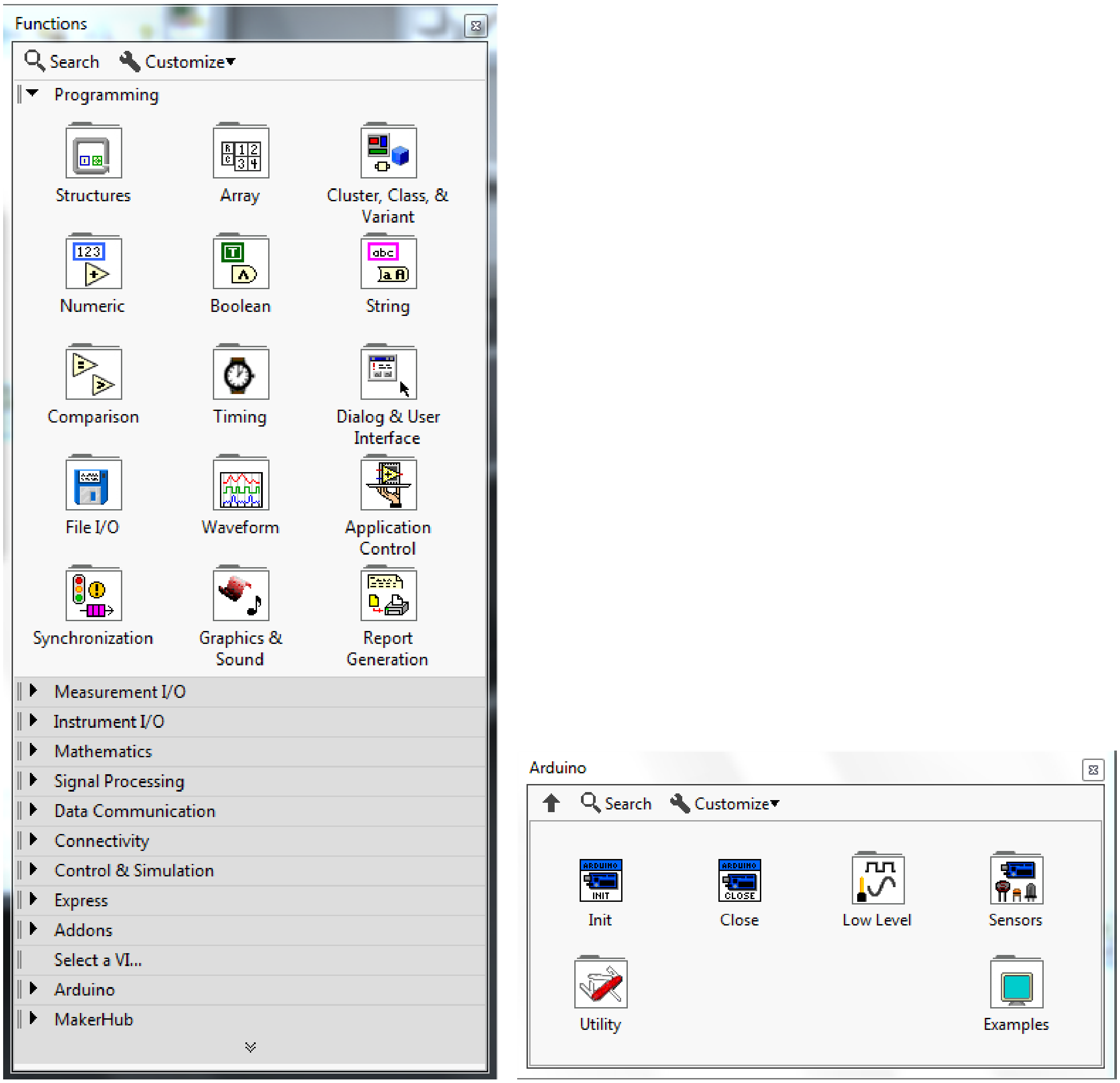


Figure 4.7: - LabVIEW interface of Arduino

Get the Arduino drivers updated and download Arduino software to feed the board with the required programming. The program can be acquired from the following directory: <u>*C:\Program Files (x86)\National Instruments\LabVIEW 2016\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA_Base*</u>

4.4 Labview Interface for Arduino (LIFA)

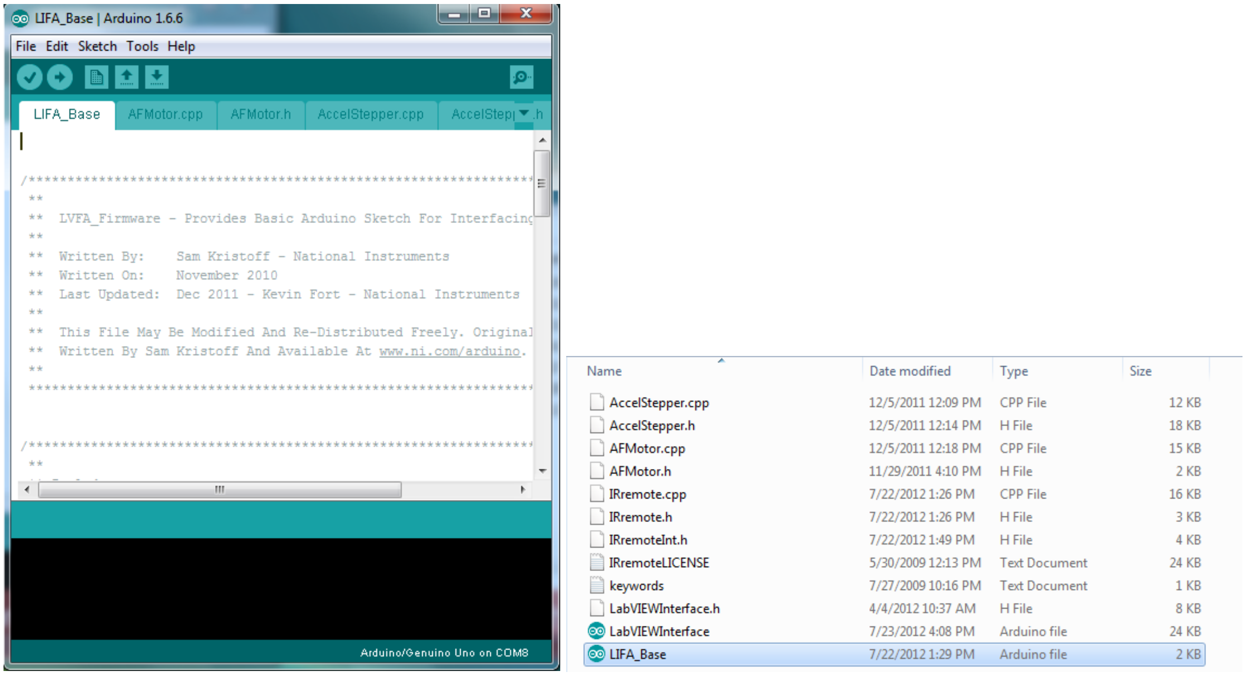


Figure 4.8: - LIFA_Base for Arduino

Try to upload the LIFA to the Arduino Board. If the error occurs, then reset the board from reset button and restart the program.

Write a program using LabVIEW interface for Arduino which is for only one motor. If the written program is not showing USB port which is basically VISA resource, then the program is not reading the right board. Most of the time it reads UNO instead of MEGA, change it manually will solve the issue. This problem will persist so restart the Labview as well as Arduino and again upload the whole program to check the issue.

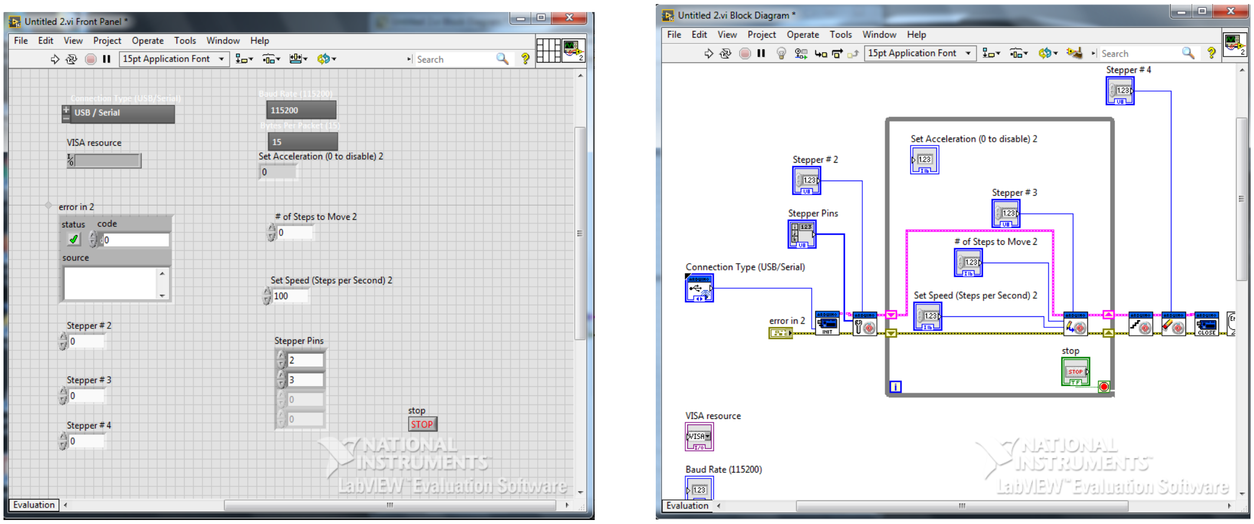


Figure 4.9: - Showing no USB connection

The figure 25 showing led example with LIFA. It works perfectly fine but to make it work with the Arduino we need to remove the RAMPS as it is not able to process signal from the Labview.

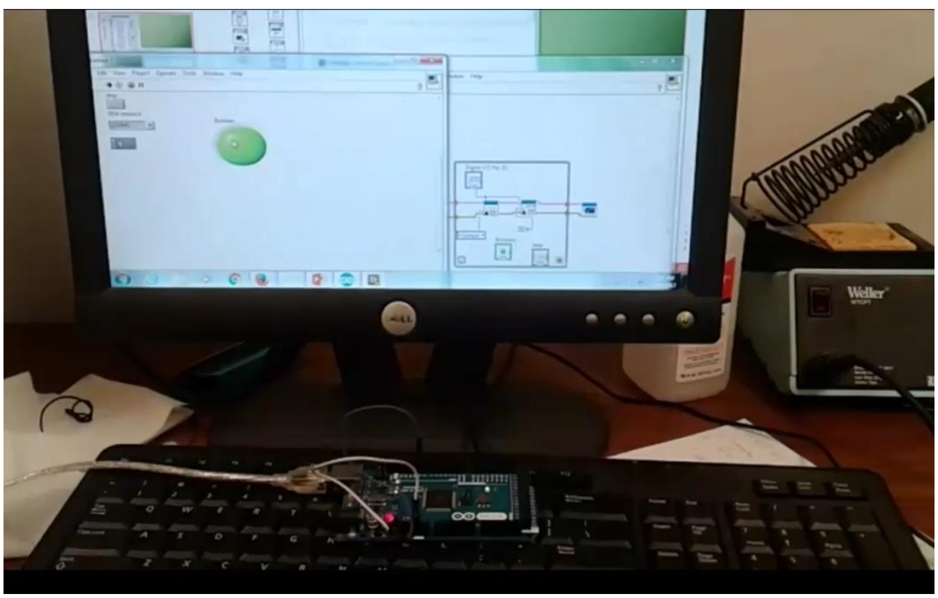
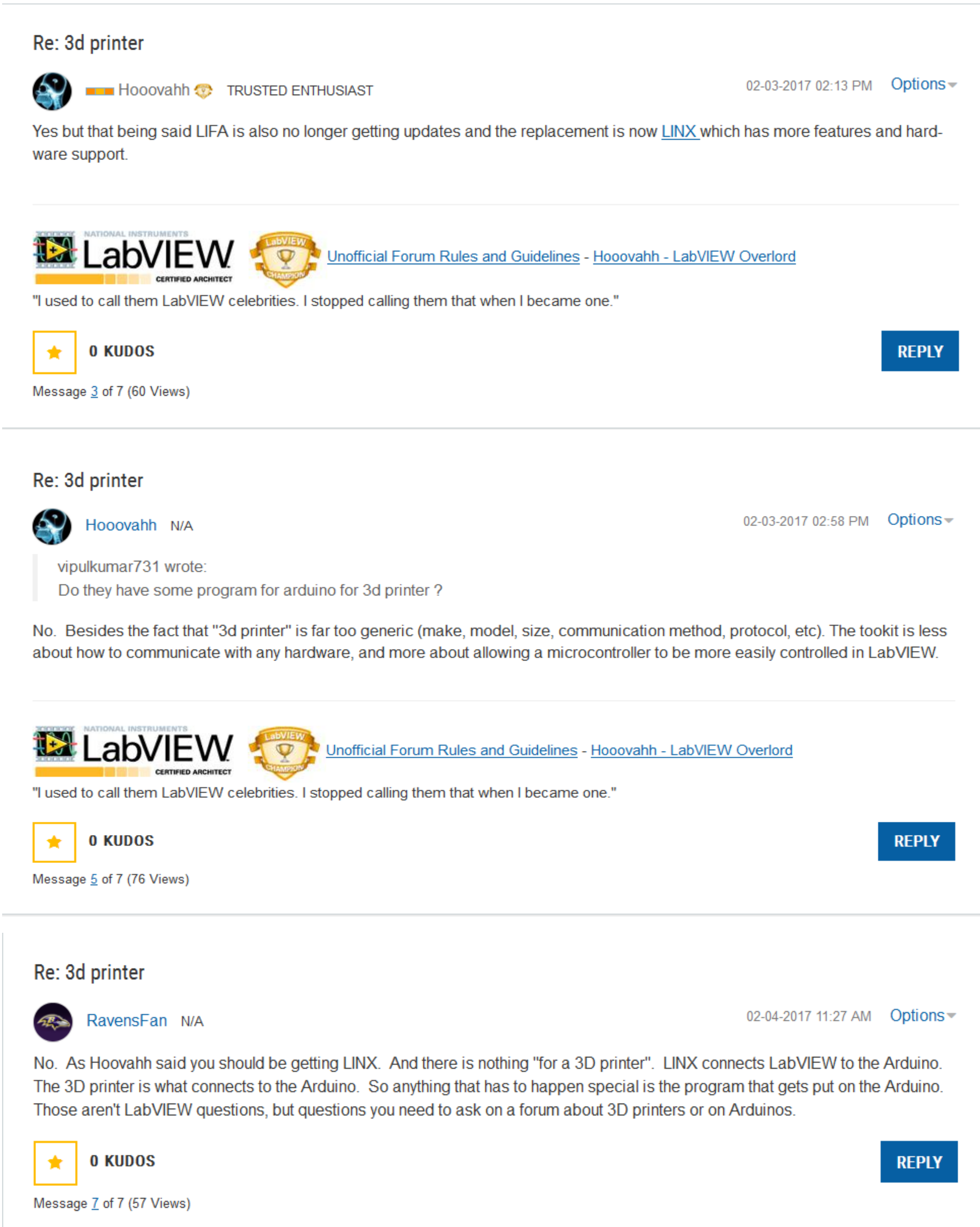


Figure 4.10: - LIFA led example

4.5 Problems with LIFA



Re: 3d printer

Hooovahh TRUSTED ENTHUSIAST 02-03-2017 02:13 PM Options

Yes but that being said LIFA is also no longer getting updates and the replacement is now LINX which has more features and hardware support.

Unofficial Forum Rules and Guidelines - Hooovahh - LabVIEW Overlord

"I used to call them LabVIEW celebrities. I stopped calling them that when I became one."

0 KUDOS REPLY

Message 3 of 7 (60 Views)

Re: 3d printer

Hooovahh N/A 02-03-2017 02:58 PM Options

> vipulkumar731 wrote:
> Do they have some program for arduino for 3d printer ?

No. Besides the fact that "3d printer" is far too generic (make, model, size, communication method, protocol, etc). The tookit is less about how to communicate with any hardware, and more about allowing a microcontroller to be more easily controlled in LabVIEW.

Unofficial Forum Rules and Guidelines - Hooovahh - LabVIEW Overlord

"I used to call them LabVIEW celebrities. I stopped calling them that when I became one."

0 KUDOS REPLY

Message 5 of 7 (76 Views)

Re: 3d printer

RavensFan N/A 02-04-2017 11:27 AM Options

No. As Hoovahh said you should be getting LINX. And there is nothing "for a 3D printer". LINX connects LabVIEW to the Arduino. The 3D printer is what connects to the Arduino. So anything that has to happen special is the program that gets put on the Arduino. Those aren't LabVIEW questions, but questions you need to ask on a forum about 3D printers or on Arduinos.

0 KUDOS REPLY

Message 7 of 7 (57 Views)

Figure 4.11: - Labview forum describes LIFA problems

4.6 LINX Interface

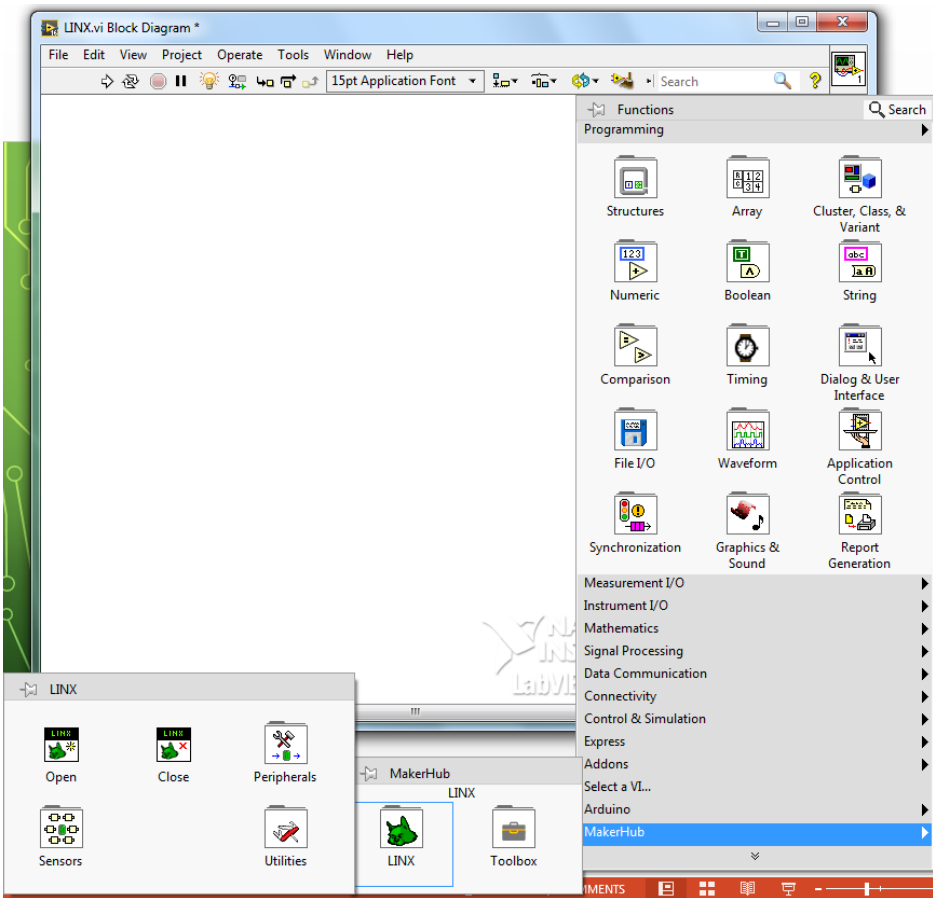
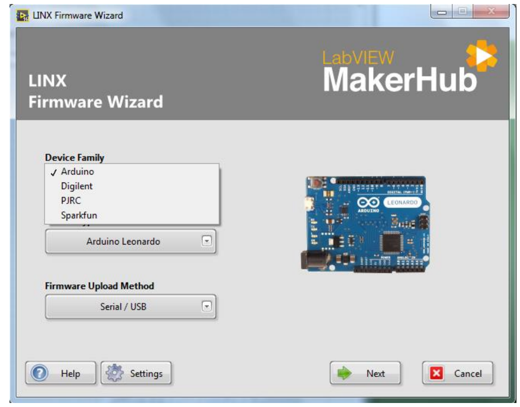
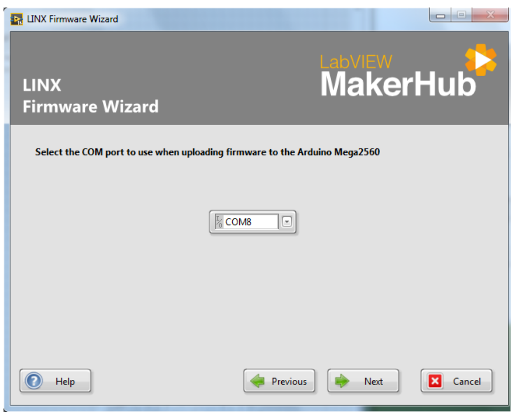


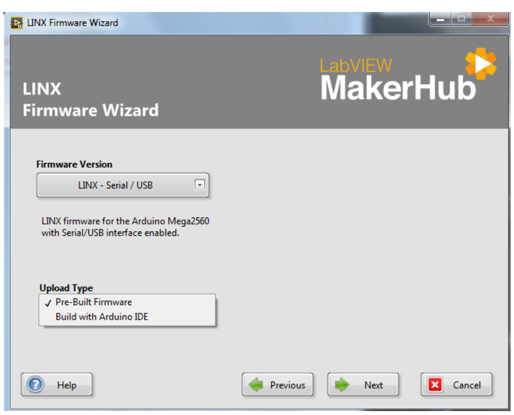
Figure 4.12: - linx interface

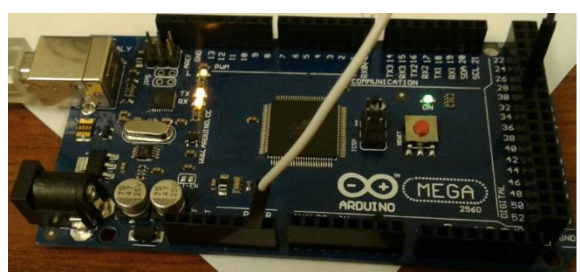
LINX provides easy to use LabVIEW VIs for interacting with common embedded platforms like Arduino, chipKIT and myRIO. Use the built in sensor VIs to start getting data to your PC in seconds or use the peripheral VIs to access your devices digital I/O, analog I/O, SPI, I2C, UART, PWM and more. Whether you're remotely controlling a chipKIT or Arduino over USB/Serial, Ethernet or Wi-Fi, or deploying VIs to run on BeagleBone Black or Raspberry Pi 2/3, LINX and LabVIEW make it easy visualize the data you're working with, debug your code, and create advanced embedded applications faster than ever before.

4.7 Uploading LINX Firmware









4.8 LINX Led Example

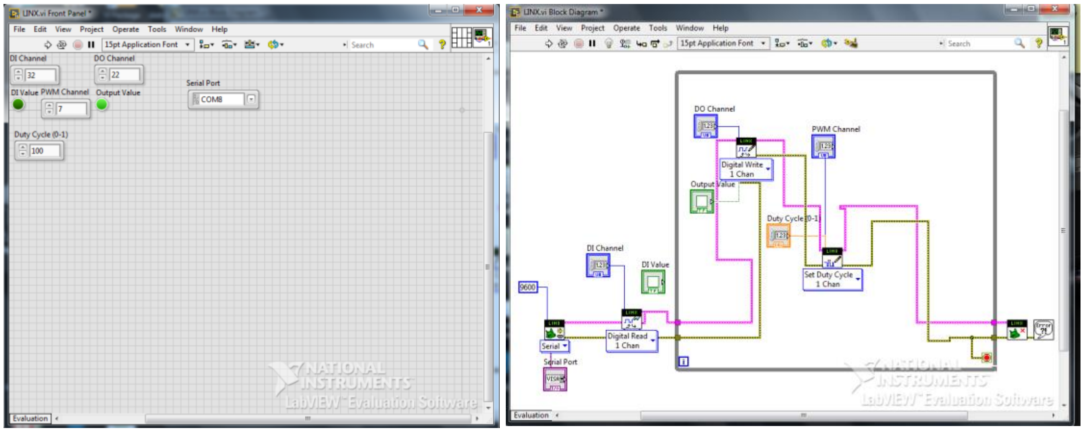
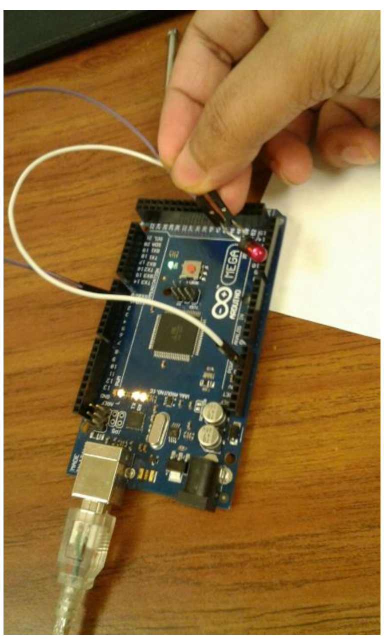
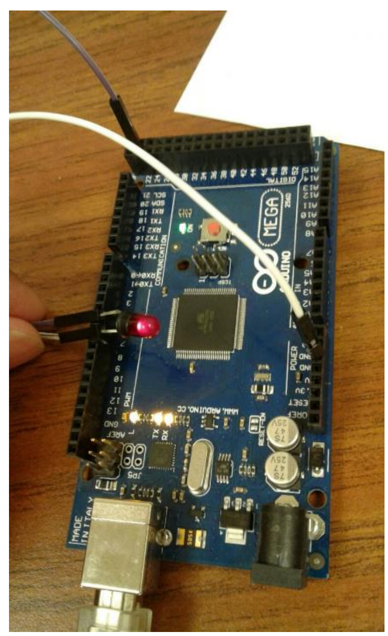


Figure 4.13: - linx led example

This example shows the difference between PWM Signal and Digital Signal using LINX in the Labview. LINX just recognizes the pin I/O, the program needs to be given the I/O pins and needs to be given the signal type.



PWM Signal



Digital Signal

4.9 LINX Problem

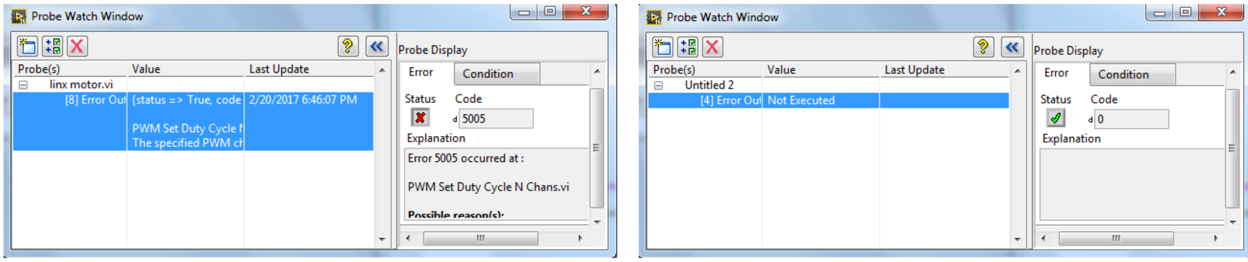


Figure 4.14: - showing linx problem

It is unable to establish connection with Arduino. Most of the time it doesn't recognize the board. Every program that uses the LIFA Toolkit, should begin with the init command end with the close commands. It has to be ensured to avoid use of the abort button as much as possible. While aborting a task, the code will not have a chance to run the close VI, and the next time code runs, the Arduino will start up in an unknown state. It may need to restart the board and re-load the LIFA firmware to correct this.

4.10 Debugging

Using a different program to check the authenticity. This program is for one motor only with a highlighted error showing function.

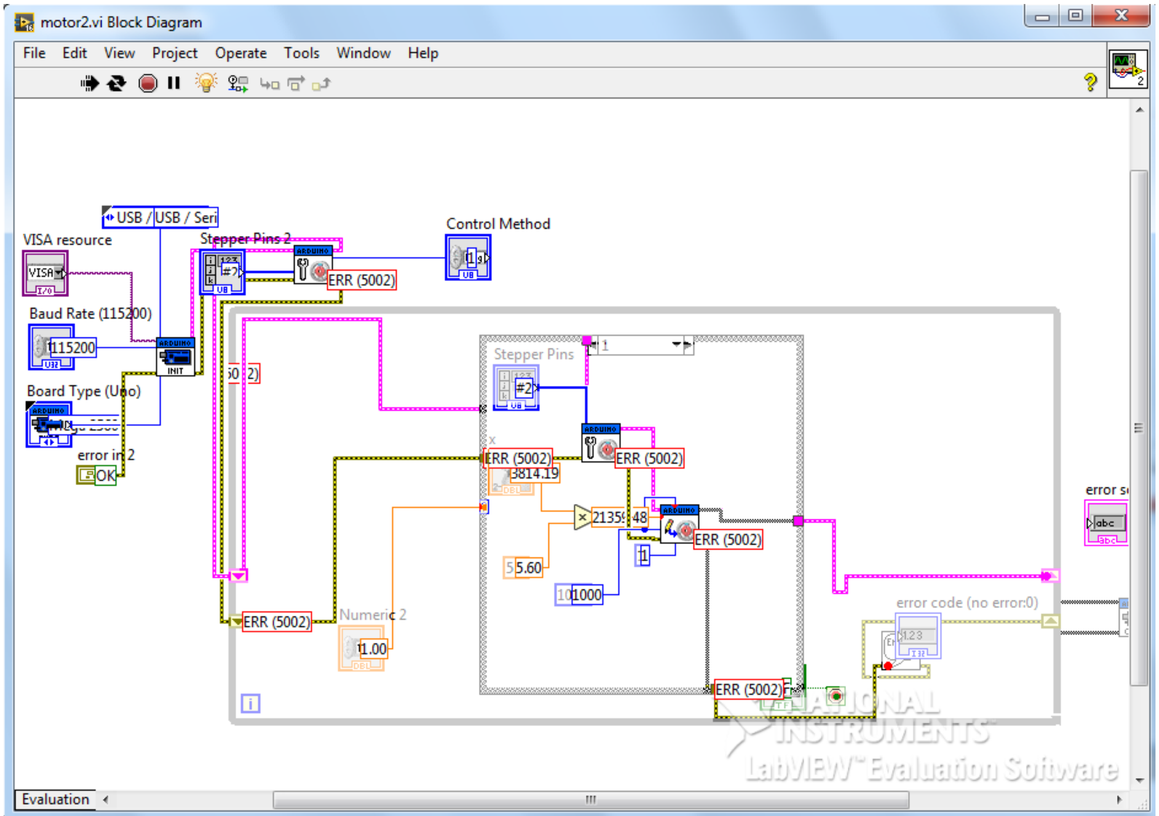
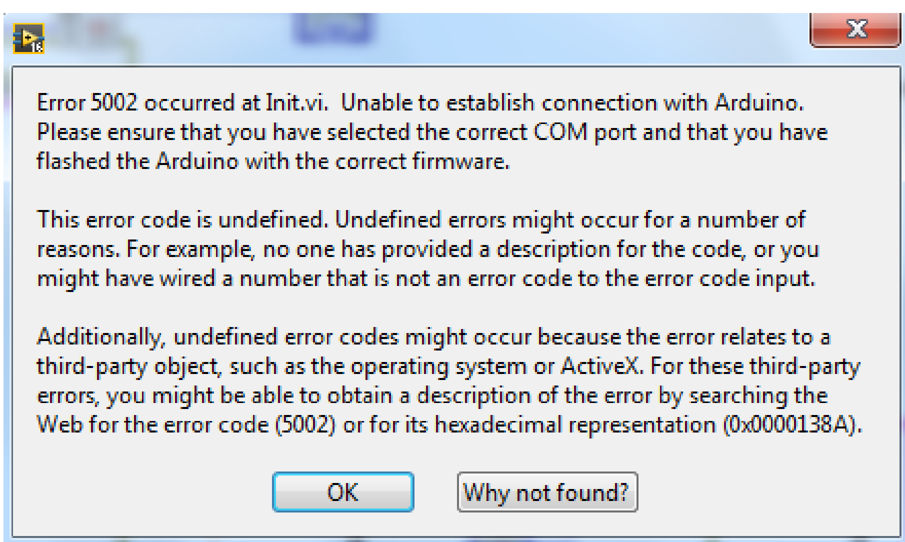


Figure 4.15: - Showing error 5002

This program is basically showing error 5002 which means that there is no establish connection with Arduino and selected the COM port is not right.



This error may be from the following reasons:

- ✓ Make sure that the correct firmware is flashed to the Arduino. The correct firmware can be found in <LabVIEW>\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA_Base.
- ✓ Make sure we have selected the correct COM port for the Init.vi or that the VI automatically locates the Arduino (this happens when no wire is connected to the VISA Resource input of the Init.vi and only works for Arduinos that are connected to the PC via USB). If we are having trouble with auto-detect mode, take the Init.vi out of auto-detection mode by wiring up the Arduino VISA resource and baud rate. Right-click on the top left terminal of the Init.vi in and select **Create»Constant.** This will create a VISA constant. Select the COM port where Arduino is connected to from the drop down menu. The correct COM port to use will be shown in the Arduino Integrated Development Environment (IDE) when we flash the LabVIEW Interface for Arduino (LIFA) firmware to the Arduino board. Right-click on the terminal underneath the VISA resource and select **Create»Constant** again. In this box set the baud rate for the Arduino. The default for most cases will be 115200.
- ✓ Check to see if the Arduino shows up under Windows Device Manager. Search Device Manager on the computer and run the executable. Expand the USB section and see if the Arduino is listed. If it is not, we may need to install the drivers for the Arduino. See the External Link: Getting started w/ Arduino on Windows for steps on doing this.
- ✓ If using a wireless connection (Xbee, BlueSMIRF) to LabVIEW try using a wired connection (USB, Serial). Poor wireless signal strength can cause the Arduino to be unreachable by LabVIEW.
- ✓ Make sure that the baud rate specified as an input to Init.vi matches the baud rate specified in the firmware by #define DEFAULTBAUDRATE X (where X is the baud rate).
- ✓ Make sure we have the latest version of NI VISA installed on the computer. We can check this by opening NI Measurement & Automation Explorer (NI MAX). Search NI MAX on the computer and run the executable. Expand the software section and verify that NI VISA is listed there. If it is not listed, download the latest version of NI-VISA from NI Driver Downloads.
- ✓ Every program that uses the LIFA Toolkit, should begin with the init command end with the close commands. Ensure that we avoid use of the abort button as much as possible. When we abort a task, the code will not have a chance to run the close VI, and the next time we run the code, the Arduino will start up in an unknown state. We may need to restart the board and re-load the LIFA firmware to correct this.

✓ Make sure that no other programs are open that are also attempting to communicate with the Arduino. If we have the Arduino IDE open, for example, the resource may already be reserved when LabVIEW tries to access the COM port.

After fixing the error 5002, using a different approach to solve the problem. The error occurred at the Arduino resource.

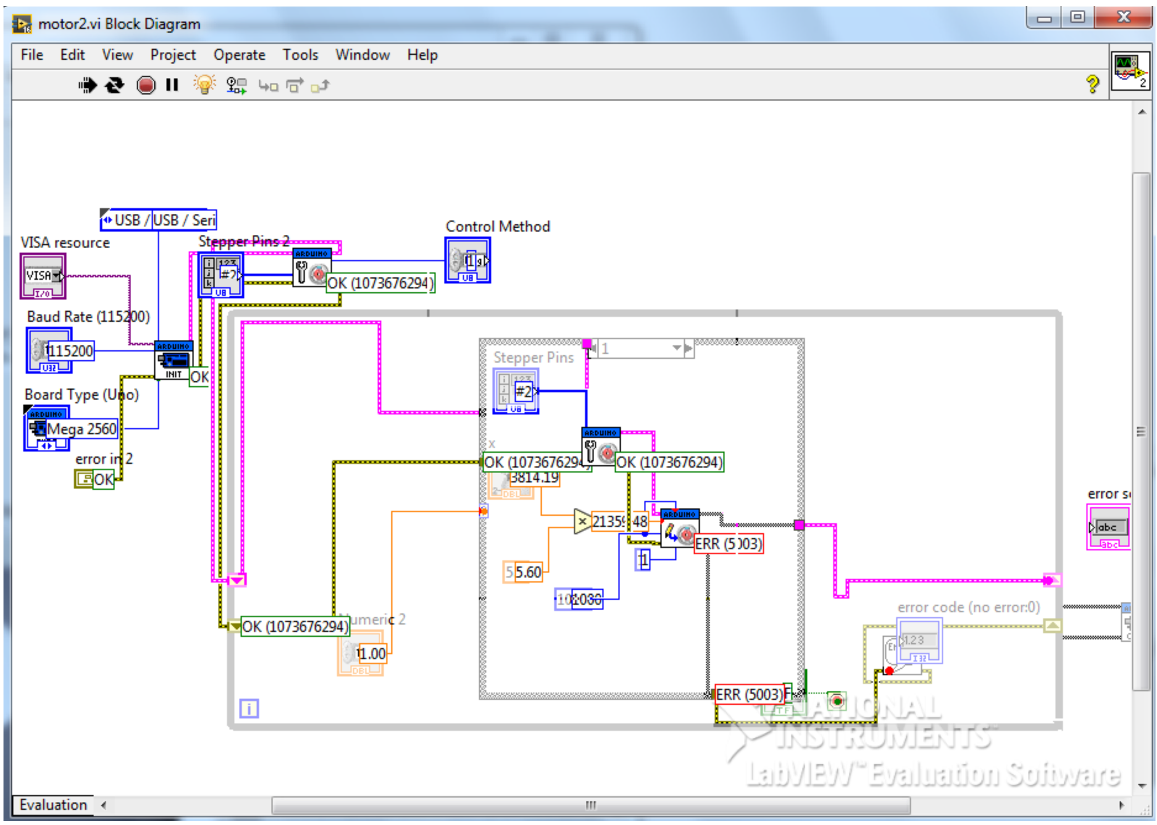
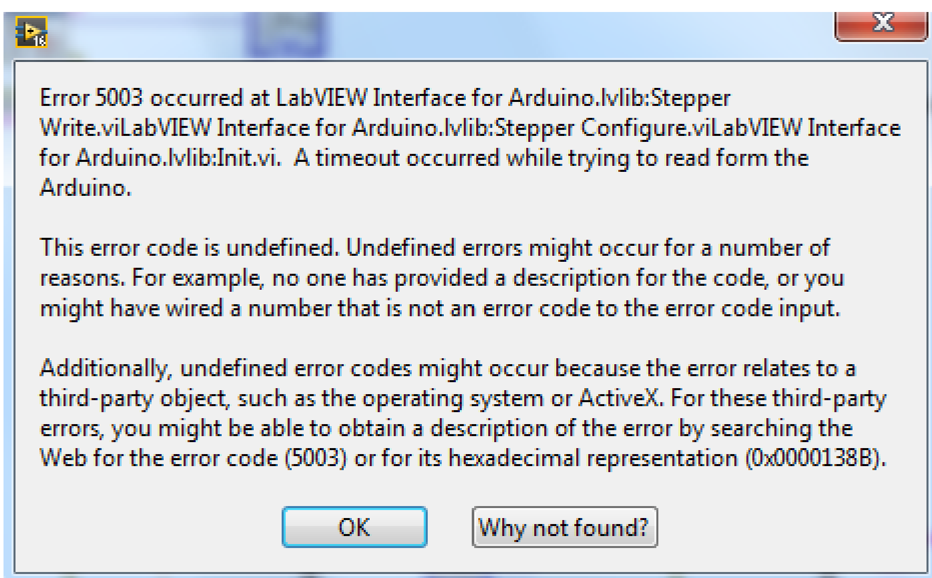


Figure 4.16: - Showing error 5003

Now this error occurred in the loop, where it cannot define stepper motor. This error may be due to certain reasons:

✓ Ensure that all cables are securely attached if using a wired connection or that we have adequate signal strength if using a wireless connection. The USB ports located on the back of a desktop computer generally have more stable power supply than the front panel's USB ports. We may want to use those ports if we are on a desktop computer. If we are using a laptop, make sure it is plugged in during the troubleshooting process.

- ✓ Check to make sure we have the correct baud rate and COM port specified for the Arduino board in order to properly define the communication protocol. Failure to specify the right settings may result in this error.
- ✓ Every program that uses the LabVIEW Interface for Arduino (LIFA) Toolkit, should begin with the Init command and end with the Close command. Ensure that we avoid the use of LabVIEW's Abort button as much as possible. When we abort a task, the code will not have a chance to run the Close VI. The next time we run the code, the Arduino will start up in an unknown state that could result in Error 5003. We may need to restart the board and re-load the LIFA firmware to correct this.
- ✓ Make sure that no other open programs are also attempting to communicate with the Arduino. For example, having the Arduino Integrated Development Environment (IDE) open may reserve the Arduino resource, resulting in Error 5003 when LabVIEW tries to access the COM port.
- ✓ We can try to open a built-in example program that interacts with the same I/O pins to see if we still get Error 5003. This will ensure the Arduino is actually capable of communicating with the computer.
- ✓ We can manually change the timeout time or the number of timeout retries by modifying the Time Out and Max Retries input terminals of the Send Receive function. Having a longer timeout value or a higher number of attempts (retries) may establish communication with the device.



After fixing this code we get an OK code 1073676294, which states that the program is running fine without any trouble.

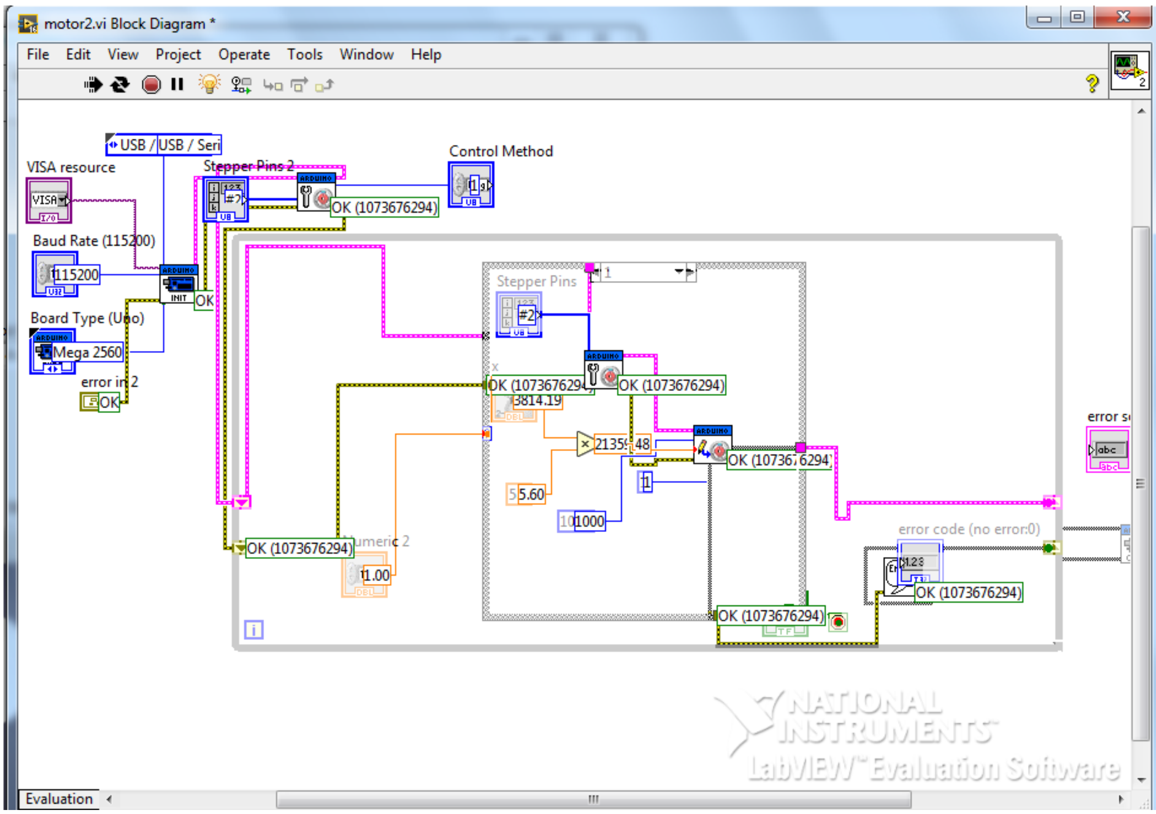


Figure 4.17: - Showing OK code 1073676294

This shows a response from Arduino board where the processing light is on which shows that it is getting command from LabVIEW.

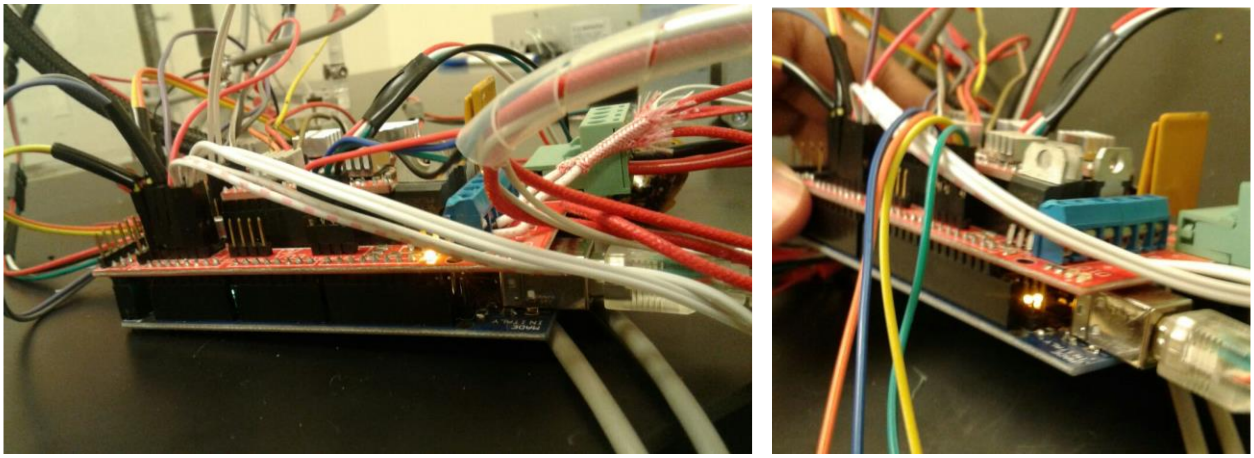


Figure 4.18: - Showing processing lights (TX, RX)

4.11 Solution

Taking the schematic of stepper motor drive and trying to make the circuit. The ground and VMOT needs to be connected with motor power supply while the step and direction can directly be manipulated through Arduino. Choose any of the pin set for stepper motor and put the reference in the Labview. The Labview will automatically detect the pin and give power to the motor.

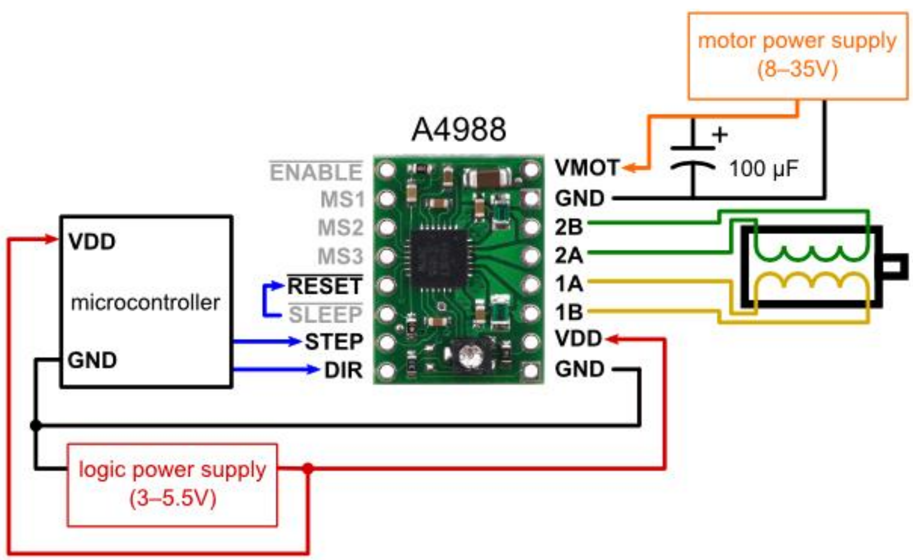


Figure 4.19: - (REF 5) -Showing schematic of stepper motor drive

4.12 Preferred Circuit Design

The preferred circuit design will show how the Labview will be running the stepper motors. The steps and direction to the each stepper can be given directly through the Labview by selecting the pin number.

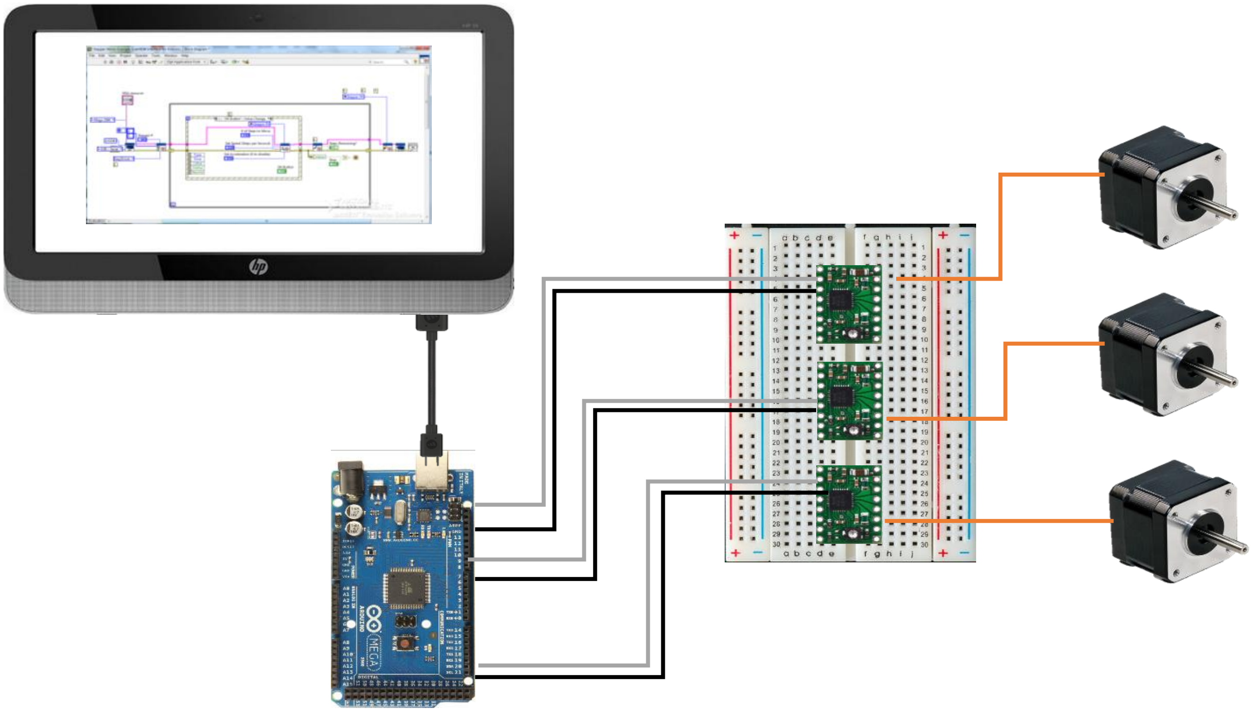


Figure 4.20: - circuit showing stepper run by labview

Our Project needed servo motors (Pittman 4224) to be run through Labview. They are very precise motors with encoders on them. Labview can run different motors including servos but they require particular microcontrollers for their working. There is a 3 wire servo example in figure 36 which shows working of servo with arduino.

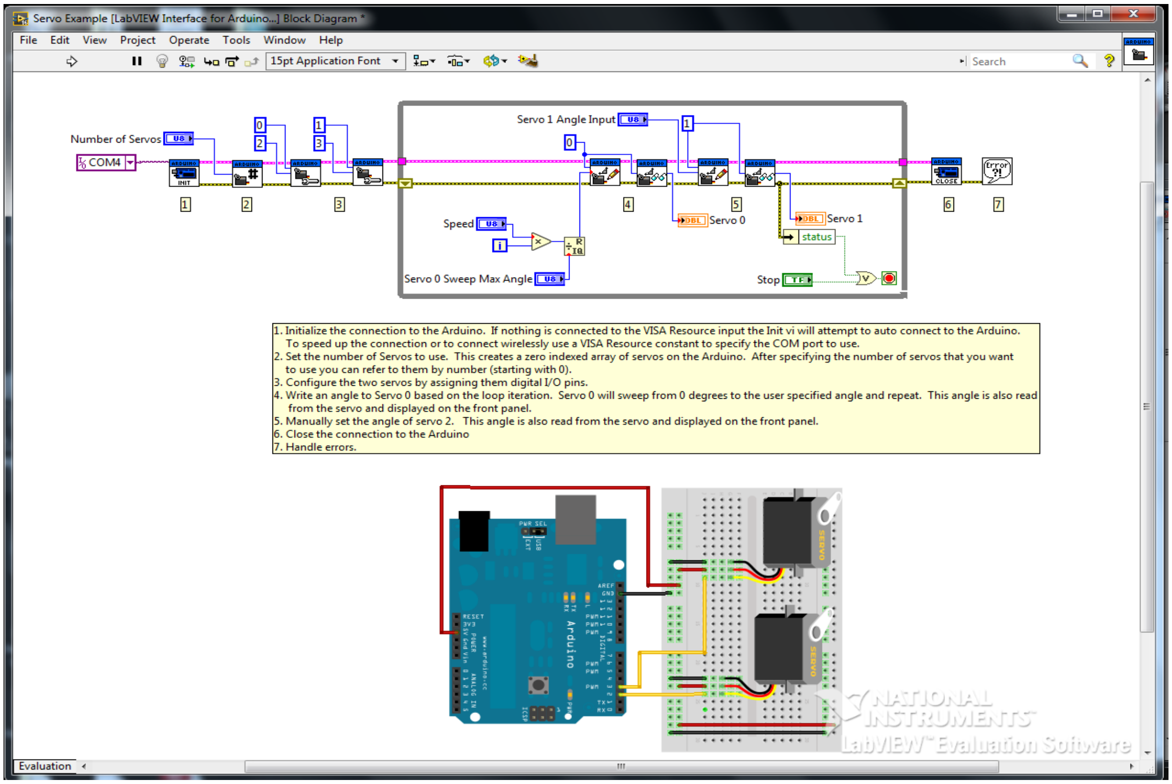


Figure 4.21: - 3 WIRE SERVO MOTOR EXAMPLE

The Labview can control different aspects which can include firmware, G-Code interpreter and interface. To work all these together, Labview needs a powerful G-code interpreter which slices the model and a firmware will receive all these data to be interpreted by an interface.

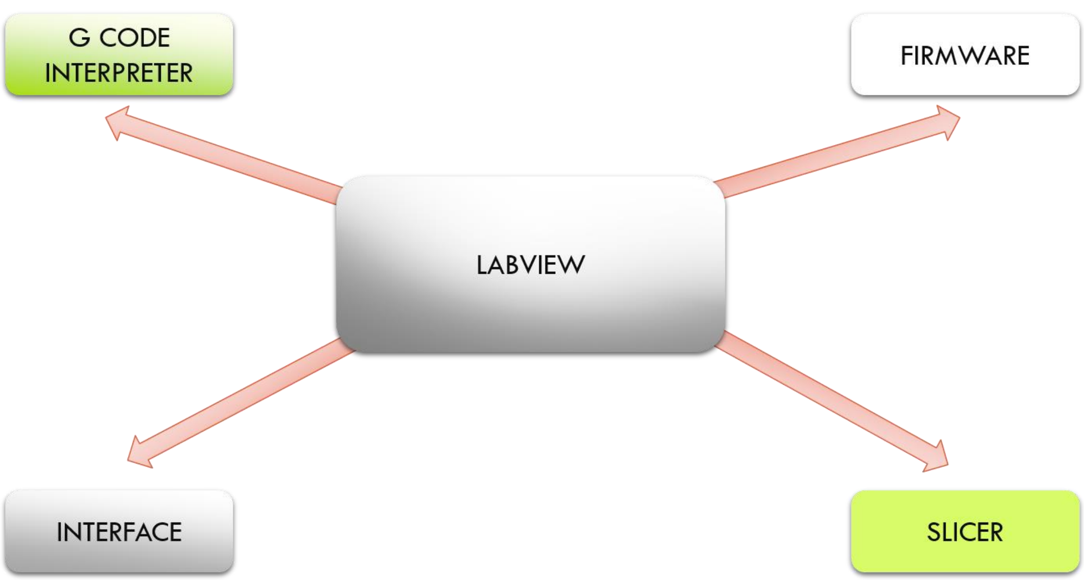


Figure 4.22: - Labview controlling different aspects

Working Example

There is a working example used by a company called YASKAWA. They uses their own interface for the servos. The drivers are used to control the servo which has an inbuilt encoders.

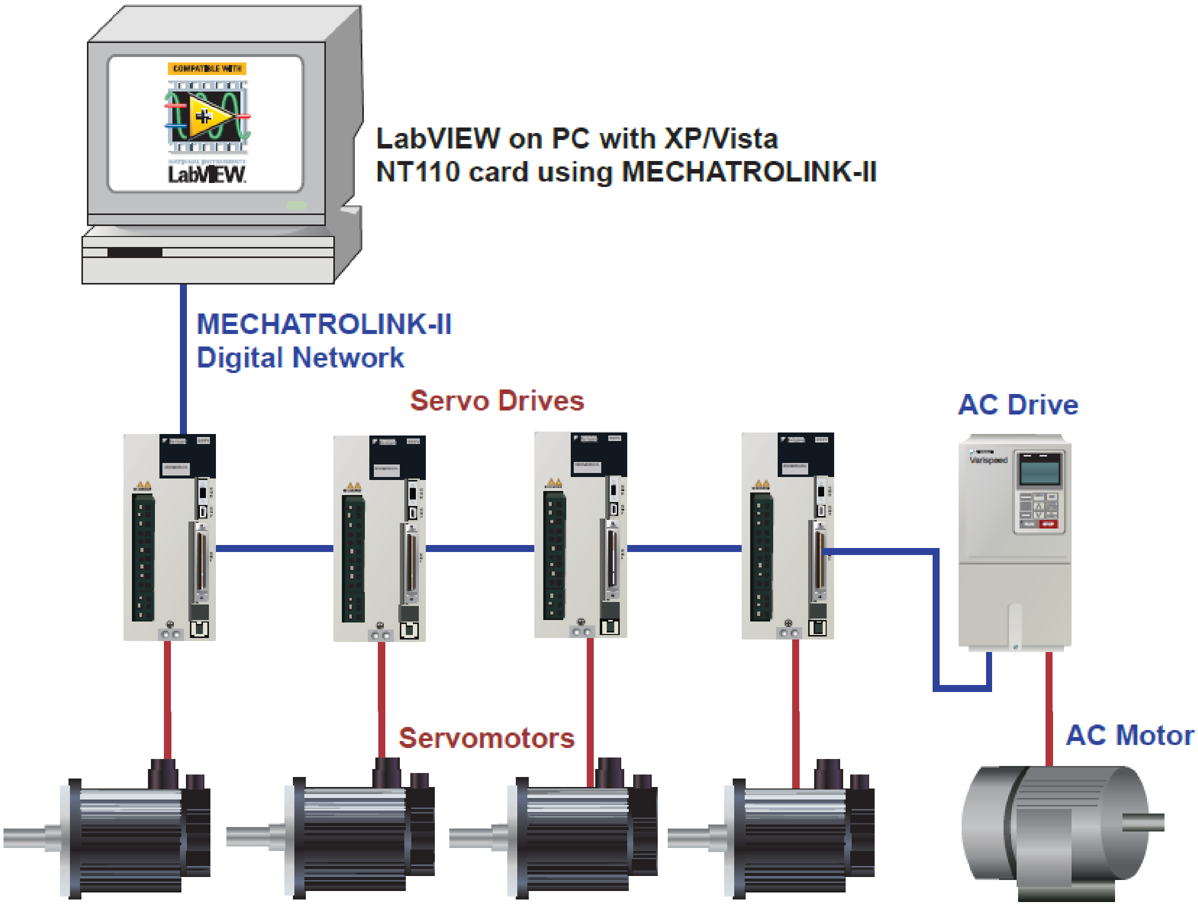


Figure 4.23: - yaskawa servo drives

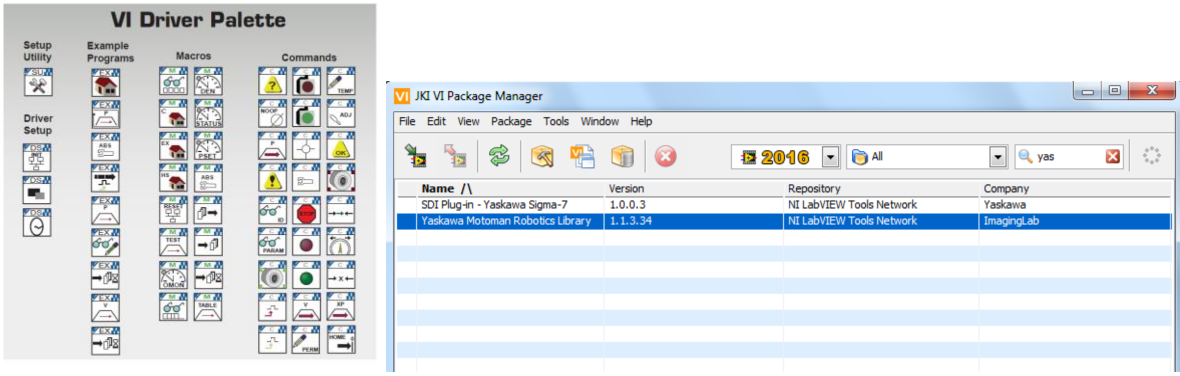


Figure 2.24: - yaskawa driver palette

The graphical representation of the process through Labview. This can be done through a firmware and using the proposed microcontroller.

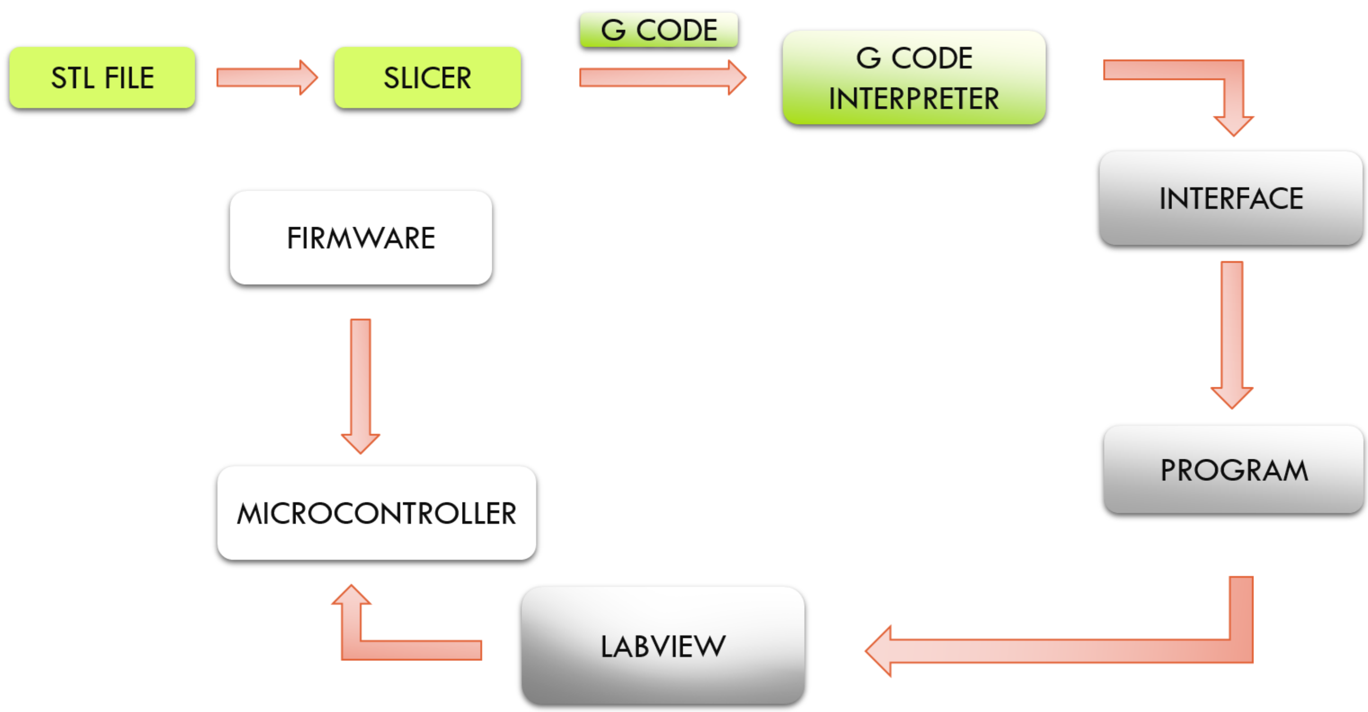


Figure 2.25: - Process through Labview

As the process is very rigid we will be using MACH3 to run the servo. The MACH3 uses a breakout board which breaks the signal and is received to the DC Servo Drive. The servo drive receives all the information of encoders and then signal is transferred through the line differential to the motors. This whole complex system is run through MACH3 via inbuilt G-Code Interpreter which is shown in figure 44.

CHAPTER 5. INTEGRATING LABVIEW WITH OTHER SOFTWARES

It is possible to integrate other softwares with Labview system. The software which will be used to control the moving table is MACH3. The Labview can integrate MACH and slicer within itself so it provides an easy access to the software. MACH3 uses its own microcontroller to control servo motors.

5.1 HDBB board

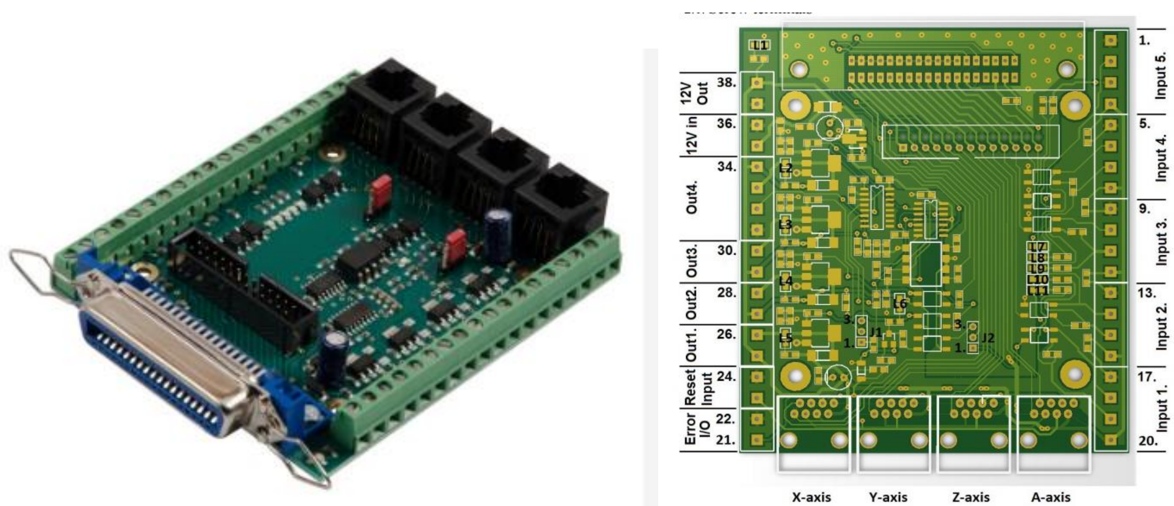


Figure 5.1: -(REF-13) HDBB board pinout

The HDBB2 is a signal breakout board making the wiring and connections to the LPT port or motion controller easy. There are 4 pieces of axis connectors to connect to our servo drives directly with straight wired „patch cables". All inputs connected via onboard optical isolator. 4 outputs have open-drain high power transistor outputs for driving external relay coils, solenoids, etc. directly upto 50Volts. All in and outputs have LED indicators which makes the installation and connections debugging easy. The board integrates a safety chargepump circuit which can be disabled/enabled with jumper. One analog 0-5V output is also onboard for software handled spindle speed control.

5.2 Servo Drivers

Whale3 Servo Drive is an interface between breakout board and high precision moving table. It receives step signals from breakout board and sends to moving table and position feedback through encoders and differential line drive modules to be mounted in hardware box/panel.

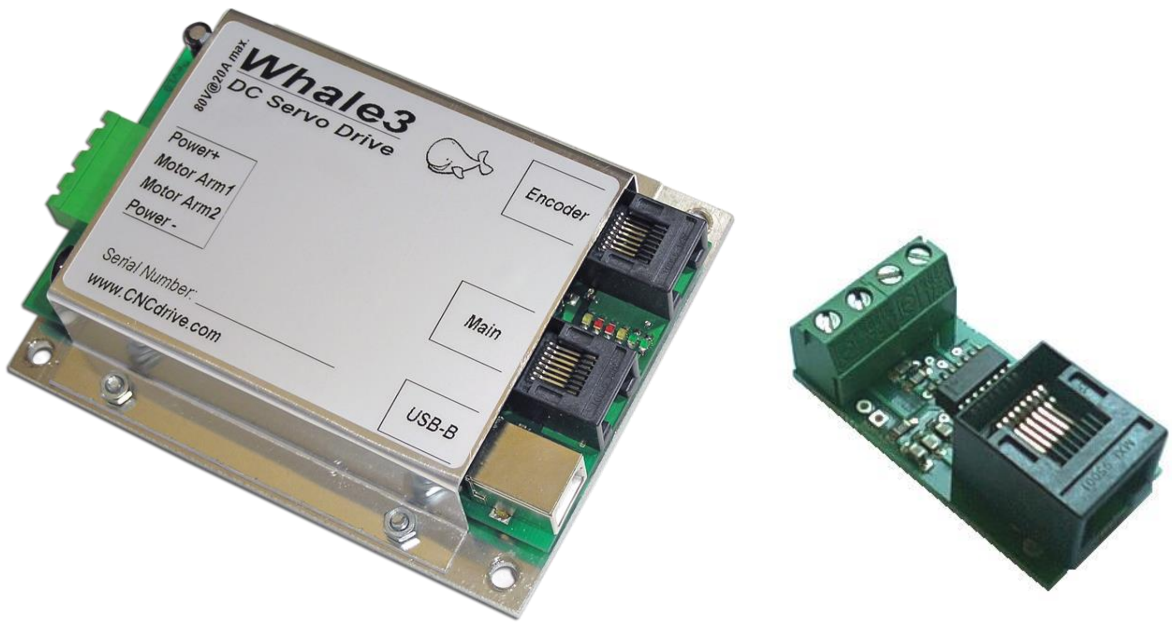


Figure 5.2: -(REF-13) servo drivers

The each Whale3 servo drives are configurated through a software called servoconfigurator. This configurator is used to position and check the working of servo drives. If it does not recognizes the servo drives, then this is a clear indication that the servo drives are faulty.

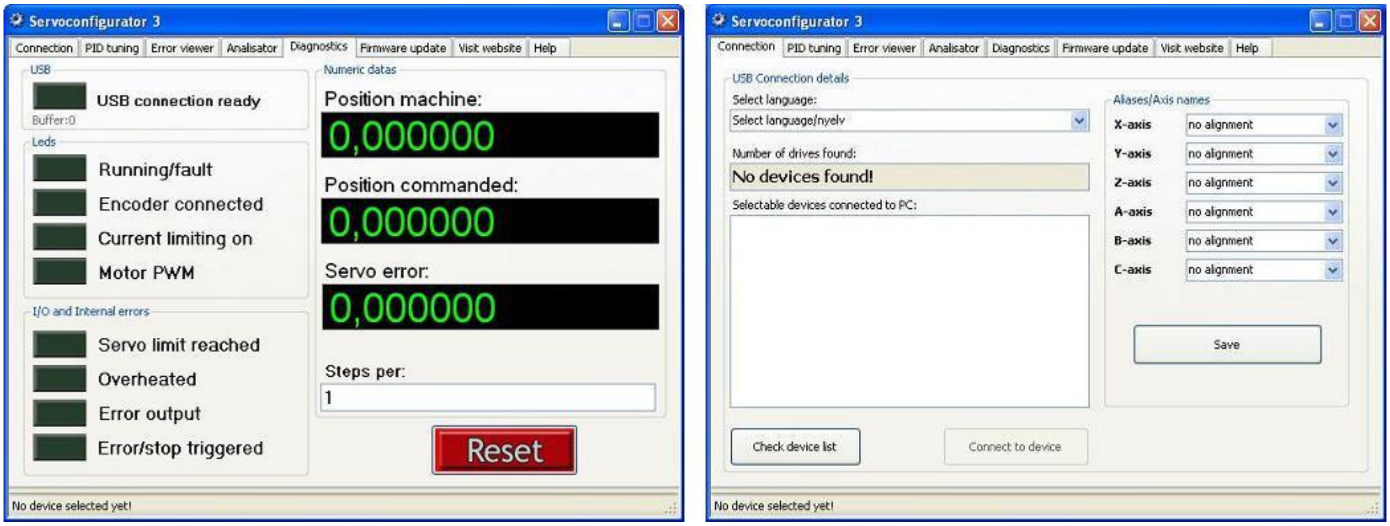


Figure 5.3: - servo configurator

Each PID control system was tuned to provide optimal performance for micron scale. The maximum error was never higher than 10 units. Each unit corresponds to 0.5 microns of translation.

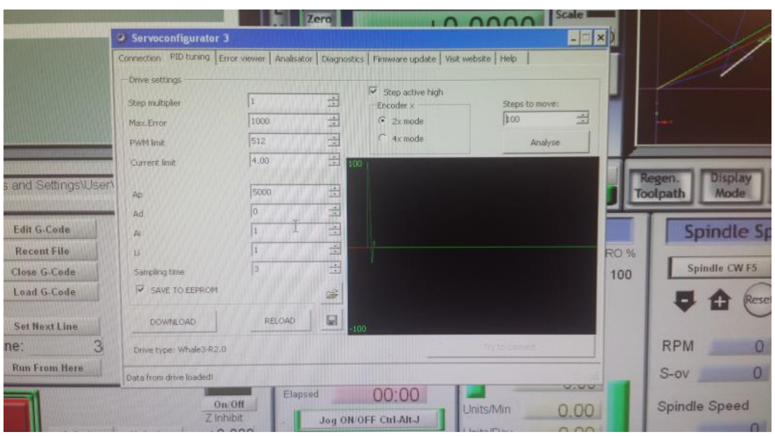


Figure 5.4: -PID control

The MACH3 uses a G-Code interpreter to run a program. This interpreter gives the step and direction to the motor which can be seen in tool display. Servoconfigurator used mainly for primary tuning of the servo motors.

This program provides position and error feedback to the user as well as status indicators. It contains a built in PID controller for variability in the control parameters (Kp, Ki, Kd) which can be directly changed via USB-B into the Whale3 (servo drive) and USB-A into the computer.

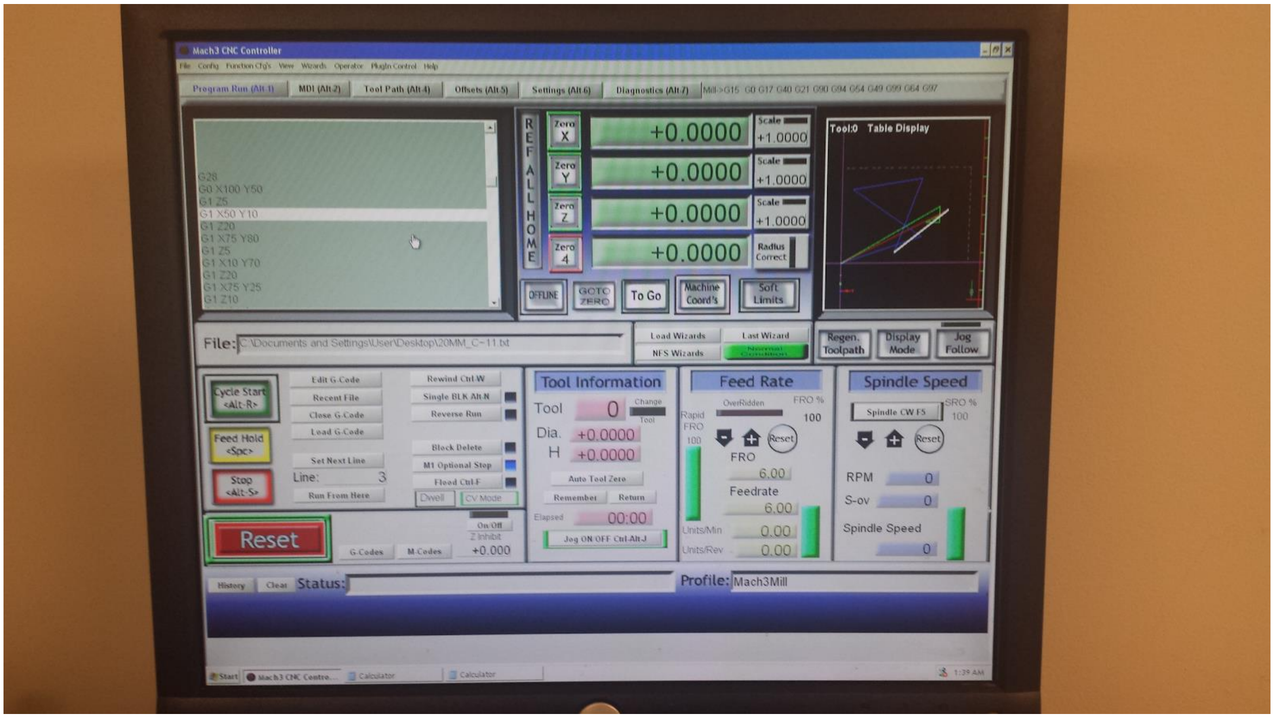


Figure 5.5: - working of mach3

Finally integration of the Mach3 with the LabVIEW control system can be done. Other features like temperature, chamber voltage, pressure, and fluid flow can be integrated and controlled by Labview systems by using particular hardware for each system.

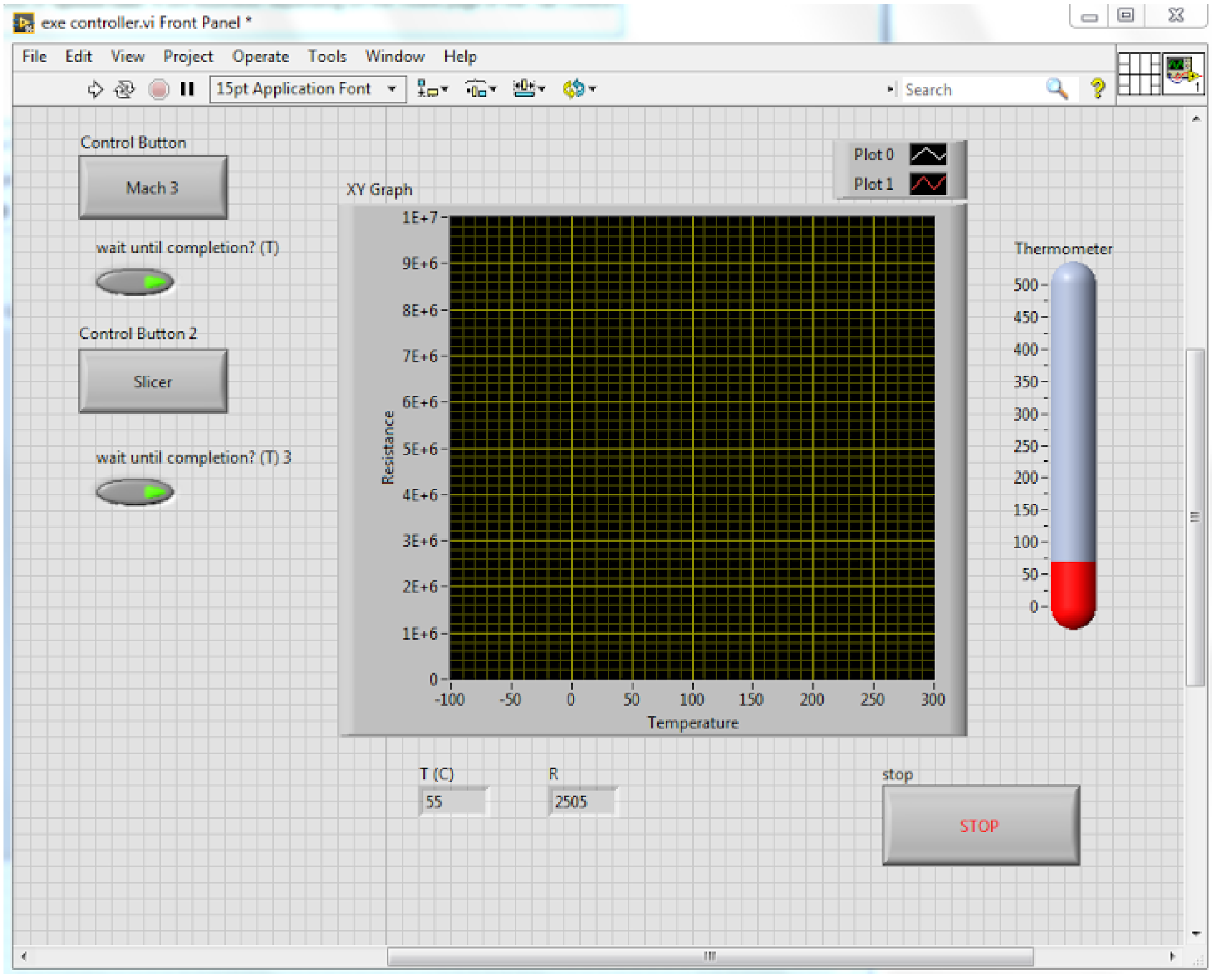


Figure 5.6: - Labview control system

As the Labview requirements are pretty specific. It cannot integrate everything within itself until it is provided with specific firmware and interface.

For the EHD printing system the requirement is to control heating element with the Labview while for the AFF it is required to control pressure.

CHAPTER 6. HEATING SYSTEM INTEGRATION

Temperature can be measured either with temperature probes like thermocouple or by thermistor. There are particular heating system probes which can be directly connected to through usb and using special DAQs we can check the readings.

6.1 Thermocouples

The problem with those probes are that they are very specific and cannot be used to control the temperature rather just used to check for temperature.

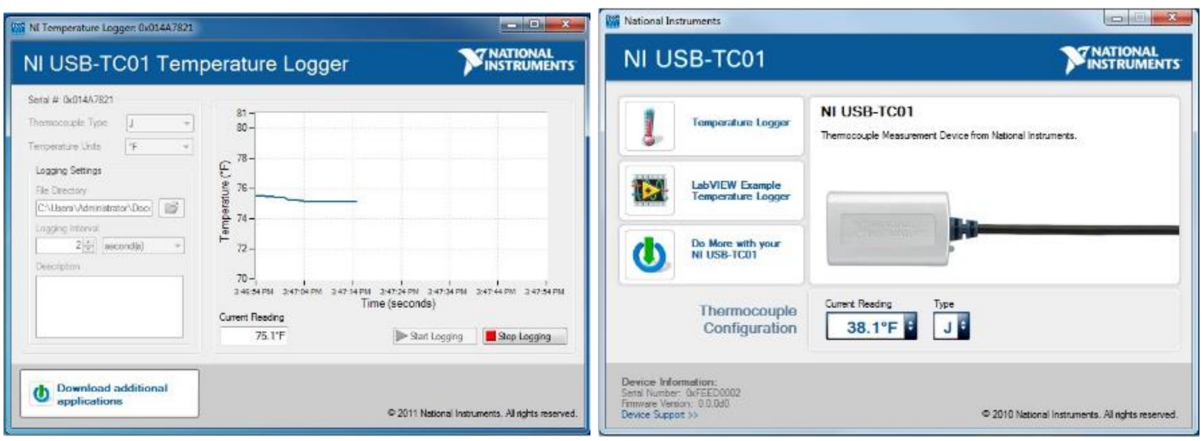


Figure 6.1: -(REF-15) NI USB TC01

The NI USB-TC01 is a thermocouple measurement device featuring InstantDAQ technology. It technology includes built-in software for viewing and logging data that automatically loads when you plug in device, which can instantly take temperature measurements with the PC. It can connect the USB-TC01 to any USB port to use the PC as a display and monitor data in real time. The USB-TC01 is compatible with J, K, R, S, T, N, E, and B thermocouples.

These thermocouple has a major drawback that CJC (Cold Junction Compensation) is required for all measurements as a reference. Thermocouples measure a temperature between two points and it's not absolute temperature. Many thermocouple devices have another built in temperature sensor to supply CJC value.

There are different types of thermocouples like K, R, S, T. They can be used for different operations and purposes.

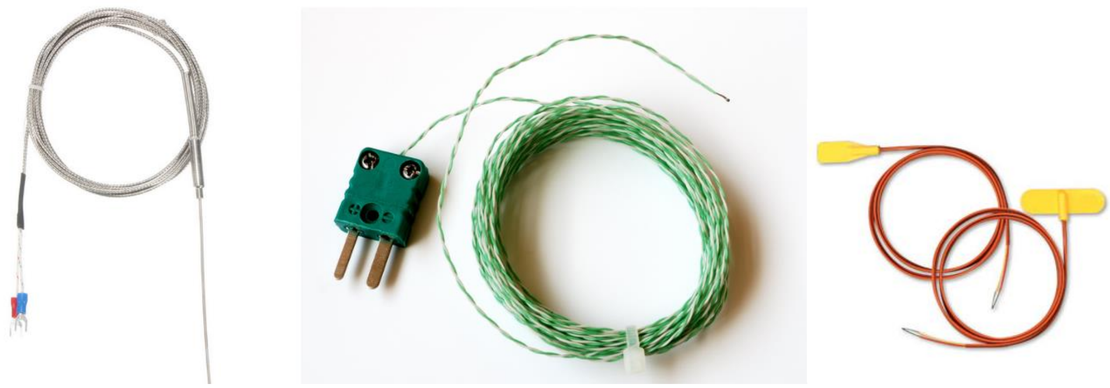


Figure 6.2: -Thermocouple K type, J type, omega SA2

The thermocouple in our lab is basically K type which is Nickel-Chromium / Nickel-Alumel. The type K is the most common type of thermocouple. It's inexpensive, accurate, reliable, and has a wide temperature range. The type K is commonly found in nuclear applications because of its relative radiation hardness. The maximum continuous temperature of this thermocouple is around 1,100C.

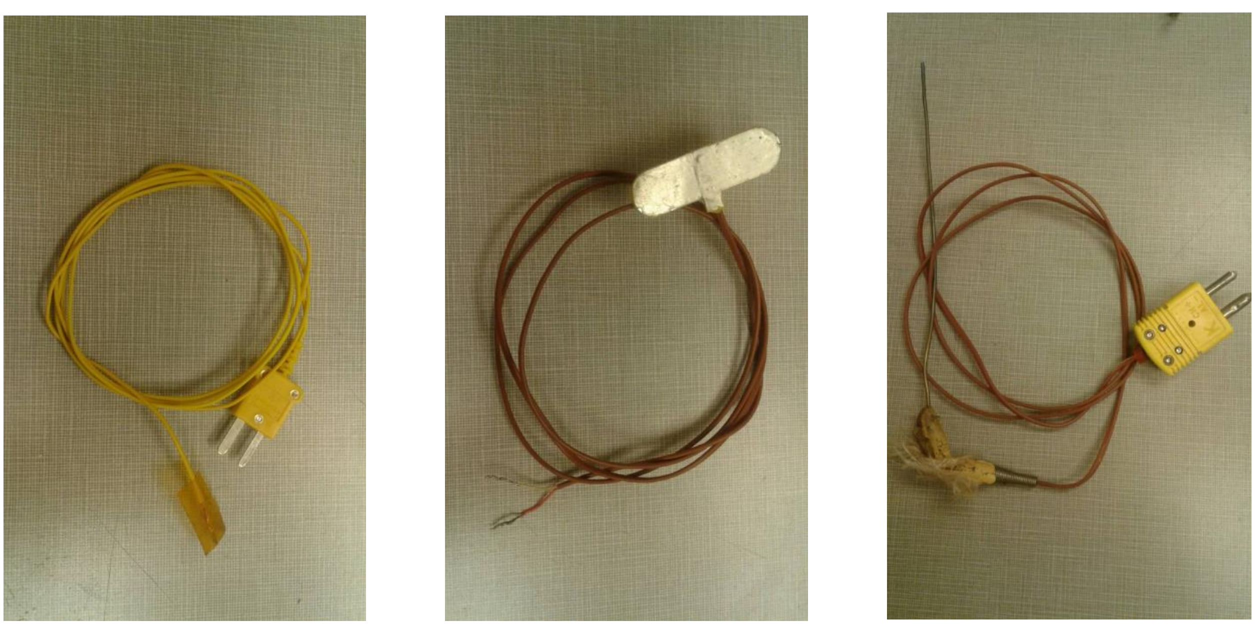


Figure 6.3: - different K type thermocouples

5.2 Thermistors

A thermistor is a type of resistor whose resistance is dependent on temperature, more so than in standard resistors. This name is derived from the more descriptive term “thermally sensitive resistor,” the original name for these devices. Thermistors are a type of semiconductor, meaning they have greater resistance than conducting materials, but lower resistance than insulating materials. The relationship between a thermistor’s temperature and its resistance is highly dependent upon the materials from which it’s composed. There are basically two types of thermistors namely NTC & PTC.

- With NTC, temperature rises as resistance decreases to protect against inrush overvoltage conditions. Commonly installed in parallel as the current sink.
- With PTC, temperature rises as resistance increases to protect against overcurrent conditions. Commonly installed in the series as a resettable fuse.

For this project two thermistors were tested one is Vishay 10k and the other is Honeywell 100K thermistor. The 10k thermistor gives quite a satisfactory results with maximum measurement of 120 °C. As the requirement for this project was about 230 °C so 100K thermistor was used, which has a higher temperature range.

Figure 6.4: - Vishay NTCLE-100E-3103 10k thermistor

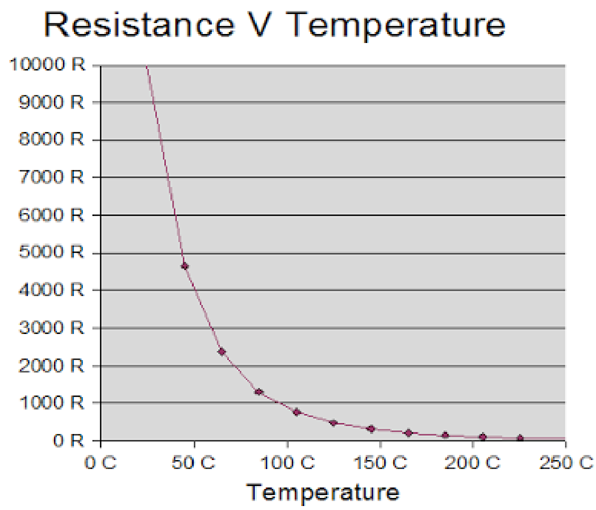
The NTCLE100E3103JB0 is a radial leaded standard Precision Negative Temperature Coefficient Thermistor which consists of a chip with two solid copper tin plated leads. It is gray lacquered and color coded, but is not insulated. It is highly stability over a long life with the temperature range of - 40 to + 125 °C.

Figure 6.5: - Honeywell 135-103FAD-J01

Honeywell makes the 135-103FAD-J01 thermistor comes with a rugged DO-35 glass encapsulation and is ideal for fully automated assembly. It has a uniform dimensions are suited for automated assembly. Thermistors are temperature-sensitive devices.

This is built to show large change in resistance to corresponding small change in temperature. Electrical resistance change of this thermal resistors is negatively correlated to the temperature change.

With an operating temperature ranging from - 60° C to 300° C, this Thermistor is designed for applications which require reliability at low cost. They have a very wide applications owing to their predictable, precise and stable properties.



100K Thermistor Output Table

°F	°C	Ohms	°F	°C	Ohms	°F	°C	Ohms
-39	-39.44	3916295	37	2.78	302466	113	45.00	41303
-37	-38.33	3627711	39	3.89	285206	115	46.11	39434
-35	-37.22	3362274	41	5.00	269035	117	47.22	37660
-33	-36.11	3117987	43	6.11	253877	119	48.33	35976
-31	-35.00	2893035	45	7.22	239664	121	49.44	34376
-29	-33.89	2685770	47	8.33	226331	123	50.56	32843
-27	-32.78	2494694	49	9.44	213819	125	51.67	31399
-25	-31.67	2318444	51	10.56	201971	127	52.78	30027
-23	-30.56	2155781	53	11.67	190946	129	53.89	28722
-21	-29.44	2004274	55	12.78	180588	131	55.00	27481
-19	-28.33	1865595	57	13.89	170853	133	56.11	26300
-17	-27.22	1737397	59	15.00	161700	135	57.22	25177
-15	-26.11	1618827	61	16.11	153092	137	58.33	24107
-13	-25.00	1509102	63	17.22	144992	139	59.44	23089
-11	-23.89	1407512	65	18.33	137367	141	60.56	22111
-9	-22.78	1313405	67	19.44	130189	143	61.67	21188
-7	-21.67	1226184	69	20.56	123368	145	62.78	20308
-5	-20.56	1145306	71	21.67	117000	147	63.89	19469
-3	-19.44	1069620	73	22.78	110998	149	65.00	18670
-1	-18.33	1000019	75	23.89	105338	151	66.11	17907
1	-17.22	935383	77	25.00	100000	153	67.22	17180
3	-16.11	875329	79	26.11	94963	155	68.33	16486
5	-15.00	819505	81	27.22	90208	157	69.44	15824
7	-13.89	767589	83	28.33	85719	159	70.56	15187
9	-12.78	719284	85	29.44	81479	161	71.67	14584
11	-11.67	674319	87	30.56	77438	163	72.78	14008
13	-10.56	632442	89	31.67	73654	165	73.89	13458
15	-9.44	593086	91	32.78	70076	167	75.00	12932
17	-8.33	556739	93	33.89	66692	169	76.11	12430
19	-7.22	522842	95	35.00	63491	171	77.22	11949
21	-6.11	491217	97	36.11	60461	173	78.33	11490
23	-5.00	461699	99	37.22	57594	175	79.44	11051
25	-3.89	434134	101	38.33	54878	177	80.56	10627
27	-2.78	408383	103	39.44	52306	179	81.67	10225
29	-1.67	384316	105	40.56	49847	181	82.78	9841
31	-0.56	361813	107	41.67	47538	183	83.89	9473
33	0.56	340581	109	42.78	45349	185	85.00	9121
35	1.67	320895	111	43.89	43273	187	86.11	8783

Figure 6.6: - 100K thermistor data

6.3 Steinhart–Hart Equation

The Steinhart–Hart equation is a model of the resistance of a semiconductor at different temperatures.

$$\frac{1}{T} = A + B \cdot \ln(Rt) + C \cdot (\ln(Rt))^3$$

where: T = degrees Kelvin

a,b, and c = coefficients derived from measurement

To know a, b and c coefficients, measure the thermistor at three different temperatures. The temperatures should be evenly spaced with atleast 10 degrees apart. Use these three temperatures to solve three simultaneous equations. These equations allow to derive a, b and c for any temperature range. Knowing a, b and c for the thermistor allows the use of Steinhart and Hart equation.

There is a major difference between thermistor, thermocouple and RTD. In contrast to RTDs, that change resistance in a nearly linear way, thermistors have a highly non-linear change in resistance and actually reduce their resistance with increases in temperature. Unlike RTDs and thermocouples, thermistors do not have standards associated with their resistance vs. temperature curves. As a result, there are many different ones to choose from. Every thermistor material provides a different resistance vs. temperature "curve". Some materials provide better stability while others have lower resistances so they can be fabricated into larger or smaller thermistors.

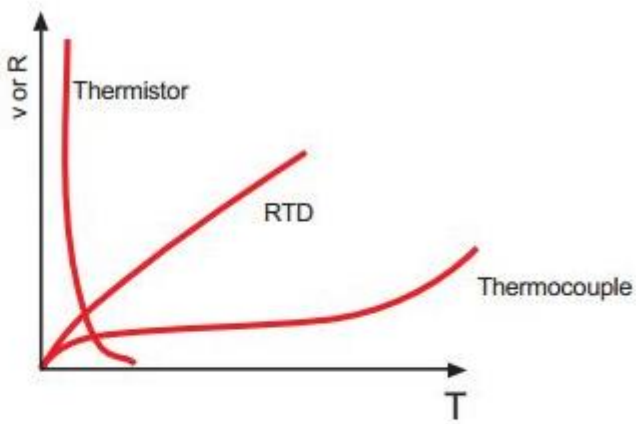


Figure 6.7: -difference between thermistor, thermocouple and RTD

6.5 Sampling with Arduino

To control the temperature we need to use a microcontroller that can generate signal and can control the heating element. For that purpose using Arduino as a microcontroller seems the best option. But the problem with the Arduino is that it cannot be used with a high power heating element which ranges above 12-15 watts.

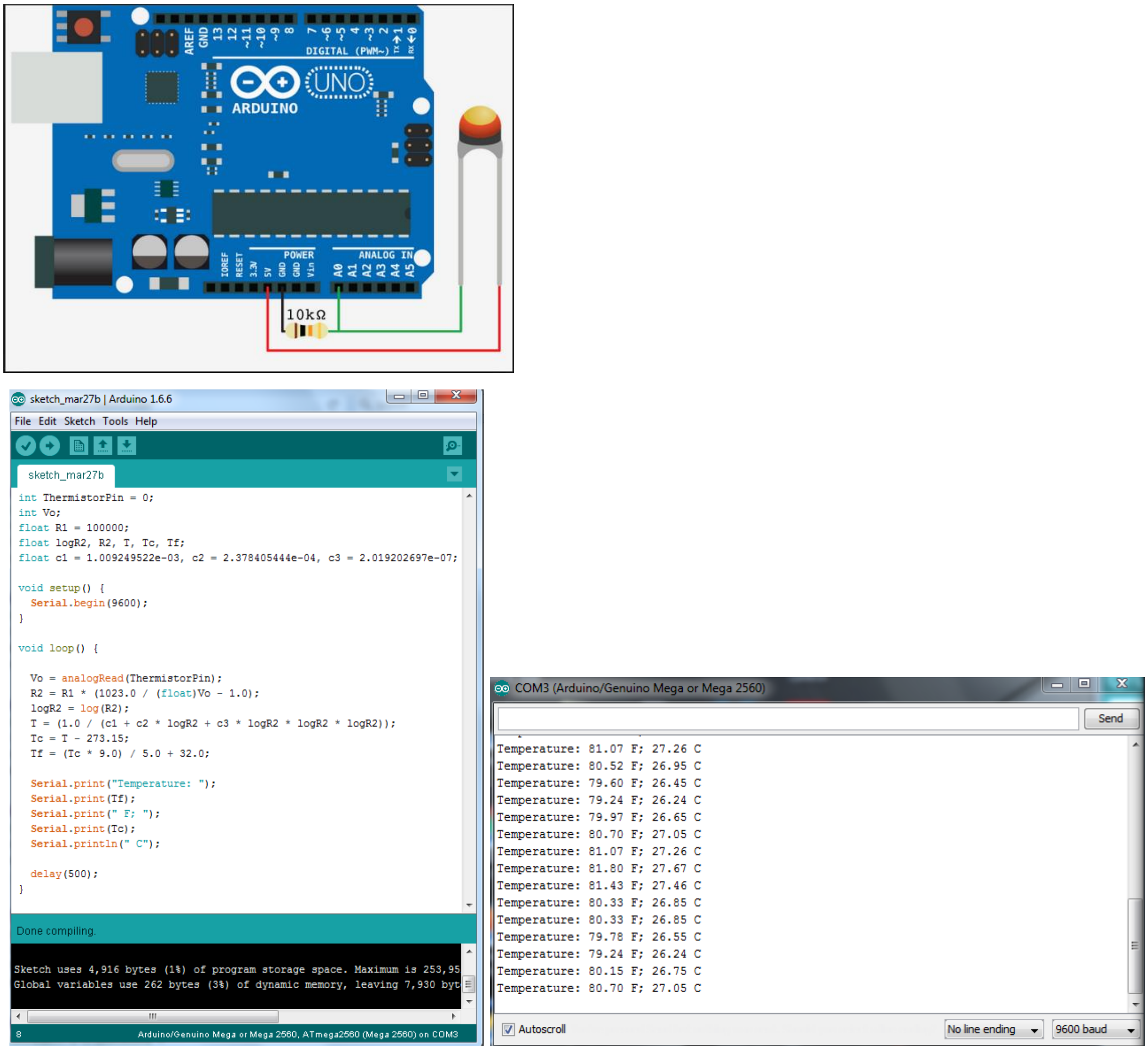


Figure 6.8: - Arduino output data

This can be done in Arduino with a simple program. Just define the value of the thermistor and write a simple program for Stein Hart Equation with it’s specific coefficient.

6.5 Sampling with LIFA

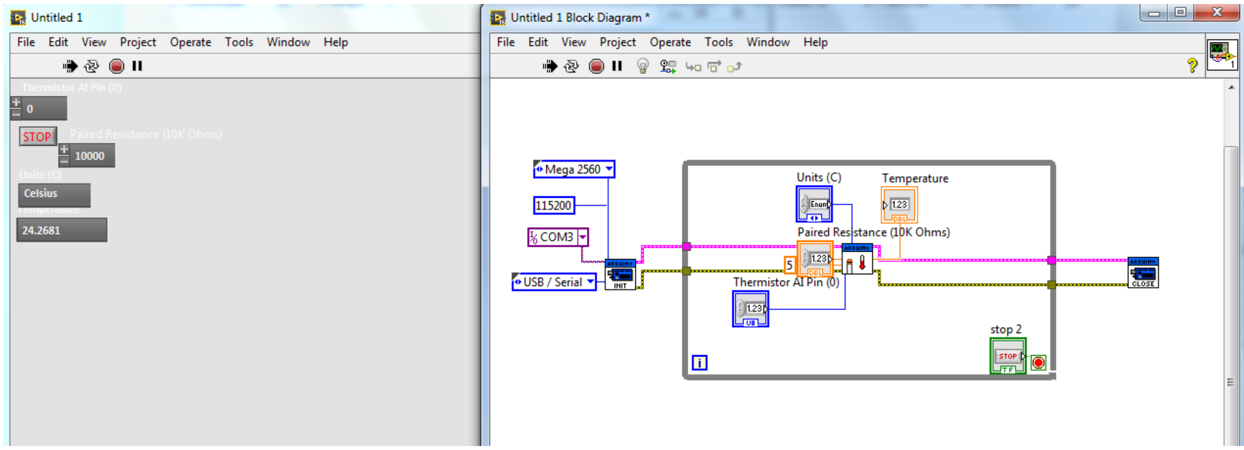


Figure 6.9: - LIFA program for thermistor

Using Labview interface for Arduino tab and making a simple program for thermistor we can see the results. The results are consistent for 10k thermistor while it changes for 100k thermistor. This is due to the Stein Hart coefficients, there is always an error in the calculation of coefficients due to the equation itself.

6.6 myDAQ

Student Data Acquisition Devices feature eight commonly used plug-and-play computer-based lab instruments, including a digital multimeter (DMM), oscilloscope, and function generator based on LabVIEW. It has access to all the ready-to-run software instruments to perform experiments and exercises with the Bode analyzer, arbitrary waveform generator, dynamic signal analyzer (fast Fourier transform), digital input, and digital output. When combined with LabVIEW and NI Multisim software, gives the power to prototype systems and analyze circuits outside traditional lectures and labs to these affordable devices which allow for real engineering.

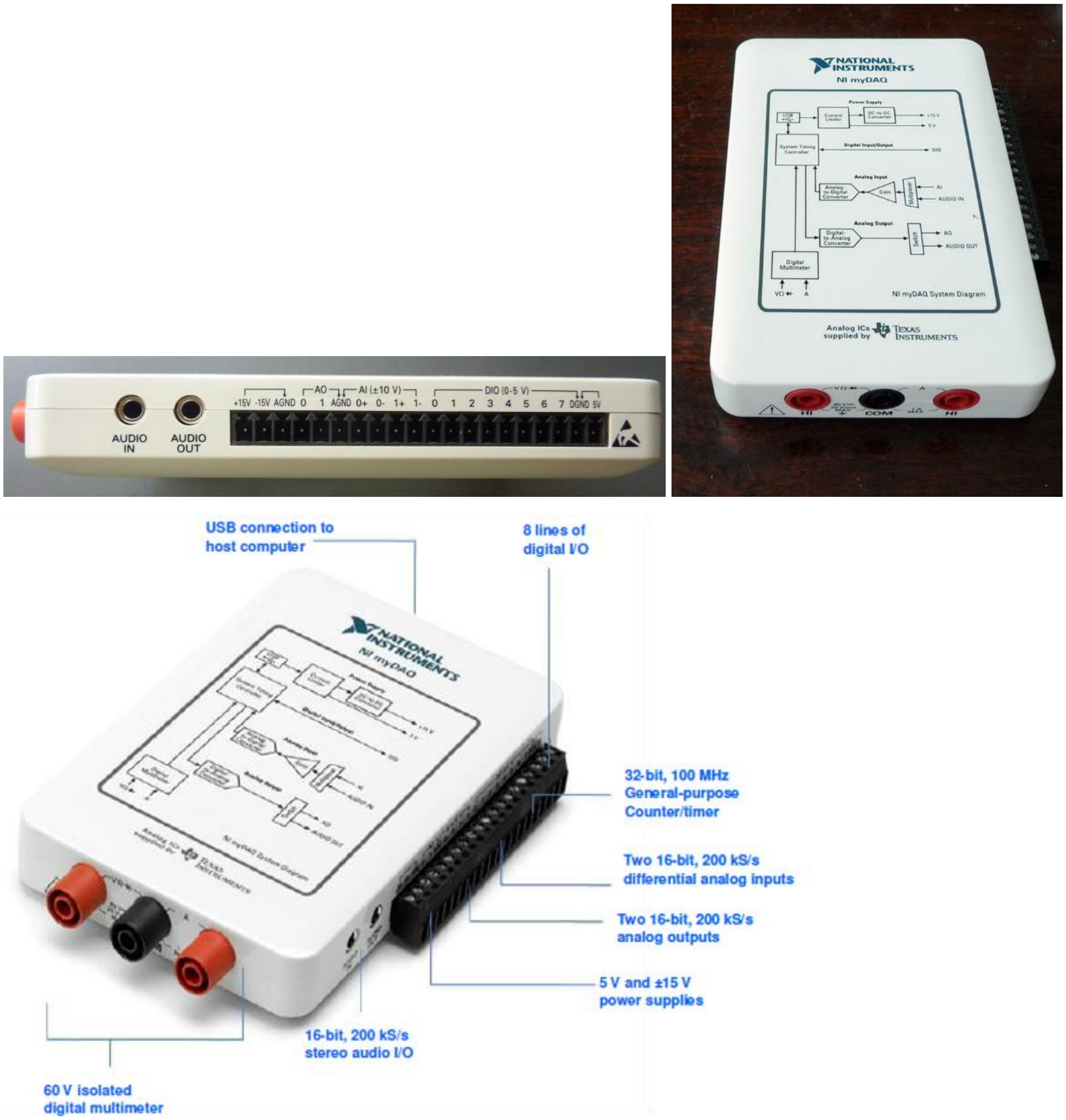


Figure 6.10: - myDAQ pinouts

This myDAQ has a major problem that it does not have CJC (Cold Junction Compensation) which is required for all measurements as a reference. Thermocouples just measure a temperature between two points and it’s not absolute temperature.

6.7 Coding myDAQ with Labview

The myDAQ can be connected to a thermistor and can be used to measure temperature. The desired thermistor is wired in a circuit as a resistor. It requires a positive input on one side and a negative input on the other side where the orientation is not important.

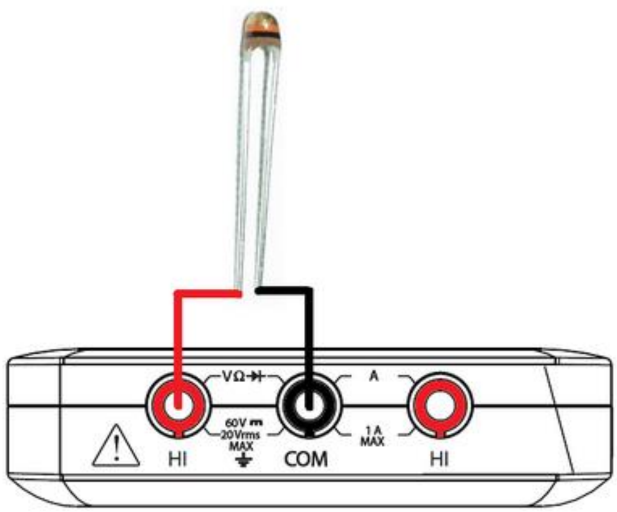


Figure 6.11: -(REF-18) wiring diagram of thermistor

Now write a simple program putting Stein Hart Equation into a loop and converting resistance to temperature.

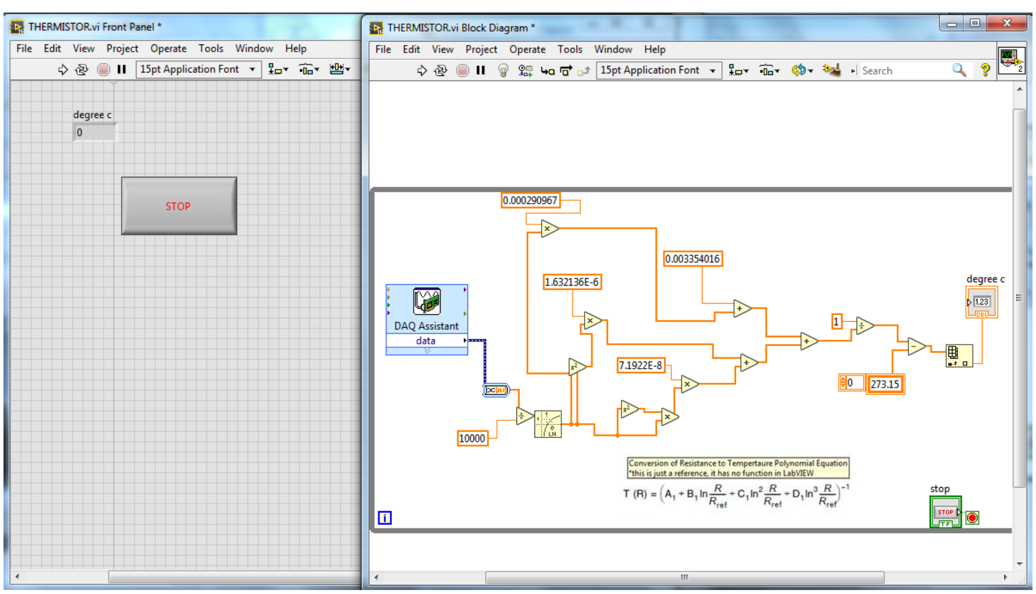


Figure 6.12: -thermistor in a loop

Update myDAQ by configuring the DAQ with the 100k thermistor using internal IEX source and 2 wire configuration. Putting 1 sample on demand will take resistance value at an instance.

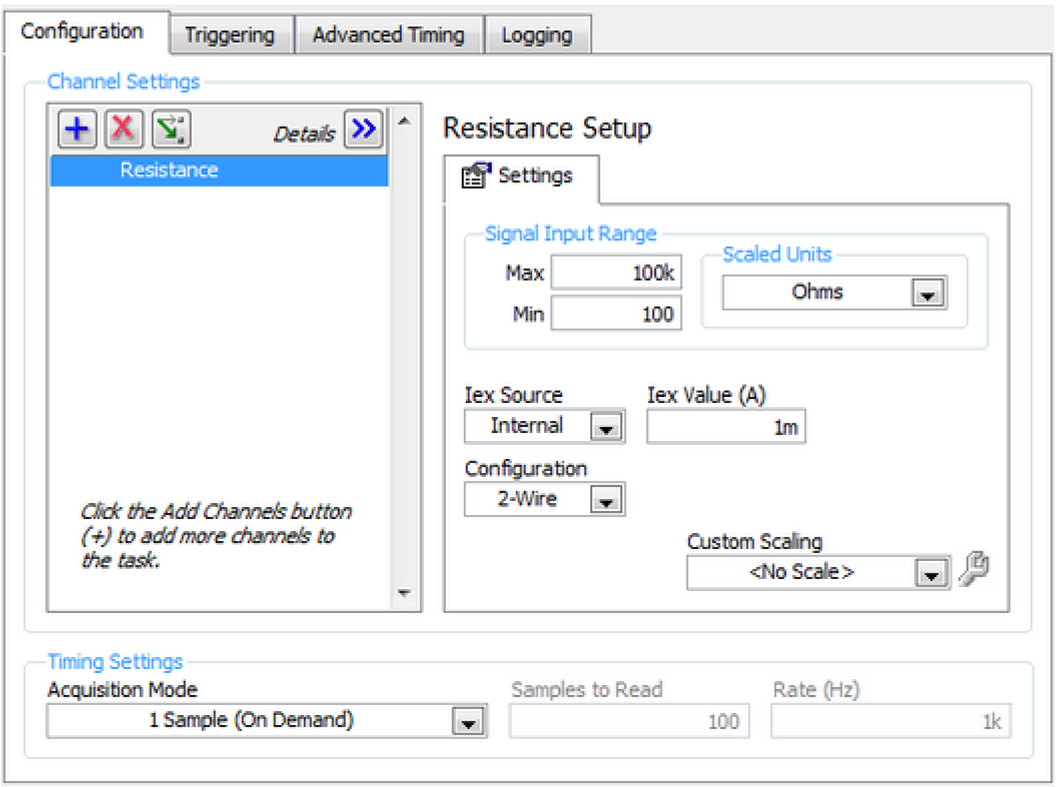


Figure 6.13: - DAQ configuration with the 100k thermistor

In LabVIEW we need to measure the resistance value coming from the thermistor of 100Ω to 100kΩ. This value is then converted to a temperature using the polynomial equation from the thermistor specifications referenced at the end of this document. Finally, the output is the result to a numeric indicator and a temperature chart on the front panel.

Inside the while loop on there is the DAQ Assistant. It's configured to read a single value from the myDAQ DMM terminals at the time it executes. Once a value is read it is passed down the wire, it is divided by 10000 and then the natural log is taken of the quotient before being again passed into the formula node. The resistance is then further converted into a temperature in Kelvin using the Stein Hart polynomial equation and constants from the sensor specifications sheet.

Now one part of the loop is complete. We need to control the temperature as till now we are only able to read it. For this we need to integrate Matlab script into the Labview.

6.8 Integrating Matlab and Labview

Here we are able to control the temperature. For this we integrated Matlab script with the Labview. The condition is specified into the Matlab script which in turns gives the output voltage as required. The condition is such that if the desired temperature value is less then we will have output on a certain selected pins.

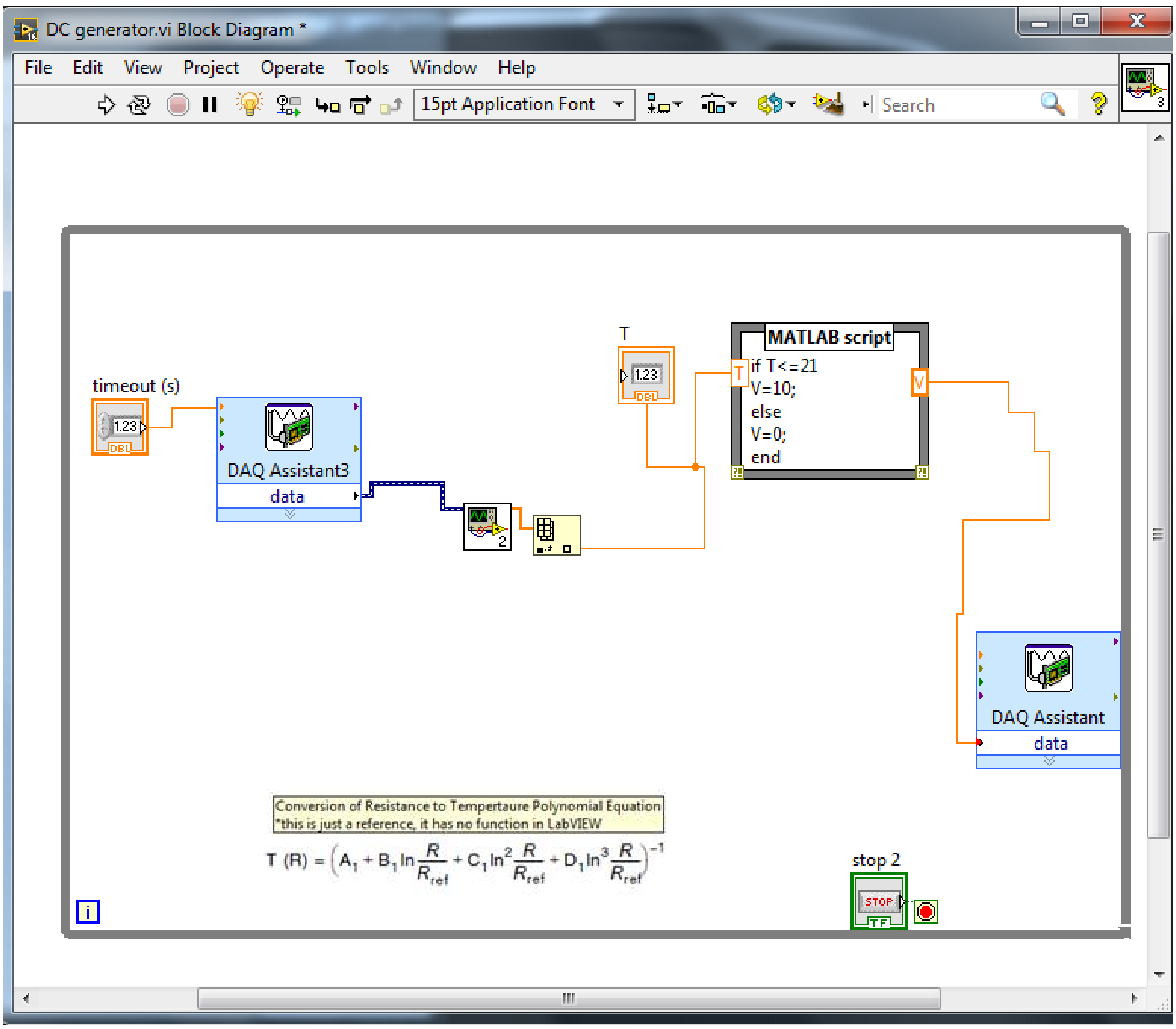


Figure 6.14: - integration of matlab with labview

This can be seen in the following figure 60 that the temperature will be constant if we apply this coding. We can add meters to read temperature.

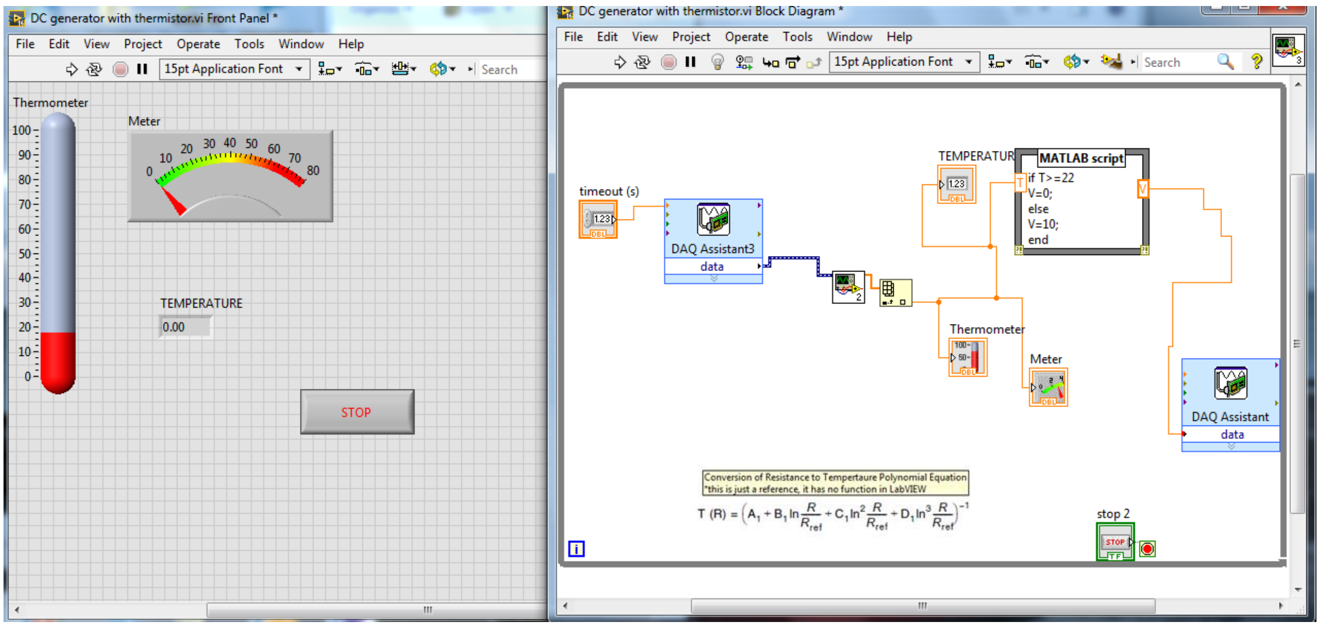


Figure 6.15: - controlling temperature

Now checking the pins for output voltage. As the selected output voltage was 10V, so there is high enough voltage but the current is very low, as low as 2mA, which is very difficult to run a heating element. It can be seen that the led lights up as the program run.

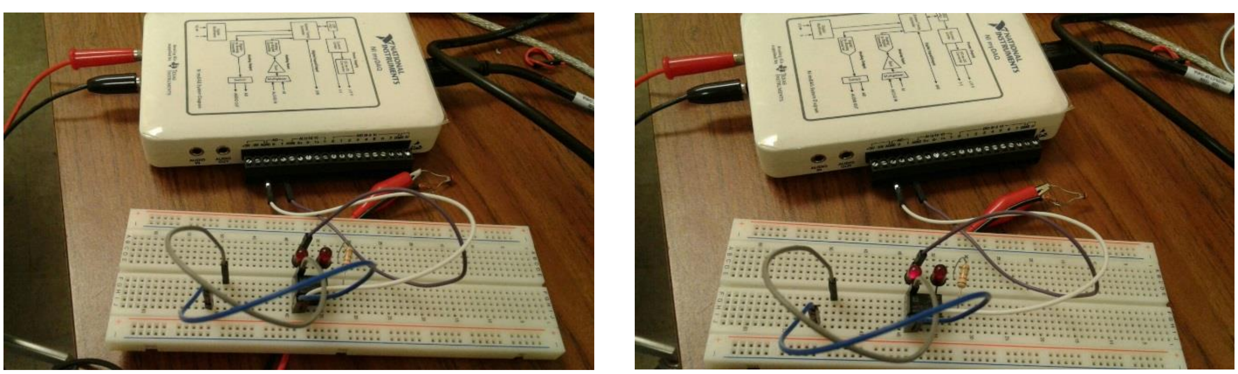


Figure 6.16: - led lights up with output pins

6.9 Graphical Method

Another idea is to take up the resistance vs temperature chart of thermistor and put these points in the Labview. This gives a T vs R graph which is exponentially decaying. Now merge the graph with the exponentially decaying graph, which are quite close. Now try to interpolate the points, as the myDAQ is able to measure resistance, we can get the corresponding value of temperature.

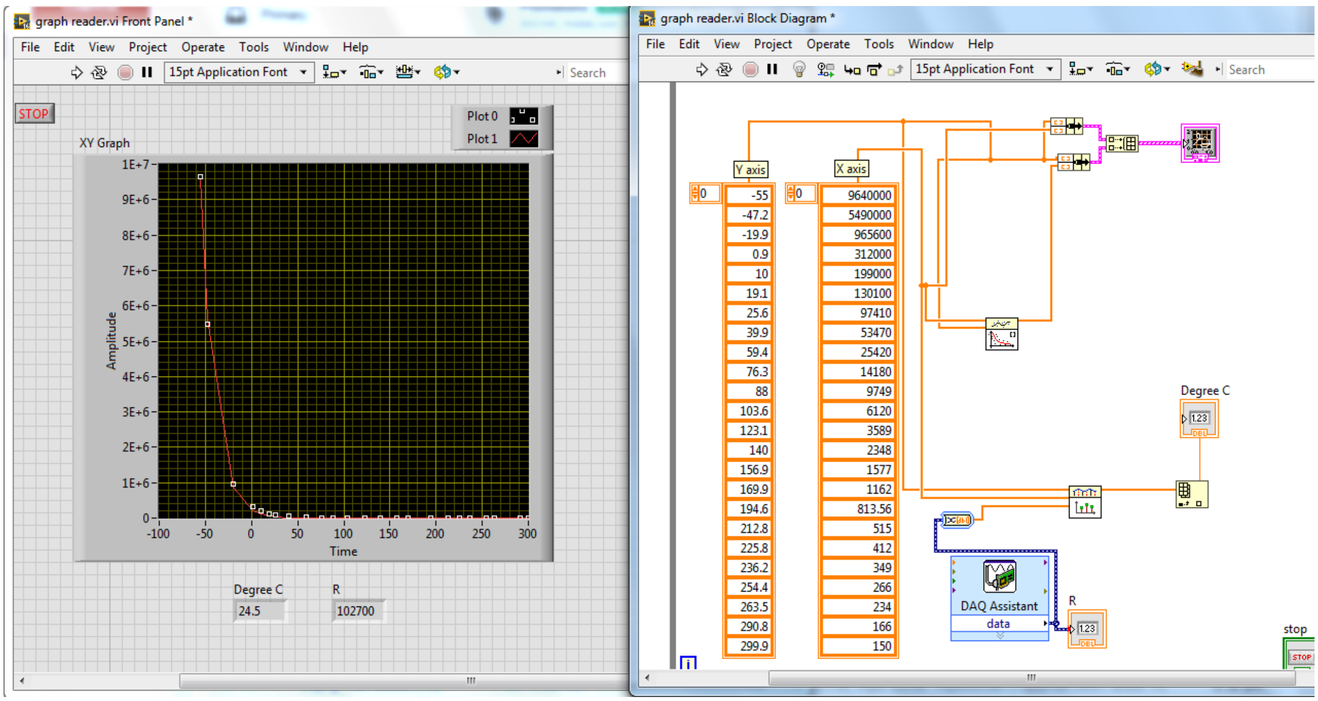


Figure 6.17: - thermistor graphical method

This interpolation gives quite a result even better than using stein hart equation. The temperature always has a tolerance of +2.3 to -2.3, which for higher temperature measurement is outstanding. This is measured from a thermometer to check if the result is consistent. The result is always consistent and has a gradual decrease or increase in the temperature.

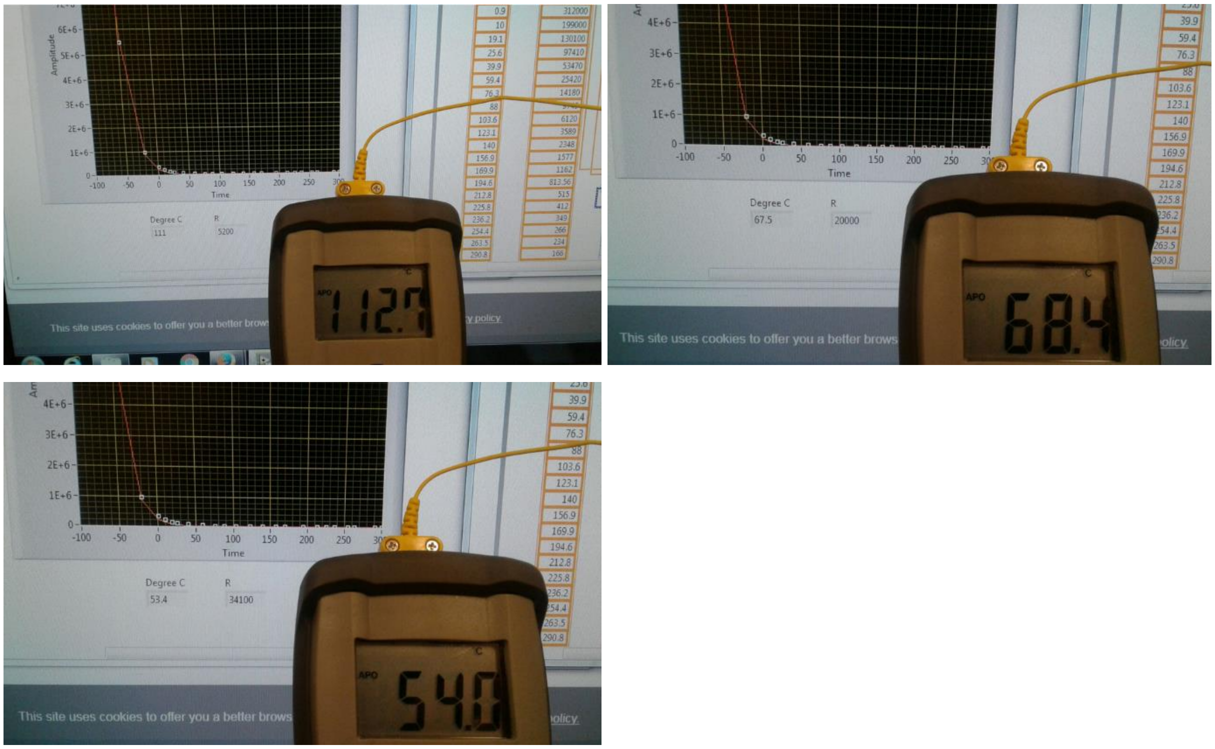


Figure 6.18: - result comparison with graphical method

6.10 Final using mydaq as Controller

As a result, comparison with graphical method all other methods are not up to the point. So taking graphical method and combining with Matlab we can now make our whole circuit. As we know the output voltage is very low so we will use transistor as a triggering and relay as a switching device.

The DC supply we are going to use is a Protek dual dc supply, which can give up to 30V and 1.5 A, which is high enough for the heating element. As this can give upto 45 watts of power, while our requirements ranges from 10-15 watts.

Figure 6.19: - Protek dual dc supply

Now the relay which we are going to use is HY1Z-5V, Signal Relay. This has a coil resistance of 125ohm with a contact current of 1A. The highest contact VDC is 30V.

This simple N.C and N.O contact will be triggered through transistor which will allow relay to switch on and off during the process. This will bypass the required current for the heating element through DC supply.

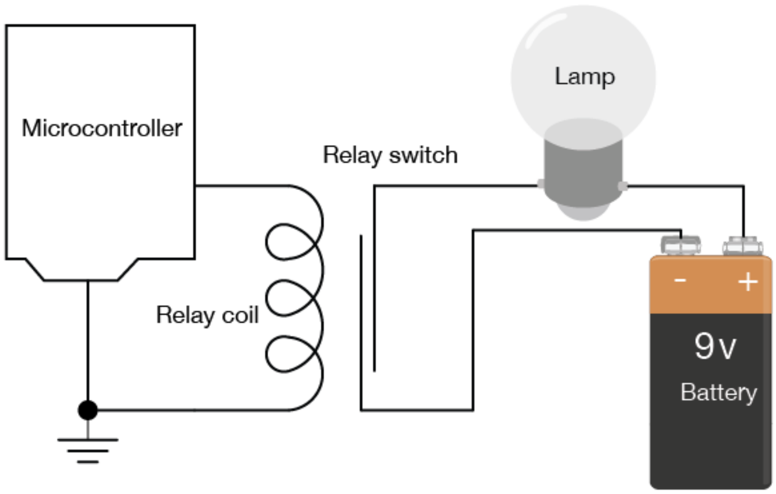
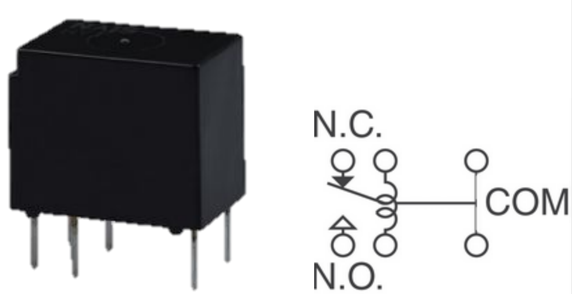


Figure 6.20: -(REF- 20) switching circuit

This has a common terminal which connects directly with N.C without being charged and connects with N.O when it's given current.



Product Information

Contact Configuration:	SPDT
Coil Voltage:	5VDC
Contact Current:	1A
Product Range:	HY Series
Relay Mounting:	Through Hole
Coil Type:	Non Latching
Contact Voltage VAC:	-
Relay Terminals:	Solder
Contact Voltage VDC:	30V
Contact Material:	Silver
Coil Resistance:	125ohm
SVHC:	To Be Advised

Figure 6.21: - HY1Z-5V Signal Relay

The transistor that will be used is 2N3906 or 2N3904 PNP. It has a emitter base voltage of 5V which is sufficient for it's working with DAQ.

Product Category:	Bipolar Transistors - BJT
Manufacturer:	Central Semiconductor
RoHS:	Details
Mounting Style:	Through Hole
Package / Case:	TO-92-3
Transistor Polarity:	PNP
Configuration:	Single
Collector- Emitter Voltage VCEO Max:	40 V
Collector- Base Voltage VCBO:	40 V
Emitter- Base Voltage VEBO:	5 V
Collector-Emitter Saturation Voltage:	400 mV
Maximum DC Collector Current:	200 mA
Gain Bandwidth Product fT:	250 MHz
Maximum Operating Temperature:	+ 150 C
Series:	2N3906
Brand:	Central Semiconductor
Continuous Collector Current:	0.2 A
Height:	5.33 mm
Length:	5.21 mm
Packaging:	Bulk
Pd - Power Dissipation:	625 mW
Factory Pack Quantity:	2500
Width:	4.19 mm
Part # Aliases:	BK

Figure 6.22: - 2N3906 PNP transistor

The transistor will allow to trigger relay as the transistor collector is connected with the relay, which provides it's high voltage. The base is connected with a 1K resistor and the emitter is connected to the ground.

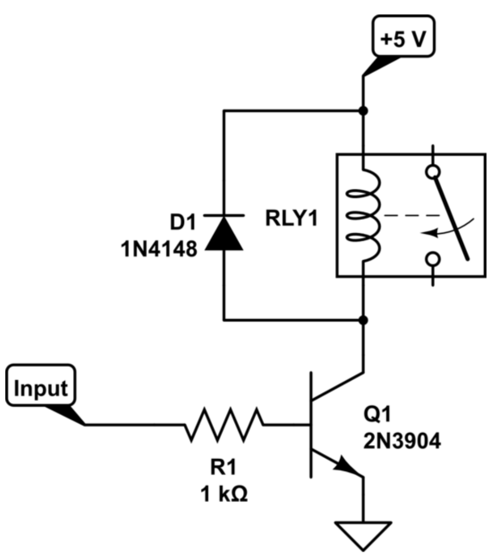


Figure 6.23: -(REF-20) transistor circuit

Now, comes main part of the heating system is the heating pad. This has an internal resistance of 15ohm with the working temperature of 140 degree Celsius. It is a polyamide thermos foil heater which has a high ranting 10 watts that generates high temperature range of 230 to 300 degree Celsius maximum. The project requires around 220 degree Celsius which will be used to melt PLA.

The dimensions are across 80 X 10 mm which is enough to surround a nozzle. This will be directly mounted on the nozzle to heat it up.

Net Weight:	3g	Country of Manufacture:	CHINA
Package Content:	1 x Polyimide Thermo Foil Heater	Working Temperature:	140C
Resistance:	15OHM	Power:	10W
Dimensions:	80 x 10mm/ 3.1 x 0.4 inch (L*W)	Material:	Polyimide, Foil
Max Temperature Up to:	200C-220C	Thickness:	<=0.2mm
MPN:	a15012800ux0370	Working Voltage:	12V
Wires Length:	10cm/ 4inch	Manufacturer Part Number:	a15012800ux0370
UPC:	711331536877		

Figure 6.24: -12V 10W Polyimide Flexible Membrane Thermo Foil Heater Heating Film 80mm x 10mm

6.11 Proposed Heating System

The proposed heating system is a loop which acts on triggering. The relay is connected to 5V battery supply to energise it. The transistor base is connected to 50K resistor which is then connected to a 5V output of the myDAQ. The emitter is connected to the negative of the battery as well as the negative terminal of the myDAQ. Where on the other side, collector is connected to the relay coil which completes one half of the circuit.

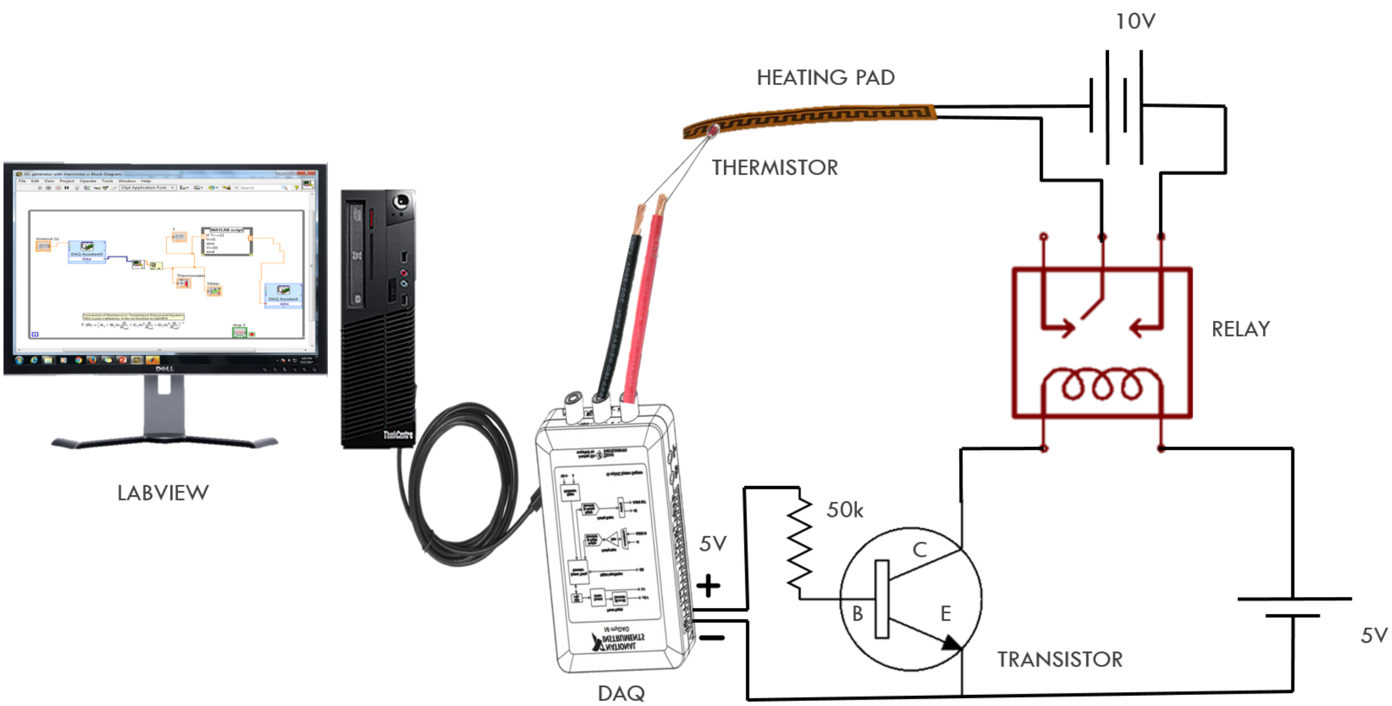


Figure 6.25: - proposed heating system

The other half of the circuit is connected with a DC power source which gives around 10V and 1A for the heating element. The myDAQ is directly connected to Labview which is run through the loop. This system is checked with both the transistors 906 & 904. The result is best derived from 906 transistor.

Other Heating Options

The Lab contains a lot of heating devices which can be used manually to control the temperature. Some of them have built in PID controllers, which controls the temperature. The thermocouple is provided with these models. Our Lab has Omega benchtop heating system which uses K type thermocouple for it's working.

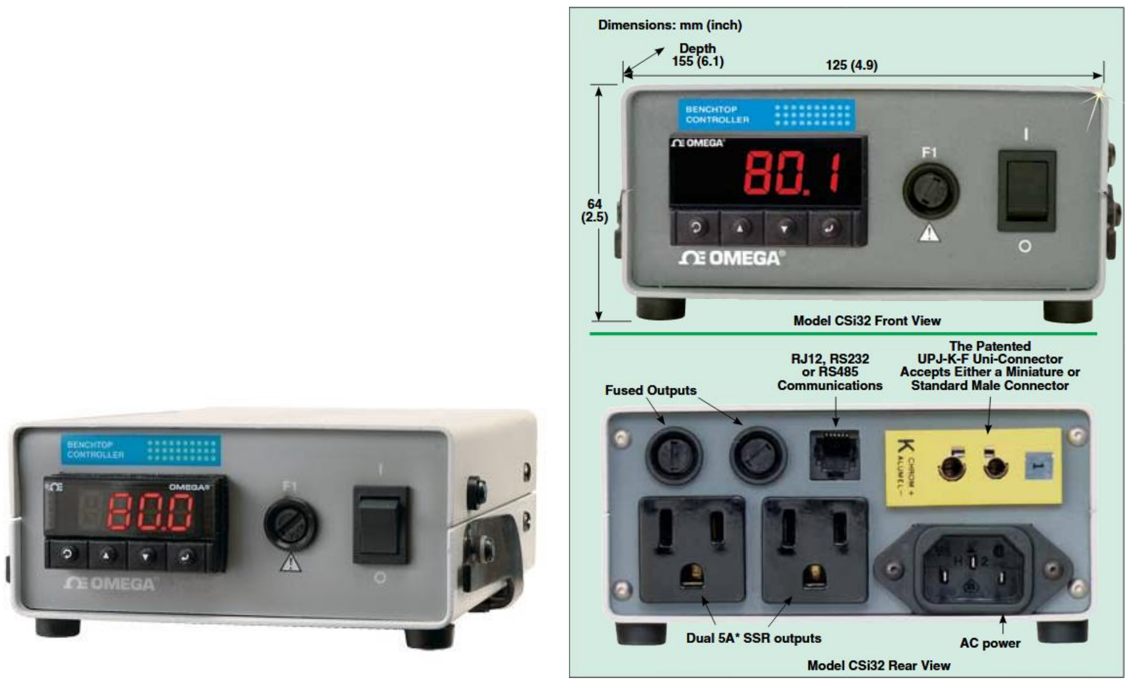


Figure 6.26: - (REF-21) omega benchtop controller

The heating element for this controller is Silicone rubber extruded heating tapes that can be used in direct contact with a metal or conductive surface. These heating tapes are low watt density electrical resistance heaters designed for temperature maintenance in applications requiring moisture and chemical resistance.

Silicone rubber tapes are made up of finely stranded resistance wires fully insulated with braided fiberglass and knitted into flat tape with fiberglass yarn. These tapes are encapsulated in a blank free silicone rubber sheath.

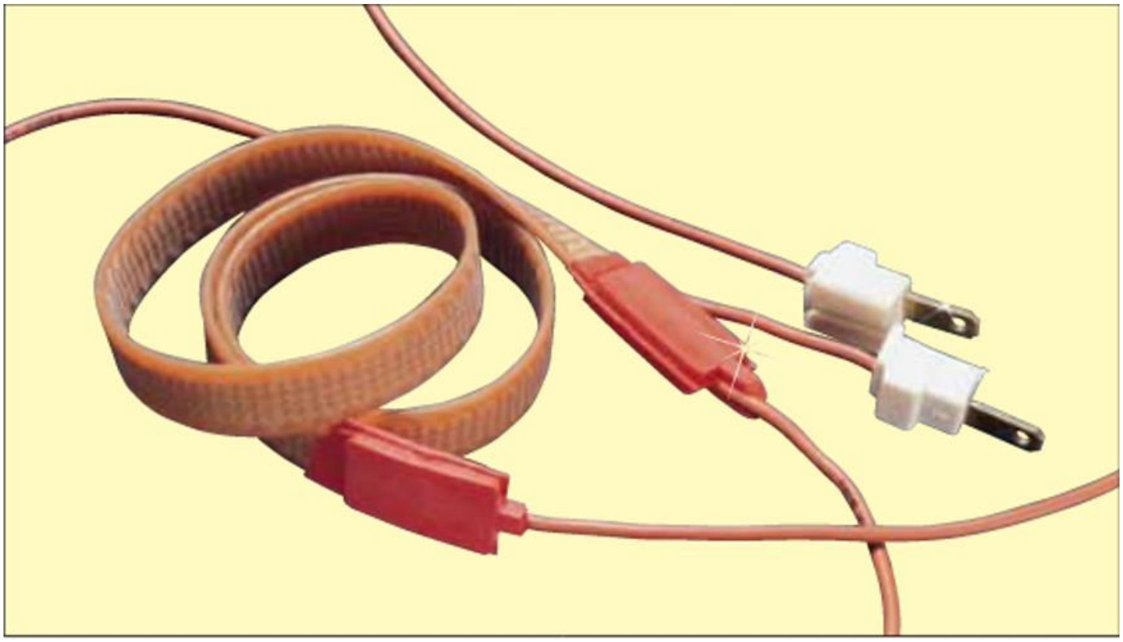


Figure 6.27: - (REF-21) Silicone rubber extruded heating tapes

CHAPTER 7. AFF SYSTEM

This contains Central Pneumatic compressor which is capable of providing 15 to 25 psi to the chamber This has a custom regulator setup for variability in airflow assistance with the printer head.

Figure 7.1: - Central Pneumatic compressor

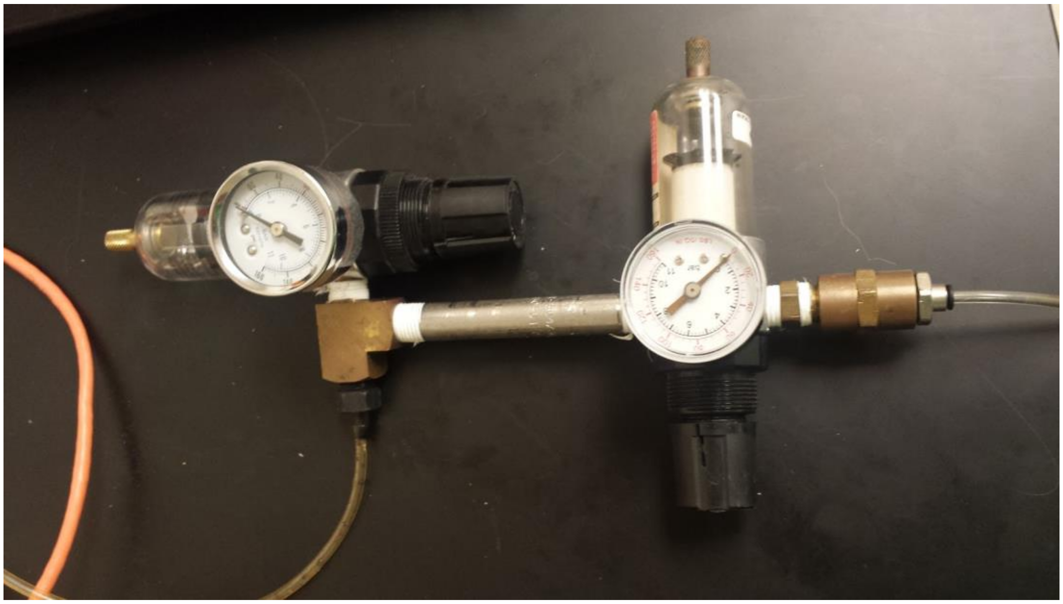


Figure 7.2: - custom regulator

The National Instrument has their own built in PID controller for pressure. So it is easy to integrate our Labview program with the pressure controller. For that we need to use NI C series module to connect to a pressure transducer.

Figure 7.3: - NI 9237

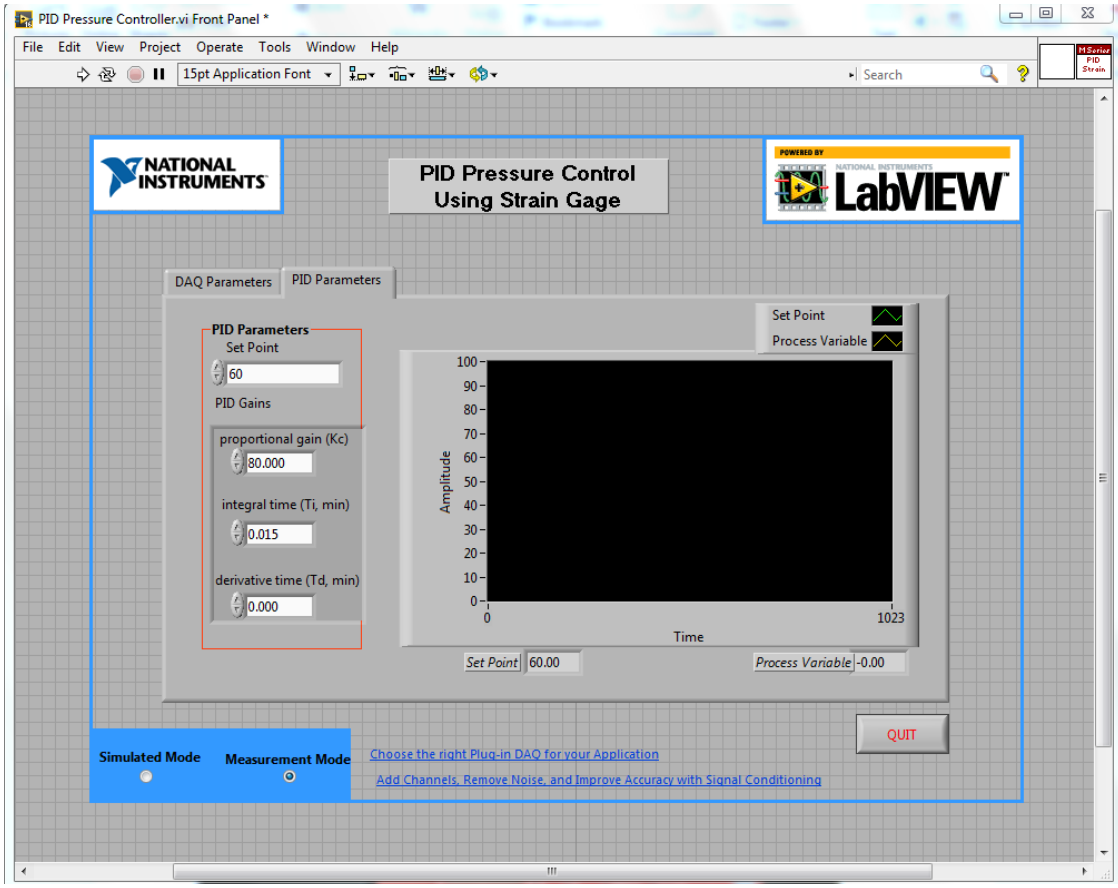


Figure 7.4: - PID pressure controller

The pressure transducer is a strain guage based model which has an inner diaphragm whose resistance changes with the change in pressure. To measure from pressure transducer we need excitation voltage. Since it's output is in millivoltages so we need some sort of signal conditioning to amplify it.

So it's better to use NI C series module as it provides both the excitation voltage and signal conditioning. Here the pressure is directly measured from Labview.

CHAPTER 8. FINAL TESTING

Now we can test our concept of EHD and AFF. To test it separately we need some syringe to provide material to the nozzle. The heating system can be mounted on the syringe to the nozzle. To make it possible we need syringe pump, pump, camera and high voltage supplier.

8.1 Syringe pump

NE-1000 programmable pump is capable of delivering as low as 0.73 µL/hr and up to 2100 mL/hr. It is built for Automation which can operates stand alone or from a computer. It Infuses and withdraws with it's applications range from simple infusions to complex pumping programs.

It can be programmed up to 41 pumping phases: change pumping rates, set dispensing volumes, insert pauses, control and respond to external signals, sound the buzzer.

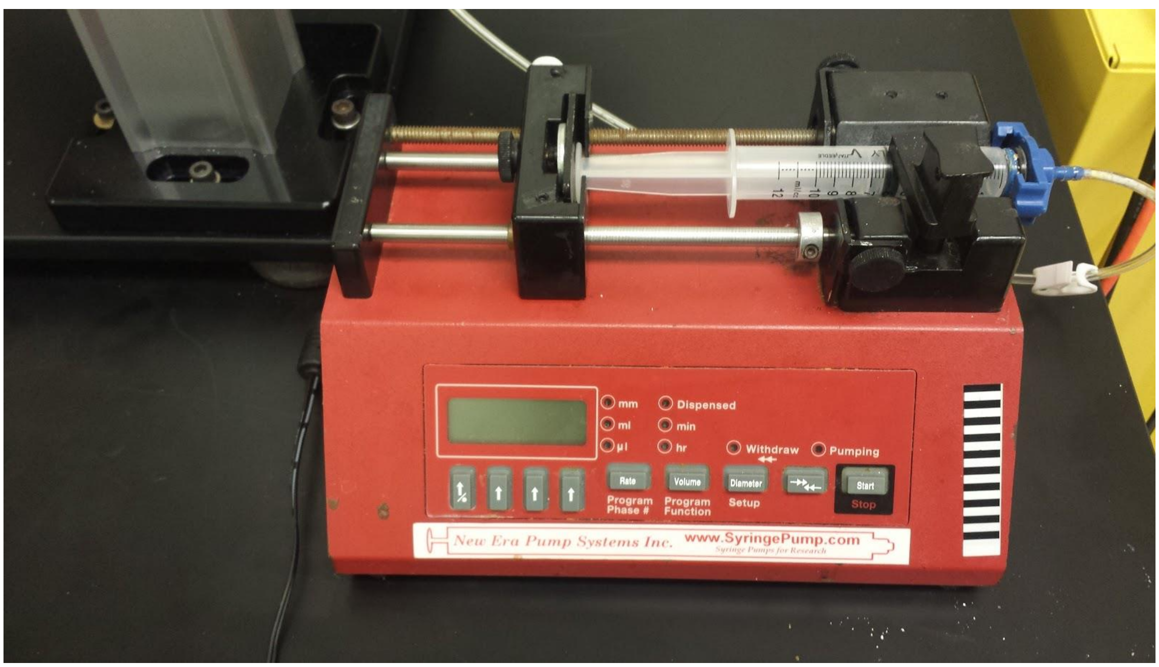


Figure 8.1: - new era syringe pump

8.2 High voltage supplier

The Model 677B is a high-voltage power amplifier/supply designed to provide accurate control of output voltages. It can be operated in one of two modes: as high-voltage power supply that responds to front panel controls to command exact output voltage or current as a high-voltage amplifier when it is configured as a noninverting amplifier with a fixed gain.

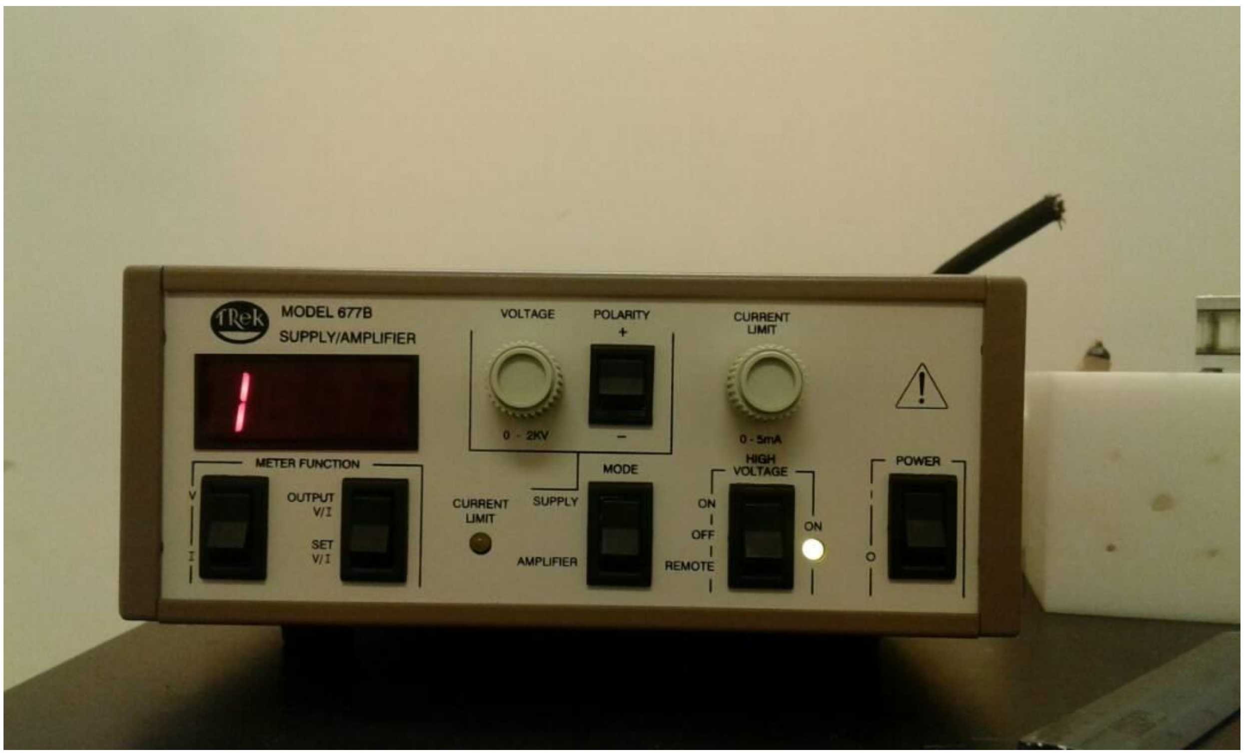


Figure 8.2: - 677B, a high-voltage power amplifier

The 677B features an all-solid-state design for wide bandwidth, high slew rate and little-noise operation. The four-quadrant, active output stage sinks or sources current into reactive or resistive loads across the output voltage range. This type of output is cricial to achieve an accurate output response and high slew rate demanded by a variety of loads such as highly capacitive or reactive loads.

8.3 Microviewer camera

There are two aven's Mighty Scope 1.3M USB Digital Microscope one with features 10x-50x, 200x magnification with white LED illumination and the other features UV LED lights operating at 405nm. They have adjustable magnification from 10x - 200x, in one compact lens.

Figure 8.3:- Mighty Scope 1.3M USB Digital Microscope

8.4 TESTING EHD & AFF

While testing EHD & AFF together we need some particular things like high voltage supplier, high definition camera, syringe pump and nozzle itself. For testing the Taylor cone from both the system we need to use these instruments manually.

Testing is done with three liquids to test the theory. These liquids are

- Isopropyl alcohol
- Vegetable oil
- wax

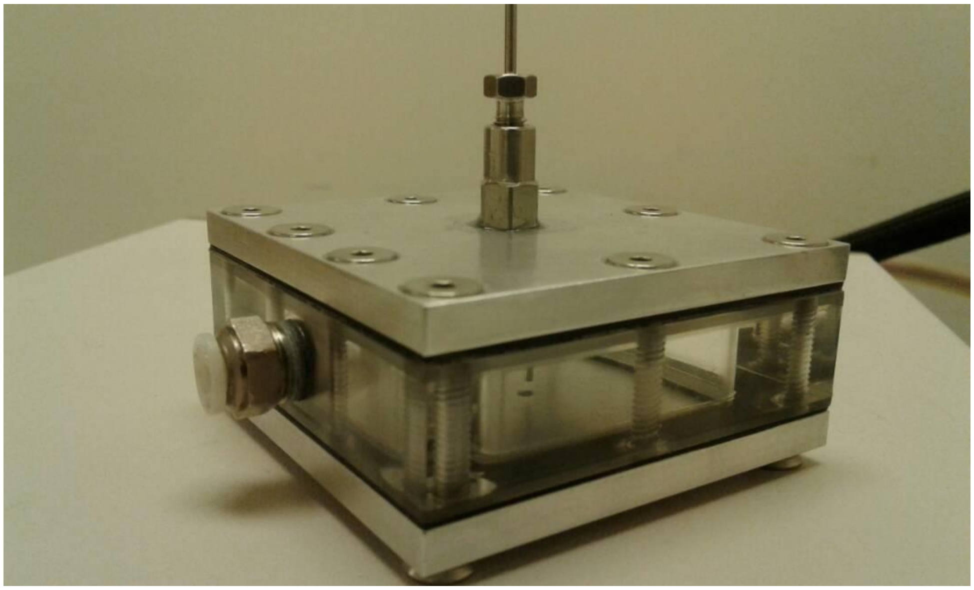


Figure 8.4:- nozzle chamber

The main part of this experiment is the nozzle which is built like a chamber. This chamber consists of metallic plates at both the end while chamber itself is made up of polycarbonate, which is an insulator.

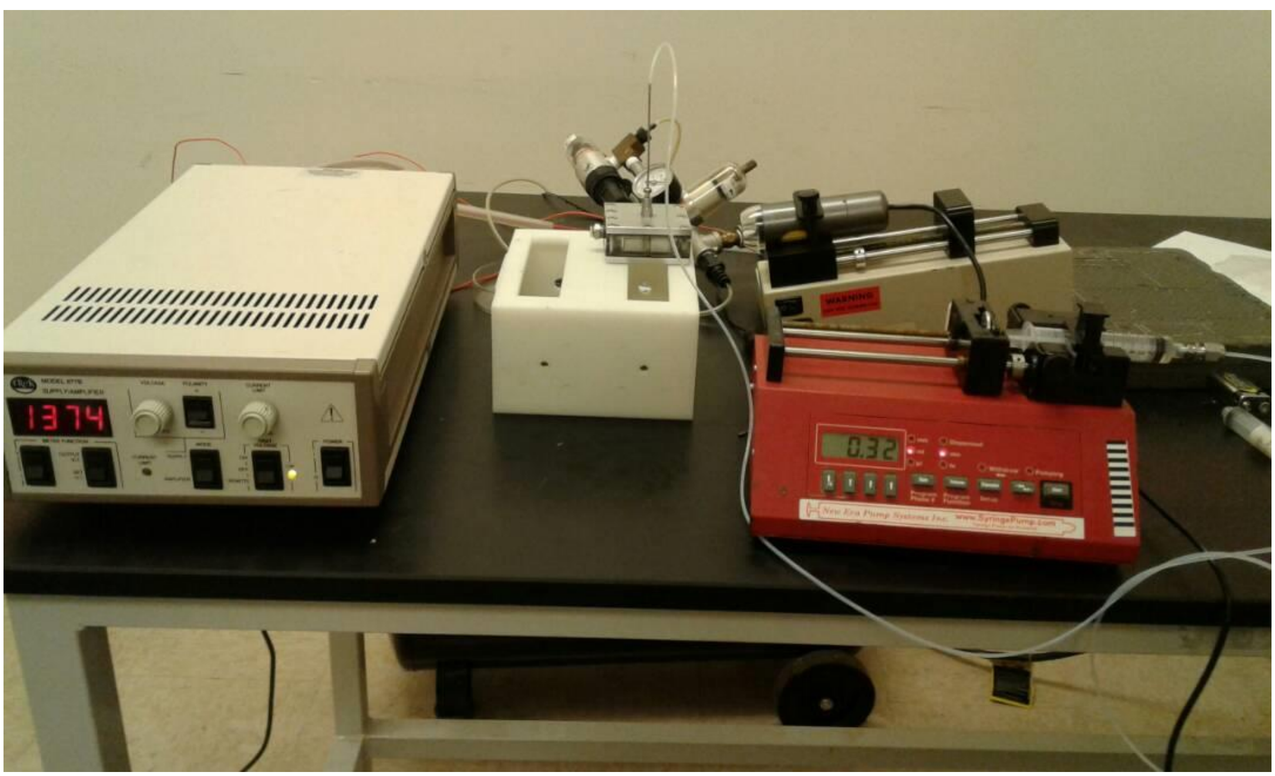


Figure 8.5:- test set up

8.5 Test result

Test results are quite satisfactory as we were able to see the Taylor cone with jet. For the EHD system the voltage requirement almost reached to 2000V for the given samples. This provided a jet which can be seen in the figure 81.

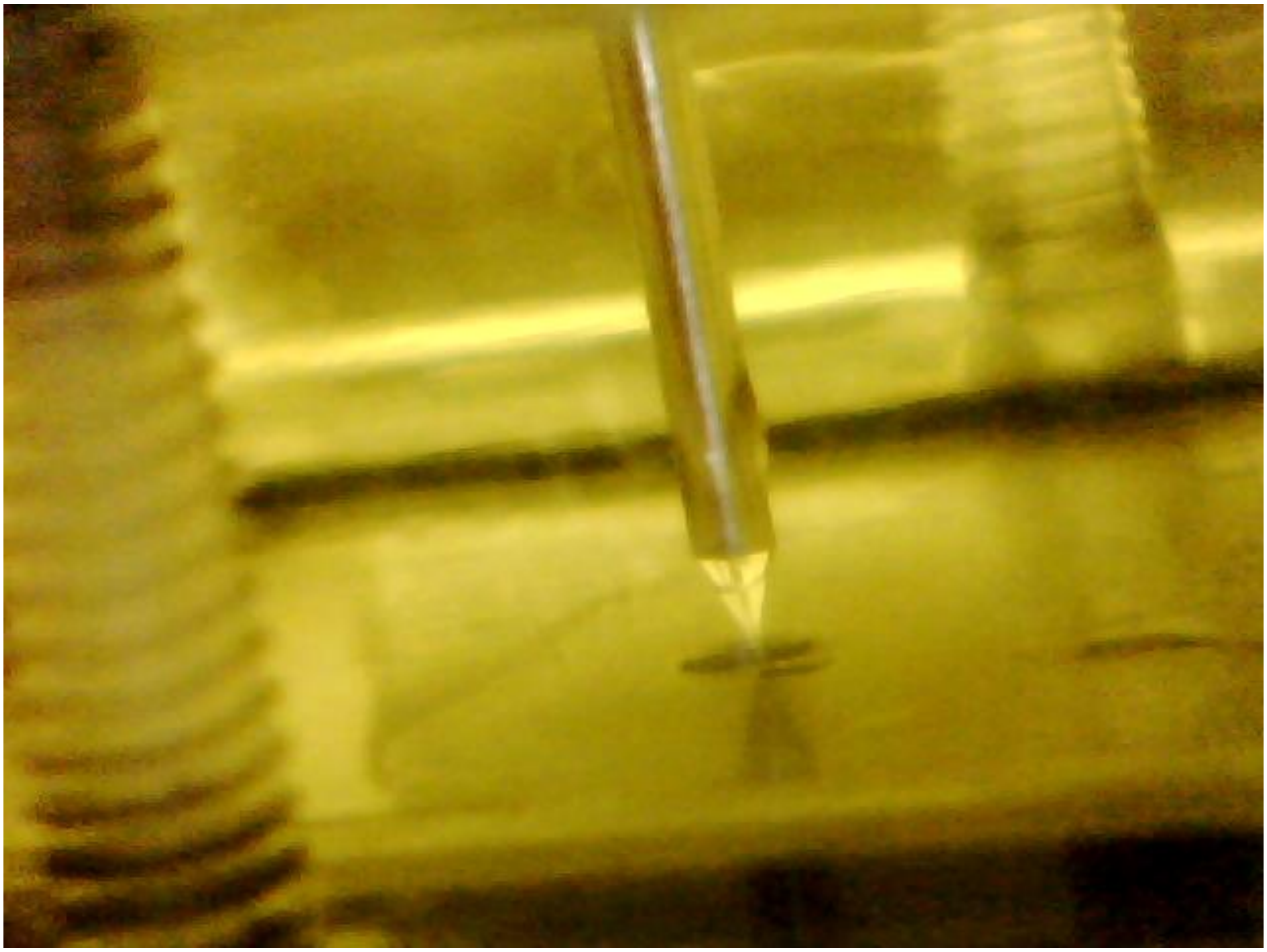


Figure 8.6:- EHD experiment

The AFF system also worked in the same way as predicted. We were able to see Taylor cone with jet. The pressure required for this experiment lies within the range of 10 to 20 psi.

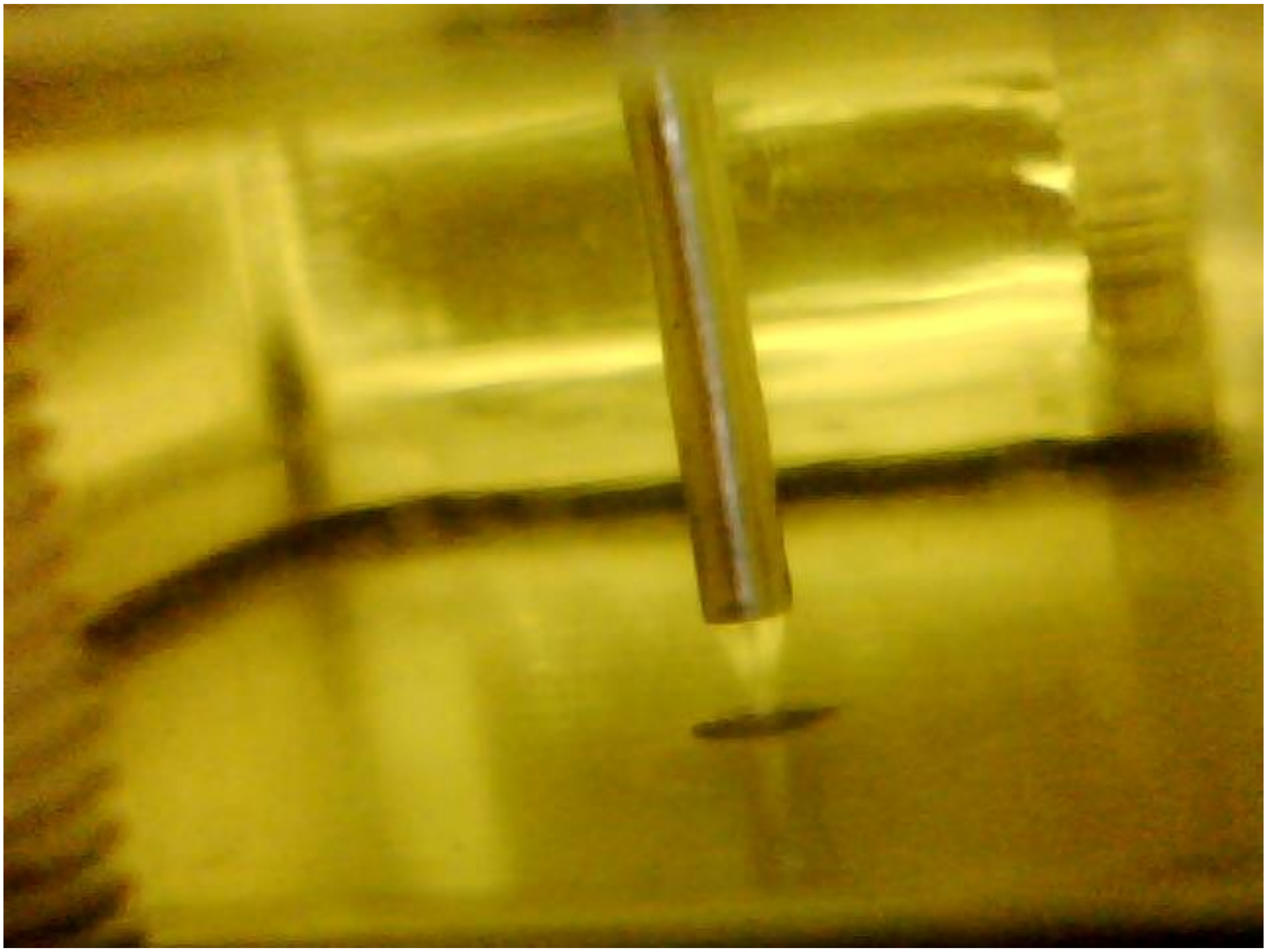


Figure 8.7:- AFF experiment

CHAPTER 9: CONCLUDING REMARK AND FUTURE WORK

9.1 Conclusion

Within the given time we fully developed our plan, constructed our printer, tested it to verify that it works, and have learned a lot about integrating it with the software. To begin we came up with a design for a printer that meets all of our requirements and have fully constructed it. To be sure that our printer build was fully functional we then used a large amount of open and closed source software in conjunction with microcontrollers like HDBB (High Definition Breakout Board) and Arduino to get it operational. Finally we began the transition to hardware and software through the development of a GUI which can run by

LabVIEW. The experiment shows very satisfactory results both for EHD and AFF. The only part required is the fully integration of all sort with the Labview. We plan on continuing the development of the GUI and the development of a firmware for use with a fully functioning and integrated 3D printer.

9.2 Future work & development

Although this project has a lot of scope to develop, future plans has to continue development into the final finished product. Our goal is to build a fully functional EHD & AFF Printer which has to be only controlled by LabVIEW. By doing this we can have our own prototype which can be developed through our project and be sure that it is fully functional and meets all expectations by the competition.

LIST OF REFERENCES

1. Lee, Michael P., et al. "Development of a 3D printer using scanning projection stereolithography." Scientific reports 5 (2015).

2. Song, Seung-Joon, et al. "A Three-Dimensional Bioprinting System for Use With a Hydrogel-Based Biomaterial and Printing Parameter Characterization." Artificial organs 34.11 (2010): 1044-1048.

3. http://www.ece.rutgers.edu/files/capstone/capstone2014/NI-3D-Printer/Final-Paper.pdf

4. Moore, Jessica L., et al. "Behavior of capillary valves in centrifugal microfluidic devices prepared by three-dimensional printing." Microfluidics and Nanofluidics 10.4 (2011): 877-888.

5. https://www.pololu.com/product/1182

6. http://home.hit.no/~hansha/documents/LabVIEW/training/Introduction%20to%20LabVIEW/Introduction%20to%20LabVIEW.pdf

7. Au, Anthony K., et al. "3D-printed microfluidic automation." Lab on a chip15.8 (2015): 1934-1941. Kollender, Jan Philipp, et al. "Highly customisable scanning droplet cell microscopes using 3D-printing." Journal of Electroanalytical Chemistry 740 (2015): 53-60.

8. Gelber, Matthew K., and Rohit Bhargava. "Monolithic multilayer microfluidics via sacrificial molding of 3D-printed isomalt." Lab on a chip 15.7 (2015): 1736-1741.

9. http://www.ni.com/pdf/manuals/373427j.pdf

10. Grunert, Ronny, et al. "ElePhant-An anatomical Electronic Phantom as simulation-system for otologic surgery." Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE. IEEE, 2006.

11. http://www.ni.com/academic/students/learn-LabVIEW/signal-processing/

12. Choi, Jae-Won, Ho-Chan Kim, and Ryan Wicker. "Multi-material stereolithography." Journal of Materials Processing Technology 211.3 (2011): 318-328.

13. http://www.cncdrive.com/products.html

14. Mallik, Abhishek, and Sauvik Das Gupta. "Modelling of MEMS based temperature sensor and temperature control in a petrochemical industry using LabVIEW." Computer and Automation Engineering, 2009. ICCAE'09. International Conference on. IEEE, 2009.

15. http://www.ni.com/product-documentation/10632/en/

16. Baohuai, WANG Xiao ZHANG. "Development of the Measuring and Controlling System of Heat Exchanger Performance Testing Equipment Based on LabVIEW [J]." Journal of Mechanical Engineering 4 (2009): 057.

17. https://www.arduino.cc/en/Main/ArduinoBoardMega2560

18. http://www.ni.com/example/31408/en/

19. http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf

20. http://plessdeane.blogspot.com/2012/04/ohms-law-circuits-potential-energy.html

21. https://www.omega.com/manuals/manualpdf/M4640.pdf

22. Rezanka, Ivan. "Ink jet printer with integral electrohydrodynamic electrodes and nozzle plate." U.S. Patent No. 4,658,269. 14 Apr. 1987.

23. Choi, Kyung Hyun, Muhammad Zubair, and Hyun Woo Dang. "Characterization of flexible temperature sensor fabricated through drop-on-demand electrohydrodynamics patterning." Japanese Journal of Applied Physics 53.5S3 (2014): 05HB02.

24. Shin, Jeongsik, et al. "EHD as sensor fabrication technology for robotic skins." SPIE Sensing Technology+ Applications. International Society for Optics and Photonics, 2014.

25. Sumida, Yoshitaka, and Shinichiro Takahashi. "Abnormally detecting device for relay." U.S. Patent No. 5,568,025. 22 Oct. 1996.

26. Schwartz, N. "Solid state relay." U.S. Patent No. 3,767,942. 23 Oct. 1973.

27. Endo, Ichiro, et al. "Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets." U.S. Patent No. 4,723,129. 2 Feb. 1988.

28. Lumma, Dirk, et al. "Flow profile near a wall measured by double-focus fluorescence cross-correlation." Physical review E 67.5 (2003): 056313.

29. Loscertales, Ignacio G., A. Barrero, and Manuel Márquez. "Production of complex nano-structures by electro-hydro-dynamics." MRS Online Proceedings Library Archive 860 (2004).

30. Joets, A., and R. Ribotta. "Electro-hydro-dynamical convective structures and transitions to chaos in a liquid crystal." Cellular structures in instabilities (1984): 294-306.

31. Boyer, John S., and Edward B. Knipling. "Isopiestic technique for measuring leaf water potentials with a thermocouple psychrometer." Proceedings of the National Academy of Sciences of the United States of America (1965): 1044-1051.

32. Călinoiu, D., et al. "Arduino and LabVIEW in educational remote monitoring applications." Frontiers in Education Conference (FIE), 2014 IEEE. IEEE, 2014.

33. Schwartz, Marco, and Oliver Manickum. Programming Arduino with LabVIEW. Packt Publishing Ltd, 2015.

34. Jamaluddin, Anif, et al. "Design real time battery monitoring system using LabVIEW interface for arduino (LIFA)." Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T), 2013 Joint International Conference on. IEEE, 2013.

35. Jaber, Alaa A., and Robert Bicker. "Real-time wavelet analysis of a vibration signal based on arduino-uno and labview." International Journal of Materials Science and Engineering 3 (2015): 66-70.

36. Petru, Livinti, and Ghandour Mazen. "PWM Control of a DC Motor Used to Drive a Conveyor Belt." Procedia Engineering 100 (2015): 299-304.

37. Kuan, Wen-Hsuan, et al. "Development of a Computer-Assisted Instrumentation Curriculum for Physics Students: Using LabVIEW and Arduino Platform." Journal of Science Education and Technology 25.3 (2016): 427-438.