Issue 6 Volume 3 ISSN: 2321-8134

IJFEAT

INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY

Design & Development of Stepper Motor Based X-Y Positioning System Using Arduino Controller.

Mr. Shubham Sahane #1, Mr. Rahul Shewale #2, Mr. Suraj Gade #3, Mr. Aniket Subhedar#4, Prof. N.K.Gavade#5

[1]*Student, Mechanical Department, SKNSITS,Maharashtra,India, sahaneshubham@gmail.com*
[2]*Student, Mechanical Department, SKNSITS,Maharashtra,India, rrshewale001@gmail.com*
[3]*Student, Mechanical Department, SKNSITS,Maharashtra,India,*
[4]*Student, Mechanical Department, SKNSITS,Maharashtra,India,*
[5]*Assistant Professor, Mechanical Department, SKNSITS, Maharashtra, India, nkg.sknsits@sinhgad.edu*

Abstract

Various Types of X-Y control Positioning System are used in industries for many purpose. In industry this mechanism is operated by PLC and Microcontroller .But the cost of PLC is high. So PLC is replaced by arduino controller. Arduino controller operates on IDE programme. The goal of this project was to develop a XY Scanning stage to translate the motion stage along the X and Y axes of the gantry and to use this information as the output for a microcontroller that can modify the commanded position of the stepper motor as the input data provided. This system use in small scale industries for plotting or cutting linear shape.

Keywords- *X-Y Scanning stage, Positioning System, Arduino Controller, IDE, Stepper Motor*

--- *** ---

1. Introduction

3D printing is an additive manufacturing process for creating objects directly, by adding material layer by layer in a variety of ways, depending on the technology used [8]. The starting point for any 3D printing process is a 3D digital model, which can be created using a variety of 3D software programmed. The model is then 'sliced' into layers, thereby converting the design into a file readable by the 3D printer. The material processed by the 3D printer is then layered according to the design and the process. The printer must be able to position the hot-end at any point to be able to precisely extrude material within the layer being printed. For this reason a special type of motion mechanism, a gantry (XY stage) is used which moves the material extruder in the X- and Y-axes, while the bed moves only in the Z-axis. Generally, in 3d printers NEMA-17 stepper motors are used to power their stages. These motors are driven by integrated-circuit motor drivers with peak currents in the range of 1 A per phase, step-direction interfaces to the host controller, and 8X or 16X micro stepping capabilities. In 3D printers, the gantry [10] is the frame structure that supports the printer head along the XY axis as the printer head moves around to print the part on the build platform. The gantries are moved by stepper motors, which use digital pulses to move and track the gantry. The stepper motors allow for high resolution movement by allowing the pulses to move the motor at a small fraction of a rotation. The gantries hold up the printer head as it moves along the build platform.

2. Literature Survey

The **Kuang-Chao Fana, Yetai Feia, Xiaofen Yua, Weili Wanga, Yejin Chen[1]** were developed an innovative CMM design, including the arch-bridge, the co-planar precision XY-stage, the spindle, the motion actuator and feedback system, and the auto focusing probe. This micro-CMM is designed for the measurement of mesoscale parts to the accuracy of nanometer range. It is aimed at achieving 1 nm resolution and 30 nm accuracy within a measuring range of $25\times25\times10$ mm. Conventional XY stage is stacked up by two linear stages composing of many components, such as ball screw, bearing, linear slide, etc. The long travel of each axis is activated by the piezo-ceramic ultrasonic motor with its AC drive mode. The fine positioning is achieved by the same motor with its DC drive mode, which can perform nanometer steps proportional to the input voltages. A linear diffraction grating interferometer (LDGI) is developed as the position feedback sensor with the resolution to 1 nm after the waveform interpolation. The motion of each axis can be calibrated using a laser interferometer in a temperature- and humidity-controlled room. The whole software system of this prototype micro- CMM is developed in the commercially available Lab-VIEW 6.0 development system. **Jooho Hwang, Chun-Hong Park, Chan-Hong Lee, Seung-Woo Kim.[3]** suggested an estimation and correction method for the two-dimensional (2D) position errors of a planar XY stage that is driven along the Y-axis by two linear motors. The 2D position errors of the stage were estimated and corrected based on measured motion errors from a conventional laser interferometer system. To compensate for the planar XY stage 2D position errors, corrections were introduced for the yaw, perpendicular, straightness and 1D position errors along each axis, which are predominantly caused by linear scale, yaw, and pitching motion errors. The effect of the motion error corrections was evaluated by diagonal measurements based on the ISO230-6 standard and six different 1D position error measurements along the X and Y-axes. By applying error motion corrections, the diagonal systematic deviation at the center point was improved from 499.7 mm to 1.16 mm, and the estimated maximum 2D position errors were improved to K0.263 and 0.530 mm in the X and Y directions, respectively. The diagonal systematic deviation at a corner point was 1.23 mm and the estimated maximum difference between the corner and center points improved from K2.603 and 2.603 mm to K0.05 and 0.12 mm in the X and Y directions respectively.

Yuki Shimizua,⁎, Yuxin Penga, Junji Kanekoa, Toyohiro Azumaa, So Itoa,WeiGaoa, Tien-Fu Lu.[4] were designed and constructed motion mechanism of an XY micro-stage for precision positioning of lightweight objects. A driving unit, which consists of two PZTs and a friction component made by permanent magnet, is mounted on the center of the stage base for driving the stage moving plate made by steel in the X- and Y-directions based on a friction drive. The moving plate is attached on the friction component by the magnetic force. Leaf springs are employed to guide the X- and Y-directional motions of the moving plate over a range of ±1 mm in both directions. The stage motion mechanism, which dominates the overall size of XY micro-stage, has a compact size of 24 mm (X) × 24 mm (Y) × 5 mm (Z). The basic performances of the stage motion mechanism have been investigated by experiments. Positioning resolution was tested when the developed motion mechanism of the stage was under closed-loop control in one-axis by employing an external position sensor. The spectral-interference laser displacement meter used was also employed as a feedback sensor. In this study, a software feedback control was employed to position the stage plate.

3. Design of the X-Y scanning Stage

The XY stage is modeled using modeling software CATIA V5. CATIA (Computer Aided Three dimensional Interactive Application) is a multiplatform CAD/CAM/CAE commercial software suite developed by the French company Dassault Systèmes directed by Bernard Charlès. Written in the C++ programming language, CATIA Version 5 uses the Sketcher workbench as its principal method is to create profiles. The entire stage is assembled considering the designed constraints

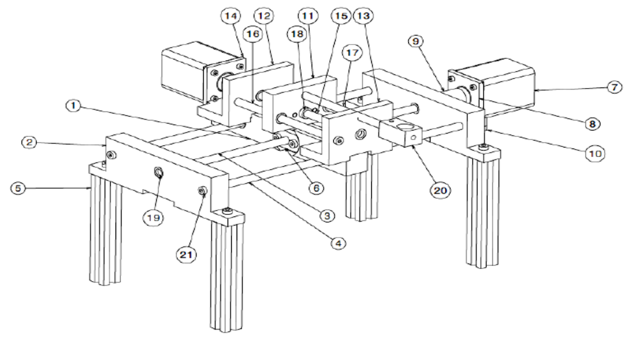


Fig.1 CAD model of X-Y stage in CATIA V5

Figure 1 show, the X-Y scanning stage consists of two motion stages. The Y-axis stage sits upon three rods located in between the two side support plate, here the three rods are the two guide rods and the lead screw, the lead screw located at the middle of guide rods allows linear motion of the X-stage along the rods axis. A stepper motor drives a shaft that is coupled to the lead screw with a motor coupler on one side of the Y-axis, allowing it to move the Y stage platform along with the X stage mounted on it. Here, the X stage is located in between the two side plates to restrict the linear motion of X-stage through X axis which are mounted on the Y stage platform. The entire XY stage assembly is resting on the support rods of squared and grooved cross section made of Aluminium

Sr no	Name of the part	Qty.	Material
1	Y motion stage	1	Al
2	Side support plate for Y motion stage	2	M.S.
3	M8 threaded rod for Y motion stage	1	MS
4	Guide rod for Y motion stage	2	MS
5	Supporting rod for XY mechanism	4	Al
6	Nut for Y motion stage	1	MS
7	Stepper motor	2	(bought)
8	Y motion motor mounting plate	1	MS
9	Motor coupler	2	Al
10	Y motion motor connecting plate	1	MS
11	X motion stage	1	Al
12	X motion motor side plate	1	Al
13	X motion motor opposite plate	1	Al
14	X motion motor mounting plate	1	Al

Issue 6 Volume 3 ISSN: 2321-8134

15	M8 threaded rod for X motion stage	1	MS
17	Guide rod for laser mounting	1	MS
18	Nut for X motion stage	1	(bought)
19	Bush for thread rod support	4	MS
20	Laser holder	1	Nylon
21	Allen screw	25	(bought)

Table 1- Part list of X-Y Positioning Stage Assembly

3.1 Components Dimensions

Name of the part	Dimensions
Y-Motion motor mounting plate	55×55×5
Y-Motion motor side plate	62×55×10
Y-motion side support plate	197×45×20
Y-Motion Stage	190×25×25
X-Motion Motor Opposite Plate	75×55×10
X-Motion Motor Side Plate	75×55×10
X-Motion Motor Mounting Plate	70×45×5
X-Motion Stage	75×45×15
A lead screw with nut	M8×1.25, length 340mm(Y-Stage) and 170mm(X-Stage)
Guide rods	Ø8×292mm and Ø8×140mm
Laser mounts assembly	Ø8×140mm, 50×20mm
Support rods	25×25 Length=125mm
Aluminum Flexible Coupling	Shaft: 5mm, 8mm Diameter: 18mm
Linear Ball Bearing	8 × 15 × 24

Table 2-Components Dimensions

4. Detailed Manufacturing Drawings

A.Y-Motion Motor Mounting Plate-

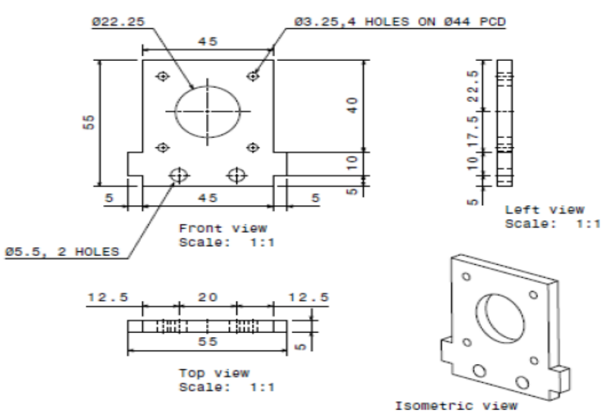


Fig.2 Y Motion Motor Mounting Plate

B. Y-Motion Motor Connecting Plate-

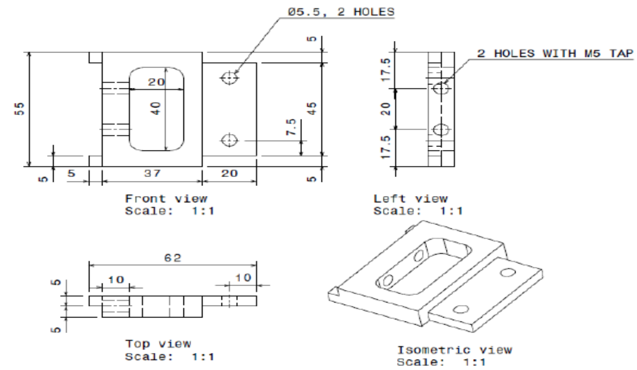


Fig.3 Y-Motion Motor Connecting Plate

C. Side Supporting Plate for Y-Motion Stage-

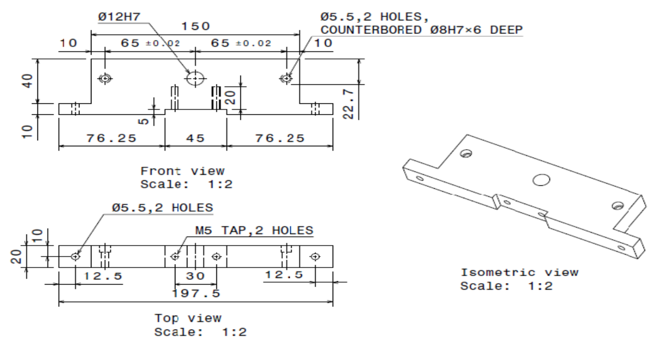


Fig.4 Side Supporting Plate for Y-Motion Stage

D. Y-Motion Stage-

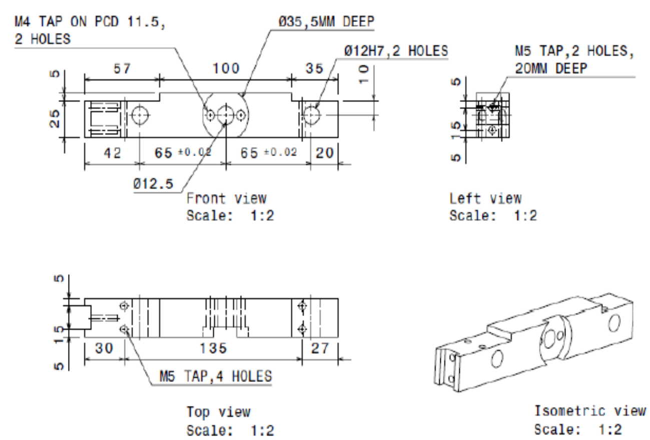


Fig.5 Y Motion Stage

E. X-Motion Motor opposite Plate-

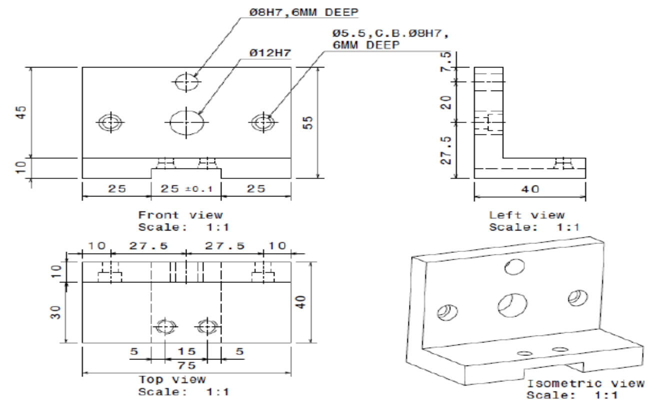


Fig.6 X-Motion Motor opposite Plate

F. X-Motion Motor Side Plate-

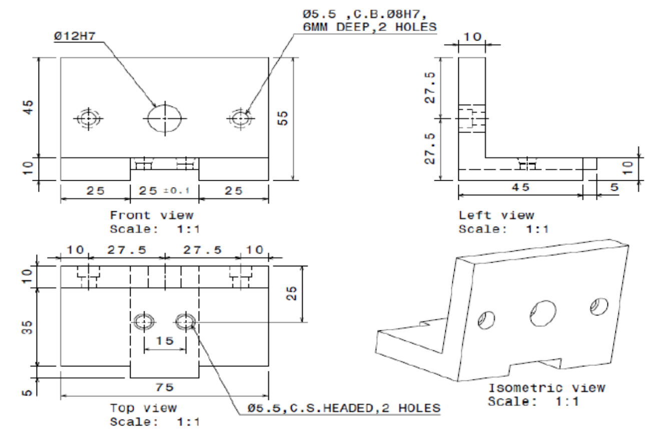


Fig.7 X-Motion Motor Side Plate

G. X-Motion Motor Mounting Plate-

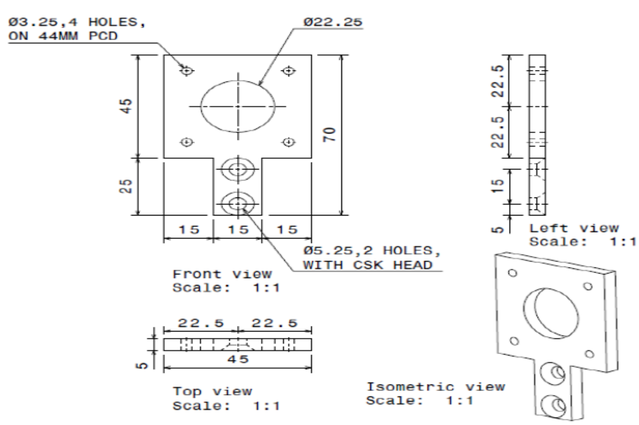


Fig.8 X-Motion Motor Mounting Plate

H. X-Motion Stage-

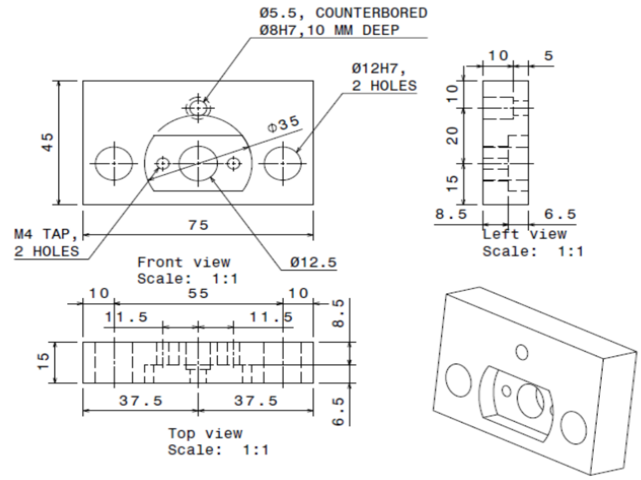


Fig.9 X-Motion Stage

4.1 Purchased Components-

A. A lead screw with nut-

Fig.10 A lead screw with nut

B. Guide rods-

Fig.11 B) Guide rods

C. Laser mounts assembly-

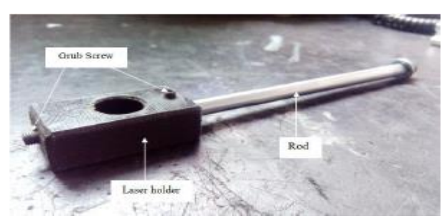


Fig.12 Laser mounts assembly

E. Aluminum Flexible Coupling-

Fig.13 Aluminum Flexible Coupling

F. Stepper Motor-

Fig.14 Stepper Motor

G. Linear Ball Bearing-

Fig.15 Linear Ball Bearing

5. Mechatronics Interfacing-

A stepper motor is a DC motor in which the rotation can be divided up into a number of smaller steps. This is done by having an iron gear-shaped rotor attached to the shafts inside the motor. Around the outside of the motor are electromagnets with teeth. One coil is energized, causing the teeth of the iron rotor to align with the teeth of the electromagnet. The teeth on the next electromagnet are slightly offset from the first; when it is energized and the first coil is turned off, this causes the shaft to rotate slightly more to toward the next electromagnet. This process is repeated by however many electromagnets are inside until the teeth are almost aligned with the first electromagnet, and the process is repeated. Each time an electromagnet is energized and the rotor moves slightly, it is carrying out one step. By reversing the sequence of electromagnets energizing the rotor, it turns in the opposite direction. The job of the Arduino is to apply the appropriate HIGH and LOW commands to the coils in the correct sequence and speed to enable the shaft to rotate. This is what is going on behind the scenes in the stepper.h library. The NEMA 17 stepper motor is connected to an easy driver motor controller board which is controlled by an Arduino Mega2560 microcontroller by means of wire connections as shown in block diagram of fig.5.1. The Arduino program is uploaded in the microcontroller which controls the number of pulses to be generated for the stepper motor provided with a required amount of voltage level by means of easy driver circuit. Fig.5.2 shows the XY stage interfacing with PC.

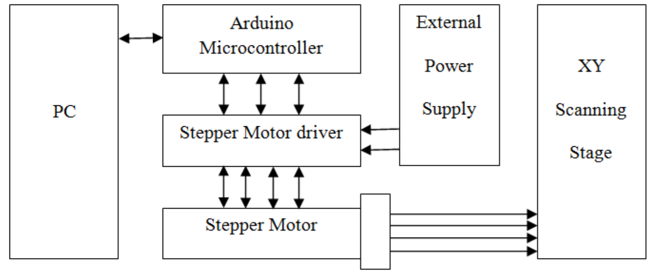


Fig.16 Block diagram of computer interfacing of XY Positioning Stage

5.1 The Arduino Hardware

The Arduino board is a small microcontroller board, which is a small circuit (the board) that contains a whole computer on a small chip (the microcontroller). From figure 3 represents Arduino Mega328 used. The Arduino Mega 328 is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila [15].

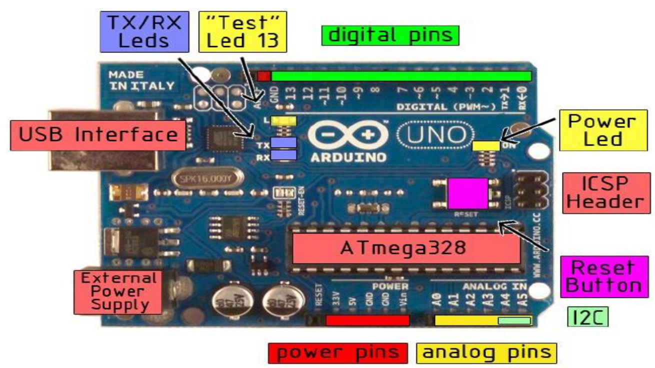


Fig.17 Schematics of Arduino Mega328 Microcontroller

5.2 The Software (IDE)

The IDE (Integrated Development Environment) is a special program running on your computer that allows you to write sketches for the Arduino board in a simple language modeled after the Processing (www.processing.org) language. The magic happens when you press the button that uploads the sketch to the board: the code that you have written is translated into the C language (which is generally quite hard for a beginner to use), and is passed to the avr-gcc compiler, an important piece of open source software that makes the final translation into the language understood by the microcontroller. This last step is quite important, because it's where Arduino makes your life simple by hiding away as much as possible of the complexities of programming microcontrollers [15].

The programming cycle on Arduino is basically as follows:

1. Plug your board into a USB port on your computer.
2. Write a sketch that will bring the board to life.
3. Upload this sketch to the board through the USB connection and wait a couple of seconds for the board to restart.
4. The board executes the sketch that you wrote.

When you open up the Arduino IDE, it will look very similar to the Windows version in the image as shown in fig.5.5 below. If you are using OSX or Linux, there may be some slight differences, but the IDE is pretty much the same no matter what OS you use.

5.3 Arduino Programming

The Arduino runs a simplified version of the C programming language, with some extensions for accessing the hardware. All Arduino instructions are one line. The board can hold a program hundreds of lines long and has space for about 1,000 two-byte variables. The Arduino executes programs at about 300,000 source code lines per sec. Programs are created in the Arduino development environment and then downloaded to the Arduino board. Code must be entered in the proper syntax which means using valid command names and a valid grammar for each code line. All Arduino programs have two functions, setup () and loop (). The instructions you place in the setup () function are executed once when the program begins and are used to initialize. Use it to set directions of pins or to initialize variables. The instructions placed in loop () are executed repeatedly and form the main tasks of the program. Therefore every program has this structure [15].

Program 1: To move the XY motion stage forward and backward with Easy Driver board.

```
//This code controls a stepper motor with the
//EasyDriver board. It spins forwards and backwards
//***************************/
int dirpin = 8;
int steppin = 9;
void setup()
{
pinMode(dirpin, OUTPUT);
pinMode(steppin, OUTPUT);
}
void loop()
{
int i;
digitalWrite(dirpin, LOW); // Set the direction.
delay(100);
for (i = 0; i<30000; i++) // Iterate for 4000 microsteps.
{
digitalWrite(steppin, LOW); // This LOW to HIGH change is
what creates the
digitalWrite(steppin, HIGH); // "Rising Edge" so the
easydriver knows to when to step.
delayMicroseconds(200); // This delay time is close to top
speed for this
} // particular motor. Any faster the motor stalls.
digitalWrite(dirpin, HIGH); // Change direction.
delay(100);
for (i = 0; i<30000; i++) // Iterate for 4000 microsteps
{
digitalWrite(steppin, LOW); // This LOW to HIGH change is
what creates the
digitalWrite(steppin, HIGH); // "Rising Edge" so the
easydriver knows to when to step.

delayMicroseconds(200); // This delay time is close to top
speed for this
} // particular motor. Any faster the motor stalls.
}
```

This XY Scanning Stage is programmed to control the linear movement of the X and Y motion stage with the help of Stepper motor and lead screw. The stepper motors rotational motion is transformed in to linear motion with help of lead screw coupled with motor shaft. As the lead screw nut holds the motion stage, results in the movement of motion stage as lead screw translates. The main aim of this program is to control the movement of the motion stage, precisely and accurately with the help of Arduino microcontroller for precision and accuracy. The rotational movement of the stepper motor is controlled with the help of special Arduino program. The pin connections are shown in fig.5.13. Some calculations regarding the steps are as follows,

Step angle = 1.8 degree

Steps per revolution = 200

One revolution = 200 steps =1.25mm linear motion stage in one rotation

It means that in one rotation of lead screw the platform will travel 1.25 mm.

Suppose, with 1000 steps for a stepper motor (means 5 revolutions of 200), then the platform travels 6.25mm.

Now,

1 step = 1/200 revolution = 0.005 mm linear motion of platform in one step.

Therefore, it is necessary for micro positioning to move the motion stage with 0.005 mm in one step which results in better translation for each stage.

Program 2: To control the speed 2 stepper motors to move the stage platforms one by one with Motor Drive Shield Board for micro-stepping mode.

```
#include <AFMotor.h>
AF_Stepper motor1(48, 1); // A 48-step-per-revolution motor
on channels 1 & 2
AF_Stepper motor2(48, 2);
void setup() {
Serial.begin(9600); // set up Serial library at 9600 bps
Serial.println("Stepper test!");
motor1.setSpeed(1000); // 1000 rpm
motor2.setSpeed(1000);
}
void loop()
{
```

```
motor1.step(3000, FORWARD, MICROSTEP);
motor1.step(3000, BACKWARD, MICROSTEP);
motor2.step(1000, FORWARD, MICROSTEP);
motor2.step(1000, BACKWARD, MICROSTEP);
```

5.4 Stepper Motors

Stepper motors are ideally suited for precision positioning of an object or/and precision control of speed without using closed-loop feedback for any automation systems. It is also translate switched excitation changes into precisely defined increments of rotor position {'step'}. Accurate positioning of the rotor is generally achieved by magnetic alignment of iron teeth on the stationary and rotating parts of the motor. The basic function of stepper motor is that its output shaft rotates in a series of discrete angular intervals or steps, one step being taken each time a command pulse is received. When a definite number of pulses are supplied, the shaft turns through a definite known angle. The name stepper is used because this motor rotates through a fixed angular step in response to each input current pulse received by its controller. These controllers can be computers, microprocessor and programmable controllers. Many computer peripherals, such as disk drives, printer & plotters or computer-controlled equipment like XY table & robot limbs, make use of steeping motor.

5.4.1 Principle of Operation

A stepping motor may be compared with a synchronous motor as far as operation is concerned: a rotating field, here generated by the control electronics, pulls a magnetic rotor along. Stepping motors are sub divided according to the manner in which the rotating field is generated, that is with unipolar or bipolar stator windings and the material from which the rotor has been constructed – permanent magnetic material or soft iron.

Step angle: The angle through which the motor shaft rotates for each command pulse is called the step angle $\phi = 360°/\text{No. of stator phases} \times \text{No of rotor teeth}$. Smaller the step angle, greater the number of step per revolution and higher the resolution or accuracy of positioning obtained. The angle can be as small as 0.72° or as large as 90°. But the most common step angle 1.8°, 2.5°, 7.5°, 15°.

5.4.2 Easy Stepper Motor Driver-

Fig.18 Easy Stepper Motor Driver

The Easy Driver is a simple to use stepper motor driver, compatible with anything that can output a digital 0 to 5V pulse (or 0 to 3.3V pulse if you solder SJ2 closed on the Easy Driver). The Easy Driver requires a 6V to 30V supply to power the motor and can power any voltage of stepper motor [12]. The Easy Driver has an on board voltage regulator for the digital interface that can be set to 5V or 3.3V. Connect a 4wire stepper motor and a microcontroller and you've got precision motor control! Easy Driver drives bipolar motors, and motors wired as bipolar. i.e.4, 6 or 8 wire stepper motors.

6. Experimental Testing

Once the Arduino programming is done and as soon as the upload button is pressed the XY stage is ready to go accordingly. Using the Arduino motor shield board with a forward- backward programming the X and Y motion stages translated according to the steps given in the program. Here the stepper motor has 1.8 step angle means when it completes one revolution, the number of steps taken by stepper motor are 360/1.8=200 steps. In one complete revolution of stepper motor the distance travelled is equal to the pitch of the lead screw. Here the pitch of lead screw is 1.25mm. Thus in 200 steps the distance travelled is 1.25 mm. Using this terminology the calculations regarding the distance travelled by the motion stages for a given number of steps in the program gets easier.

Now a simple experiment is done to measure the travelled distance with a digital dial indicator instrument. The pointer of dial is rested on the one of the flat surface in the direction parallel to the motion of platform as in fig.6.1. The forward and backward movement is controlled with an Arduino program with a number of steps and speed. Measurements were taken for 200, 400 & 600 RPM with varying number of steps from 100 to 2000 i.e. for twenty such readings.

7. Result & Discussion

From the experimentations the readings are interpreted in tabular form and respective motion values can be analyzed with the graphical representation.

no. of steps	motion expected (mm)	Observed X motion readings		
		at 200 RPM (mm)	at 400 RPM (mm)	at 600 RPM (mm)
100	0.625	0.598	0.498	0.486
200	1.25	1.198	1.098	1.08
300	1.875	1.835	1.7	1.66
400	2.5	2.398	2.25	2.12
500	3.125	3.08	2.89	2.88
600	3.75	3.65	3.55	3.45
700	4.375	4.305	4.28	4.15
800	5	4.85	4.6	4.55
900	5.625	5.5	5.435	5.34
1000	6.25	6.205	5.98	5.9
1100	6.875	6.8	6.74	6.63
1200	7.5	7.48	7.33	7.22
1300	8.125	8.098	7.9	7.88

1400	8.75	8.65	8.45	8.32
1500	9.375	9.305	9.01	9.09
1600	10	9.98	9.88	9.8
1700	10.625	10.6	10.5	10.56
1800	11.25	11.2	11.12	11.23
1900	11.875	11.869	11.78	11.83
2000	12.5	12.402	12.23	12.308

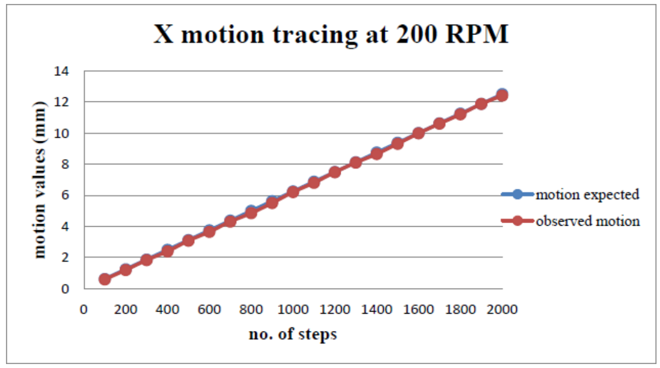
Table 3- Readings taken for the X-motion tracing

no. of steps	motion expected (mm)	Observed Y motion readings		
		at 200 RPM (mm)	at 400 RPM (mm)	at 600 RPM (mm)
100	0.625	0.485	0.365	0.4
200	1.25	1.154	1.088	1.02
300	1.875	1.843	1.53	1.45
400	2.5	2.346	2.185	2.8
500	3.125	3.062	2.98	2.88
600	3.75	3.586	3.39	3.35
700	4.375	4.165	4.08	4
800	5	4.745	4.6	4.58
900	5.625	5.425	5.1	5.09
1000	6.25	6.105	6.02	5.99
1100	6.875	6.765	6.54	6.405
1200	7.5	7.438	7.2	7.156
1300	8.125	7.985	7.63	7.57
1400	8.75	8.59	8.48	8.45
1500	9.375	9.185	9.102	9.09
1600	10	9.88	9.58	9.45
1700	10.625	10.44	10.15	10.023
1800	11.25	11.09	10.85	10.78
1900	11.875	11.68	11.52	11.45
2000	12.5	12.385	12.2	12.18

Table 4- Readings taken for the Y-motion tracing

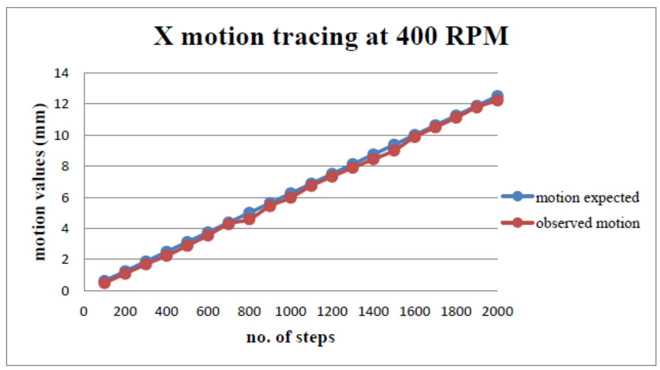
7.1 Graphical Representation

From the values obtained graphs were plotted against the number of steps on X axis with observed and expected values on Y axis. It is observed from the graphs that the X motion stage values are much closer to expected values than the Y motion stage values.



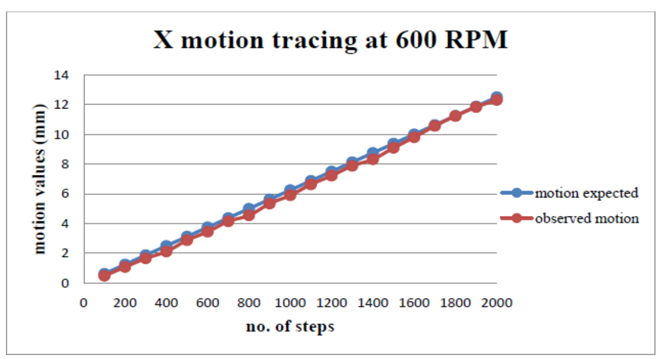
Graph 1- X-motion tracing at 200 RPM

The graph above shows that the observed motion points are nearly convergent with the expected motion points at 200 rpm.



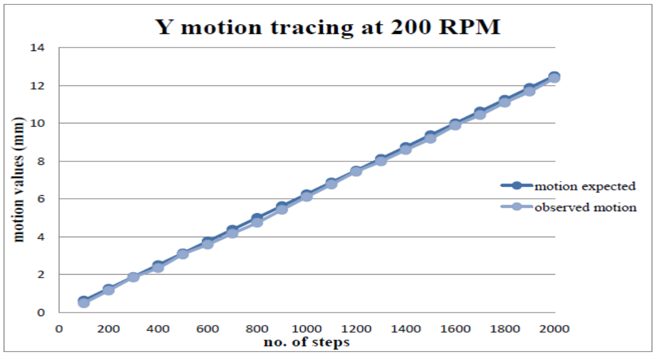
Graph 2- X-motion tracing at 400 RPM

As the speed goes on increasing the values starts deflecting from expected motion values with varying number of steps.



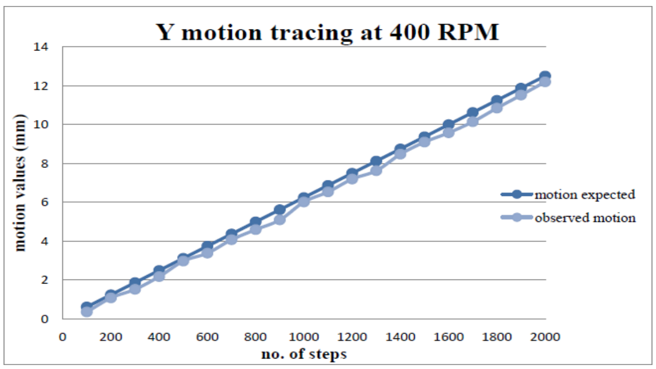
Graph 3-X-motion tracing at 600 RPM

At 600 rpm the remarkable deflections are observed. Just 3 to 4 points are convergent rest of all getting deflected.



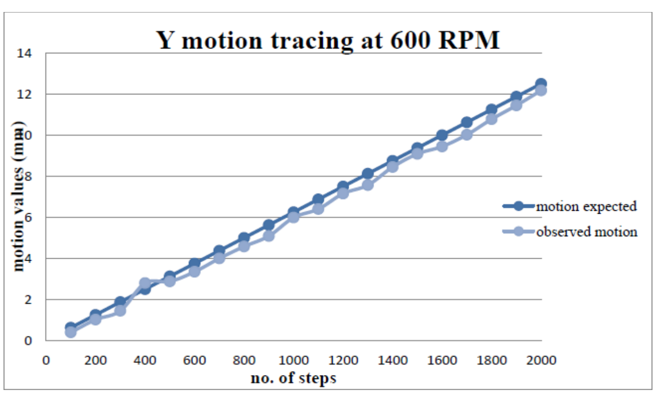
Graph 4- Y-motion tracing at 200 RPM

Nearly convergent points but the number less than X motion values were obtained with Y motion tracing at 200 rpm.



Graph 5-Y-motion tracing at 400 RPM

The convergence rate getting lesser as the speed goes on increasing.



Graph 6- Y-motion tracing at 600 RPM

Here the remarkable divergence is observed at 600 rpm. The convergent points are getting lesser.

8. Conclusion

1. The XY Scanning Stage is micro-positioning stage which is driven by two bipolar stepper motors and controlled by Arduino Mega2560 microcontroller. We have designed and developed this XY Scanning Stage to precisely position the motion stage platforms along the predefined directions given by the special programs developed using Arduino software. Here, the two basic programs are developed to test the motion of the stages. The first program translates the motion stage in forward and backward motion at given delay time to reverse the direction. It uses the Easy Driver circuit and controls the single stage motor at a time. Second program is based on the Arduino shield board which controls the two motors one by one under a micro stepping mode with a speed control.

2. Experimentation is done to measure the X and Y motions with a dial gauge at the speeds varying from 200 RPM to 1000 RPM and varying steps from 100 to 2000. The results obtained are within the acceptable limits. The accuracy within 10 microns is achieved.

3. This work is beneficial in many micro-positioning and precision applications especially in 3D Printers. It also provides the cost effective way for system interpretation and electronics.

8.1 Future Scope

Generally, the XY stages are used in precision positioning industries. They can also find the applications in micromachining, micro fabrications, and stereo- lithographic tables. This developed XY stage Scanning Stage is definitely beneficial in such micromachining applications as laser cutting, scanning profiles or geometries. As a part of XY positioning gantry in 3D printer this stepper motor based XY stage is the cost effective way in additive manufacturing. One can develops a vertical Z axis positioning stage with this XY stage which is again a low cost 3D printer.

9. References

[1]Kuang Chao Fan Yetai Fei, Xiaofen Yu, Weili Wang, Yejin Chen, Study of a noncontact type micro-CMM with arch-bridge and nanopositioning stages, Robotics and Computer-Integrated Manufacturing 23 (2007) 276–284

[2]Chien-Hung Liu, Wen-Yuh Jywe, Yeau-Ren Jeng, Tung-Hui Hsu, Yi-tsung Li, Design and control of a long-traveling nano-positioning stage, Precision Engineering 34 (2010) 497–506

[3]Jooho Hwang, Chun-Hong Park, Chan-Hong Lee, Seung-Woo Kim, Estimation and correction method for the two-dimensional position, errors of a planar XY stage based on motion error measurements, International Journal of Machine Tools & Manufacture 46 (2006) 801–810

[4]Yuki Shimizu, Yuxin Peng, Junji Kaneko, Toyohiro Azuma, So Ito, Wei Gao, Tien-Fu Lu, Design and construction

of the motion mechanism of an XYmicro-stage for precision positioning, Sensors and Actuators A 201 (2013) 395– 406.

[5]Dongwoo Kang, Kihyun Kim, Dongmin Kim, Jongyoup Shim, Dae-Gab Gweon, Jaehwa Jeong, Optimal design of high precision XY-scanner with nanometer-level resolution and millimeter-level working range, Mechatronics 19 (2009) 562–570.

[6]Rong-Fong Fung, Yi-Lung Hsu, Ming-Shyan Huang, System identification of a dual-stage *XY* precision positioning table, Precision Engineering 33 (2009) 71–80

[7]Chih-Liang Chu, Sheng-Hao Fan, A novel long-travel piezoelectric-driven linear nanopositioning stage, Precision Engineering 30 (2006) 85–95

[8]3D Printing Basics: The Free Beginner's Guide, http://3dprintingindustry.com/3dprintingbasicsfreebeginnersguide/

[9]Chandi Campbell,Cartesian, Delta, and Polar: The Most Common 3D Printers

[10]Gantry in 3D Printers, https://www.whiteclouds.com/3dpediaindex/gantry3dprinters

[11]Arduino SA. Arduíno - homepage. http://www.arduino.cc/.

[12]Easy Driver Stepper Motor Driver, https://www.sparkfun.com/products/12779

[13]Stepper Motor Quickstart Guide SparkFun, https://www.sparkfun.com/tutorials/400

[14]Adafruit Motor Shield, https://learn.adafruit.com/adafruit-motor-shield

[15]Massimo Banzi, “Getting started with Arduino”, second edition.

[16]R.S.Khurmi, J.K.Gupta, Machine Design, 14th Edition.