2-axis Egg-Plotter Control Using Digilent Analog Discovery 2

by

Liew Ee Mei

Faculty of Applied Sciences

Tunku Abdul Rahman University College

Kuala Lumpur

2017/18

2-axis Egg-Plotter Control Using Digilent Analog Discovery 2

by

Liew Ee Mei

Project supervisor: Prof. Hor Yew Fong

This is a project dissertation submitted to the Faculty of Applied Sciences in partial fulfilment of the requirement for the award of Bachelor of Science Degree, Tunku Abdul Rahman University College.

Department of Physical Science

Faculty of Applied Sciences

Tunku Abdul Rahman University College

Kuala Lumpur

2017/18

Acknowledgment

We give all honor and praise to my project advisor, Prof. Hor Yew Fong and Prof. Lim Hee Chuan who has guided, assisted and advised me for the entire duration of the projection. Prof. Lim always give me supports and guide to me how to do my project in purpose to produce a good outcome from research that been studied.

We also express my gratitude and thanks to my teammate, Mr Chu Jaan Horng who work hard to produce a good project with all afford and responsibility. My teammate also helped and guided me during project.

Finally I would like to thanks the TARUC for providing the instrument or the completion of this object.

Abstract

2-axis Egg-Plotter is an embedded system that operates based on the principle Computer Numerical Control (CNC). Robotic 2D Plotter basically works with two stepper motors and a servo motor. The machine plots the input fed from the computer and plots the art, image and text on the surface of the sphere object using Digilent Analog Discovery 2 as the controller. The Robotic 2D plotter has a two axis control and the arm to bring up or lower the pen. Each axis are powered and driven by using the A4988 driver. Pen control is accomplished using a servo motor. The software used to program the Digilent Analog Discovery 2 is namely as NI LabVIEW (2016). The software used to generate the G-code file is called Inkscape (version: 0.48.5). The open-source software used to simulate the G-code file is called CAMotics. The accurate and efficient organisation and proper function of the programs along with the circuit builds up an efficient 2-axis Egg-Plotter (CNC).

Contents

List of Figure

Chapter 1 Introduction

1.1. Overview

The source of this stimulus of this project comes from the difficulties draw on spherical or egg-shaped objects from the size of a ping-pong ball to the size of a tennis ball.

Rather than using human's hands to draw on the spherical or egg-shaped objects which have difficulty drawing a fine pattern on a size of a ping-pong ball and low efficiency of the human's hand repeating the pattern on spherical or egg-shaped objects.

This project works based on principle Computer Numeric Control (CNC) machine. CNC refers to a machine whose robotic control of multi-axis machines such as mills, laser cutters or printer that make work without human help. CNC is a term for a device that established to control the motion and operation of machine tools. CNC is worked via discrete numerical value fed into the machines. Most of the CNC machines are driven through G-code language. G-code provides some features such as feed rate and coordinates. There are several advantages to using CNC machines. The process is more accurate than manpower and it can be duplicated exactly to the same degree of precise any several times style.

The machine tool movements used in producing a goods. The machine tools have two elementary types of movement: straight line movement and contouring movement. The straight line movement is the connecting point to point and do not have curvature. The contouring movement is connecting points of equal height. The Cartesian coordinate system is used in CNC to numerically give direction and plane of movement.

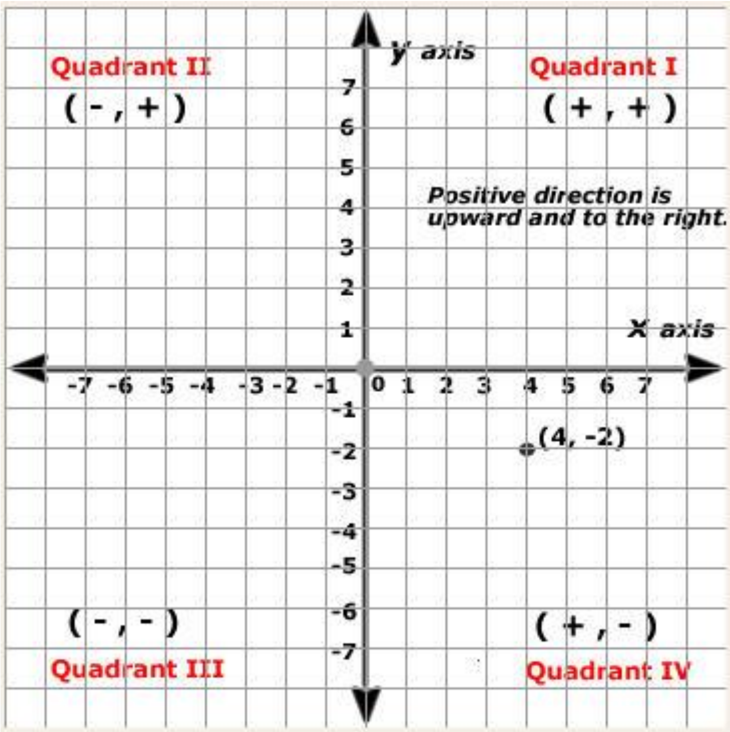


Figure 1 Cartesian coordinate system.

From: https://sites.google.com/site/pd2math201516/home/polynomial-functions/relations-and-functions

This system labelled the any stated points along the three perpendicular axes. In this project, the x-axis is rotated the egg with horizontal, the y-axis is from or towards the egg's poles and the z-axis is the vertical movement of the pen. The plus (+) and minus (-) signs represent the direction from the origin along the axis as shown in Figure 1.

2-axis Egg-Plotter is the branch of technology that handles design, building, operation and application of a plotter, as well as the computer system for their control. The construction of this project will frequently integrate the principle of mechanical engineering, electronic engineering, and computer science. 2-axis Egg-Plotter is an embedded system using Digilent Analog Discovery 2 as a controller. 2-axis Egg-Plotter is a plotter that provides the fastest way to professionally yield very fine drawings.

Egg-Plotter will be able to print an image by moving a pen across the surface of a spherical object. Egg-Plotter can print complex line art as well as text, but the process slowly due to the limitation of stepper motor and some delay.

1.2. Objectives

The objectives of this report:

a. To develop an egg-plotter by NI USB 6009 DAQ device.
b. To build an egg-plotter system by NI LabVIEW software to program the A4988 stepper drivers to drive the stepper motor and a SG-90 servo motor.

1.3. Literature Survey and/or theory

Machine plotter on a spherical or egg-shaped object has been completed many time formerly, but the most present solution seems to be 2-axis multi-purpose CNC plotter machines. Another project that is more similar to an egg-plotter is the Eggbot as shown in Figure 2. Eggbot is an open source egg painter that is capable design art on eggs surface with a pen. The motivation here is instead in the software by what method to translate art into machine movements.

The origin of Eggbot was created by Bruce Shapiro in spring of 1990. The early Eggbot drawn on a round object can be done with relatively simple mechanics.

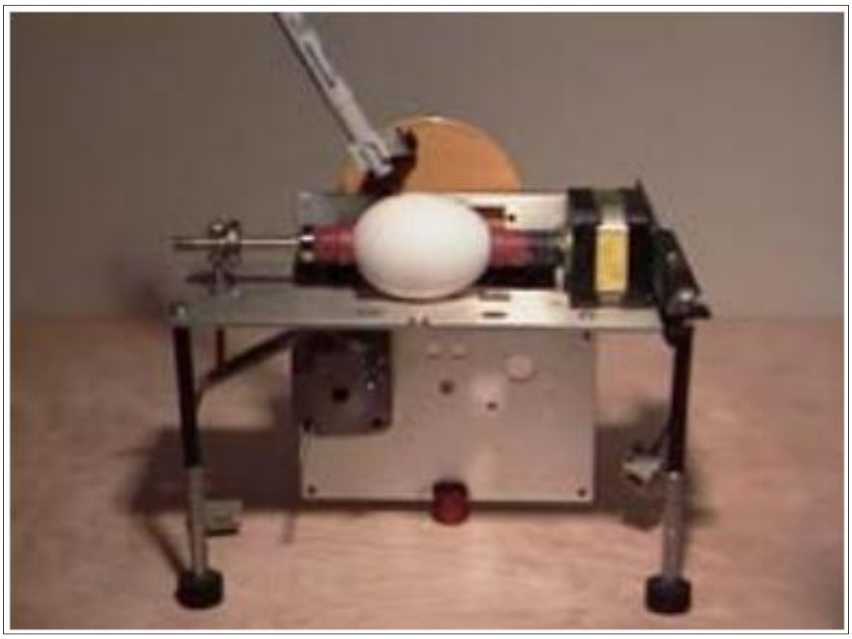


Figure 2 an early Eggbot.

From: https://egg-bot.com/history/

In this chapter, introduction of this project, objective and literature survey have been presented. In the next chapter, discusses the design and the implement of egg-plotter.

Chapter 2 Design and Implementation

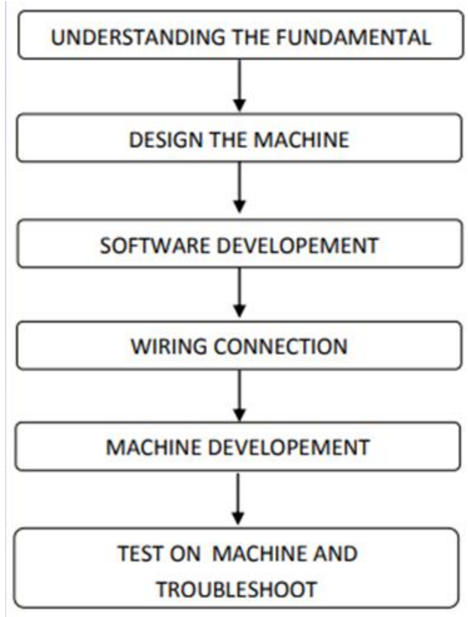


Figure 3 flow of the whole system

It is essential to design the architecture of software, hardware and electronic modules in order to implement the 2-axis egg-plotter. The flow of the whole system from beginning to end as shown in Figure 3. In this chapter, the design and implement of egg-plotter will be described. Before beginning to plan the system, must understand the fundamental to operate the stepper motor and servo motor. The software module operates in real-time. The software module generates the low level commands to the electronic module. The electronic module is doing some wiring connection and takes the low level commands as the input and processes the commands. The electronic module generates the control signals and passes to the motor. Finally, can test the system and troubleshoot it. This project contains software part, electronic part and hardware part.

2.1. Software Part

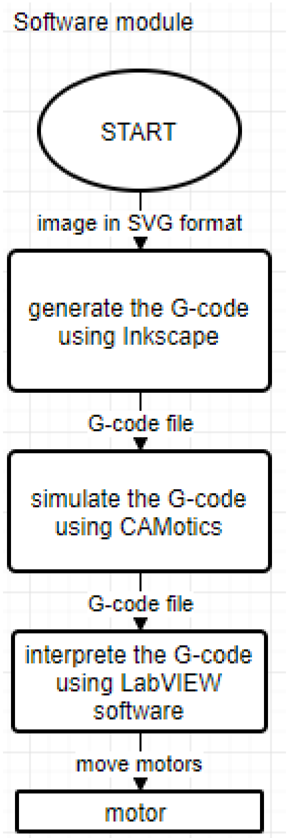


Figure 4 software flow chart

The software module is a part of a computer system that consists of computer instructions. The flow of the software part as shown in Figure 4 .The software used in this project comes under open source. Open source offers the chance to contribute new features and bug fixes back into the public. The programs and tools picked for this project are open source and practice international standards. The project software system consists of: Inkscape (version 0.48.5), CAMotics and LabVIEW 2016.

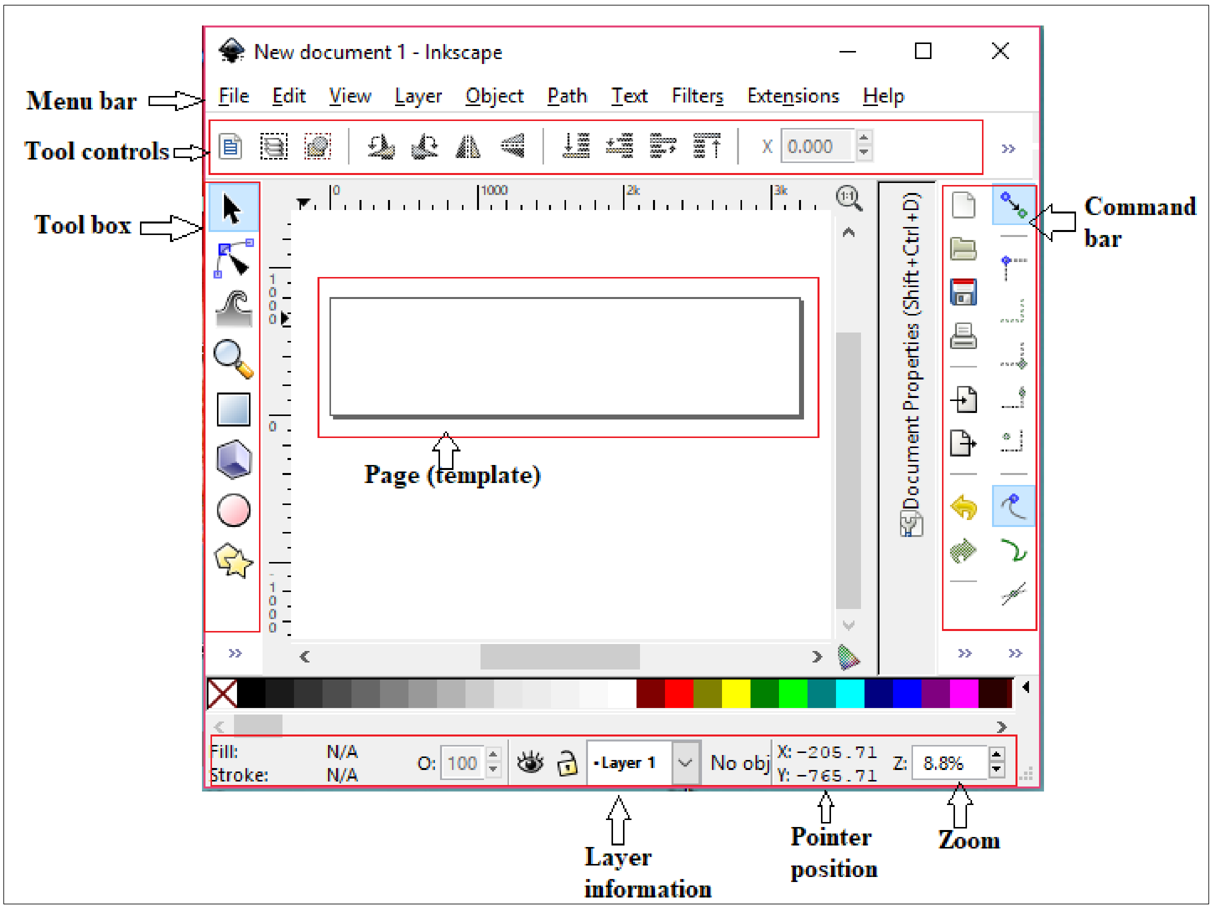


Figure 5 Inkscape window

The Inkscape software provided two basic types of graphic images which are bitmap images and vector images. Bitmap images are defined in term of rows and columns of individual pixels. The vector image is defined in term of lines. The difference of these graphic images can be clearly evident when the image is blow up as shown in Figure 6. The bitmap resolution is significantly less than the display resolution, the resultant image will show rough line. Inkscape is a vector graphics editor and it can used to create or edit the vector graphic as shown in Figure 5. The vector graphic format used in Inkscape is SVG format.

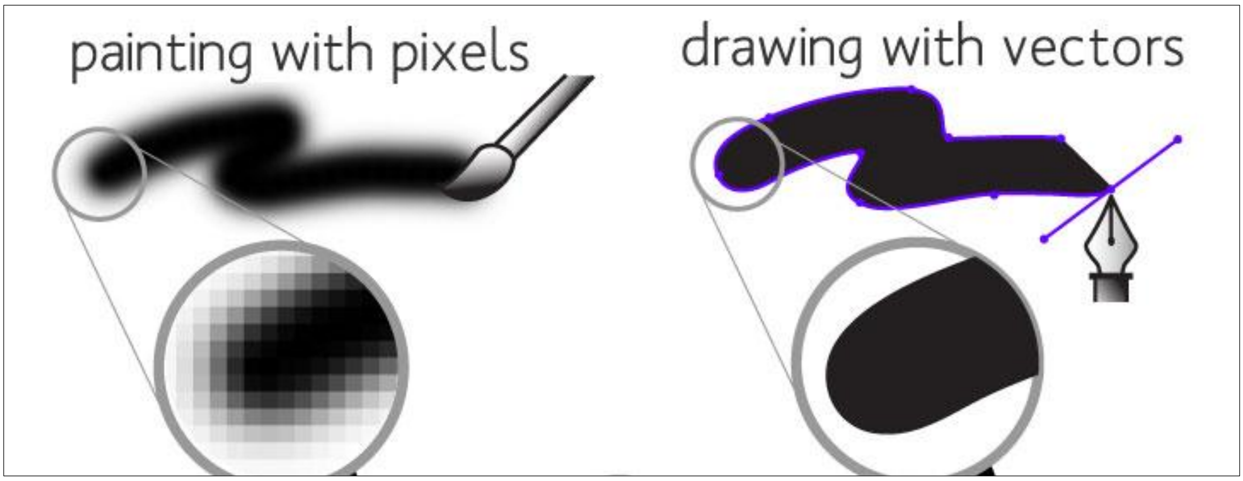


Figure 6 Bitmap image and the vector image are blow up

From: http://vector-conversions.com/vectorizing/raster_vs_vector.html

Turn your ideas into prints, need download a plugin extension from github <*https://github.com/martymcguire/inkscape-unicorn*>. This plugin is Gcodetools extension generates the G-code from the SVG input and writes it to a file as a side effect of the SVG transformation. Install the plugin is very simple, copy the files located in SRC folder and paste it under this path *C:\ProgramFile\Inkscape\share\extensions*. This extension provided some features to adjust the angle the servo motor, XY axes feedrate in millimeters per minute unit, origin point, the number of copies and pause on layer changes as shown from Figure 7 to Figure 9.

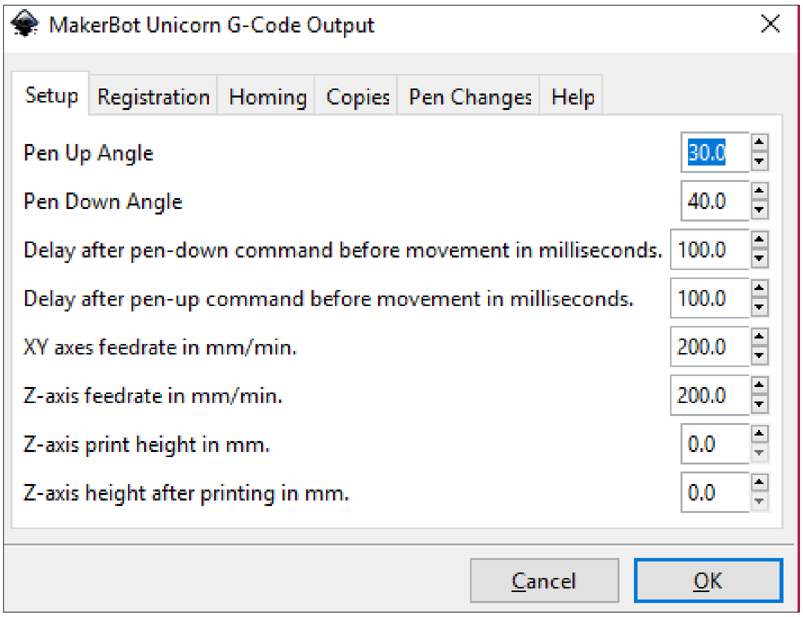


Figure 7 Setup the angle of servo motor, delay after pen up/ down, xy axes feed rate

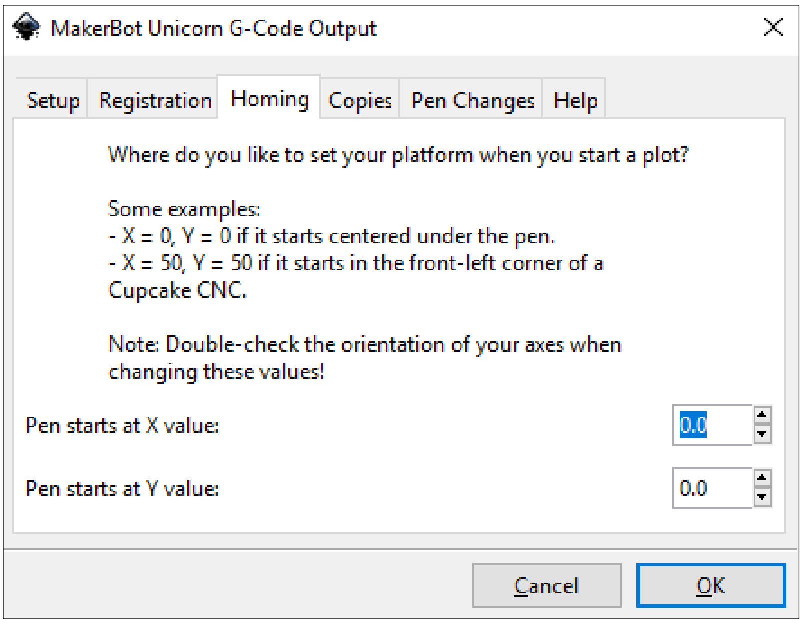


Figure 8 initial point

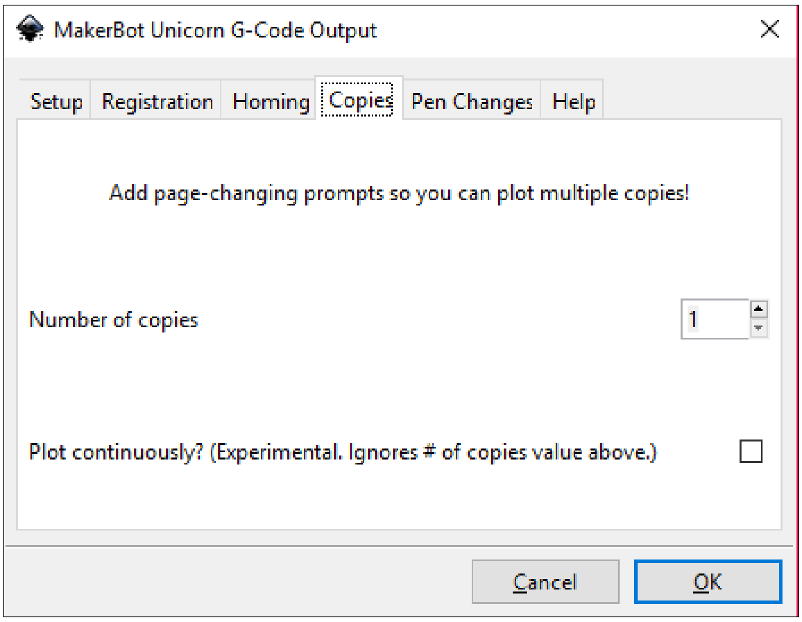


Figure 9 setting number of copy

Inkscape used 2 dimensional Cartesian coordinate system in term of (x, y) axis. In Inkscape software, x-axis running horizontally across the computer screen and y-axis running vertically. The x-axis value increase from left to right and y-axis increase from top to underside. The Inkscape drawing template requires user resizing the template. With this 1:1 resizing resolution, each pixes is equal to one stepper motor step. One stepper motor step is 1.8 degree divide by 16 (0.1125 degree). Identify the Inkscape's x-axis is the egg-rotation and y-axis is the pen arm travel. Inkscape drawing template with a width of 3200 pixels and a height of 1000 pixels. The width of the template is matching to the complete circumference of a sphere object, and the height matches to the extreme range of pen travel. Technically, these stepper motors have only 200 individual steps per revolution.

The detail of stepper motor and driver will discuss at the electronic part section. However, used a driver that provided one-sixteenth microsteps the software sees the stepper motors as having 16x200 steps. Use absolute position (0,0) as a homing point (origin) as shown in Figure 8. Presume that the pen is placed at the point (width/2, height/2). Inkscape provides a layer plate that makes it more user-friendly and easier to manage. Inkscape's layer can document the image with given a name to the current image. The advantage of layer plate is the pen will pause on the layer and pop out a window. Before each layer begin to draw an option will be presented to the user to change the pen colour.

Once, successfully generate the G-code file then use CAMotics as a simulation software to simulate the G-code. The CAMotics is an open source software which simulates 3-axis CNC milling or engraving as describe in Figure 10. The constraint of this software is simulated only pictures of the cutting operation. The below Figure 11, shows the simulated motions from the G-code by CAMotics.

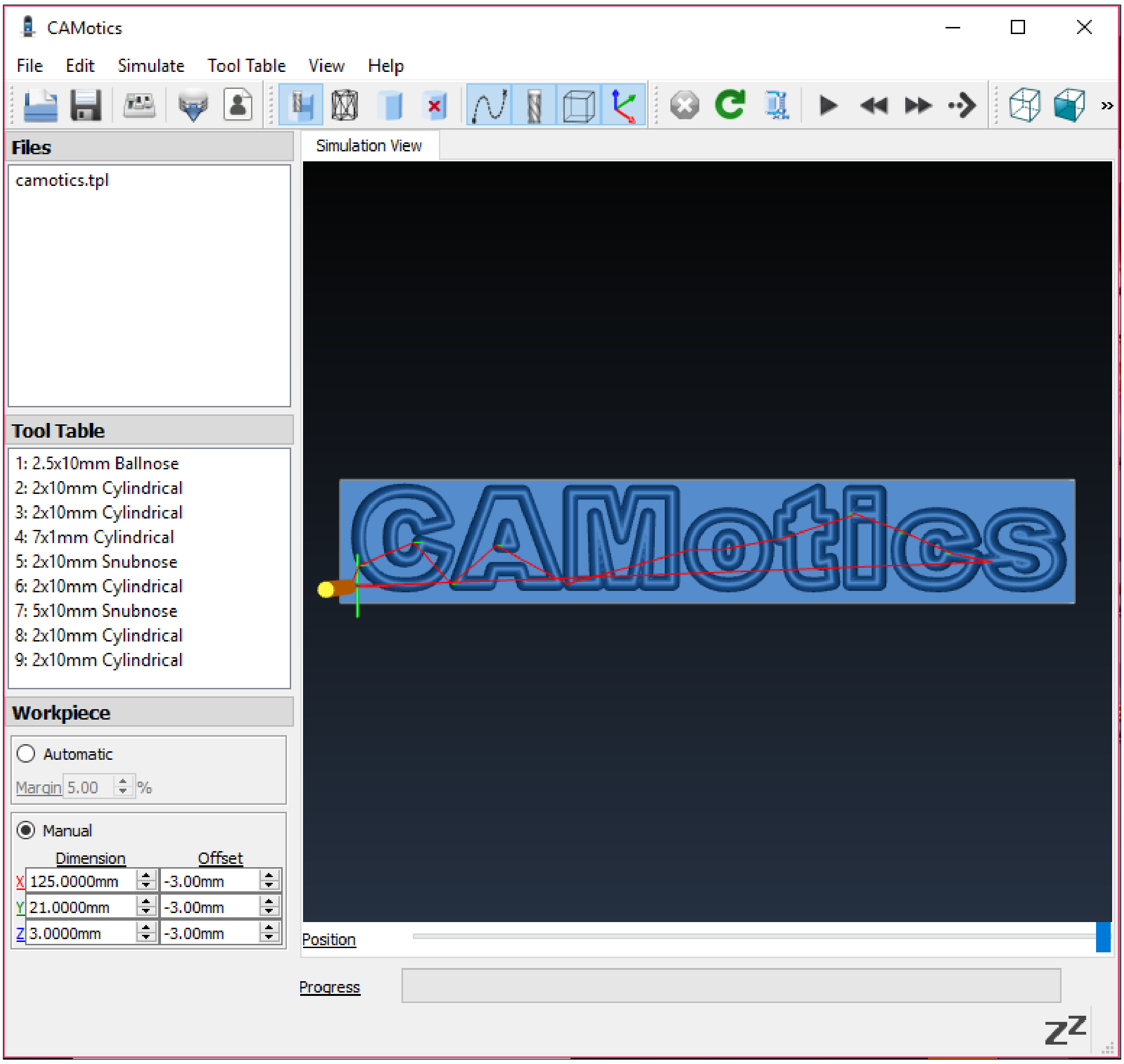


Figure 10 CAMotics software

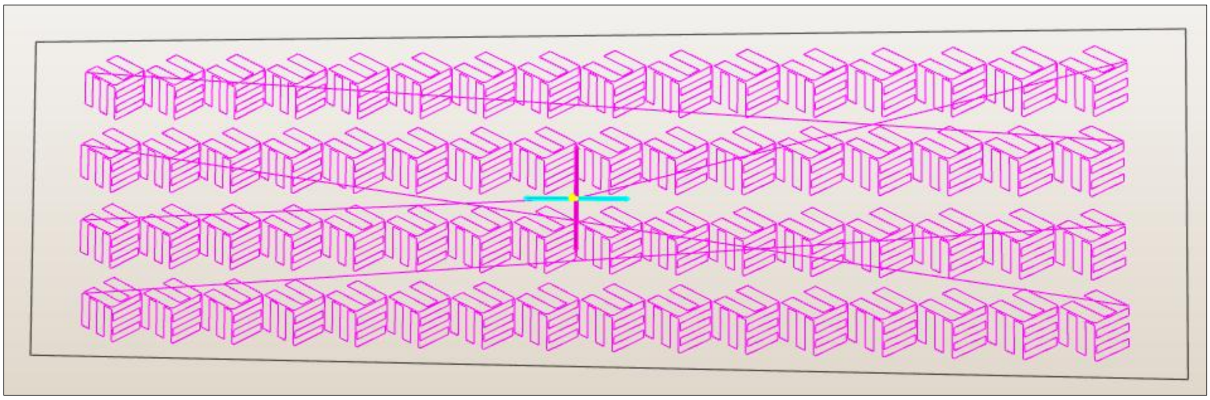


Figure 11 Simulation result by CAMotics software

The NI LabVIEW 2016 software used to program the Digilent Analog Discovery 2 as shown in Figure 12. Before that, user can install the Waveforms 2015 as shown in Figure 13. This software provides a user interface for a 2-channel oscilloscope, logic analyser, arbitrary waveform generator, digital I/O, and more. The default application that allows user to spontaneously run the instruments on your Analog Discovery unit is WaveForms 2015 by Digilent.

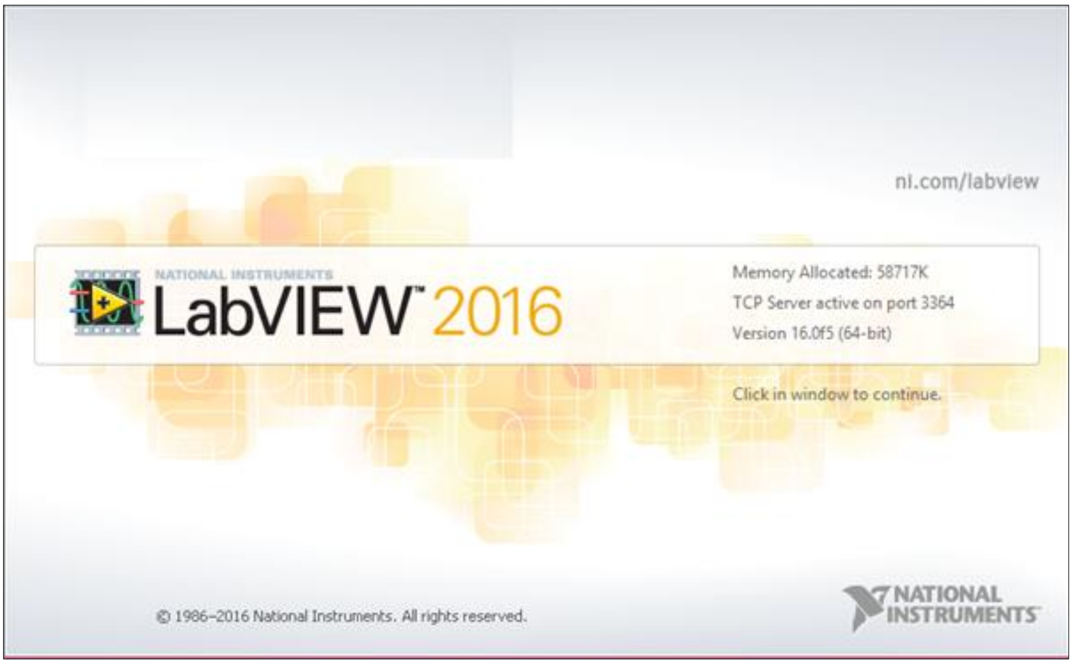


Figure 12 NI LabVIEW 2016

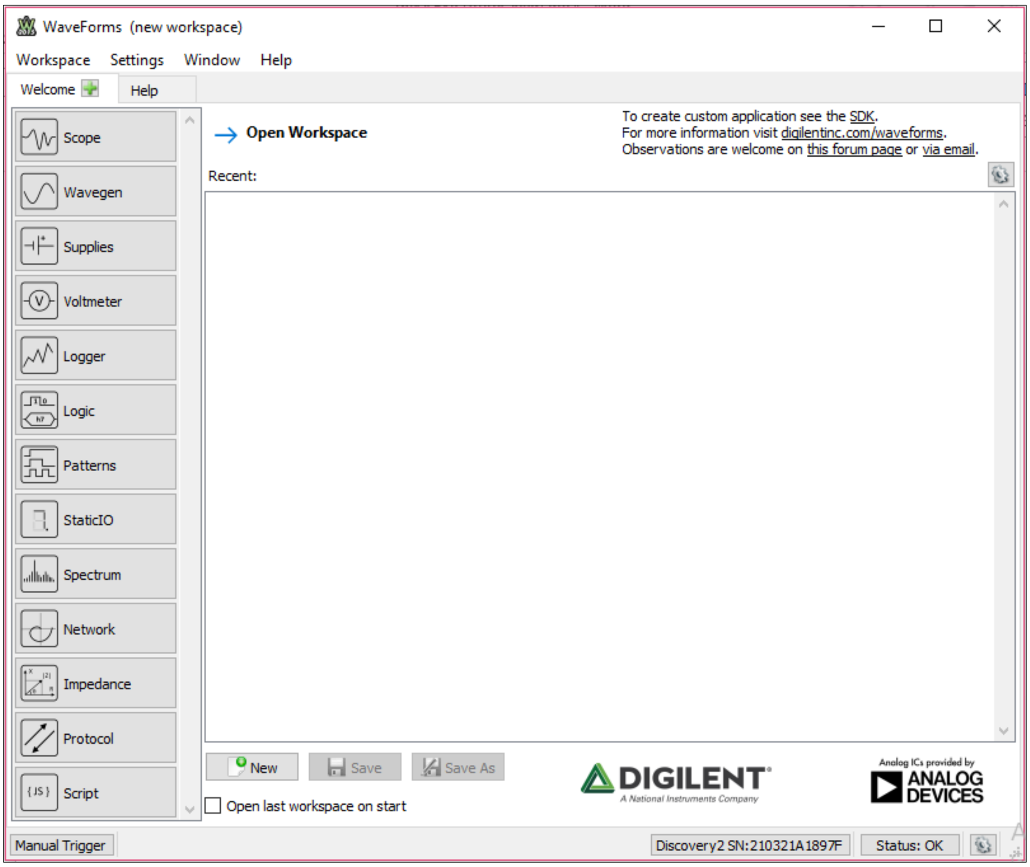


Figure 13 Waveforms 2015 window

In this project, using the Digilent Analog Discovery 2 as a heart of egg-plotter that able to generate the signals to move the motors.

This instrument able to program the signals with LabVIEW software. The Analog Discovery 2 gives the command from the computer as G-code to the Analog Discovery 2 instrument to get the plotter output. Before starting to program the LabVIEW, need installs the Digilent WaveForms VI's package (shown in Figure 15) from VI package manager 2017 (shown in Figure 14) that includes a comprehensive set of functions that allows users to program Analog Discovery 2.

Figure 14 VI package manager 2017 software

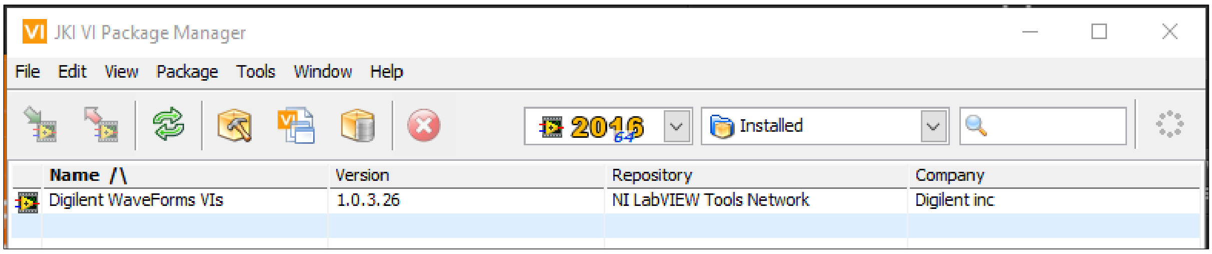


Figure 15 VI package manager window

After install the Digilent WaveForms VI's package, the VI's package is added into the NI LabVIEW function palette as shown in Figure 16.

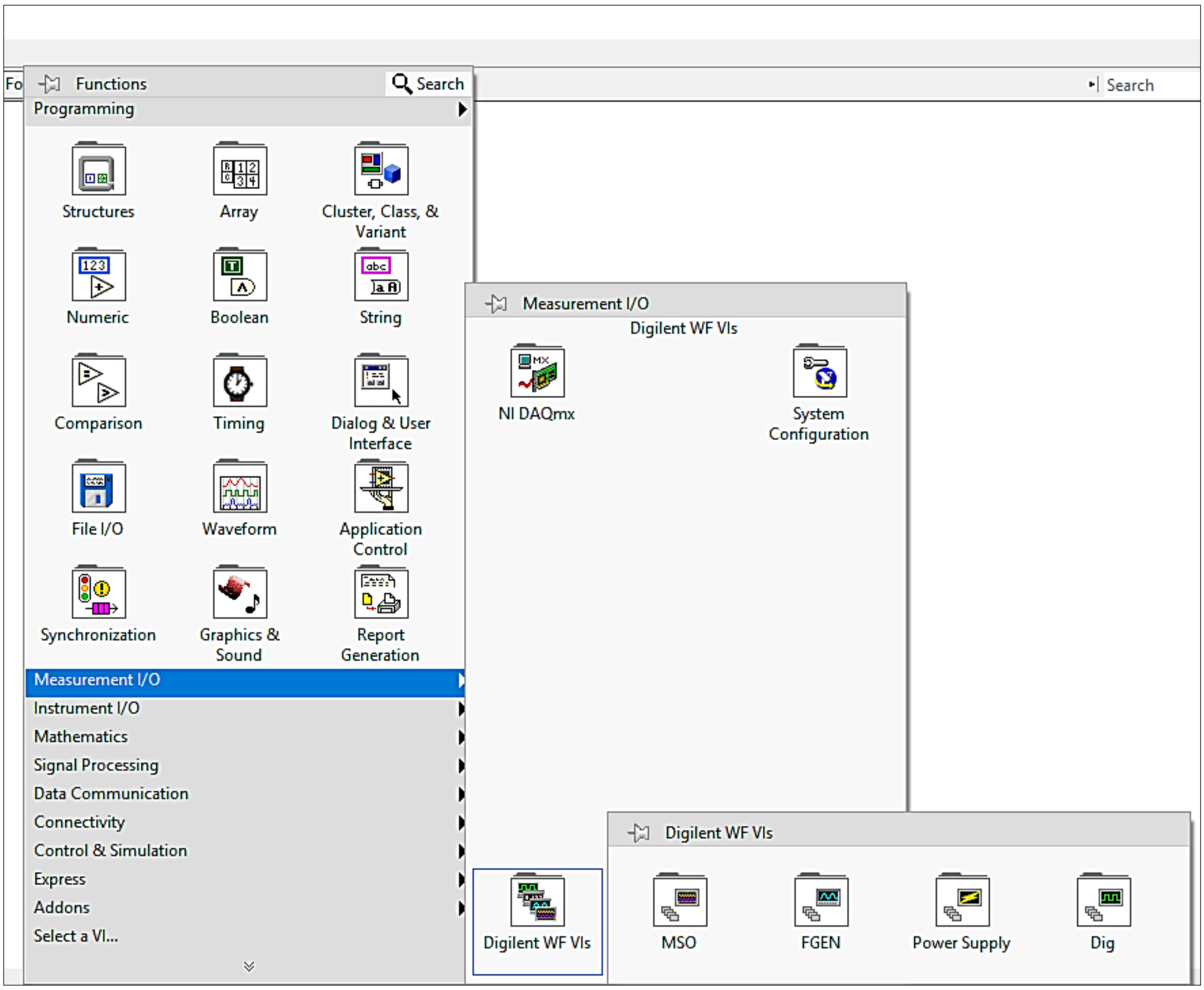


Figure 16 Function Palette

The next step is interpreting the G-code using LabVIEW software. G-code is programming language that provided instructions to the machine tool such as motors where to move and what kind of path to follow. The G commands (first letter is G in G-code) often given instruction to control kind of gesture such as rapid positioning, linear feed, circular feed, fixed cycle or offset value. The M command is an action code and usually people say that the 'M' stands for machine. The LabVIEW program is a very basic G-code translator that implements the G1, G92, G4, M300 and M01 commands that enough to print out an image.

The G1 commands provide coordinated linear motion to the destination point at the current feed rate. The basic linear drawing algorithm used in computer graphics is

Bresenham's line Algorithm. This algorithm was developed to draw lines on digital plotter. This algorithm normally used to draw lines in a bitmap image. This algorithm is fast since only used integer addition, subtraction and bit shifting operation. The Bresenham's algorithm considers giving an initial point (x_1, y_1) and end point (x_2, y_2). Determine the driving axis to be x-axis if $|\Delta x \geq \Delta y|$ or the y-axis if $|\Delta y > \Delta x|$. The driving axis is applied as the axis of control for the algorithm and is the axis of maximum motion. The coordinates corresponding to the other axis is only incremented as necessary. Regard the following presented as shown in Figure 17(a), to pull out a line from (0, 0) to (3, 1). Since x is the driving axis in this example, it then increments the x coordinate by one. Rather than keeping track of the y coordinate (which increases the slope by m = $\Delta y/\Delta x$, each time the x increases by one), the algorithm maintains an error bound ϵ at each stage, which represents the negative of the distance from the point where the line exits the pixel to the top edge of the pixel (see the figure).

The following shown calculate the error bound ϵ at each stage:

$$1 - \epsilon = m$$

$$-\epsilon = m - 1\ , \text{-}\ \epsilon\ represents\ the\ negative\ distance, the\ sign\ (-) can\ be\ ignored.$$

$$\epsilon = \frac{\Delta y}{\Delta x} - 1$$

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

This value, ϵ is first set to m − 1, and is incremented by m each time the x coordinate is incremented by one. The value of error, ε requires the use of floating arithmetic to calculate the slope, m of the line. The value of error able convert to integer arithmetic by multiplying ϵ with $\Delta x, \bar{\epsilon} = \Delta x \epsilon = \Delta y - \Delta x$. If $\bar{\epsilon}$ becomes greater than zero, the line has moved upwards one pixel, and that increment y coordinate and readjust the error to represent the distance from the top of the new pixel as shown in Figure 17(b)– which is done by:

$$-\epsilon = \epsilon \text{ -1, - } \epsilon\ represents\ the\ negative\ distance.$$

$$\bar{\epsilon} = \Delta x \epsilon$$

$$\bar{\epsilon} = \bar{\epsilon} - \Delta x$$

If the $\bar{\epsilon}$ is less than zero, increment x coordinate and readjust the error by:

$$-\epsilon = \epsilon + \text{m, - } \epsilon\ represents\ the\ negative\ distance.$$

$$\bar{\epsilon} = \bar{\epsilon} + \Delta y$$

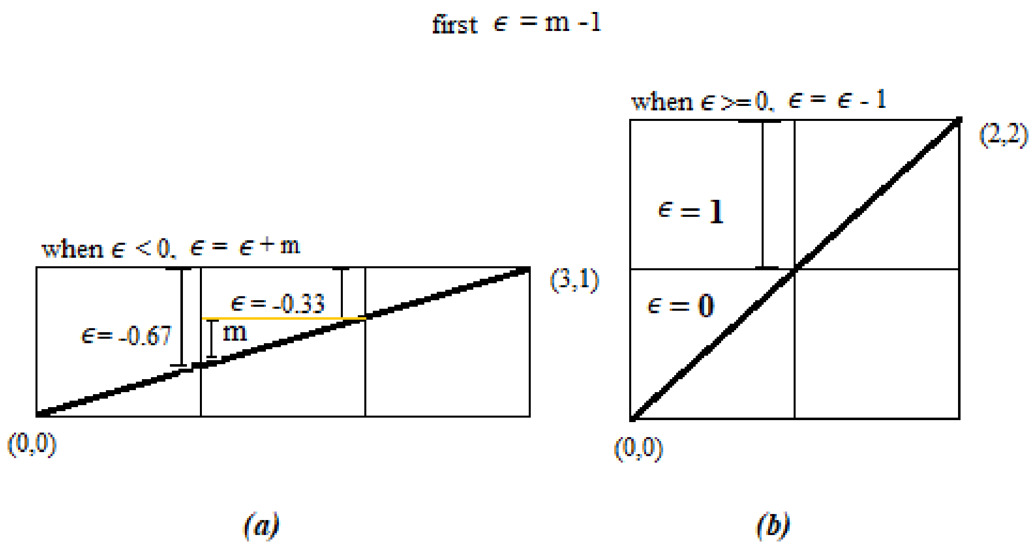


Figure 17 linear line implement Bresenham Line Algorithm

The G92 X~ Y~ command is to create the current point have the coordinates, where the axis words have the current axis numbers. The G4 P~ command will preserve the axes frozen for the period of time in millisecond definite by the P number. The M300 S~ command will change servo motor angle specified by the S number. The M01 command will let the system pause until the user make an option.

The LabVIEW program will begin to let the user to enter the total number of G-code line and import the G-code file path. Next the user can start to run the system. If the user does not enter any value for the total number of G-code line will pop out a window to alert the user to key in value. The program will display the current instruction, number of instruction has been executed, initial value of coordinate x, initial value of coordinate y, initial angle of servo motor, speed of the stepper motor in millisecond (ms), delay of whole system in millisecond (ms) and the current angle of servo motor. Then, egg-plotter machine start to plot the image, art or pattern on the surface of sphere object. The front panel of the system as shown in Figure 18. The block diagram of the system as shown in Figure 19.

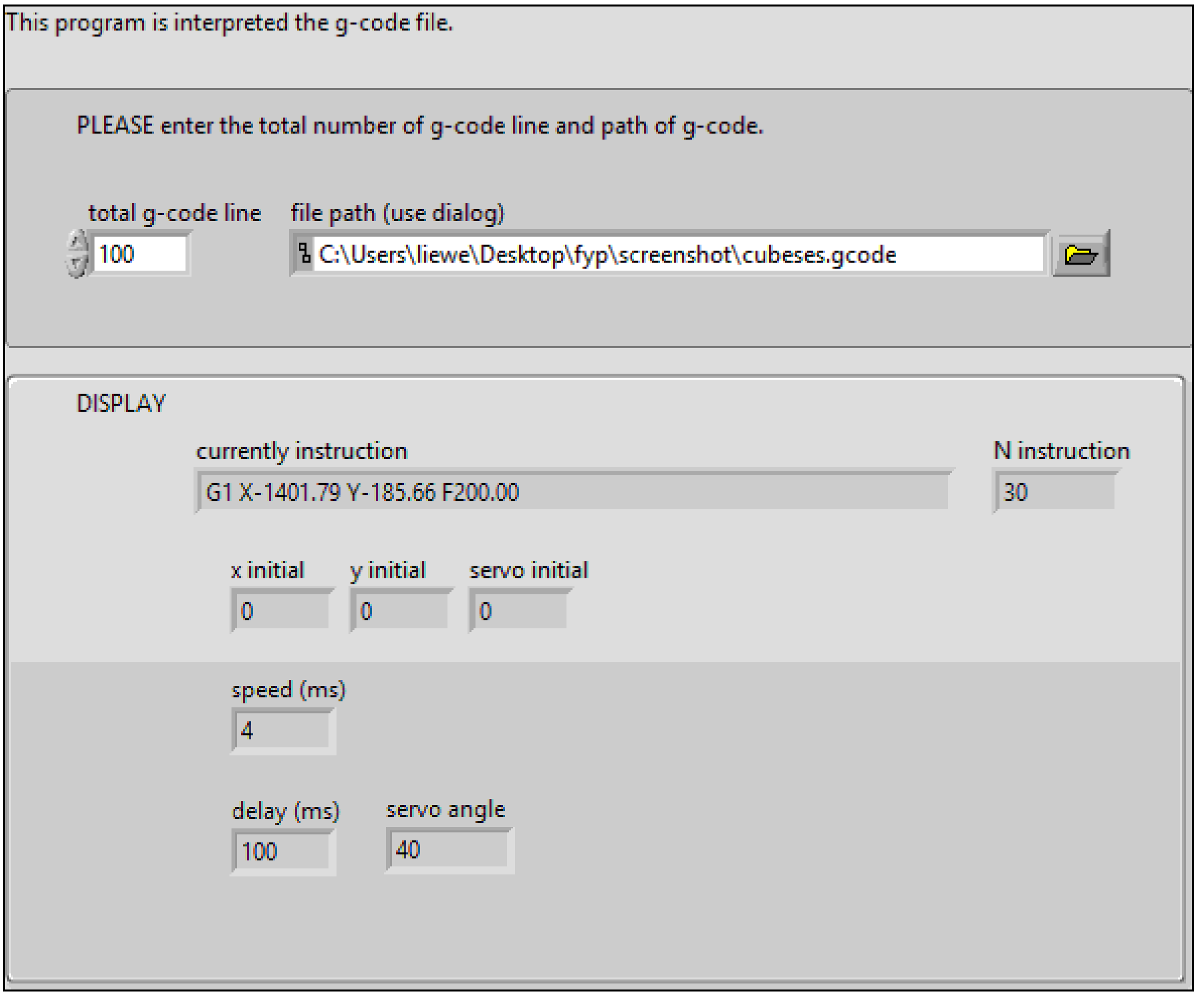


Figure 18 front panel of the system

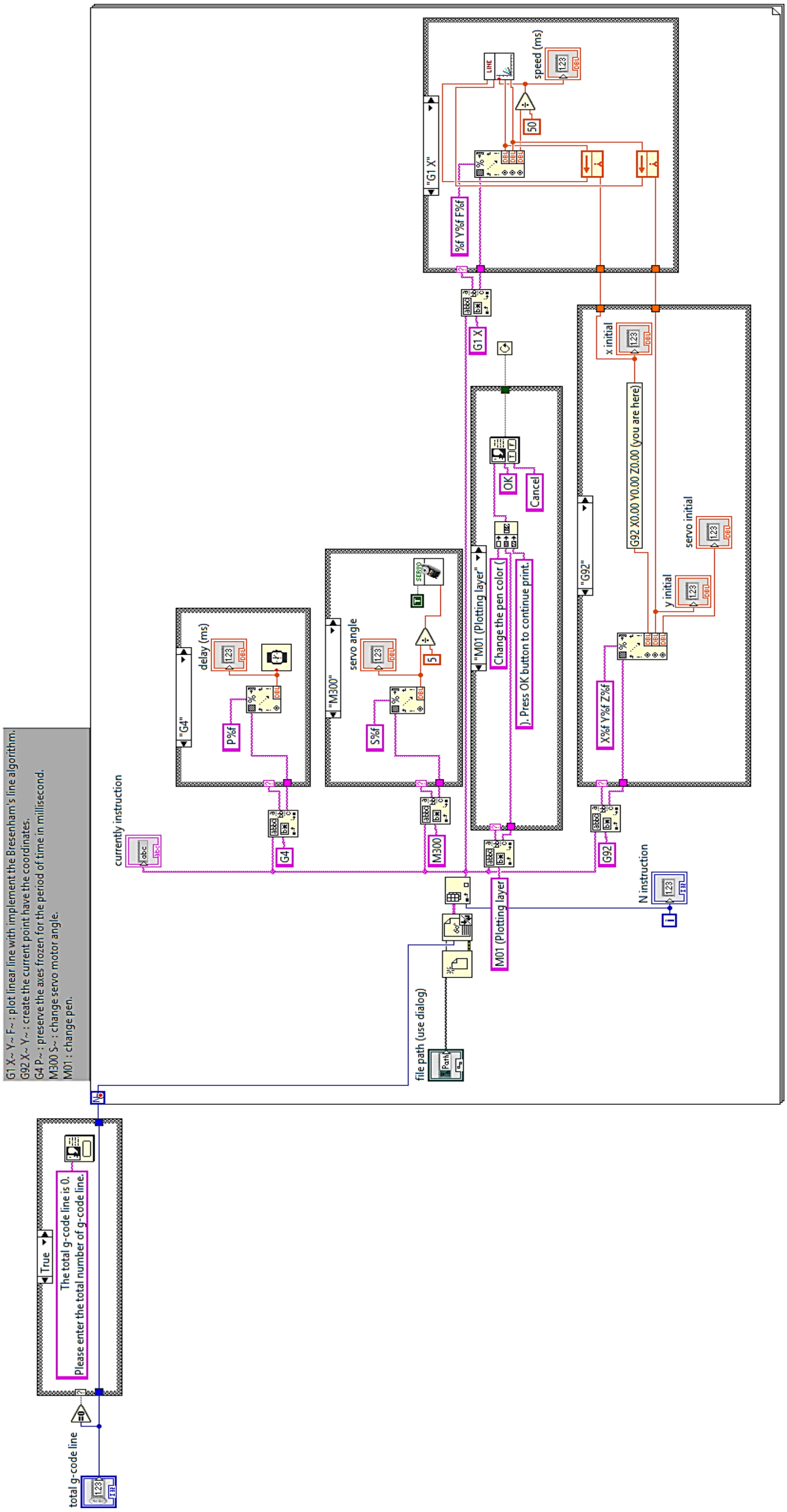


Figure 19 block diagram of the system

2.2. Electronic Part

This section discuss about the wiring connection and the spec of the motor as well as controller. The x axis and y axis is motivated by a stepper motor driven by a A4988 motor driver circuit. The z axis is driven by a servo motor. The Digilent Analog Discovery 2 is a USB oscilloscope and multi-function instrument that permits users to measure, visualize, generate, record and control mixed-signal circuits of all varieties. Analog Discovery 2 provided two programable power supplies(0v to +5VDC and 0v to -5vDC), two channel waveform generator (±5V, 12MHz+ bandwidth), and 16-channel virtual digital I/O ports. The overview pins on this instrument as shown in Figure 20. The Analog Discovery 2 connects to USB port PC via a standard mini USB cable and allows the PC control the program during prints.

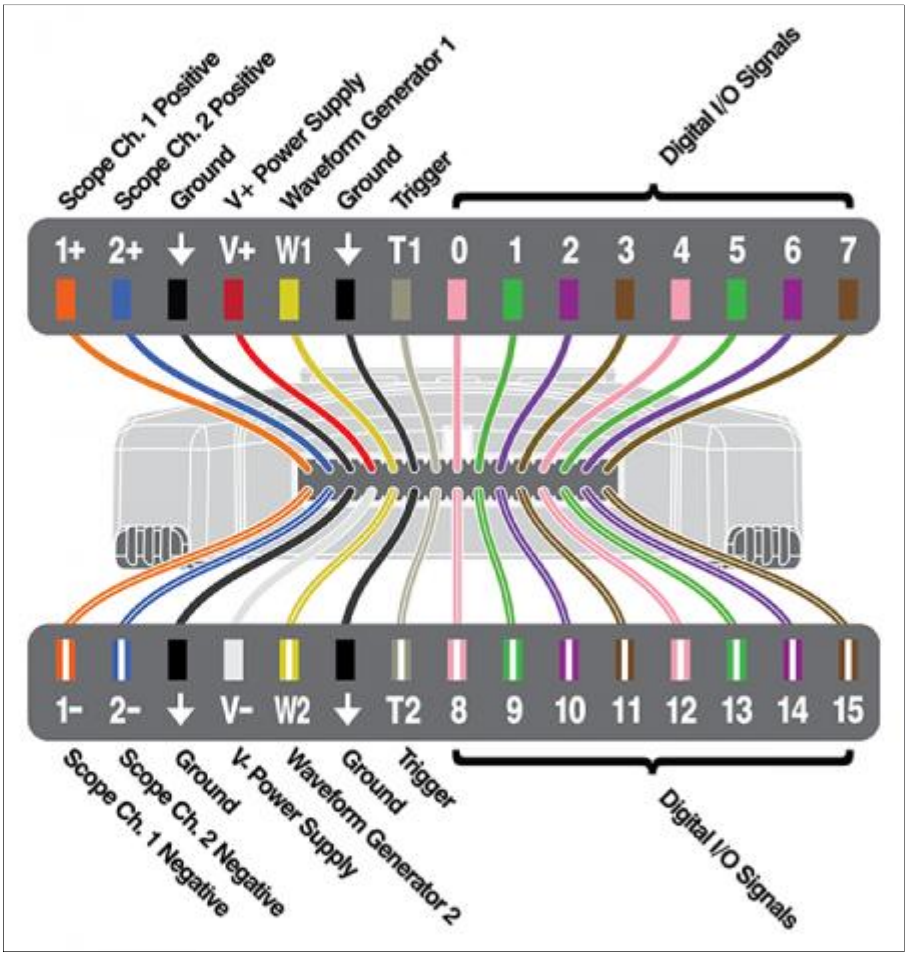


Figure 20 pin layout of Analog Discovery 2. From: Analog Discovery 2 Reference Manual

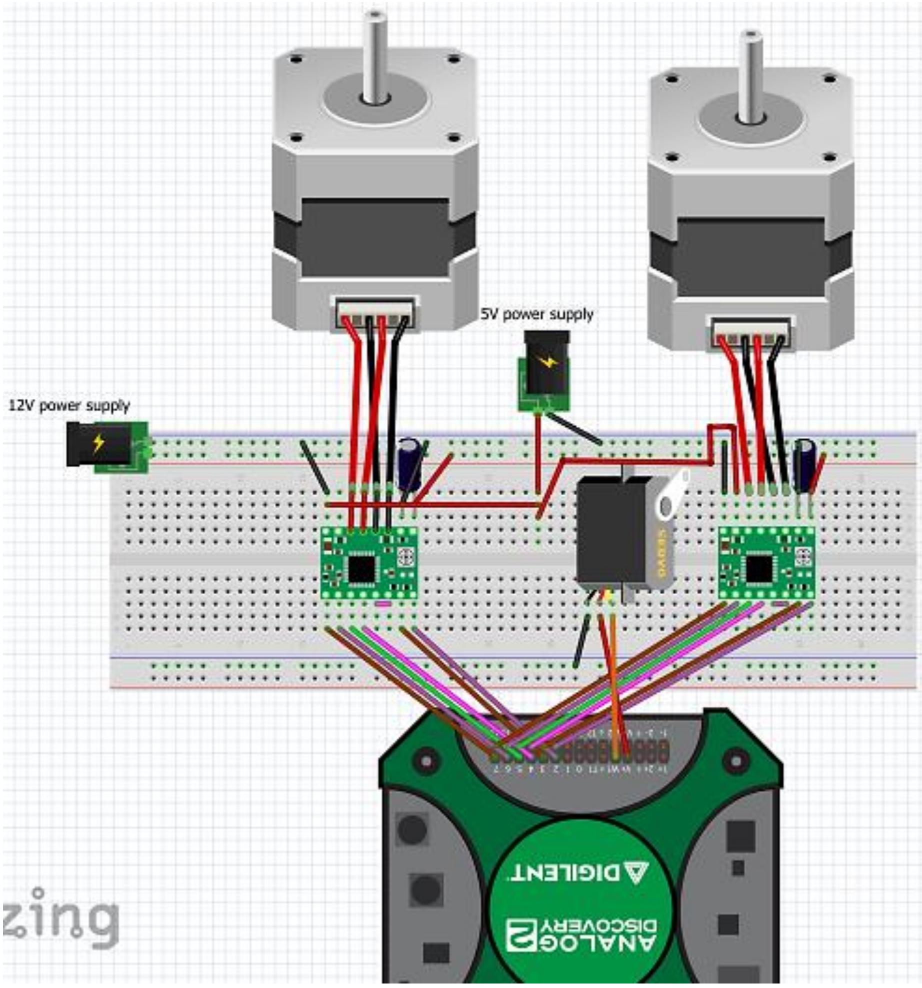


Figure 21 schematic diagram of whole system

Figure 22 back side of A4988 driver.

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

The A4988 motor driver is a very flexible since it can be arrange to specified clockwise or anticlockwise direction of rotation of the motor shaft and the number of degrees. The A4988 driver only drive the bipolar stepper motor. Users provide a pulse to the STEP pin, the stepper motor will rotate the shaft by 1, 1/2, 1/4, 1/8 or 1/16 of a step at a time based on precision wanted by users. Each A4988 stepper driver was connected to each stepper motor as shown in figure XX. The user can program the DIR pin of driver with given high level logic (1) to rotate the motor in clockwise direction or low level logic (0) to to rotate the motor in anticlockwise direction. Each driver consists of a dual H-bridge achieved by an electronic device that allows user to specify the direction of rotation of the electromagnetic field and of the stepper-motor shaft. The stepper motor can take a full step or fractions ranging from 1/2 to 1/16 depend on the combination logic on MS1 pin, MS2 pin and MS3 pin. The step size combination logic for pin (MS1, MS2, and MS3) from the five step resolutions according to the Figure 23.

In this project, choose the step size is 1/16 resolution so that the drawing art is fine and high resolution. The driver has three different inputs for controlling its power states: RST, SLP, and EN. The RST pin is a floating pin and user can connect it to the adjacent SLP. The EN pin has an internal build in a NOT gate and implement low level logic.

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Figure 23 combination logic for five resolution.

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

The VDD and Ground pins connect to a power supply of 3 to 5.5 V and in this project that will be DC to DC converter provide +5 V. The 2B, 2A, 1A and 1B pins are for connecting the motor. The 1A and 1B pins will be connected to one coil of the motor and the 2A and 2B pins to the other coil of the motor. The Ground and VMOT pins for powering motor and connect it to a power supply from 8 to 35 V and also need to use decoupling capacitor with at least 47 µF for protecting the driver board from voltage spikes. In this case, will be an external power supply provides a 12V with 1.0A of current and utilised a 100 µF capacitor as a driver protector. Each bridge can supply a current of 2A and the V_{REF} pin of driver allows user to adjust the bridge output current as shown in Figure 24. The Imax value is the maximum of stepper. The calculation formula for the I_{MAX} is showing at the following:

$$I_{MAX} = V_{REF} / (8 \cdot R_{CS})$$

rearranged to solve for V_{REF} :

$$V_{REF} = 8 \cdot I_{MAX} \cdot R_{CS}$$

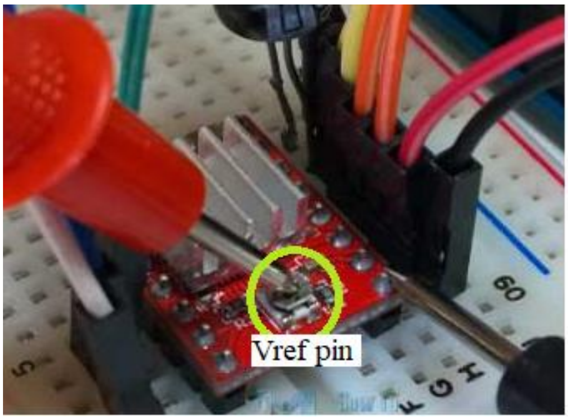


Figure 24 Vref pin of A4988 driver. From:http://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/

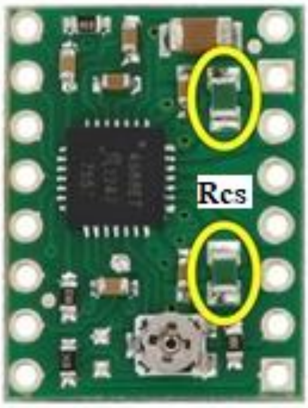


Figure 25 Front view of A4988 driver.

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

A good starting point for choice stepper motors is the 42SHD0217-24B model because it provided the 1.8 degrees in one full-step and the holding torque (3.2 kg-cm) able to hold the tennis ball weight even the egg. The stepper motor needs a power supply to power by with 12V but should even work within 8V to 35V. This stepper motor is a hybrid stepper motor and it can operate by both unipolar and bipolar stepper motor drivers. Each phase draws 1.2A current at 4V. In this project, make it as a bipolar stepper motor and center-tap yellow and white wires can be left disconnected as shown in Figure 26. User can drive the unipolar stepper motor by transistor circuit as shown in Figure 27. The lead of unipolar stepper motor must arrange in sequence. The both common center-tap connect to the +5V. The unipolar stepper motor less energize then the bipolar stepper motor.

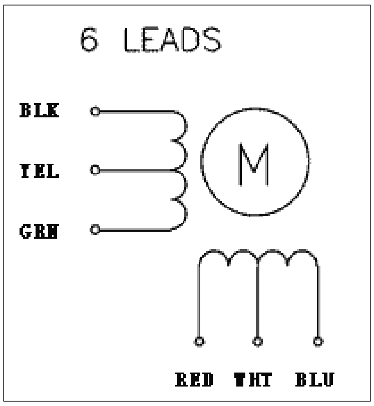


Figure 26 wiring diagram of hybrid stepper motor

From: https://www.pololu.com/product/1200

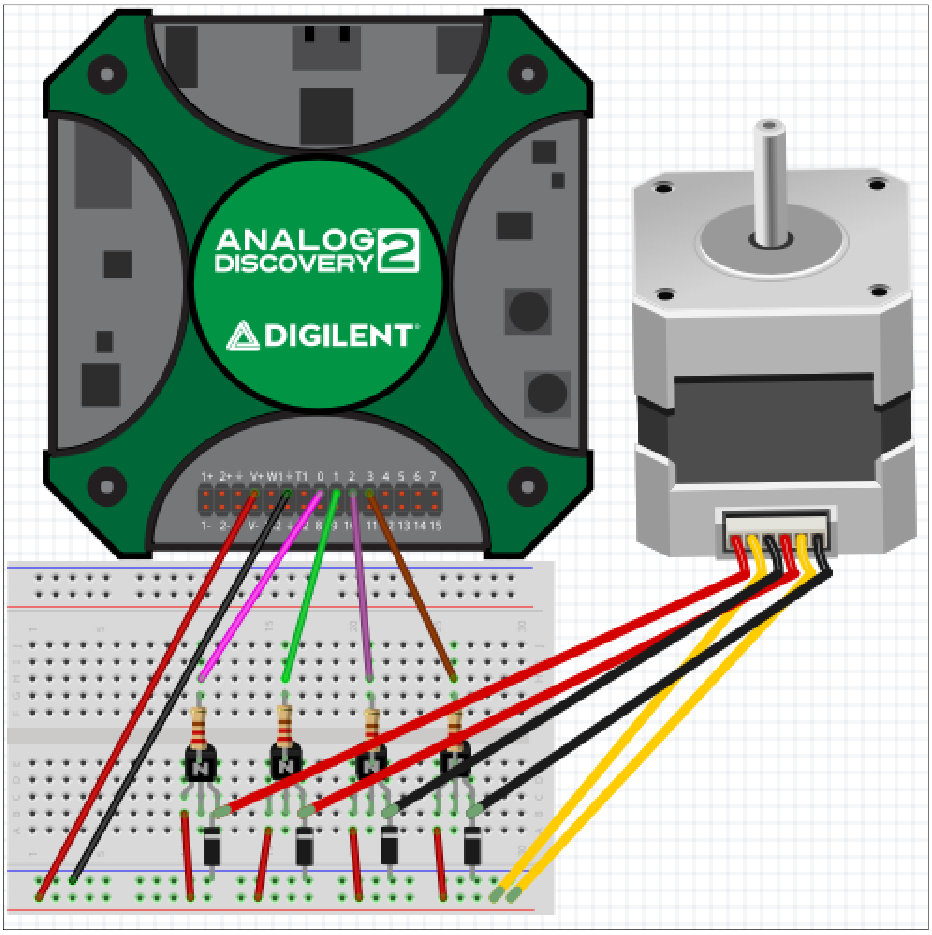


Figure 27 use the transistor circuit to drive the unipolar stepper motor.

The following will discuss about the servo motor. A servo motor is a device that can push or rotate an object with great precision. Usually used the servo motor to rotate and object at some specific angles or distance. There consist of two different types of servo motor which are DC motor and AC motor. In this project, used the DC servo motor (model: SG90) for lifting up or down the marker pen when drawing art on sphere object as shown in Figure 28. This servo motor can rotate approximately 180 degrees and the torque is 1.8kgf·cm sufficient to lift the pen holder and the Sharpie ultra-fine pen (total weight is 161gram±).

Figure 28 SG-90 servo motor

From: http://akizukidenshi.com/download/ds/towerpro/SG90.pdf

SG90 servo motor is controlled by Pulse Width Modulation (PWM) means its angle of rotation is controlled by the duration of the applied pulse to its orange wire. The waveform generator 1 pin (W1) of Analog Discovery 2 instrument connects to the orange wire of servo motor and generate the PWM at 50Hz. The servo motor expects to receive a PWM signal with the period of 20 millisecods (50Hz) in this period the servo motor only response to a high logic level for a period from 1 to 2 milliseconds. The Figure 29 shows the controlling the duty cycle of PWM at the 50Hz and the motor angular rotation. The servo motor works with +5V power supply. In this project, used the programmable power supply of Analog Discovery 2.

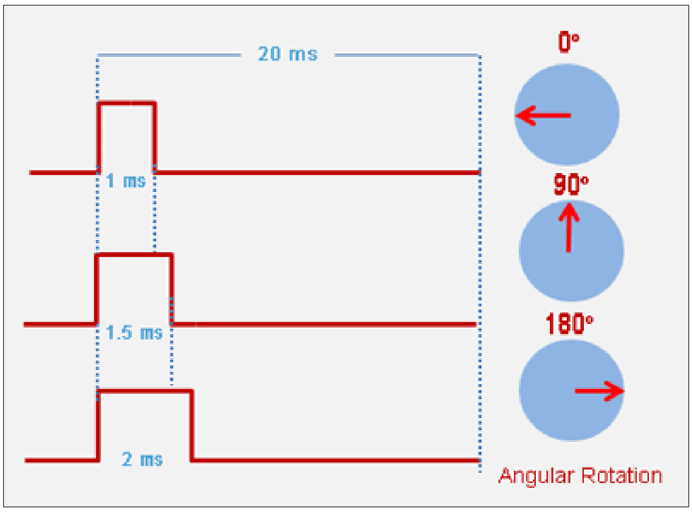


Figure 29 rotation angle depend on duty cycle

From: https://circuitdigest.com/article/servo-motor-basics

2.3. Hardware Part

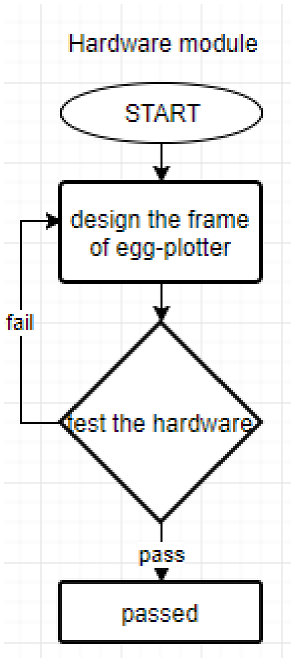


Figure 30 Hardware flow chart

This section will discuss about the hardware module. The whole process design the hardware will describe in Figure 30. The first design is made of wood as shown in Figure 31.

The second design of the two dimensions of mechanical design of the egg-plotter refers to the sphere-o-bot. This sphere-o-bot was designed by Attila Nagy. The 3D model can be obtained from this website https://www.thingiverse.com/thing:1683764. Choose the second design and frame is printed by the 3D printer so that the error can be minimised and save time consume . The two dimensional mechanical design of the sphere-o-bot is pictured in Figure 32.

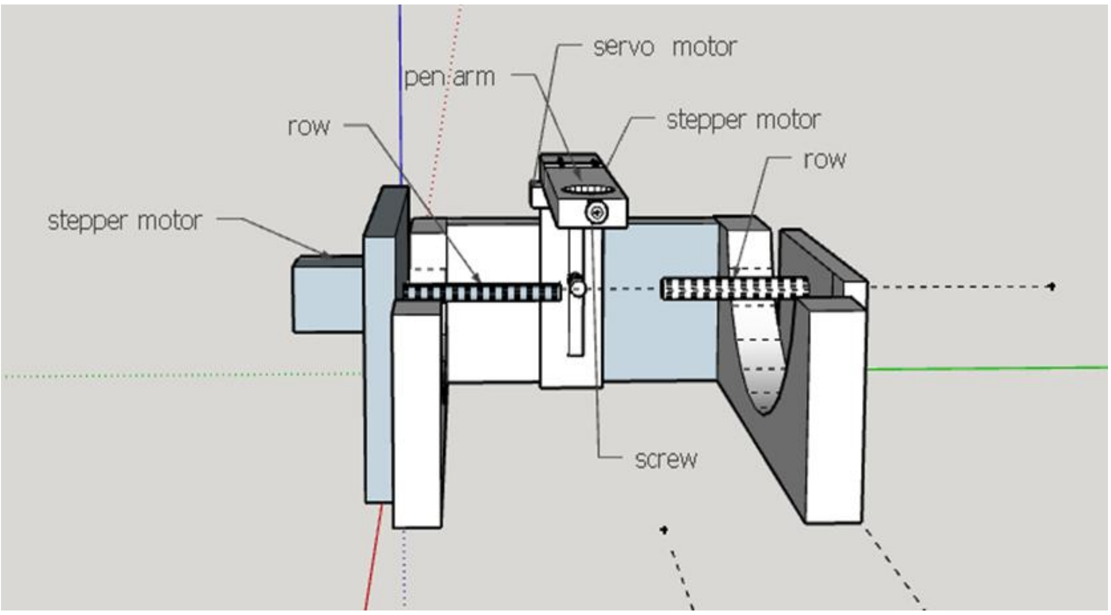


Figure 31 first design

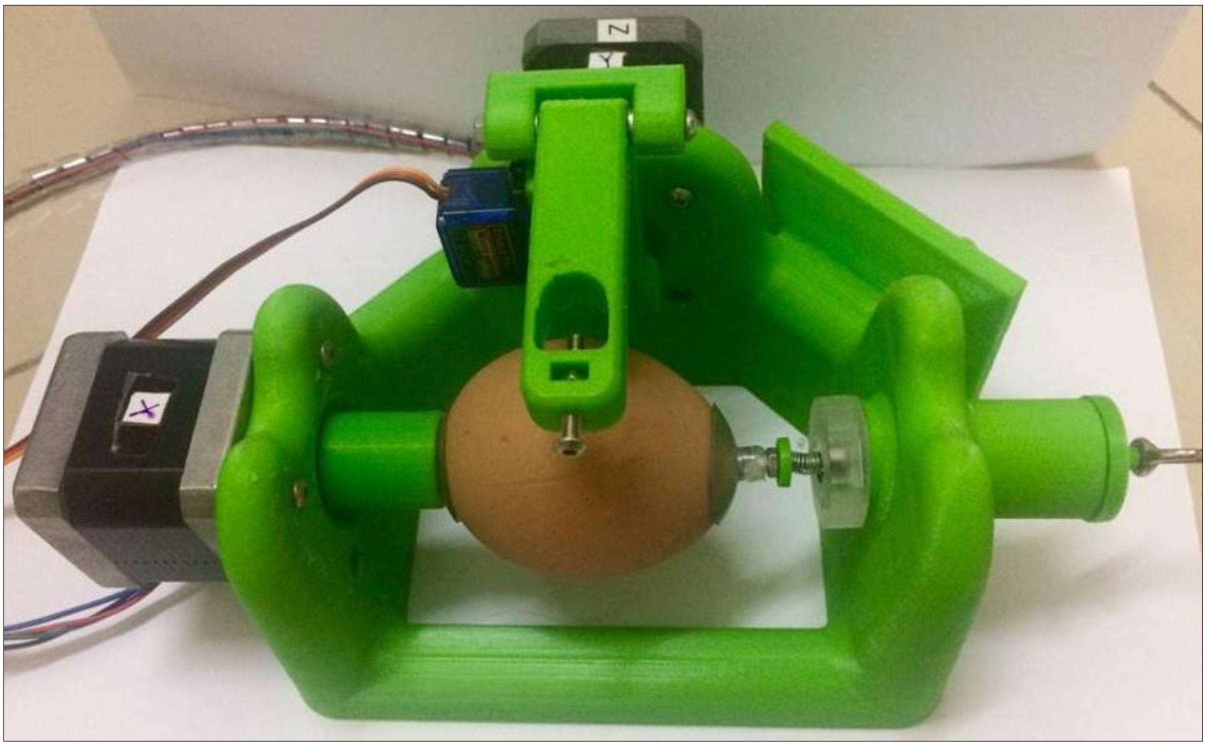


Figure 32 second design

Chapter 3 Results and Discussion

3.1. Results

The program in this project is able to plot desired complex art, text and image on the egg-shape or sphere object. Based on Figure 33, is a picture to demonstrate the effect of the art with a different colour cube. This Figure 35, shown the cube art design in Inkscape software and transform it to G-code file. Unfortunately, the cubes doesn’t seem to be constant a line properly due to the size of the egg as shown in Figure 34. This course of study is planned with open-loop system does not go through the error feedback. Overall, the cubes art result looks nice as pictured in Figure 33.

Figure 33 cubes art on egg

Figure 34 highlight the error

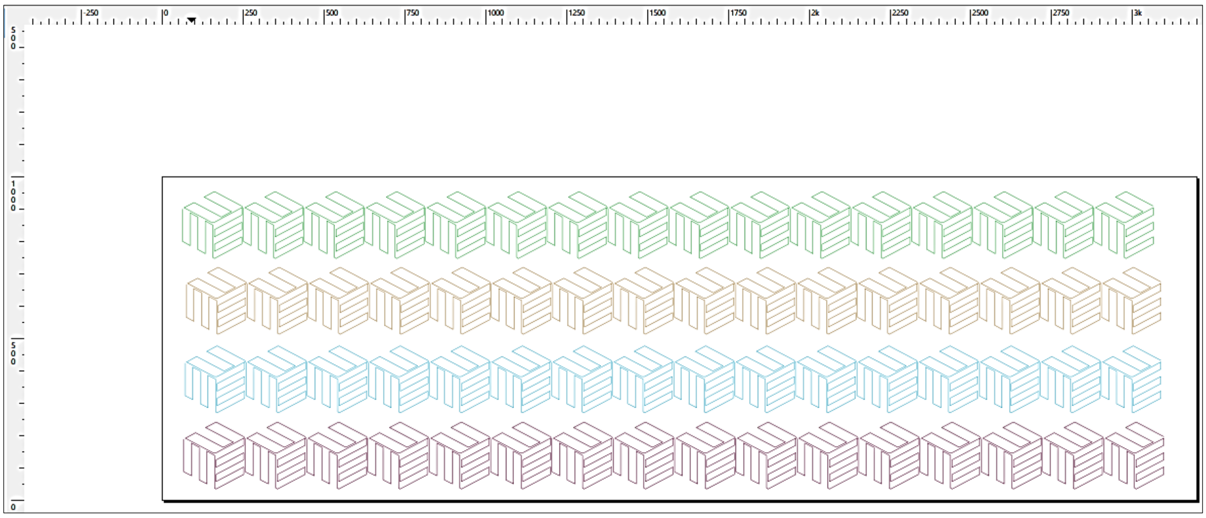


Figure 35 cubes art design in Inkscape software

Based on Figure 36, is a picture to show the effect of the art on the egg surface. The cube art projects in the Inkscape software as indicated in Figure 38. The Figure 37 highlight the error on the egg surface. The error is due to the backlash of the geared motor. The stepper motor at some point will miss-position due to the result has some imperfection. In that respect is an important point choose the good stepper motor. This stepper motor is provided by the laboratory. For this cube art result is acceptable only some miss-place effect of the motor.

Figure 36 cube art effect on egg

Figure 37 highlight the error

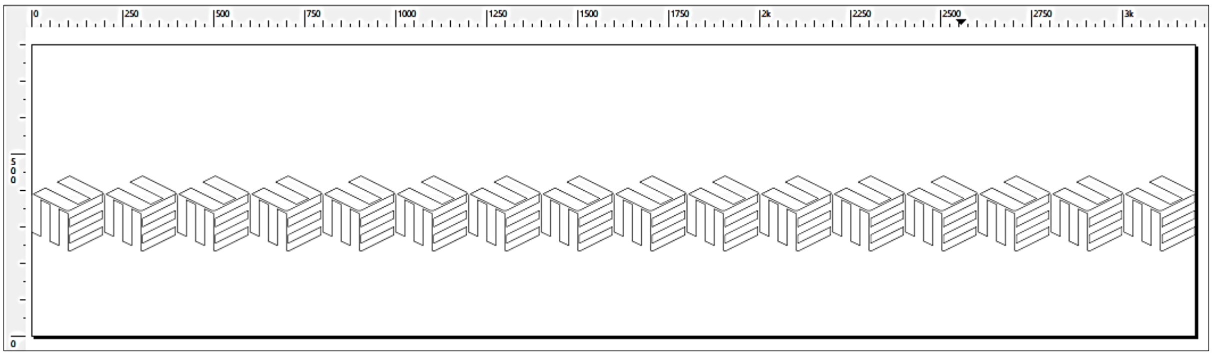


Figure 38 cube art design in Inkscape software

Based on Figure 39, is a picture to show the effect of the complex pattern on the egg surface. The complex pattern is designed in the Inkscape software as shown in Figure 41. The Figure 40 highlight the error on the egg surface. The error is due to the size of the egg and this system is implement the open-loop system do not receive feedback. Another error is misaligned such that the starting point of the pen arm was poor and the pen arm bumped into the suction-cup. When the pen arm reaches the end of its range, it may bend and try to continue plotting there, or the motor may skip a step and lose alignment. Users have to make sure that the pen arm has free travel over entire range. For this complex pattern result is barely reached, but some patterns are intersecting and words have a shadow effect.

Figure 39 words have shadow effect

Figure 40 patterns overlap each other

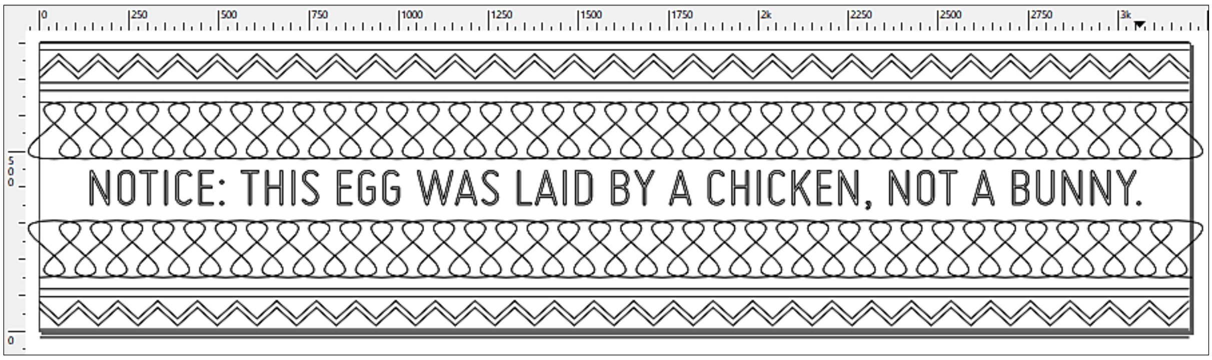


Figure 41 complex pattern design in Inkscape software

Based on Figure 42, is a picture to prove the outcome of the image along with the egg surface. The STAR WARS image (as shown in Figure 44) is converted to SVG format and save it as a G-code file from the Inkscape software. The simulation result is pictures of the drawing operation as shown in Figure 45. The Figure 43 highlight the error on the egg surface. The error is due to the size of the egg and the pen holder occasionally cannot lift up. The pen holder and the pen arm can attach with a rubber band so that create a bounce force as shown in Figure 46. For this image result is acceptable only some patterns are overlapping.

Figure 42 plot the STAR WARS image on egg

Figure 43 highlight the error

Figure 44 STAR WARS image

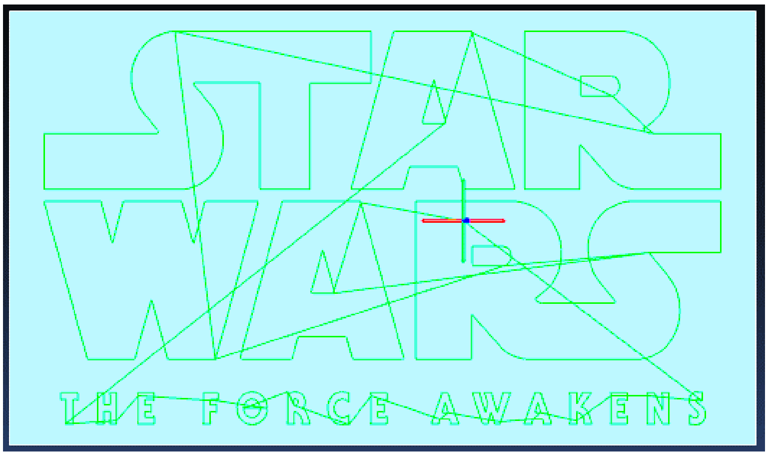


Figure 45 simulation result in CAMotics software

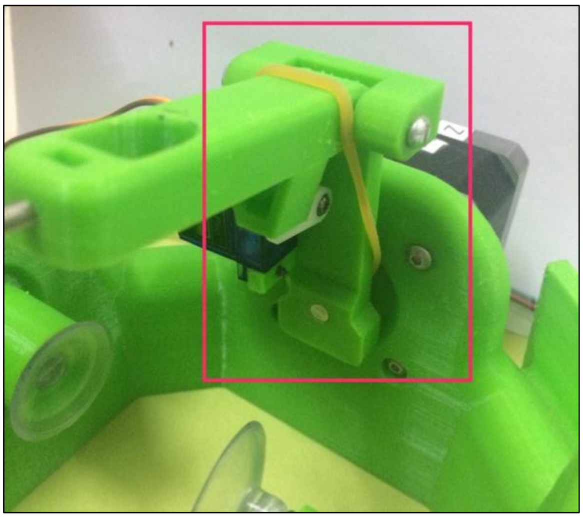


Figure 46 pen arm and pen holder attach with a rubber-band

To conclude, the condition of a stepper motor, align the egg, tilting pen raising, and diameter of the egg-shape or sphere object can affect the outcome. The backlash of the geared motor is the reason for the inaccuracies of the egg-plotter. When the ends of figures do not line up, this strongly indicates an alignment or slippage problem as shown in Figure 47. To control the alignment, rotate your egg with the motors disengaged and see if there is any perceptible wobble. It's significant that the egg be well centered in the egg cups, otherwise, the egg may precess as it is rotated. Sometimes it helps to lift the Eggbot up and sight down the long axis of the egg as you turn it. This Figure 48 shows that the egg is line properly.

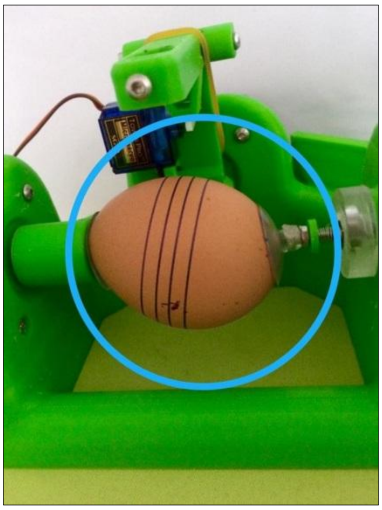


Figure 47 ends of figures do not line up

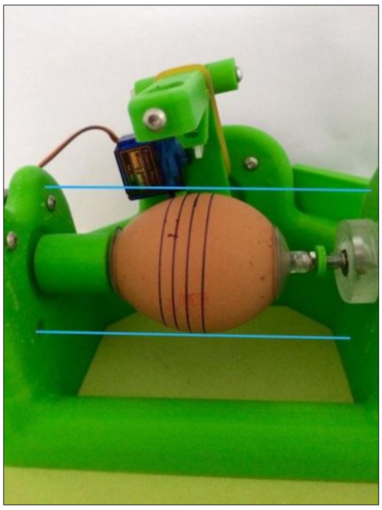


Figure 48 ends of figures line up properly

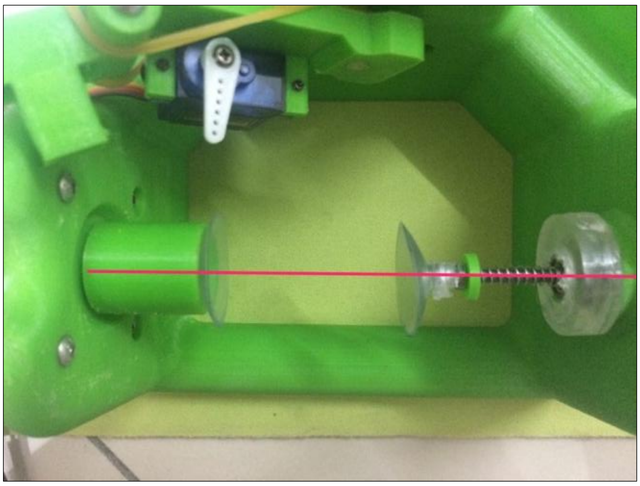


Figure 49 align the egg in straight line

This mechanical system has some error on the shaft holdler due to shape of the hole as highlighter in Figure 50.

Figure 50 the hole of the shaft holder

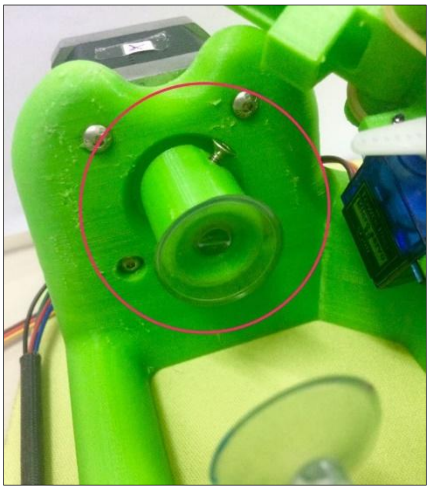


Figure 51 shaft holder

Figure 53 shows the right position for the distal pen arm and pen. Notic that when the tip of the pen reaches the egg, the distal pen arm extends out straight and flat, down towards the center of the egg. If, on the other hand, the pen arm is not horizontal when in the down position, it will cause the drawing to be twisted and create curve line as pictured in Figure 52.

Figure 52 pen arm is not horizontal

This position of pen arm shown in Figure 53 is in the correct position, since the pen tip points directly down and towards the thickest part of the egg, when the distal pen arm extends straight out and flat.

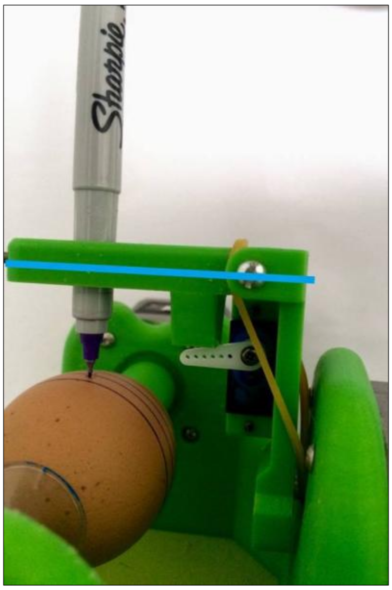


Figure 53 pen arm in correct position

Another issue is the result of stepper motor moving too fast. The "delay after pen down" setting too fast will create the pen to shake or the pen hasn't reached on the egg. The speed of the motor moving too fast caused the pen to shake when drawing.

3.2. Discussion

The LabVIEW program in this project is developed to program the A4988 drivers to drive the stepper motor and servo motor as well as interpret the G-code command.

3.2.1 System Description

At the beginning of the program, the user must enter the total number of G-code line and G-code file path so that the program iterate that G-code file. If the user does not enter any value, a message will prompt to ask the user to key in the total number of G-code line. The message is in a 'True' case where the message will be prompted out when the total g-code line is equal to zero as shown in Figure 54. The total g-code line wires to the  count terminal of the For Loop. After the message dialog had prompt out, it is necessary to determine whether the user has keyed in the total g-code line as shown in Figure 56. This case structure to ensure the program is able to proceed after the user had keyed in the total g-code line.

In the 'False' case is wired the total g-code line to the  count terminal of For Loop as shown in Figure 55.

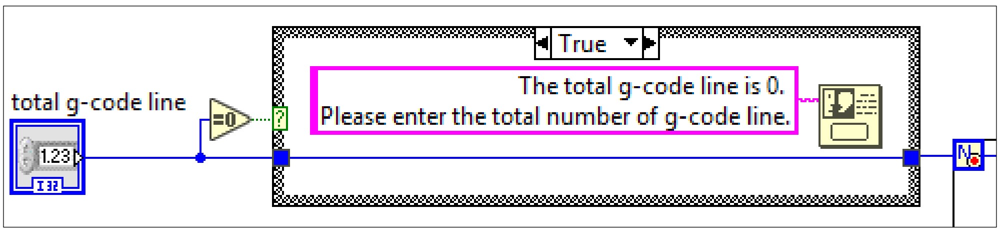


Figure 54 'True' case for prompt message to enter the total g-code line

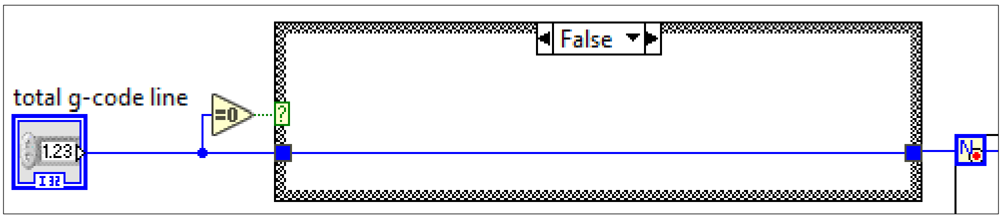


Figure 55 'False' case for wire the total g-code line to the count terminal of For Loop

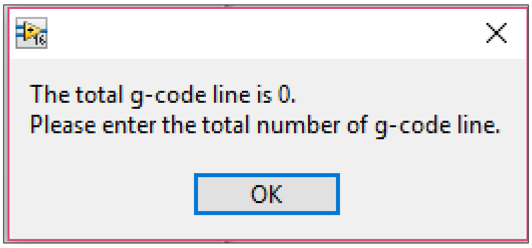
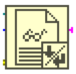


Figure 56 a prompt window to display message to enter the total g-code line

Succeeding, the program will enter into the For Loop. Inside the For Loop the G-code file has been open and read as shown in Figure 57. The Read from Text File Function will output G-code file in array form which start from the first line to end of file. The maximum number of lines read by  Read from Text File Function is the total number of G-code file enter by the user.

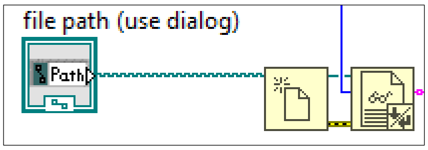


Figure 57 block diagram for open and read the G-code file.

After the read the out the G-code in array form, the array form of G-code will input to the Index Array Function and return an element of G-code array at index as shown in Figure 58. The index is the  number of iterations of For Loop (G-code file). The element of the G-code array will executed according to the command.

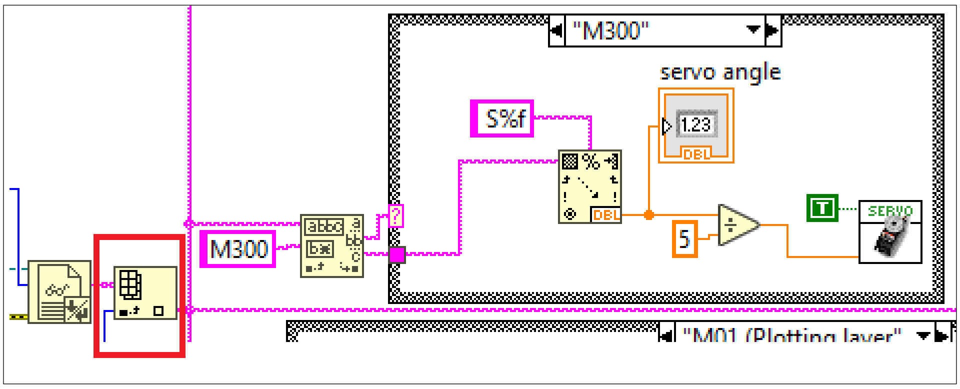


Figure 58 return the element of G-code array at index.

When the element of G-code array as the input string of the Match Pattern Function and match the ‘G4’ regular expression. The G4 P~ command freezes the system in milliseconds. The matched substring of the Match Pattern Function wire to the case selector of case structure.

Based on Figure 59, this case structure interprets the G4 command and delay the whole program for a period in millisecond specify by P value. The value of P returns the portion of the string that remains after scanning the remaining the sub-string of elements of G-code array by the Scan From String Function. The value of P in floating point format. When the element of the G-code array does not match the ‘G4’ regular expression, then enter into the ‘False’ case do nothing as shown in Figure 60.

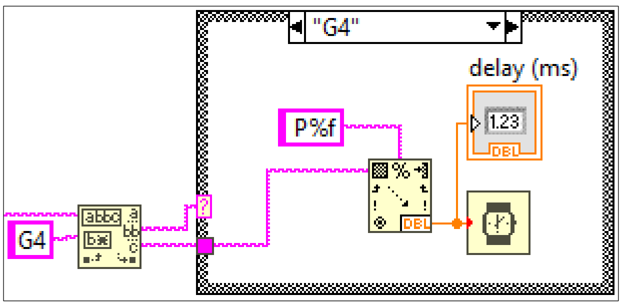


Figure 59 case structure for interpret the G4 P~ command

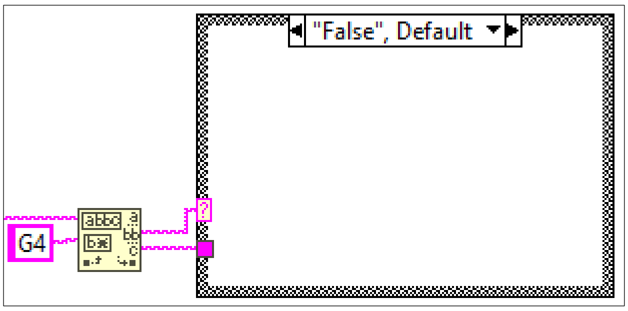


Figure 60 'False' case do nothing for interpret the G4 P~ command

Based on Figure 61, this case structure interprets the M300 S~ command. The value of S defines the angle rotation of servo motor and in term of floating point. The value of S as the input of the servo motor sub VI. The P value is divided by 5 so that P value can wire to the duty cycle of PWM. The range of duty cycle of PWM from 3 up to 10. When the element of the G-code array does not match the ‘M300’ regular expression, then enter into the ‘False’ case do nothing as shown in Figure 62.

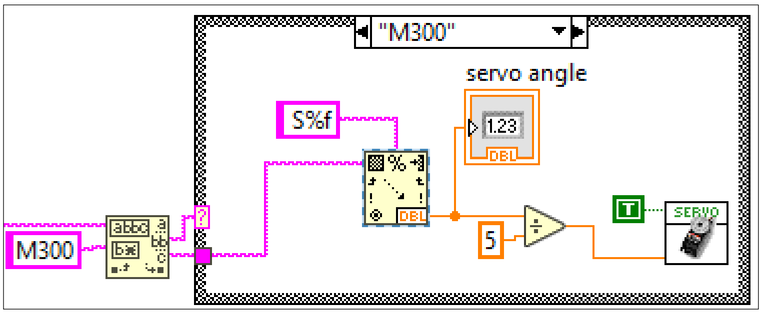


Figure 61 case structure for interpret the M300 S~ command

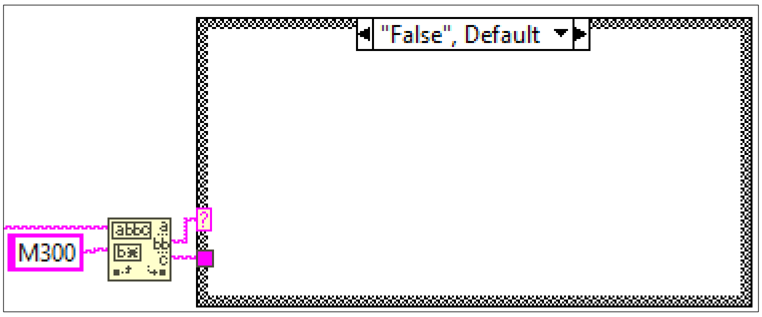


Figure 62 'False' case do nothing for interpret M300 S~ command

Based on Figure 63, this case interprets the M01 command. In the ‘M01 (Plotting layer’ case will prompt an option dialog box to let the user change the pen colour as shown in Figure 65. Subsequently the user changes the pen colour, presses the ‘OK’ button to continue loop the For Loop. When the conditional terminal has continued if ‘True’ Boolean, the loop executes until the terminal receives a ‘false’ Boolean. If the user presses the ‘Cancel’ button, the whole program will stop because receive a ‘false’ Boolean at conditional terminal. When the element of the G-code array does not match the ‘M01 (Plotting layer’ regular expression, then enter into the ‘False’ case and send a ‘true’ Boolean to the conditional terminal as shown in Figure 64.

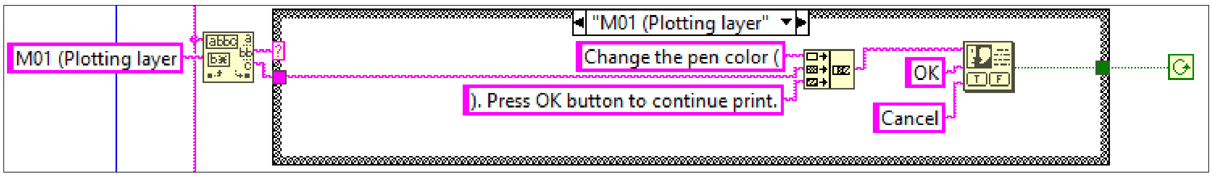


Figure 63 case structure for interpret the M01 command

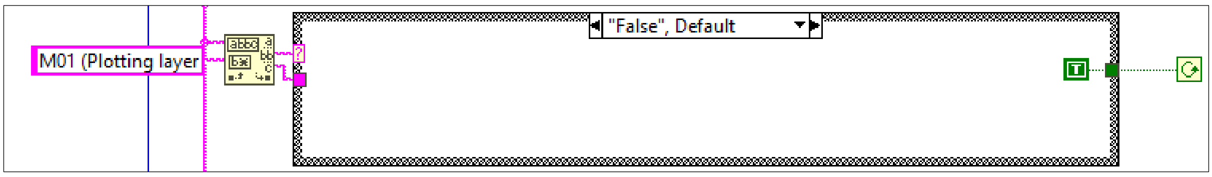


Figure 64 'False' case for interpret M01 command

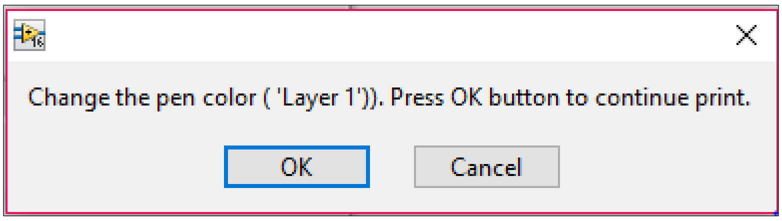
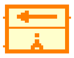


Figure 65 a dialog box to display a message to let user change the pen colour

Based on Figure 66, this case structure interprets the G92 X~ Y~ Z~ command. This command produces the current point had coordinate. This command usually uses to create the initial point where the pen initial point to that coordinate. The z-axis is for vertical movement of the pen. When the element of the G-code array does not match the 'G92' regular expression, then enter into the 'False' case and x-y coordinate value set to zero as shown in Figure 67. This x and y coordinate value is the  initial value of feedback node.

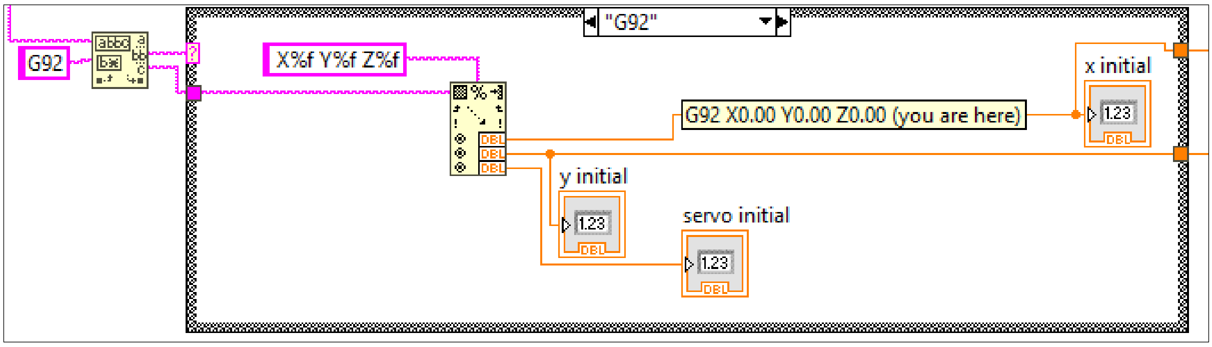


Figure 66 case structure for interpret the G92 X~ Y~ Z~ command

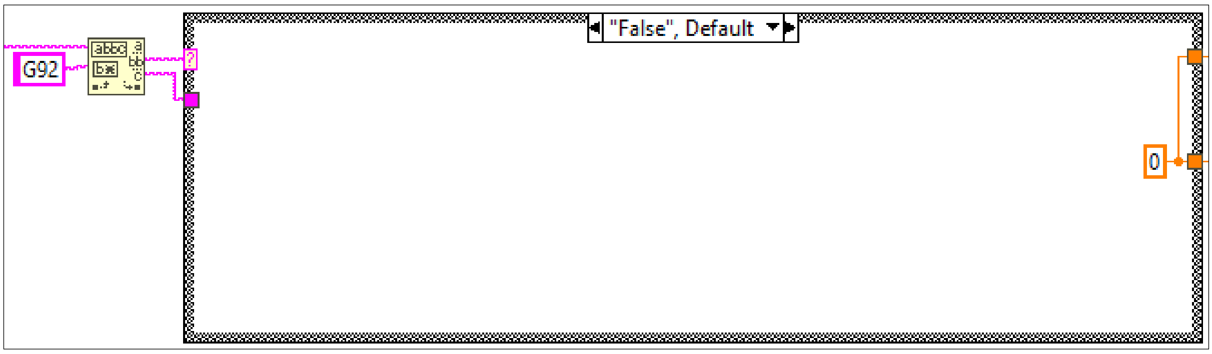
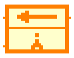
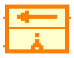


Figure 67 'False' case for interpret G92 command

Based on Figure 68, this case interprets the G1 X~ Y~ F~ command. This command is plotting a linear line at feed rate. The value of X is defined the end point of x-coordinate. The value of Y defines the end point of the y-coordinate. The value of F defines the feed rate. The feedback node is stored data from the previous iteration. The previous iteration data of

 feedback node store the initial point of the linear line. The current the iteration data of  feedback node store the end point of the linear line. The feed rate is divided by 50 then wire to the speed of the stepper motor move in term of a millisecond (ms). When the element of the G-code array does not match the ‘G1 X’ regular expression, then enter into the ‘False’ case do nothing as shown in Figure 69.

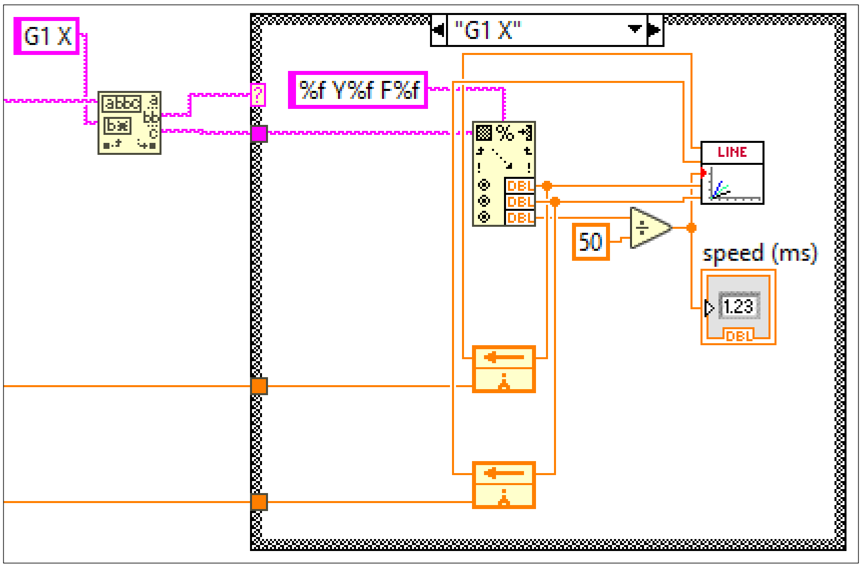


Figure 68 case structure for interpret the G1 X~ Y~ command

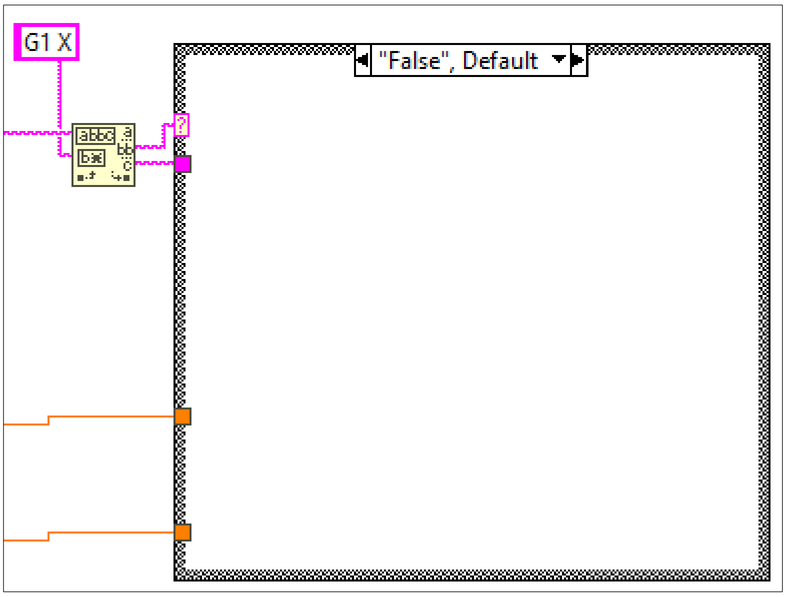


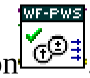
Figure 69 'false' case structure for interpret the G1 X~ Y~ command

3.2.2 Sub VI Description

3.2.2.1 Rotate the angle of servo motor sub VI

This sub VI is to configure the pin on Analog Discovery 2, initialise the waveform, adjust the duty cycle of PWM, and program the positive power supply as shown in Figure 72. This sub VI is taking the input from the S value of M300 command. The enable output switch always switch on. The waveform generator pin 1 of Analog Discovery 2 wires to the orange wire of SG-90 servo motor. The positive power supply of Analog Discovery 2 is connected to the red wire of SG-90 servo motor. The configuration of the power supply is done by 

this function. This function 

need specifies the name of the channel as ps/+5V, voltage level at 5V, and the current limit is 0.1A. Before that, enable the power supply with writing a 'True' Boolean to the enable output pin of DWF PS Enable All Outputs function 

. This servo motor expected to receive a PWM signal with period 20ms. The 

DWF FGEN Initialize function is to configure the waveform. The PWM signal is a square waveform at 50Hz with 5V peak-to-peak. In this sub VI the duty cycle of the waveform is adjustable at the range between 3 to 10, where 3 is 3 percentage and 10 is 10 percentage. The percentage of the time period the PWM signal is HIGH or ON is called the duty cycle of the signal.

The duty cycle can be calculated by the following:

$$for\ an\ example\ frequency\ is\ 50Hz$$

$$Time\ period = \frac{1}{50} = 20ms$$

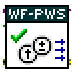
$$Assume\ that\ duty\ cycle\ is\ 10\%$$

$$\frac{t_{ON}}{t_{OFF} + t_{ON}} = 0.1,\ t_{OFF} + t_{ON} = 20ms$$

$$t_{ON} = 0.1x20 = 2ms,\ t_{OF} = 20 - 2 = 18ms$$

Result of PWM signal at 10% duty cycle.

After configure the waveform and power supply, then start to run the waveform for a period of 500ms. In experimental find out the need some delay between 300ms to 500ms before 

stop to run the waveform. Next, disable the power supply with writing a 'False' Boolean to DWF PS Enable All Outputs function 

. Lastly, stops the session and deallocates any resources acquired during the session. This function 

is release the handle for waveform generator and this function 

is release the handle for the power supply.

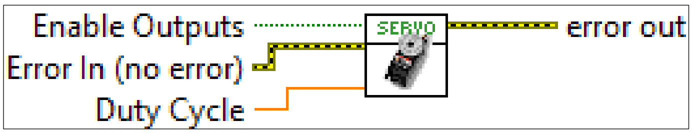


Figure 70 icon of rotate the angle of servo motor sub VI

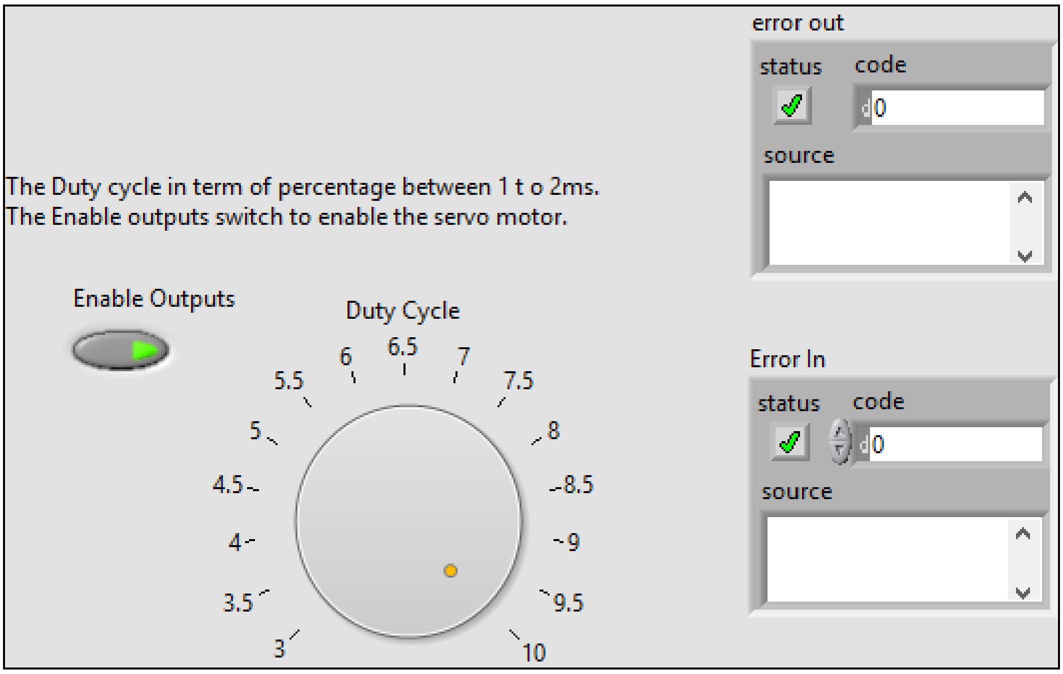


Figure 71 front panel of rotate the angle of servo motor sub VI

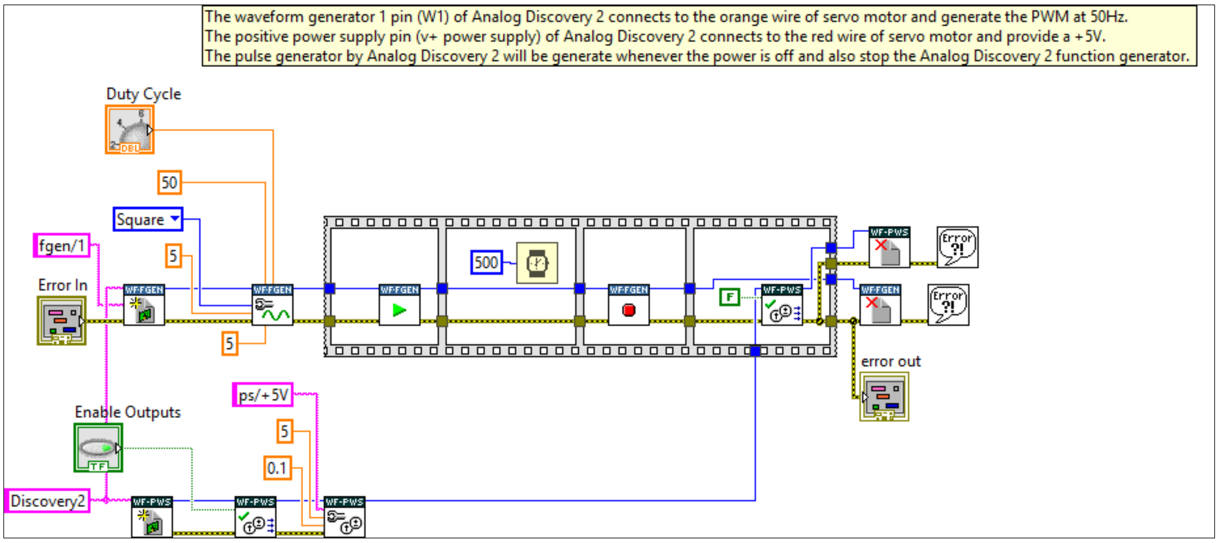


Figure 72 block diagram of rotate the angle of servo motor sub VI

3.2.2.2 Bresenham Line Algorithm sub VI

This sub VI is implement the G1 X~ Y~ F~ command to draw a linear line as shown in Figure 73. Firstly, this sub VI calculates the difference between initial point and end point (dy, dx) and the direction done by the calculate the different length and the direction of the coordinate sub VI. If the dy greater than dx will enter into the 'False' case. Based on Figure 75, 'False' case is dealing with the y-axis driving axis. The 'False' case always increment the y-axis by one and the x-axis will be increment according the value of the error ($\bar{\epsilon}$). If the error value ($\bar{\epsilon}$) greater than zero will increment the x-axis by one, else maintain the x-axis at that point as shown in Figure 76. In the 'True' case is deal with x-axis drives axis as shown in Figure 77. This case always increment the x-axis rather than keeping the track of y-axis. Increment y-axis also depend on the calculated value of the error ($\bar{\epsilon}$). If the error value ($\bar{\epsilon}$) greater than zero will increment the y-axis by one, else maintain the y-axis at that point as shown in Figure 78.

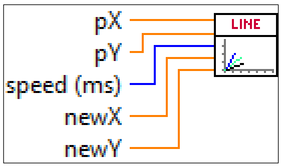


Figure 73 icon of Bresenham Line Algorithm sub VI

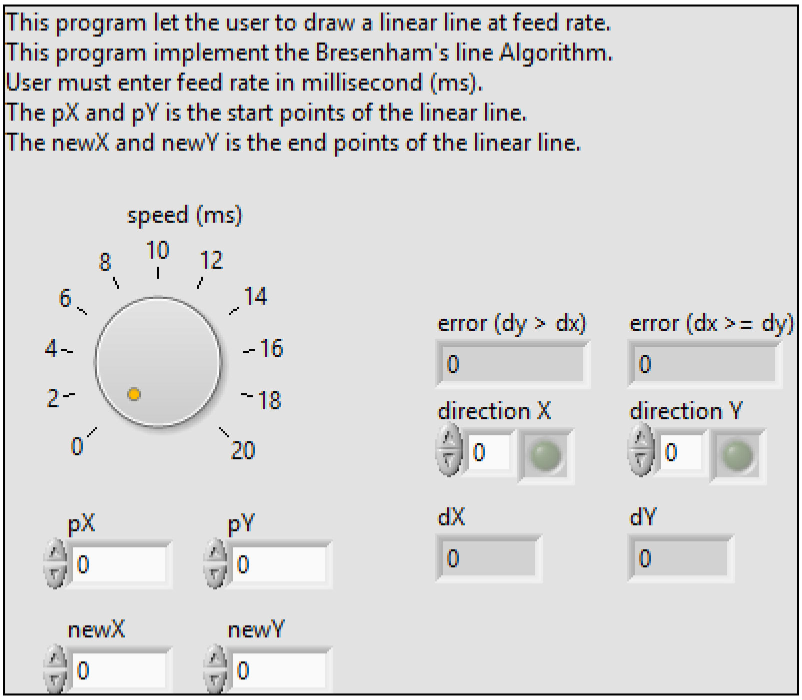


Figure 74 front panel of Bresenham Line Algorithm sub VI

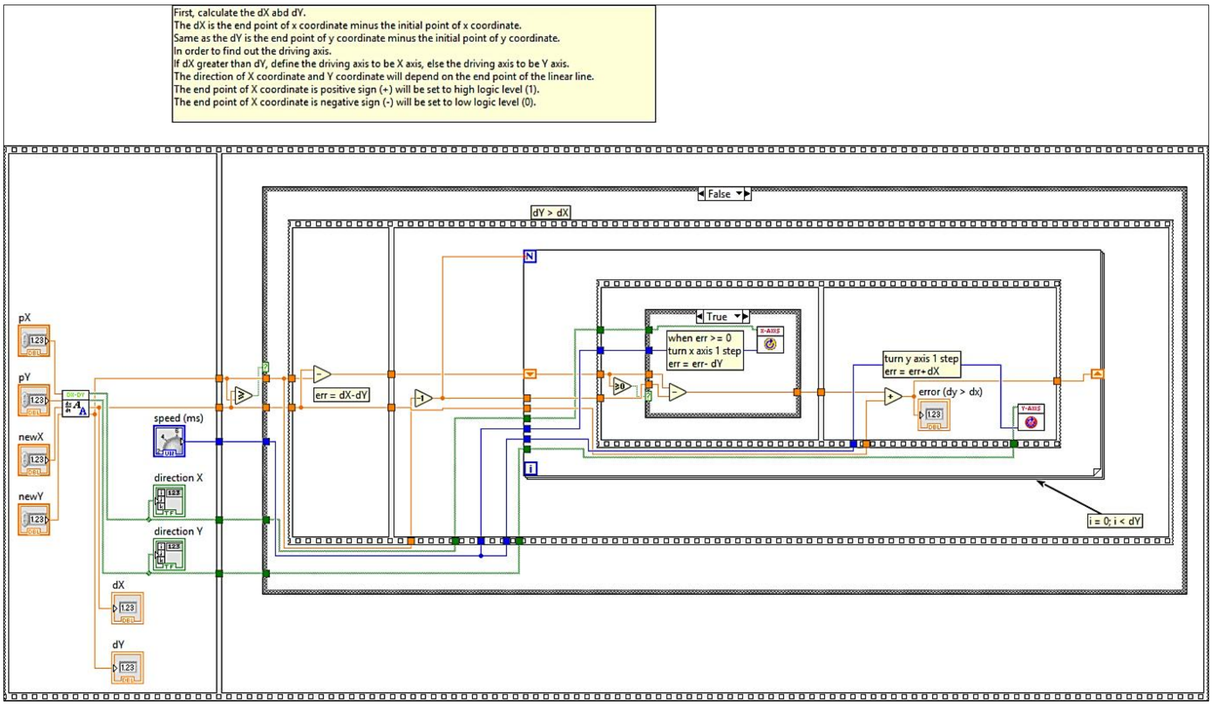


Figure 75 y-axis driving axis of block diagram of Bresenham Line Algorithm sub VI

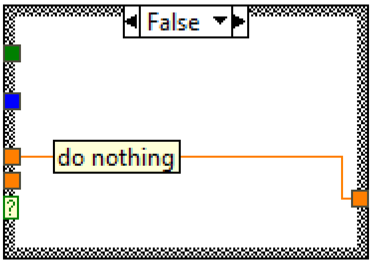


Figure 76 ‘false’ case of y-axis driving axis of block diagram of Bresenham Line Algorithm sub VI

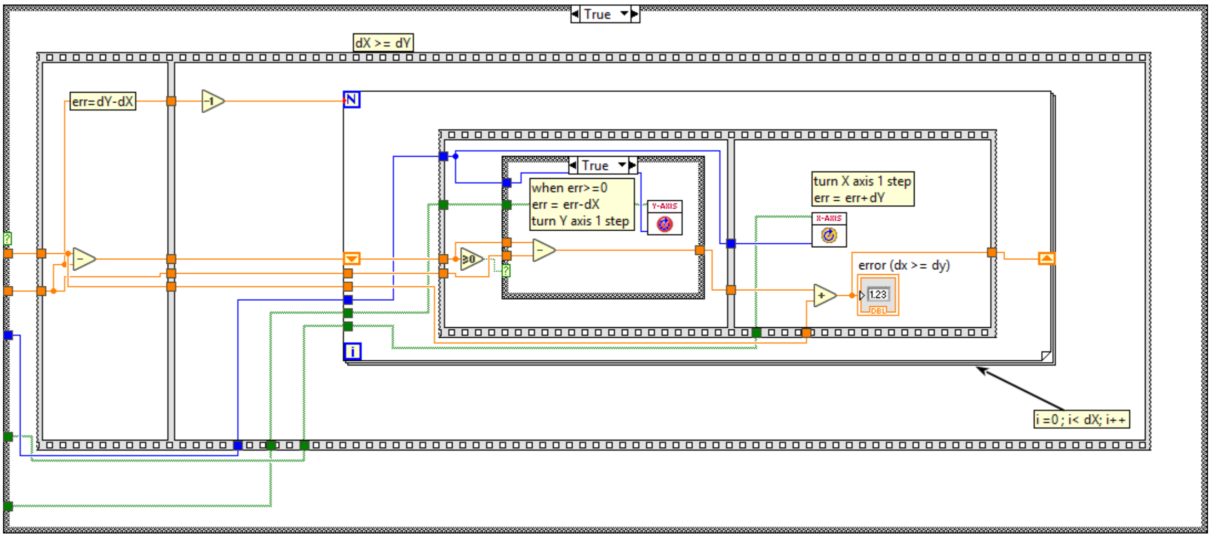


Figure 77 x-axis driving axis of block diagram of Bresenham Line Algorithm sub VI

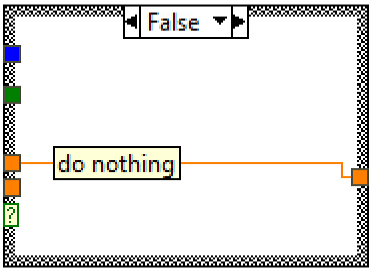


Figure 78 ‘false’ case of x-axis driving axis of block diagram of Bresenham Line Algorithm sub VI

3.2.2.3 Calculate the different length and the direction of the coordinate sub VI

This sub VI calculates the length between initial points of x-coordinate and end points of x-coordinate and also calculated the length between initial points of y-coordinate and end points of y-coordinate. This sub VI also defined the x-axis direction and the y-axis direction in term ‘True’ or ‘False’ Boolean. The icon, front panel, and block diagram of this sub VI as shown in Figure 79 to Figure 81.

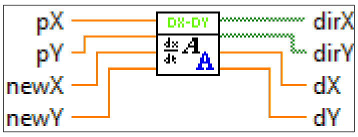


Figure 79 icon of the calculate the different length and the direction of the coordinate sub VI

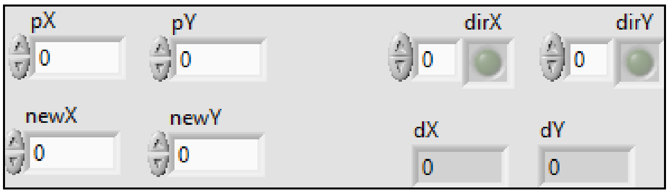


Figure 80 front panel of calculate the different length and the direction of the coordinate sub VI

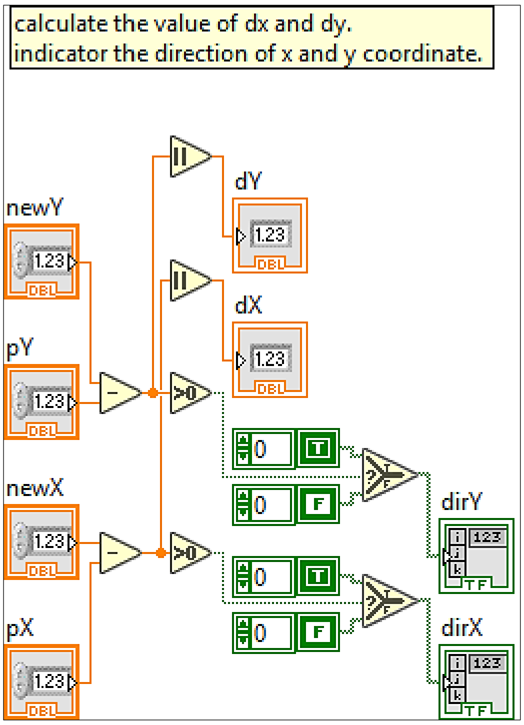


Figure 81 block diagram of calculate the different length and the direction of the coordinate sub VI

3.2.2.4 Configuration and program the A4988 driver sub VI

The x-axis sub VI is to initialise the Digilent Analog Discovery 2, configure the pin on the Digilent Analog Discovery 2 and program the A4988 driver to drive stepper motor to move one step. The icon of this sub VI as shown in Figure 82. This x-axis VI rotates the x-axis stepper motor in one step. This sub VI speed pin takes the input from the feed rate of x-coordinate linear line. The direction pin takes the input from the direction of the x-coordinate linear line. This sub VI sets the step resolution of A4988 driver to 1/16 with implement 3-bits of high logic level and connects to the digital pin 3 to 5 of Analog Discovery 2. The enable pin of A4988 driver connects to the digital pin 0 of Analog discovery 2 with implementing a low logic level. The step pin of A4988 driver connects to the digital pin 1 of Analog discovery 2 with implementing a pulse sent to this pin the motor moves one step. Between each logic level need add some delay from which the speed of the motor will depend.

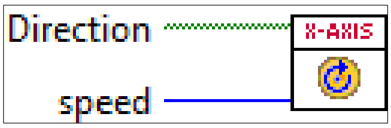


Figure 82 x-axis sub VI

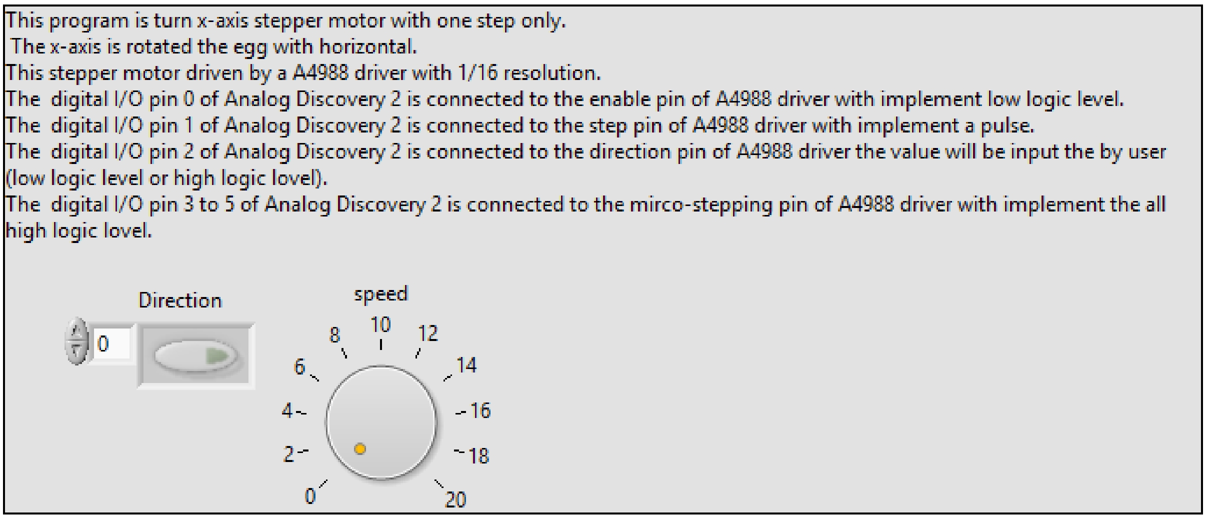


Figure 83 front panel of x-axis sub VI

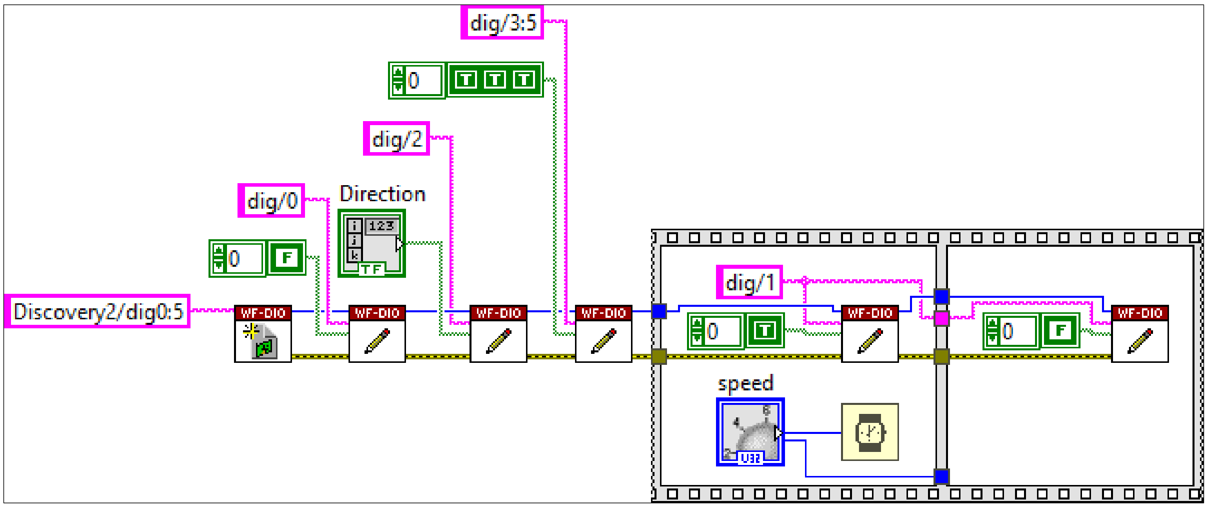


Figure 84 block diagram of x-axis sub VI

The function of y-axis sub VI exact same as the x-axis sub VI except the configuration pin on Analog Discovery 2 as shown in Figure 86. This sub VI is take the input from the feed rate of y-coordinate of linear line and the direction of the y-coordinate of linear line. The Analog Discovery 2 digital pin 8 connects to the enable pin of a4988 driver. The controller, digital pin 9 connects to the step pin driver, controller, digital pin 10 connects to direction pin driver, and controller, digital pin 11 to 13 connect to the micro-stepping pin driver.

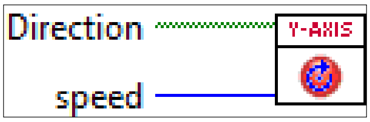


Figure 85 icon for y-axis sub VI

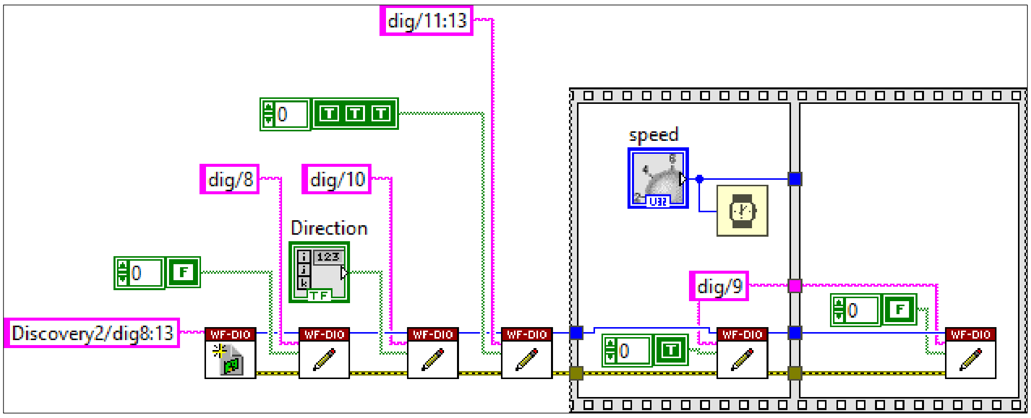


Figure 86 block diagram of y-axis sub VI

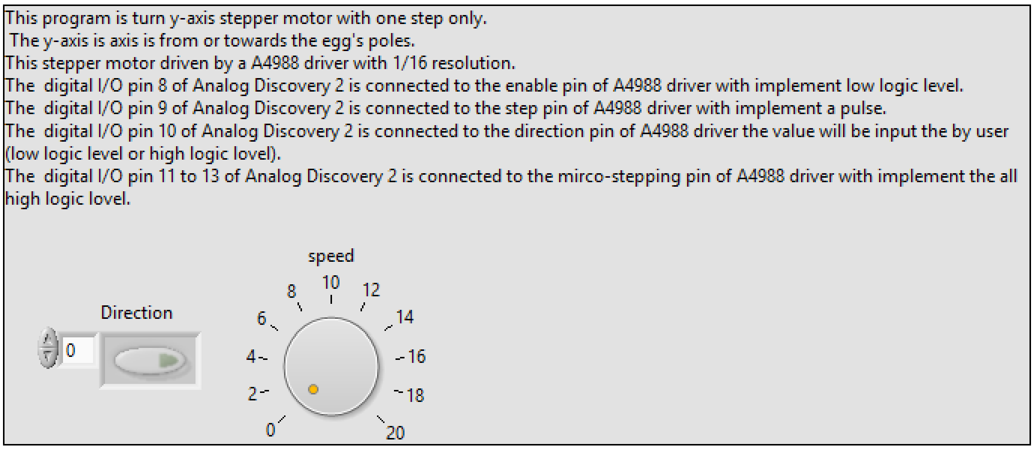


Figure 87 front panel of y-axis sub VI

3.2.3 Testing

The height of the template of the Inkscape is defined as the extreme range of pen travel. In experimental, found out that the maximum value is 1000 pixels. For safety and good results, 800 pixels is the best value for the height. For 1000 pixels have some error and blemish on the result.

Chapter 4 Conclusion

4.1. Conclusion

In conclusion, the program of this project is developing an egg-plotter control using Digilent Analog Discovery 2 to automate plot the image, text, or complex art on sphere object or egg-shape by NI LabVIEW software. This project proved that able automate to print text, image or art on egg-shaped or sphere object.

Instead of using human's hand to draw the pattern, art or text and repeat the same pattern on egg-shaped or sphere object. This is to make sure the pattern can repeat almost the same on different sphere object.

To increase the flexibility of this project, this project is designed by using the Inkscape software to generate a G-code file from the SVG format image. This feature lets user the design any size of pattern, image, or text inside the Inkscape's template, then printed on the egg-shape or sphere object. This project is fully automated done by the machine.

The advantage of this project is saving manpower to draw the art on the egg-shaped or sphere object. Another benefit of this project is gaining experience in several fields, like mechanics, stepper motors, use of a vector based drawing program, LabVIEW programming and design the frame.

The results that obtained from this project accuracy was not expected, but overall the result is acceptable.

4.2. Future Plans

The future plan of this project is designed in closed-loop system. This system can be designed to automatically achieve and maintain the desired shape of the object by comparing it with the actual shaped. This will increase the accuracy of the result.

Next, a 3D printer is used to sculpture any shape. Based on the algorithm and prototype of this project to build a 3D printer. 3D printer is a 3-axis machine and based on CNC principle. Implement the same algorithm to print a linear line. The same fundamental operate the stepper motor just add one more stepper motor at z-axis.

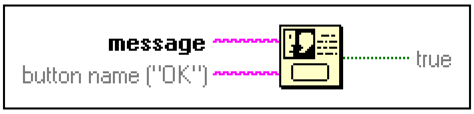
References

1. History of the EggBot. (n.d.). [ebook] pp.1 to 12. Available at: http://egg-bot.com/history/ [Accessed 25 Aug. 2017].
2. Tower Pro (n.d.). SG90 9 g Micro Servo. [ebook] pp.1 to 2. Available at: http://akizukidenshi.com/download/ds/towerpro/SG90.pdf [Accessed 25 Aug. 2017].
3. Hybrid stepping motor 42SHD0217-24B. (n.d.). [image] Available at: http://g04.s.alicdn.com/kf/HTB1PpNjHFXXXXbQXXXXq6xXFXXXI/202171482/HTB1PpNjHFXXXXbQXXXXq6xXFXXXI.jpg [Accessed 15 Jan. 2018].
4. Analog Discovery 2 Reference Manual. (2015). [ebook] Available at: https://reference.digilentinc.com/_media/reference/instrumentation/analog-discovery-2/ad2_rm.pdf [Accessed 15 Jan. 2018].
5. A4988 driver. (n.d.). [ebook] Available at: https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf [Accessed 15 Jan. 2018].
6. Analog Discovery 2 – NI Edition. (n.d.). [ebook] NI. Available at: http://www.ni.com/pdf/manuals/AnalogDiscovery2NIDatasheet.pdf [Accessed 15 Jan. 2018].

Appendices

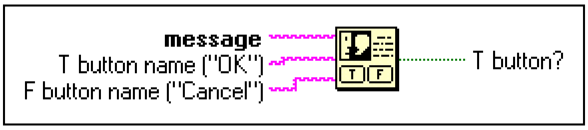
List of VIs

i. One Button Dialog Function



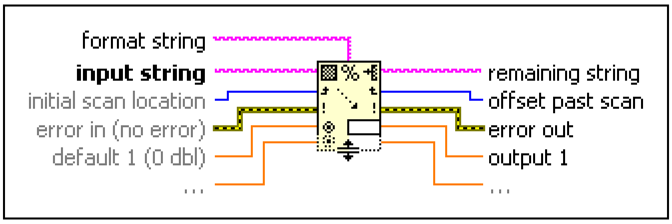
Displays a dialog box that contains a message and a single button.

ii. Two Button Dialog Function



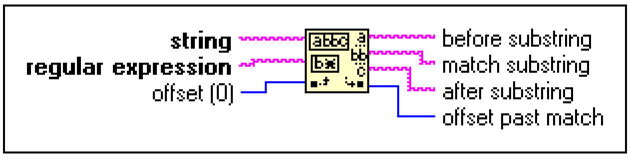
Displays a dialog box that contains a message and two buttons.

iii. Scan From String Function



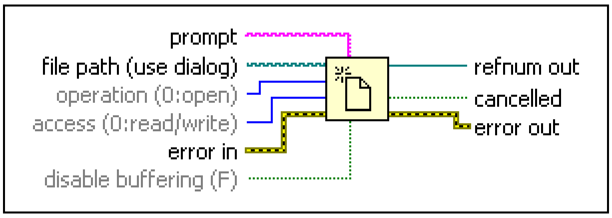
Scans the input string and converts the string according to format string.

iv. Match Pattern Function



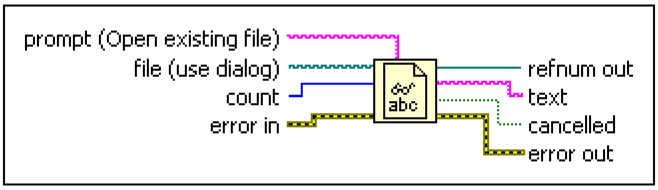
Searches for regular expression in string beginning at offset. If the function finds a match, it splits string into three substrings. A regular expression requires a specific combination of characters for pattern matching.

v. Open/Create/Replace File Function



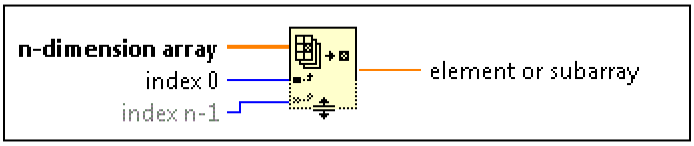
Opens an existing file, creates a new file, or replaces an existing file, programmatically or interactively using a file dialog box. This function does not work for files inside an LLB.

vi. Read from Text File Function



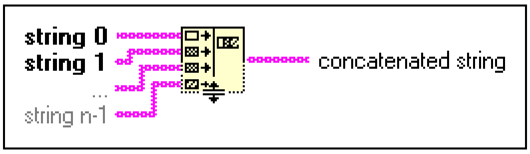
Reads a specified number of characters or lines from a byte stream file. This function does not work for files inside an LLB.

vii. Index Array Function



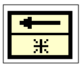
Returns the element or subarray of n-dimension array at index.

viii. Concatenate Strings Function



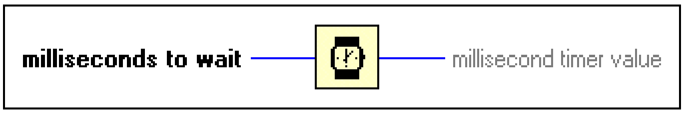
Concatenates input strings and 1D arrays of strings into a single output string. For array inputs, this function concatenates each element of the array.

ix. Feedback Node



Stores data from one VI execution or loop iteration to the next.

x. Wait (ms) Function

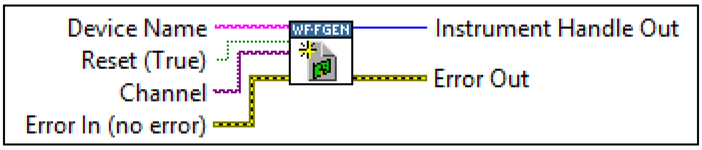


Waits the specified number of milliseconds and returns the value of the millisecond timer. Wiring a value of 0 to the milliseconds to wait input forces the current thread to yield control of the CPU.

List of Digilent Waveform VIs

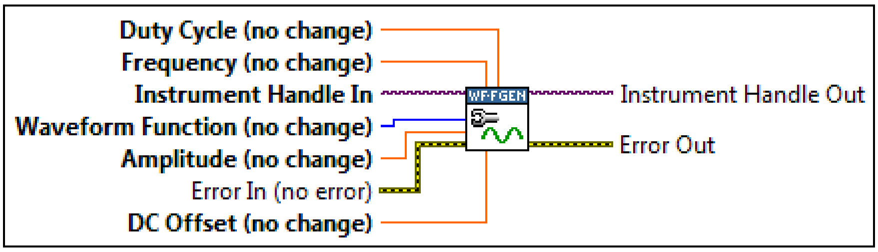
List of Function Generator VIs

i. DWF FGEN Initialize (VI)



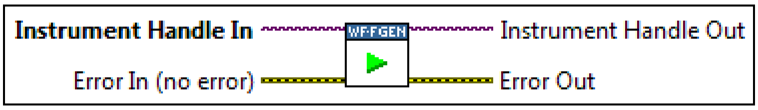
Establishes communication with the device. This VI should be called once per session.

ii. DWF FGEN Configure Standard Waveform (VI)



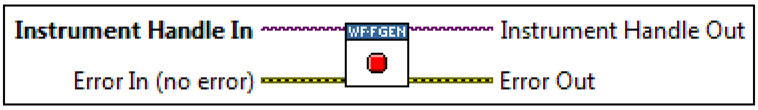
Configures the instrument to output a standard waveform. Any unwired configuration terminals do not change the existing values of those parameters.

iii. DWF FGEN Run (VI)



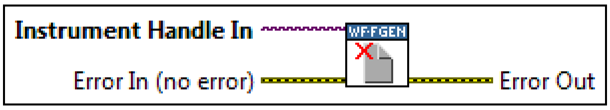
Transitions the session from the Stopped state to the Running state.

iv. DWF FGEN Stop (VI)



Transitions the acquisition from either the Triggered or Running state to the Stopped state.

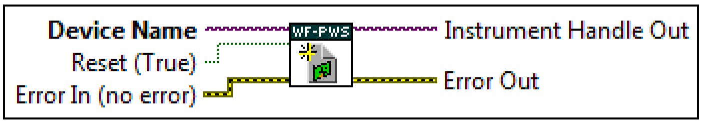
v. DWF FGEN Close (VI)



Stops the session and deallocates any resources acquired during the session. If output is enabled on any channels, they remain in their current state and continue to output data.

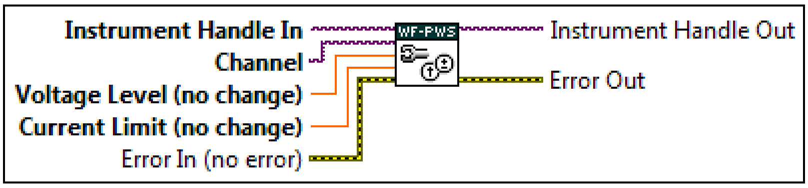
List of Power Supply VIs

i. DWF PS Initialize (VI)



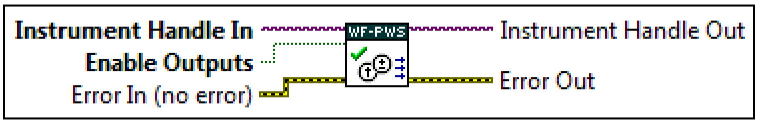
Establishes communication with the device. This VI should be called once per session.

ii. DWF PS Configure Voltage Output (VI)



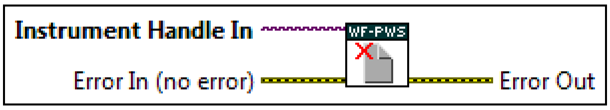
Configures a voltage output on the specified channel. This VI should be called once for every channel you want to configure to output voltage.

iii. DWF PS Enable All Outputs (VI)



Enables or disables all outputs on all channels of the instrument.

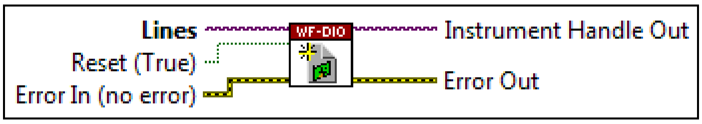
iv. DWF PS Close (VI)



Stops the session and deallocates any resources acquired during the session. If output is enabled on any channels, they remain in their current state and continue to output data.

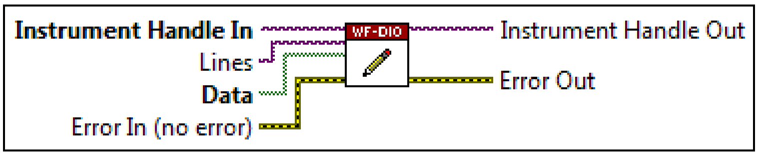
List of Digital VIs

i. DWF Dig Initialize (VI)



Establishes communication with the device. This VI should be called once per session.

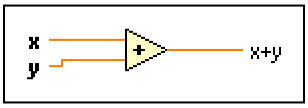
ii. DWF Dig Write (VI)



Writes data to the specified lines.

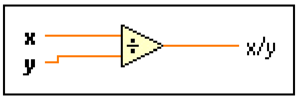
List of Numeric VI Function

i. Add Function



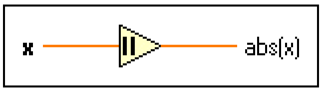
Computes the sum of the inputs.

ii. Divide Function



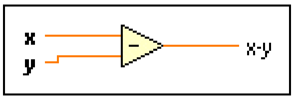
Computes the quotient of the inputs.

iii. Absolute Value Function



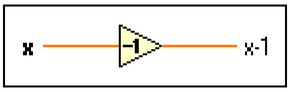
Returns the absolute value of the input.

iv. Subtract Function



Computes the difference of the inputs.

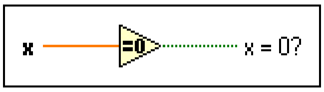
v. Decrement Function



Subtracts 1 from the input value.

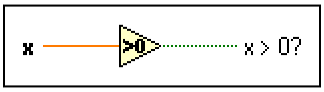
List of Comparison VI Function

i. Equal To 0? Function



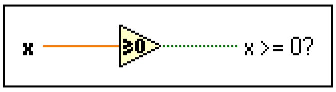
Returns TRUE if x is equal to 0. Otherwise, this function returns FALSE.

ii. Greater Than 0? Function



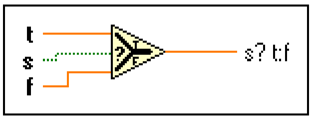
Returns TRUE if x is greater than 0. Otherwise, this function returns FALSE.

iii. Greater or Equal To 0? Function



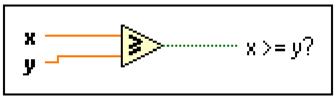
Returns TRUE if x is greater than or equal to 0. Otherwise, this function returns FALSE.

iv. Select Function



Returns the value wired to the t input or f input, depending on the value of s. If s is TRUE, this function returns the value wired to t. If s is FALSE, this function returns the value wired to f.

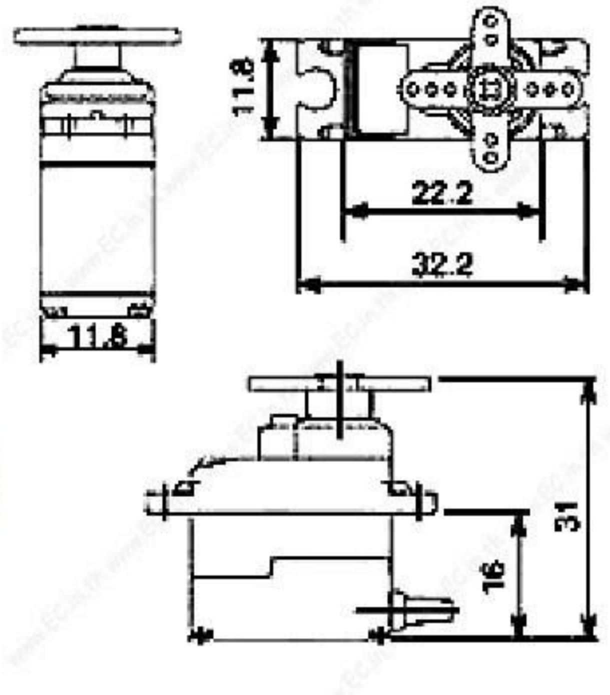
v. Greater or Equal? Function



Returns TRUE if x is greater than or equal to y. Otherwise, this function returns FALSE.

Spec of SG-90 servo motor

SG90 9 g Micro Servo

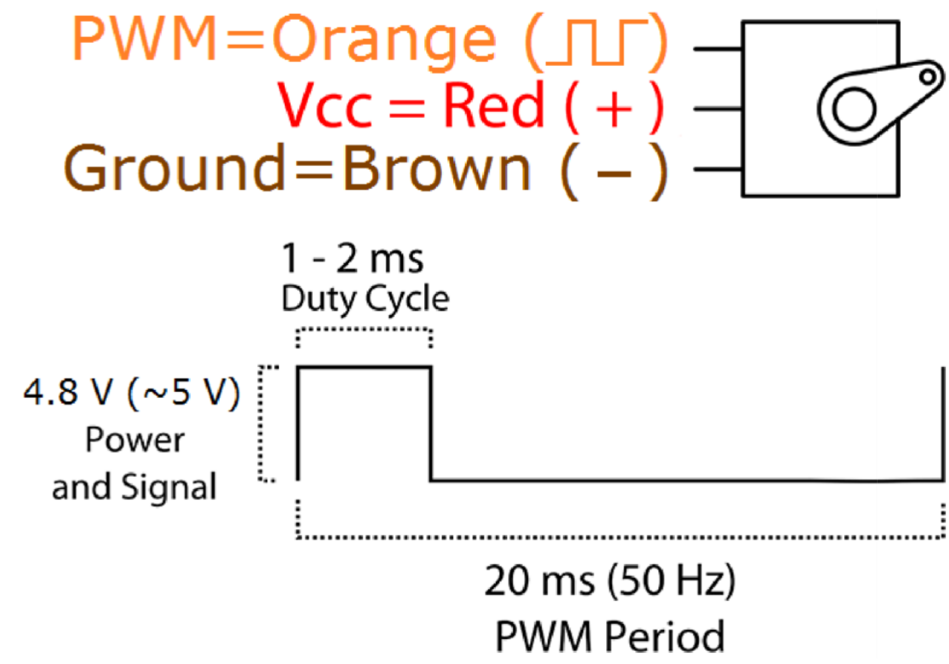


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs

- Temperature range: 0 ºC – 55 ºC

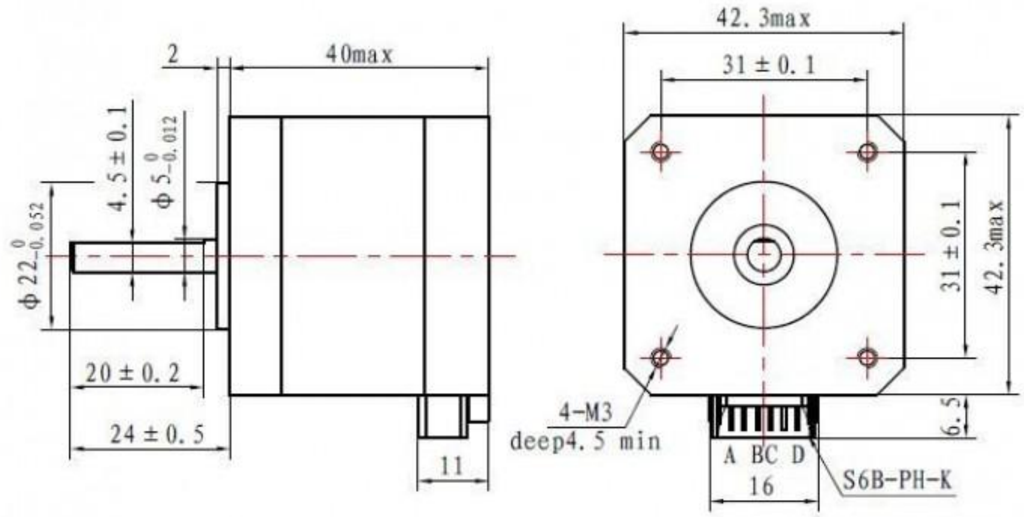


Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Spec of Stepper Motor

HYBRID STEPPING MOTOR 42SHD0217-24B Casun®

General specification		Electrical specification	
Step angle	1.8°	Rated voltage	3.75V
Number of phase	2	Rated current	1.5A
Insulation resistane	100MΩmin. (500V DC)	Resistance per phase	2.5Ω ± 10%
Insulation class	Class B	Inductance per phase	5.0mH ± 20%
Rotor inertia	57g.cm²	Holding torque	500mN.m
Mass	0.24kg	Detent torque	15mN.m



A4988

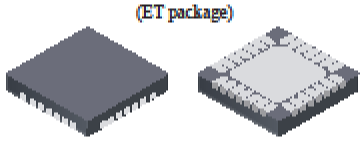
DMOS Microstepping Driver with Translator And Overcurrent Protection

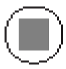
Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^1/_2$, $^1/_4$, $^1/_8$, and $^1/_{16}$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)





Approximate size

Description

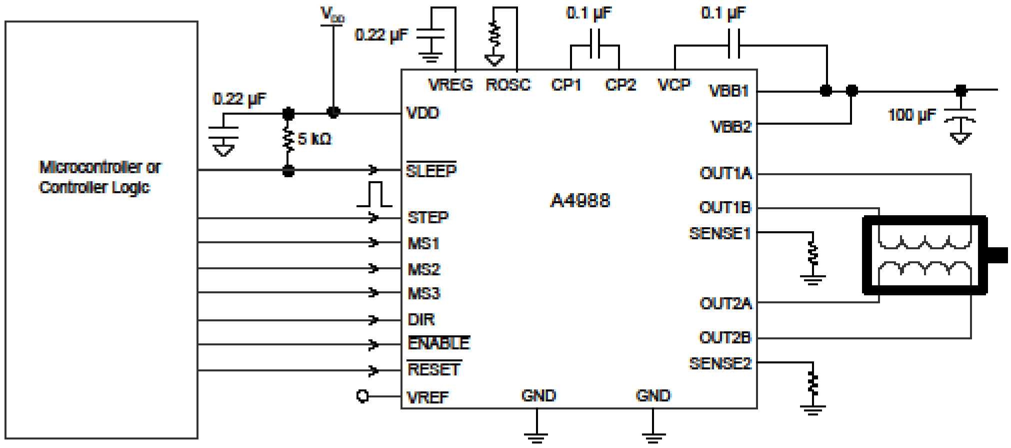
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page…

Typical Application Diagram



4988-DS, Rev. 5

Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix –T), with 100% matte tin plated leadframes.

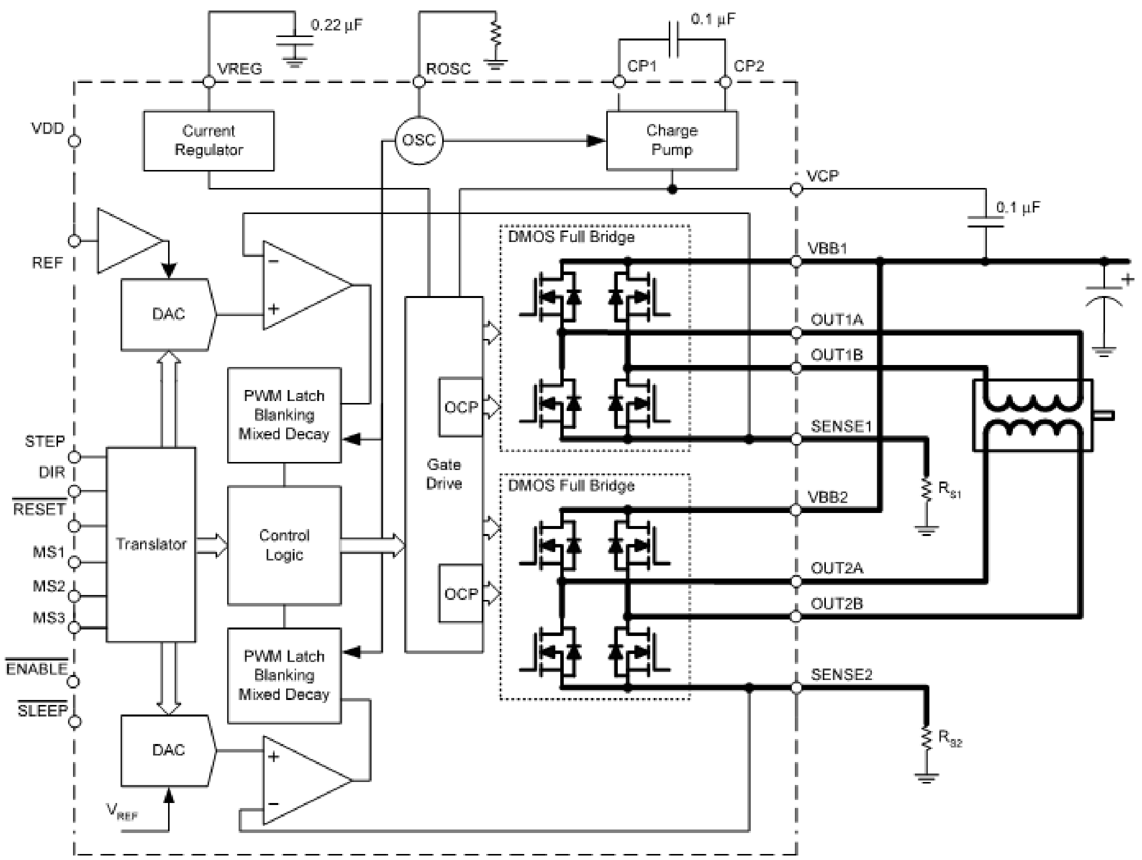
Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BB}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		–0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		–0.3 to 5.5	V
Motor Outputs Voltage			–2.0 to 37	V
Sense Voltage	V_{SENSE}		–0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	–20 to 85	ºC
Maximum Junction	$T_J(max)$		150	ºC
Storage Temperature	T_{stg}		–55 to 150	ºC

Functional Block Diagram



ELECTRICAL CHARACTERISTICS[1] at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ.[2]	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, I_F = –1.5 A	–	–	1.2	V
		Sink Diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		V_{DD}×0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD}×0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD}×0.7	–20	<1.0	20	µA
	$I_{IN(0)}$	V_{IN} = V_{DD}×0.3	–20	<1.0	20	µA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	µs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	µs
		R_{OSC} = 25 kΩ	23	30	37	µs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	µA
Current Trip-Level Error[3]	err_I	V_{REF} = 2 V, $\%I_{TripMAX}$ = 38.27%	–	–	±15	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 70.71%	–	–	±5	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 100.00%	–	–	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold[4]	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

[1]For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

[2]Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

[3]V_{ERR} = [(V_{REF}/8) – V_{SENSE}] / (V_{REF}/8).

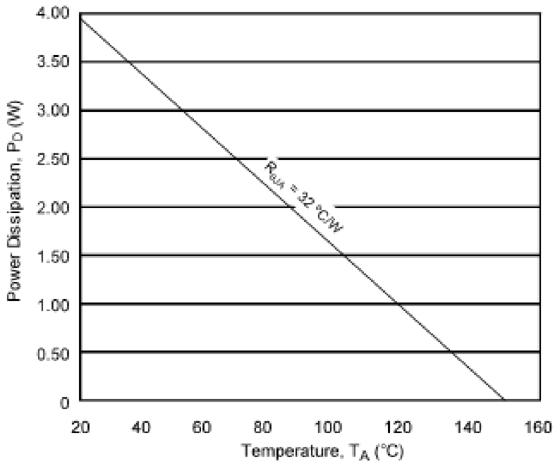
[4]Overcurrent protection (OCP) is tested at T_A = 25°C in a restricted range and guaranteed by characterization.

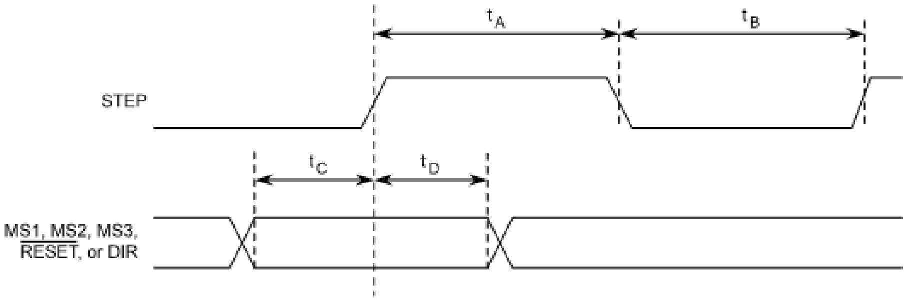
THERMAL CHARACTERISTICS

Characteristic	Symbol	Test Conditions*	Value	Units
Package Thermal Resistance	$R_{\theta JA}$	Four-layer PCB, based on JEDEC standard	32	°C/W

*Additional thermal information available on Allegro Web site.

Power Dissipation versus Ambient Temperature





Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	μs
STEP minimum, LOW pulse width	t_B	1	μs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns

Figure 1: Logic Interface Timing Diagram

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Functional Description

Device Operation. The A4988 is a complete microstepping motor driver with a built-in translator for easy operation with minimal control lines. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth, and sixteenth-step modes. The currents in each of the two output full-bridges and all of the N-channel DMOS FETs are regulated with fixed off-time PWM (pulse width modulated) control circuitry. At each step, the current for each full-bridge is set by the value of its external current-sense resistor (R_{S1} and R_{S2}), a reference voltage (V_{REF}), and the output voltage of its DAC (which in turn is controlled by the output of the translator).

At power-on or reset, the translator sets the DACs and the phase current polarity to the initial Home state (shown in Figures 9 through 13), and the current regulator to Mixed Decay Mode for both phases. When a step command signal occurs on the STEP input, the translator automatically sequences the DACs to the next level and current polarity. (See Table 2 for the current-level sequence.) The microstep resolution is set by the combined effect of the MSx inputs, as shown in Table 1.

When stepping, if the new output levels of the DACs are lower than their previous output levels, then the decay mode for the active full-bridge is set to Mixed. If the new output levels of the DACs are higher than or equal to their previous levels, then the decay mode for the active full-bridge is set to Slow. This automatic current decay selection improves microstepping performance by reducing the distortion of the current waveform that results from the back EMF of the motor.

Microstep Select (MSx). The microstep resolution is set by the voltage on logic inputs MSx, as shown in Table 1. The MS1 and MS3 pins have a 100 kΩ pull-down resistance, and the MS2 pin has a 50 kΩ pull-down resistance. When changing the step mode the change does not take effect until the next STEP rising edge.

If the step mode is changed without a translator reset, and absolute position must be maintained, it is important to change the step mode at a step position that is common to both step modes in order to avoid missing steps. When the device is powered down, or reset due to TSD or an over current event the translator is set to the home position which is by default common to all step modes.

Mixed Decay Operation. The bridge operates in Mixed decay mode, at power-on and reset, and during normal running according to the ROSC configuration and the step sequence, as shown in Figures 9 through 13. During Mixed decay, when the trip point is reached, the A4988 initially goes into a fast decay mode for 31.25% of the off-time, t_{OFF}. After that, it switches to Slow decay mode for the remainder of t_{OFF}. A timing diagram for this feature appears on the next page.

Typically, mixed decay is only necessary when the current in the winding is going from a higher value to a lower value as determined by the state of the translator. For most loads automatically-selected mixed decay is convenient because it minimizes ripple when the current is rising and prevents missed steps when the current is falling. For some applications where microstepping at very low speeds is necessary, the lack of back EMF in the winding causes the current to increase in the load quickly, resulting in missed steps. This is shown in Figure 2. By pulling the ROSC pin to ground, mixed decay is set to be active 100% of the time, for both rising and falling currents, and prevents missed steps as shown in Figure 3. If this is not an issue, it is recommended that automatically-selected mixed decay be used, because it will produce reduced ripple currents. Refer to the Fixed Off-Time section for details.

Low Current Microstepping. Intended for applications where the minimum on-time prevents the output current from regulating to the programmed current level at low current steps. To prevent this, the device can be set to operate in Mixed decay mode on both rising and falling portions of the current waveform. This feature is implemented by shorting the ROSC pin to ground. In this state, the off-time is internally set to 30 µs.

Reset Input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input sets the translator to a predefined Home state (shown in Figures 9 through 13), and turns off all of the FET outputs. All STEP inputs are ignored until the $\overline{\text{RESET}}$ input is set to high.

Step Input (STEP). A low-to-high transition on the STEP

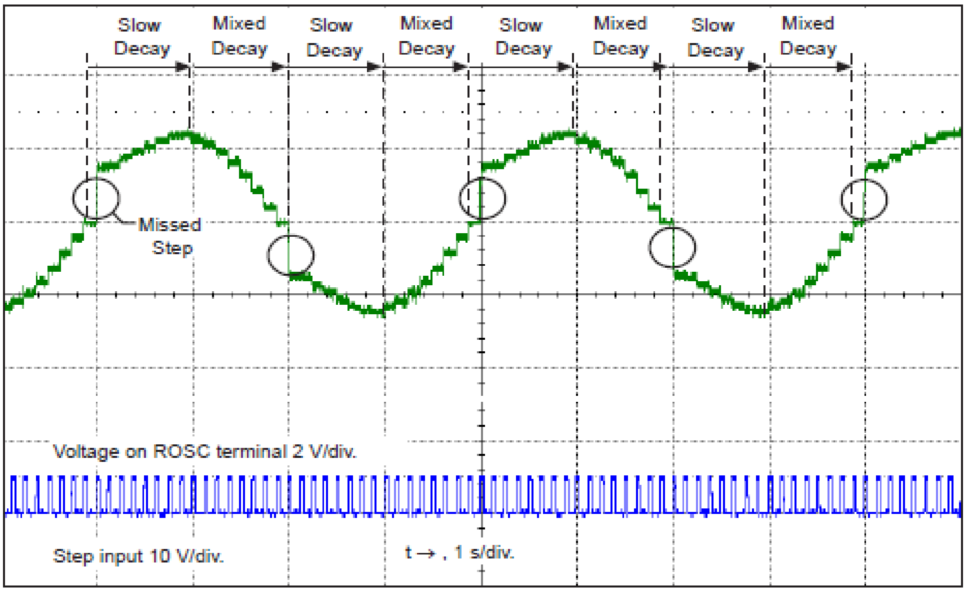


Figure 2: Missed Steps in Low-Speed Microstepping

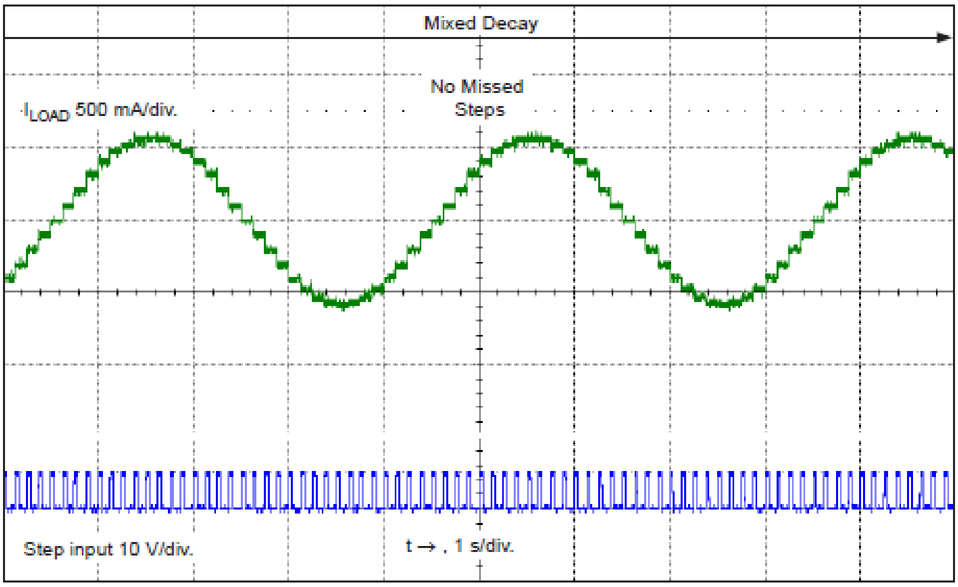


Figure 3: Continuous Stepping Using Automatically-Selected Mixed Stepping (ROSC pin grounded)

Digilent Analog Discovery 2 Reference Manual

Analog Discovery 2 – NI Edition

ANALOG DISCOVERY 2 – NI EDITION OVERVIEW

The Analog Discovery 2 – NI Edition is a pocket-sized device that transforms any PC into an electrical engineering workstation. Driven by the WaveForms™ software front panel, the USB-powered Analog Discovery lets you build and test analog and digital circuits in any environment with the functionality of a stack of lab equipment:

- 2-Channel Oscilloscope
- 2-Channel Waveform Generator
- 16-Channel Logic Analyzer
- 16-Channel Digital Pattern Generator
- Digital I/O
- Voltmeter
- Spectrum Analyzer
- Network Analyzer
- ±5VDC Adjustable Power Supplies

DETAILED SPECIFICATIONS

Analog Inputs

- Two fully differential channels; 14-bit converters; 100 MSPS real-time sample rate
- 500uV to 5V/division; 1MΩ, 24pF inputs with 9MHz analog bandwidth (30MHz using BNC Adapter Board)
- Input voltages up to ±25V on each input (±50V differential); protected to ±50V
- Up to 16k samples/channel buffer length
- Advanced triggering modes (edge, pulse, transition types, hysteresis, etc.)
- Trigger in/trigger out allows multiple instruments to be linked
- Selectable channel sampling mode (average, decimate, min/max)
- Mixed signal visualization (analog and digital signals share same view pane)
- Real-time FFTs, XY plots, Histograms and other functions always available
- Multiple math channels support complex functions
- Cursors with advanced data measurements available on all channels
- All captured data files can be exported in standard formats
- Scope configurations can be saved, exported and imported

Arbitrary Waveform Generator

- Two channels; 14-bit converters; 100 MSPS real-time sample rate
- Single-ended waveforms with offset control and up to ±5 V amplitude
- 9MHz analog bandwidth and up to 16k samples/channel (12MHz using BNC Adapter Board)
- Easily defined standard waveforms (sine, triangle, sawtooth, etc.)
- Easily defined sweeps, envelopes, AM and FM modulation
- User-defined arbitrary waveforms can be defined using standard tools (e.g. Excel)

Digital Pattern Generator

- 16 signals shared between analyzer, pattern generator, and discrete I/O
- 100 MSPS, with buffers supporting up to 16K transitions per pin
- Algorithmic pattern generator (no memory buffers used)
- Custom pattern editor supports up to 16K transitions per pin
- 3.3V outputs
- Data file import/export using standard formats
- Customized visualization options for signals and busses

Digital I/O

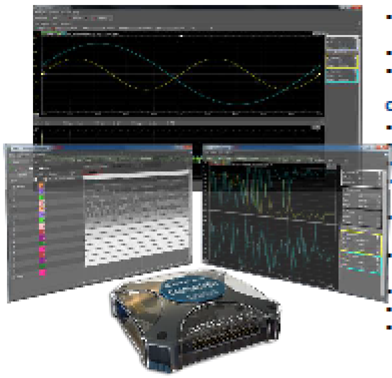
- 16 signals shared between analyzer, pattern generator, and discrete I/O
- LVCMOS (3.3 V) logic level inputs and outputs
- PC-based virtual I/O devices (buttons, switches & displays) drive physical pins
- Customized visualization options available

Power Supplies

- Two power supplies that can be powered from USB port or through external power
- ±0.5V to ±5V up to 500mW combined (700mA max) when powered through USB, and up to 2.1W per channel (700mA max) when connected to external power

Logic Analyzer

- 16 signals shared between analyzer, pattern generator, and discrete I/O
- 100 MSPS, with buffers supporting up to 16K transitions per pin
- LVCMOS logic level inputs
- Multiple trigger options including pin change, bus pattern, etc.
- Trigger in/trigger out allows multiple instruments to be linked
- Interpreter for SPI, I2C, UART, Parallel bus
- Captured signals can be saved and exported in standard file formats



Spectrum Analyzer

- Performs FFT or CZT algorithm on analog input channels and displays power spectrum
- Frequency range adjustments in center/span or start/stop modes
- Linear or logarithmic frequency scale
- Peak tracking option finds peak power and adjusts display to keep peak in center of display
- Vertical axis supports voltage-peak, voltage-RMS, dBV and dBu display options
- Windowing options include rectangular, triangular, hamming, Cosine, and many others
- Cursors and automatic measurements including noise floor, SFDR, SNR, THD, and many others
- Data file export using standard formats

Network Analyzer

- Waveform generator drives circuits with swept sine waves up to 10MHz
- Input waveforms settable from 1Hz to 10MHz, with 5 to 1000 steps
- Settable input amplitude and offset
- Analog input records response at each frequency
- Response magnitude and phase delay displayed in Bode, Nichols, or Nyquist formats

Voltmeters

- Two independent meters (shared with Analog input channels)
- Automatic measurements include DC, AC RMS and True RMS values
- Single-ended and differential measurement capability
- Up to ±25V on each pin (±50V max peak-peak)
- Auto-range feature selects best gain range

Other features

- Compatible with Windows, Mac, and Linux
- USB powered; all needed cables included
- High-speed USB2 interface for fast data transfer
- Software Development Kit provided for custom applications
- Waveform Generator output can be played on stereo audio jack
- Two external trigger pins can link triggers across multiple devices
- Cross triggering between instruments
- Help screens, including contextual help
- Instruments and workspaces can be individually configured; configurations can be exported

ALL SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

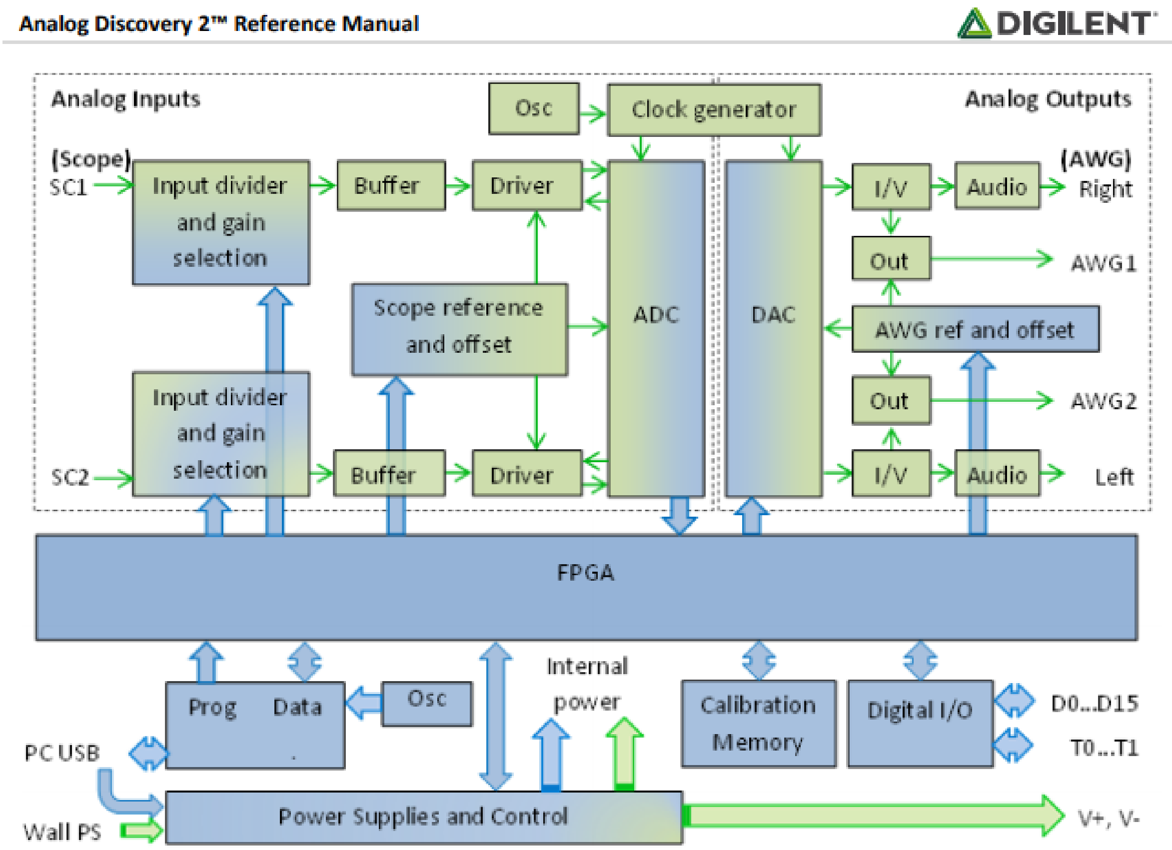


Figure 88 Analog Discovery 2 block diagram. From: Digilent Analog Discovery 2 Reference Manual

5 Digital I/O

Figure 22 shows half of the Digital I/O pin circuitry (the other half is symmetrical). J3 is the Analog Discovery 2 user signal connector.

General purpose FPGA I/O pins are used for Analog Discovery 2 Digital I/O. FPGA pins are set to SLOW slew rate and 4mA drive strength, with no internal pull.

PTC thermistors provide thermal protection in case of shortcuts. Schottky Diodes double the internal FPGA ESD protection diodes for increasing the acceptable current in case of overvoltage. Nominal resistance of the PTCs (220Ω) and parasitical capacitance of the Schottky diodes (2.2pF) and FPGA pins (10pF) limit the bandwidth of the input pins. For output pins, the PTCs and the load impedance limit the bandwidth and power.

Input and output pins are LVCMOS3V3. Inputs are 5V tolerant. Overvoltage up to ±20V is supported.

Figure 89 Description Digital I/O. From: Digilent Analog Discovery 2 Reference Manual

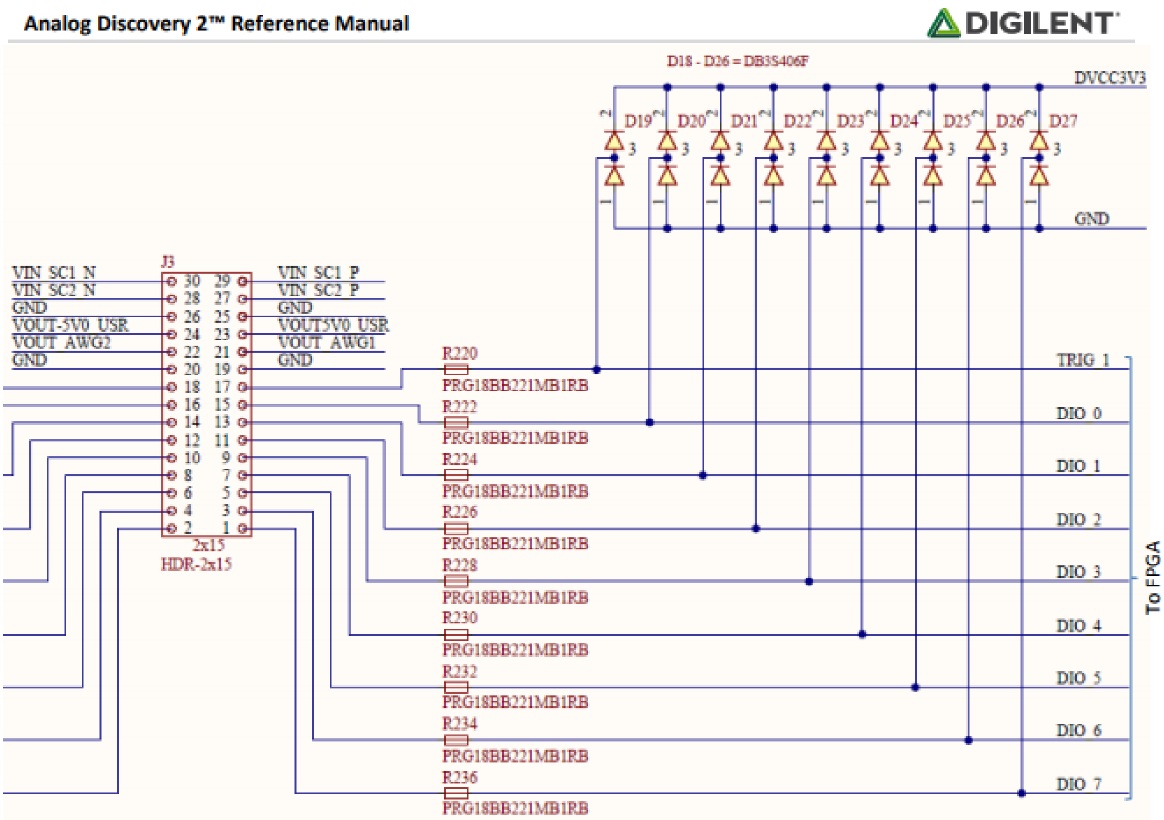


Figure 90 Digital I/O. From: Digilent Analog Discovery 2 Reference Manual

9.4 Digital Pattern Generator

- 16 signals shared between analyzer, pattern generator, and discrete I/O[xxxvi]
- 100 MSPS
- Algorithmic pattern generator (no memory buffers used)[xxxvii]
- Custom pattern editor with buffers supporting up to 16K transitions per pin[xxxviii]
- 3.3V outputs
- Data file import/export using standard formats[xxxix]
- Customized visualization options for signals and busses[xl]

9.5 Digital I/O

- 16 signals shared between analyzer, pattern generator, and discrete I/O[xli]
- LVCMOS (3.3 V) logic level inputs and outputs
- PC-based virtual I/O devices (buttons, switches & displays) drive physical pins[xlii]
- Customized visualization options available[xliii]

9.6 Power Supplies

- Two fixed power supplies derive power from USB port
- +5V up to 50mA and -5V up to 50mA (100mA total)

Figure 91 Features and performance of Analog Discovery 2. From: Digilent Analog Discovery 2 Reference Manual