Current Journal of Applied Science and Technology

39(45): 84-105, 2020; Article no.CJAST.63733
ISSN: 2457-1024
(Past name: British Journal of Applied Science & Technology, Past ISSN: 2231-0843, NLM ID: 101664541)

Development of a Small Scale Material Handling Machine for Automated Storage and Retrieval System (ASRS)

Saif Ullah Iqbal[1], C. B. Yeo[1] and Umar Nirmal[1*]

[1]*Centre of Advanced Mechanical and Green Technology, Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia.*

Authors' contributions

This work was carried out in collaboration between all authors. Author SUI performed a comprehensive review on the development of small scale material handling machine for ASRS. Author CBY supervised and coordinated the work while author UN designed the contents and layout in this article. All authors read and approved the final manuscript

Article Information

DOI: 10.9734/CJAST/2020/v39i4531161
Editor(s):
(1) Dr. Elena Lanchares Sancho, University of Zaragoza, Spain.
Reviewers:
(1) Alemu Gurmessa Gindaba, Wollega University, Ethiopia.
(2) Shivam Singh, Indian Institute of Technology Delhi, India.
Complete Peer review History: http://www.sdiarticle4.com/review-history/63733

Original Research Article

Received 17 October 2020
Accepted 21 December 2020
Published 31 December 2020

ABSTRACT

The current works deals with the design and development of an Automated Storage and Retrieval System (ASRS) machine for Flexible Manufacturing System (FMS). The work involved investigating ASRS machine features and operating procedures, evaluating related hardware, software and communication modules for the machine. The work explored on the different options of hardware and software modules offered in the current market and further selected the suitable one for the ASRS machine development. Several design considerations and the limitations faced during the process of the project development and implementation are given. Arduino was used as the coding system for the Arduino UNO board while stainless steel 305 and aluminium was used as the main frame in fabricating the ASRS machine. Lastly, a final working prototype of a fully-developed ASRS machine with dimensions of: 80cm by 73cm by 94cm (length by width by height) is presented.

Keywords: Small scale; material handling; ASRS; FMS; prototype; pick and place machine; Arduino UNO; stainless steel 304; aluminium.

**Corresponding author: E-mail: nirmal@mmu.edu.my, nirmal288@zoho.com;*

1. INTRODUCTION

An ASRS involves a selection of computer-controlled structures in order to robotically place and retrieve loads from well-defined stocked places. Their fundamental preferences over usual (pickers to quantities) storeroom systems are vast space practice, decreased work charges, short recovery periods and an improvement in controlling the inventory of stocks. Automated Storage and Retrieval Systems (ASRSs) are generally utilized in stockrooms and primarily in distribution centres in most part of the world. There is a good ground for the idea that this type of automation drive to keep (or even escalate) their wide application in the likely future, along with the discussed model industry trends and certainly all the above benefits. There has been much advancement in Automated Storage and Retrieval technology in the last forty years and the ASRS number is predicted to grow rapidly in the next few decades. The progressing multiplication of item assortment perceptible in numerous businesses expands the assortment of stock-keeping-units (SKUs) to be taken care of in a normal distribution centre. For example, the plant in Dingolfing of German car manufacturer BMW gets material in excess of 13,000 compartments conveyed by around 600 providers on over 400 trucks every day. Taking care of such an enormous measure of parts by solid in-house coordination forms appears to be not really reasonable without mechanized warehousing [1-2]. An ASRS stores stock all the supplementary thickly and, in this manner, dispenses with the requirement for such energy to cool, heat, ventilator, and lighting abundance of storing space. One of the Material Handling Industry of America (MIH, 2009), the food business sets aside to 30% cooling cost in chilled distribution centres put on Automated Storage and Retrieval Systems [3-4]. With a maturing society in many created nations and an expanding authoritative weight (e.g., EU Machinery Directive, 2006/42/EC, 89/391/EEC, Occupational Safety and Health Act) ergonomic perspectives need extraordinary thought for storing. The above statistics should be a wake-up call for industries to consider ways they can make employee safety significant in the workplace. One of the solutions is to introduce automation. By automating manual repetitive tasks, workers can be reassigned to safer areas where they can exercise imagination, adaptability and decision-making skills. As noted by the World Economic Forum, robots are perfect for physically demanding or dangerous tasks for humans to carry out and, economically, they can work around the clock at a lower cost than human workers among which, the Automated Storage and Retrieval Systems appears to be favourable [5].

Generally, an Automated Storage and Retrieval System is programmed for managing pallets without the interference of a human worker, consequently, the structure is called fully automated. Both in manufacture and supply, locations of Automated Storage and Retrieval Systems are used for putting products (e.g., raw materials or partially finished products) in storage, and for retrieving those products back from storage to complete the task. In designing an Automated Storage and Retrieval System, numerous physical plan and control issues must be tended to in the correct method to completely exploit every one of its professionals. The ASRS will hone the maker's opposition as ASRS can diminish the inventories and process duration, and ideally use the capital and work [6]. Fundamentally, an ASRS system consists of storage shelves, the control system, storage and retrieval column, human user interface and deposit and withdraw bay. An item would be placed and retrieved from the storage shelves by mean of the storage and retrieval column machine where it is to retrieve or store the objects in the storage system. The automation system generally consists of a conveyor system as automation for storage and retrieval machines and frame structure as automation pathway [7]. The automation systems will in charge of vertically, inside/outside and horizontally moving the storage and retrieval mechanism that performs the storage or retrieval course. Therefore, to control the system, The Graphical User Interface is the platform for the user to command a process to either store or retrieve the selected boxes placed on the withdraw and deposit bay or sort the boxes in the shelves of the complete rack itself. The material picking module of ASRS is to meet the targeted user specific operational requirement. After the design phase, the Storage and Retrieval mechanism will be fabricated and used to conduct the storage and retrieve process in an ASRS. The material picking module of ASRS is intended to meet the focused-on consumer-explicit operational necessity. This module consists of a racks structure along with the storage and retrieval column to conduct the storage and retrieval commands by means of ASRS.

Most of the Automated Storage and Retrieval System accessible in the market is normally high-priced or large in size and it is just focusing on the large enterprise [8,9]. For the medium and small organizations, the issue of the floor plan and economic report frustrate them to execute the Automated Storage and Retrieval System into the operational organization. Mistakes may occur unintentionally loading and retrieving objects at the exclusive area and notice the things in racking can now and then be difficult. About these issues, an Automated Storage and Retrieval System is planned to suit the operational prerequisite of the little and medium firms, just as to diminish the error rates and expanded reliability quality while saving labor costs and floor space [10,11]. Therefore, the current works is aimed to design and develop a state-of-the-art automated storage and retrieval system. The machine will be linked with a developed GUI program to allow users to interact with the machine in a 'user-friendly' manner.

2. METHODOLOGY

2.1 Hardware

In this section, there are sections in assembly movements, selected components design stability and the design itself and its fabrication process, these subparts with its own specific function for serving its own operational purposes. The general assembly of these subparts will able to achieve the project objectives to perform the storage and retrieval operations precisely.

2.1.1 Structure assembly materials

The structure material for this module is chosen as Stainless Steel 305 and Aluminium 5083, while Aluminium is typically not as strong as Stainless Steel and Stainless Steel has excellent corrosion resistance for two reasons. First, Stainless Steel has chromium added which forms an invisible corrosion resistant film around the Steel. Second, it is non-porous which increases the corrosion resistance. The Stainless Steel are used specifically for load carrying segments, and the Aluminium are used just for coupling or to maintain the overall project weight as compare to Stainless Steel, Aluminium is far way lighter compared to stainless steel. The Stainless Steel 305 rectangular tube (5cm x 2.4cm) and (2.5cm x 1.2cm) are used for Shelve frame base and shelve module structure respectively and aluminium plate 5083 are used for coupling of different assemblies.

2.1.2 Information on components used

Following are the hardware components are used for the current work and their brief descriptions.

2.1.2.1 Linear screw

According to Thomson Industries [12-14], the thread of Linear Screw uses a helix angle to make the linear motion. The coefficient of friction between the nut and the screw are aligned with the performance of a lead screw depends heavily on, which relies on the material utilized for the nut and screw. For this project, we used Linear Screw for the three-dimensional shuttle in x, y and z axes so to move the assembly of Storage and Retrieval to left-right, up-down and forward-backward.

2.1.2.2 Hard chrome piston rod

The hard chromium piston rods are produced by Suzhou Weipeng Precision Machinery Co., Ltd. is in Suzhou with JIS S45C material, where the poles initially experience exactness processing and preparing, and are then put through guard surface chromium treatment, permitting a surface accuracy dimension of f7, and a surface hardness coming to HV800 least and up, which help to enhance wear opposition and help to broaden the existence cycle of the bars, hence making a good motion. The module is dual-mounted with the chrome piston rods, for frictionless slider motion of Storage and Retrieval assembly in x, y and z axes. The pair of rods helps the assembly to move precisely in a specific direction using Linear Ball Bearing.

2.1.2.3 Stainless steel support rod

The Stainless Steel rod is installed as a structural support of Z column (Crane) which is fitted vertically, on top with Aluminium plate of Motor base [15].

2.1.2.4 Linear ball bushings

Linear bushing with mechanism utilizes the rolling motion of ball features. As linear motion is achieved with a simple mechanism, the linear bushing can be used in a wide variety of uses, including carriage, food treating, and semiconductor manufacturing tools. Linear slide bushings use a round channel for the guiding axis, causing in space savings, which allows for compacted schemes. The Low Friction, channel surface is precision ground. Since the contact surface between the ball features and the raceway surface is reduced, linear slide bushing bearings provide low friction related to other linear motion tools. In this module we are using two sizes of Linear Ball Bushings, i.e.,16mm and 20mm for the for the hard chrome plated rods helping the assembly motion [16].

2.1.2.5 Mounting bolts

There are different types and sizes of Allen key and Hex Head bolts used in this project. Further, it is categorized as Button Head Cap, Socket Head Cap, Counter Sunk and Hex Head Cap. In total, the module has 43 Button head cap, 21 Counter Sunk Head, 16 Socket Head Cap and 8 Double Ended bolts of Allen Key plus 4 Hex Head Cap bolts are fitted. So as the net total 92 bolts are fixed in the whole module. Refer Appendix A for more information on the bolds used.

2.1.2.6 Levelling bottom pad

The module is fitted with four Levelling Pads while levelling feet are almost always solid metal; levelling pads usually have a rubber or urethane pad at the bottom, which will provide grip to keep the machine from moving and can help stabilize cabinetry, shelving, machinery and more. Levelling pads may be used instead of levelling feet when a floor may be scarred or marred by a steel foot.

2.1.2.7 Electrical / electronic housing

An electrical box is a cabinet for either electrical or electronic kit to mount controls, knobs, and displays and to stop electrical shockwave to kit users and keep the subjects from the surroundings; This housing is made from Acrylic Sheet 25cm by 15cm, Three parallel plates are connected through Long bolt of 12mm leaving a gap inside of 8cm and 12.5cm respectively, creating a design of open shelves The Bottom shelf is assigned for power supply and the top shelf is to mount the control Motor Drivers, Arduino UNO board and Interface Circuit between the limit Switches and Arduino UNO Board. The top shelf sheet is mounted with Safety Switch for any dangerous situation.

2.1.2.8 Drag chain

Cable carriers, correspondingly well-known as drag chains, drive chains, or cable chains depending on the producer, are guides planned to surround and guide bendable electrical cables and hydraulic or pneumatic hoses connected to moving robotic equipment. In this module we are using three sizes of Drag Chains 7 x 7 mm, 10 x 15 mm and 15 x 30 mm. For the x-axis, the cables are using 7 x 7 mm size of Drag chain with a length of 450 mm, for z-axis the cable is guided into 10x15 mm Drag Chain with a length of 730 mm whereas for y-axis the cable is hoses into 15x30 mm Drag Chain with a length of 990 mm. The y-axis is carrying all the cables leading from x and z axes where it is acting as a bottleneck to main controller housing. Refer Appendix B for more information on the drag chains used.

2.1.2.9 Coupling

It is a component used for bonding two different rods/shafts composed by their split ends in order to conduct power. Generally, it does not usually let disconnection of shafts through operation, though there are torque controlling couplings in which it slips-up or detaches when some torque perimeter is surpassed. There is only single coupling mounted on z-axis between the motor and lead screw, which was fabricated from Stainless Steel rod of diameter 25mm using a Turning and Milling Machine.

2.1.2.10 Wiring spiral duct

Wire duct application is same as drag chain where it differs only in a way that it's flexible and can be shaped in any form; generally, it is a kind of cable management product that permits the users to route bulky packs of cabling in a relaxed & organized style. The project wires are bed in with 700 mm and 400mm long size with a diameter of 15mm and 8mm respectively.

2.1.2.11 Frame structure

The Frame is designed with 3 x 3 column and rows. The rows and columns are designed in such a way to achieve the diagonal movement demonstration plus reduction half right angle triangle based to reduce cost and achieve the demonstration objectives. However, the shelf is rack railed with Stainless Steel rail size 10mm wide enough to handle heavy loaded items. The base of the frame is made of Stainless Steel rectangular Tube 503 of size 50 mm x 240 mm, the base dimensions are 600 mm x 600 mm on top of which the shelves are fabricated with Stainless Steel rectangular tube 503 of size 250 mm x 120 mm with each shelf internal dimensions of, i.e.,(L x W x H) 200 mm x 180 mm x 150 mm = 5.4 x 10^6 mm^3. Multiplying this with 8 shelves 43.2 10^6 mm^3 of Storage Capacity, whereas the last shelve at column 3 row 2 is Pick and Place shelf and could be connected to a conveyor or human manual loading. Choosing the Stainless Steel as fabrication for the frame was to make it strong enough to handle fairly heavy loads. Due to the storage and retrieval process, this material has a longer life compare to other options. Fig. 1 illustrates the CAD model of the frame structure design.

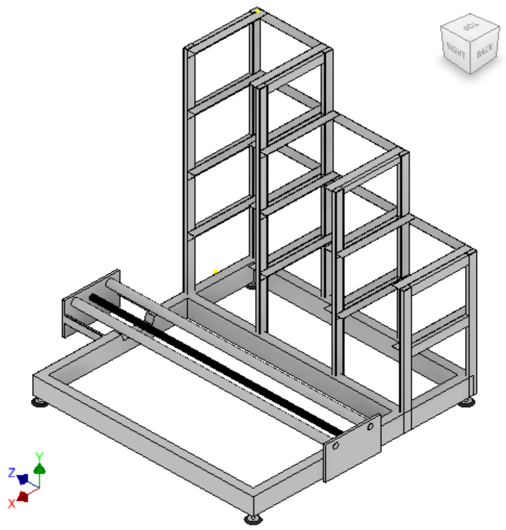


Fig. 1. Frame structure design using CAD

2.2 Assembly Motion

The Module assembly is designed in such a way so the motion of the mechanism could be smooth and frictionless. The machine has three assembly motions categorized as coordinated system x, y, and z. The x-axis coordinate assembly is responsible to store and retrieve the packages from the racks by moving back and forth; c.f. Fig. 2. The assembly maximum moving zone is 360mm. The x-axis is mounted through linear ball bushing with z-axis to lift up or down the mechanism of Storage and Retrieval.

The z-axis coordinates assembly is responsible to move the Storage and Retrieval mechanism up and down to a specific row by following the specific command from the user input; c.f. Fig. 3. The maximum travel area for this axis is 640 mm. This assembly is mounted to y-axis assembly.

The y-axis is the core assembly and it is mounted with frame base to move the other two assemblies in specified directions; the z-axis is mounted with y-axis by linear ball bushings; c.f. Fig. 4. This axis has the maximum allowed movement zone of 650mm either to move the full assemblies together towards left or right to specify the column numbers.

2.3 Design and Fabrication

In designing such a module the first thing to be taken care of is listing down the material required, cut into the exact size required and then assembling them either by mean of welding or bolt joint. Second and most important, the orientation of the tubes to design the frame in such a way to get a higher moment of inertia, as a higher moment of inertia means higher resistance to bending. Following Fig. 4, the orientation of the fabricated tube suggests the greatest moment of inertia in the *Iz-axis* compared to the other two axes. The following section will provide brief information on the design and development of the module.

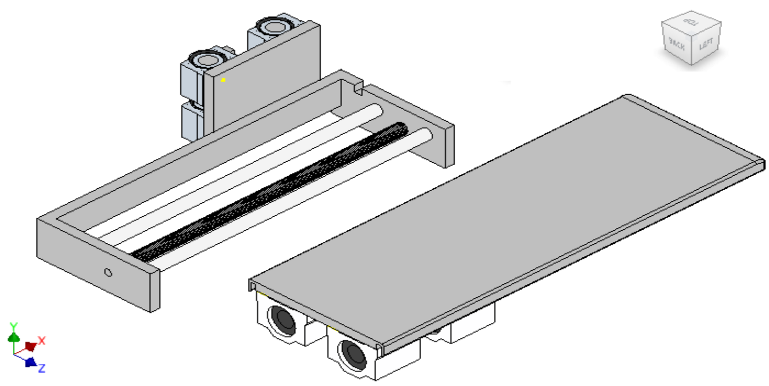


Fig. 2. Assemble motion at the X-Axis

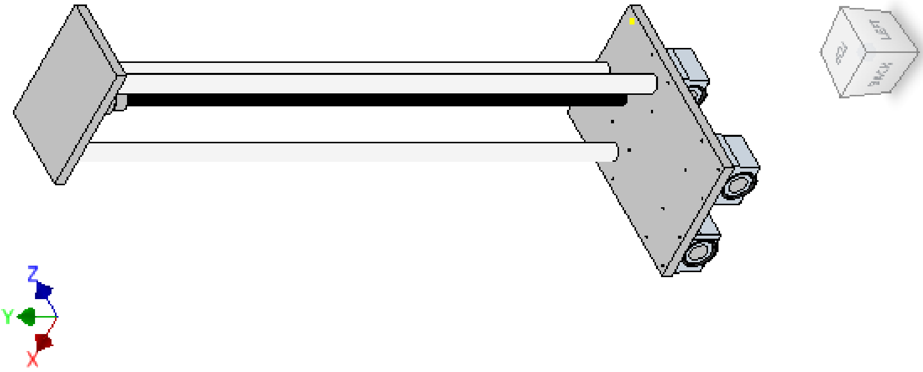


Fig. 3. Assemble motion at the Z-Axis

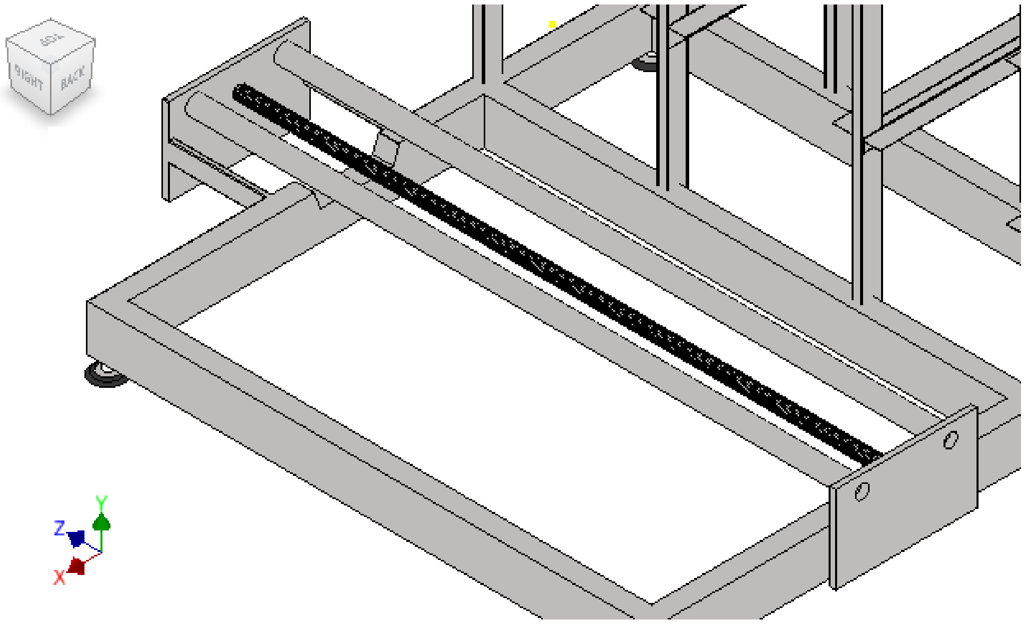
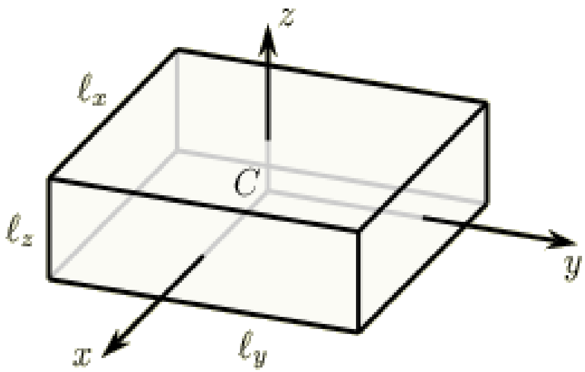


Fig. 4. Assemble motion at the Y-Axis

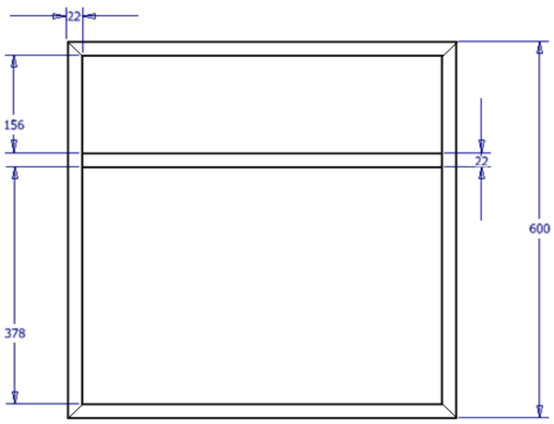


$$I_x = \frac{1}{12}m({\ell_y}^2 + {\ell_z}^2)$$
$$I_y = \frac{1}{12}m({\ell_z}^2 + {\ell_x}^2)$$
$$I_z = \frac{1}{12}m({\ell_x}^2 + {\ell_y}^2)$$

Fig. 5. Information of moment of inertia of rectangular tube

2.3.1 Frame and base design

The frame consists of base and shelves column, i.e., nine racks, the 2D dimensions of the base is shown in Fig. 6 respectively. The rectangular frame comprises a rectangular tube of size 50mm x 240mm, each side the tube was cut at 45-degree angle of length 600mm making four equal length bars and brazing them with tungsten torch wire with a cumulative length of 2400mm providing a great solidity to the whole framework.



Remark: All dimensions in mm

Fig. 6. Base dimensions

2.3.2 Rack design and 3D dimensions

On top of the base frame, the shelves are designed with a rectangular tube of dimensions 250 mm x 120 mm making of three columns, Fig. 7 shows the size of each column tube, i.e., the first column height is 600mm consuming four equal lengths of 600mm rectangular tubes, the second column consists of two rectangular tubes of height of 450mm welded with base and the third column approaches the height of 300mm fitted with two rectangular tubes attached from the base. The racks are made with a gap 150mm in height and 200mm in length. The width of each column is 180mm resulting in eight storage and retrieval for storage and one shelf for deposit and pick up bay. With shelves, the Y-axis assembly holder in welded as the core moment of other assemblies. The end plates are mounted on sides of the base frame to hold the two-piston chrome-plated rods with the linear screw, on the left plate a drive motor is connected to move the assembly towards left or right. Fig. 8 shows the base with shelves and Y-axis assembly holder. They are connected with Y, Z and X axes assembly mechanisms, following the same structure method for x assembly. For z holder, the end supporting plated is connected with Y-axis assembly and the other end in supported by an Aluminium plate with a support of Stainless Steel rod including the linear screw and two piston chrome plated rods.

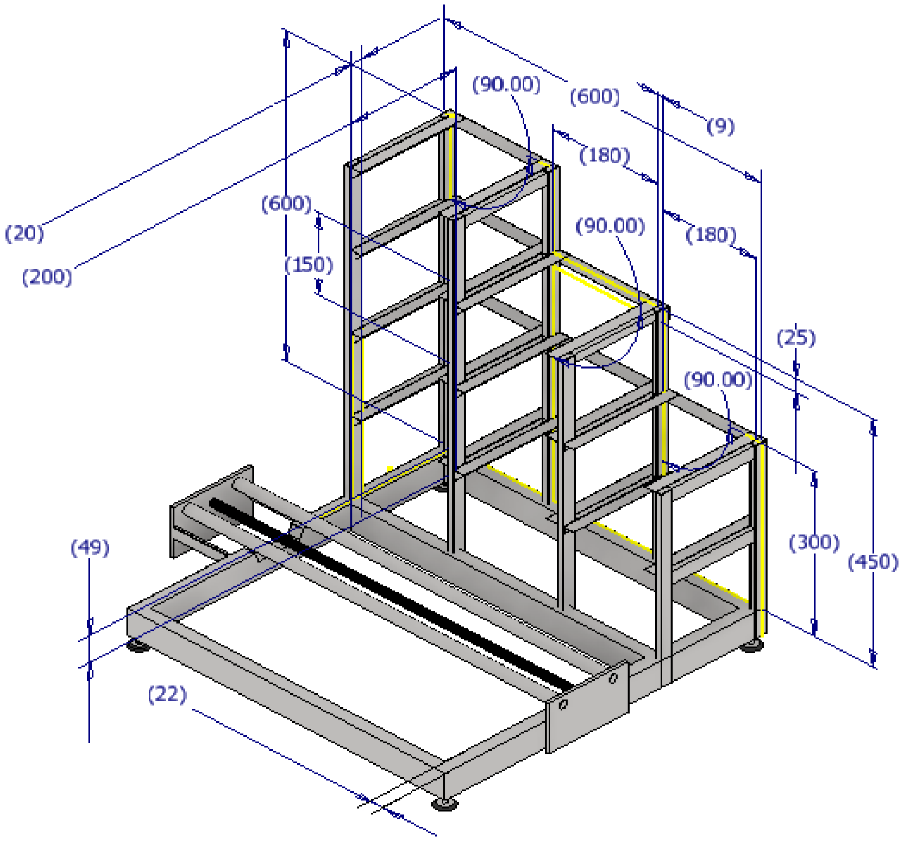
2.3.3 Control system: Electrical works

- **Drive motors:** In this module, there are three stepper motors, i.e., NEMA 23, NEMA 23 and NEMA 34, used by X, Y and Z axes respectively. The motors are installed at the end of each assembly to move the assembly in a certain direction with a specified number of steps. The Motors operates with a holding torque (2 Phases) on 1.89 Nm and 8.5 Nm for NEMA 23 and NEMA 34 respectively. Please refer to Appendix C for motor details.

- **Limit switches:** The module consists of six limit switches. Each assembly is fitted with two different switches which are Home Position switch. The Home position switch is synchronised is a way that whenever the system is reset it all the respective assembly move to the Home Positions, i.e., X-axis Home, Y axis Home, and Z-axis Home and stop there until the next command is entered by the user to perform the action. In case on any wrong movement, let's assume the assemblies move the opposite directions by not considering its Home Position, and it moves to the assemblies Safety Switch, i.e., X-axis Safety, Y axis Safety, and Z-axis Safety and by hitting it will move back to the opposite direction to reinitialize its home position and protect the assemblies from wear and tear.

- **Power supply:** This module is supplied with 12 volt and 25 amperes with the three outputs supplied to the Motor Drivers, where NEMA 23 Motor Driver consumes 3 ampere and NEMA 34 Motor Driver consumes up to 4.5 ampere.

- **Safety switch:** For the current work, safety Switches are used for emergency purposes to immediate shutdown of the system for safety purpose. The Safety Switch is connected between the AC input live cable from a power source to Linear Power Supply, connected as Normally Open (NO).



Remark: All dimensions in mm

Fig. 7. Rack with Y-axis assembly dimensions

Fig. 8. Main structure of the ASRS machine

2.3.4 Control system: Electronic works

- **Arduino UNO:** Arduino Uno is a micro controller board dependent on ATmega328P (datasheet). It has fourteen computerized in and out pins (of which six can be utilized as PWM output), six simple bases of info, a 16 MHz quartz speed of the clock, a USB association, a power holder, an ICSP header and a reset control. It contains everything expected to help the micro controller; basically, associate it to a PC with a USB link or power it with an AC to DC connector or battery to launch. You can tinker with your UNO without stressing excessively over accomplishing something incorrectly, most dire outcome imaginable you can swap the chip for a couple of dollars and begin once again. "Uno" signifies one in Italian and was picked to stamp the launch of Arduino Software (IDE) version 1.0. The version and the board itself of Arduino Software were the reference renditions of Arduino, presently developed to fresher discharges. The Uno board is the first in a progression of USB on the board, and the reference display for the Arduino index stage; for a broad rundown of current, past or obsolete boards see the Arduino list of boards. Fig. 9 demonstrates the Arduino UNO board [17], and the technical data of the board is presented in Table 1. Table 2 illustrates the pin configurations for the Arduino UNO board used.

Fig. 9. Arduino UNO board used

Table 1. Technical specifications of the Arduino UNO board

μ-controller	ATmega328P
Operational Voltage	5V
Input Voltage	7-12V
Limit of Input Voltage	6-20V
Digital Input Output Pins	14 (of which 6 provide PWM output)
PWM Digital Input Output Pins	6
Analog Input Pins	6
DC Current per Input Output Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 Kilo Byte
RAM	2 Kilo Byte
ROM	1 Kilo Byte
Crystal Clock Speed	16 MHz
Built-in LED Pin	13
Dimension and Weight of Board	L = 68.6 mm, Width = 53.4 mm, Weight = 25g

Table 2. Output pins configuration of Arduino UNO board

Pin	Descriptions
5V	EN +, DIR + and PUL +
GND	EN –, DIR – and PUL –
2	Z Step Pin (PUL +)
3	Z Direct Pin (DIR +)
4	Limit Switch
5	Limit Switch
6	Limit Switch
7	Limit Switch
8	Limit Switch
9	Limit Switch
10	Y Step Pin (PUL +)
11	Y Direct Pin (DIR +)
12	X Step Pin (PUL +)
13	X Direct Pin (DIR +)
A0 (14)	Enable Pin EN+ of X, Y, and Z

- **Motor drivers TB6600:** The TB6600 Arduino Stepper Motor Driver is a simple to-utilize proficient stepper motor driver, which could control a double phase stepping motor. It is perfect with Arduino and different microcontrollers that can output a 5 Volts digital pulse signal. The TB6600 driver has a wide range control input, from 9 to 42 Voltage DC power supply. Also, it can supply an output of 4 Ampere peak current, which is sufficient for most of the stepping motors. The driver underpins speed and course control. You can set its small-scale step and an output current with 6 DIP switch. There are 7 small scales steps (1, 2/A, 2/B, 4, 8, 16, 32) and 8 sorts of current control (0.5, 1, 1.5, 2, 2.5, 2.8, 3, 3.5) Ampere on the whole. Each flag terminal receives fast pick coupler disconnection, upgrading its anti-high-frequency impedance [18,19]. More information on the motor driver is provided in Appendix D respectively. Fig. 10 presents an example of a connection diagram of Motor Driver with Arduino, Stepper Motor and Power Supply.

- **Interface circuit b/w limit switches and arduino UNO:** The Limit switch pins are connected each to ground to Arduino Pin through a resistor of 1 K ohm resistor, the 5 volts is supplied to limit switch through a resistor. The circuit consists of 6 resistors, 2 input pins of GND and 5 volt VCC, 12 terminal blocks from 6 six switches and 6 terminal blocks from Arduino Pins. The schematic diagram of the interface circuit is provided in Fig. 11 while the entire control unit of the module is shown in Fig. 12 respectively.

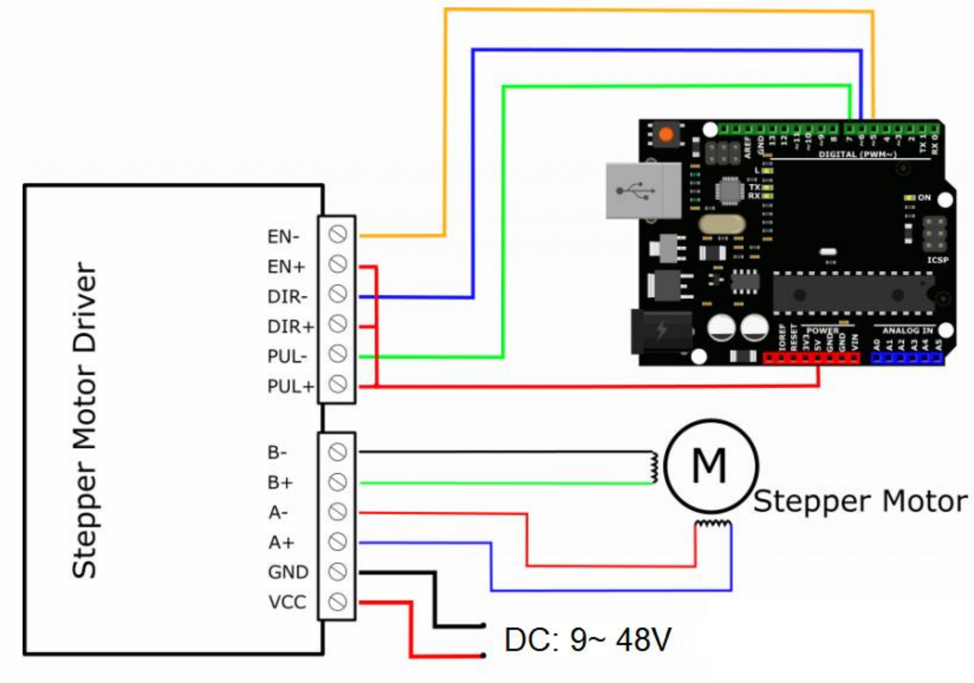


Fig. 10. Connection diagram

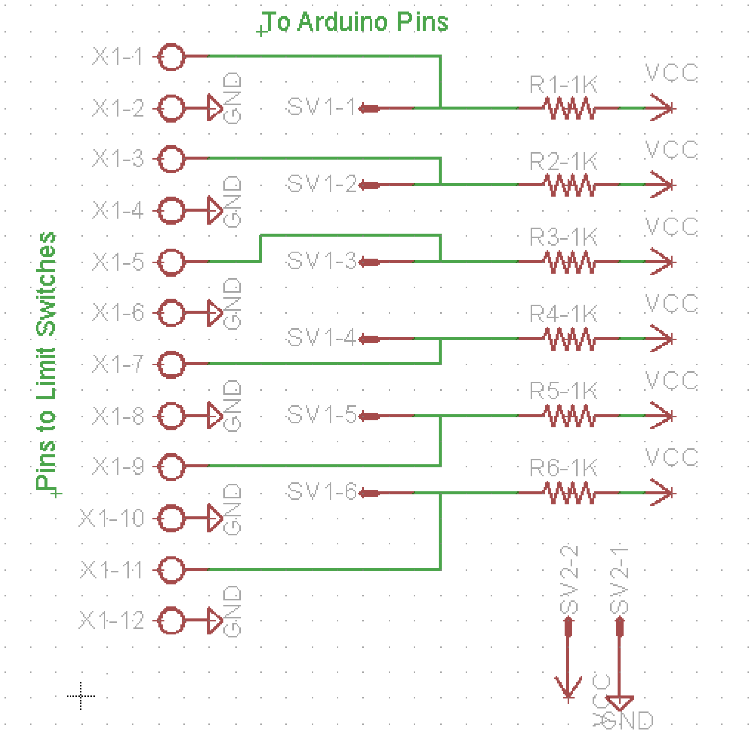


Fig. 11. Schematic diagram of the interface circuit

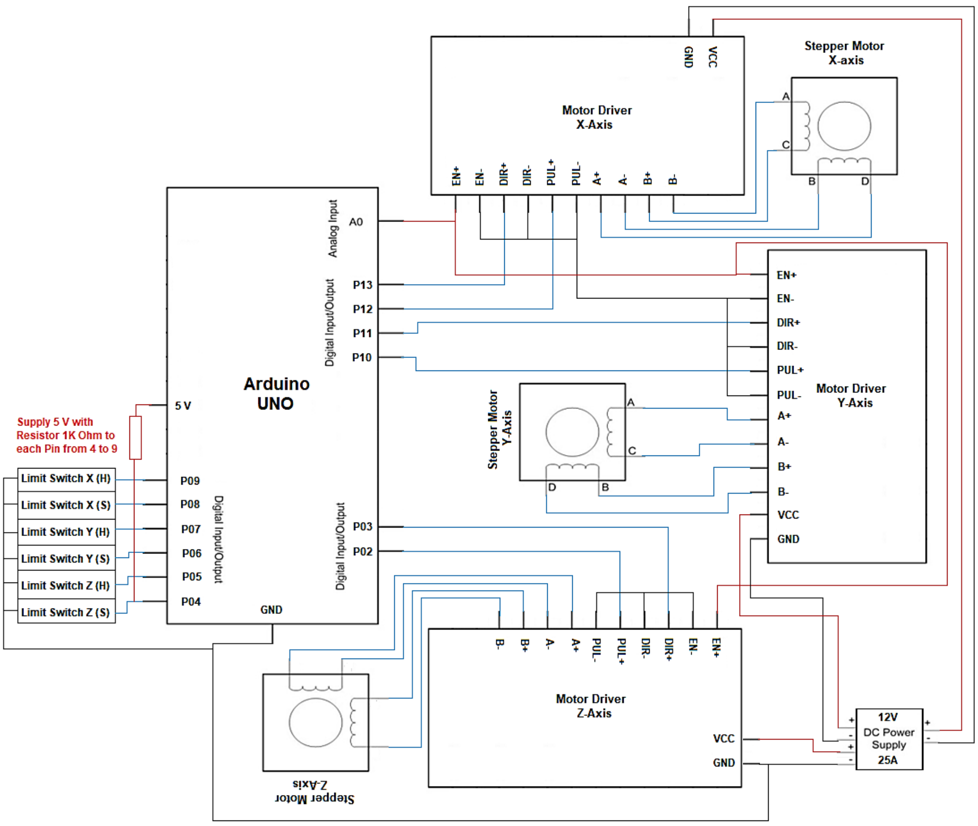


Fig. 12. Control circuit of the module

2.3.5 Software

In the programming procedure, Arduino is chosen to be utilized for incorporating and leading the logical control activity of the coding. The general software IDE is used to write the coding for controlling the module, the language used is basic C#. A simple method to comprehend the Cartesian facilitate framework in connection to machine is utilizing the Right Hand Rule. Hold your hand out palm up with your thumb and forefinger pointed outwards, and your centre finger pointed upwards. Spot your turn before assembly, lined up with the assembly's segment and you'll see the axes line up clearly. The movement of system along the coordinate system is always based on how assembly moves, not the shelves itself. For instance, increasing the Y coordinate value would transport the assembly towards left, but when looked at from the viewpoint of the shelves, it's moving right along the assembly. Decreasing the Z-axis coordinate would move the Storage and Retrieval mechanism up, increasing it would move it down reference with the shelves. Every three dimensional system has its own in-house origin place called Assembly Home. When the module initially boots-up, it generally has no clue where it is in physical space and requires calibration to get it to its Home position. When this process happens, all three axes of the machine move toward their extreme mechanical limit. Once a limit is touched, a signal gets directed to a controller which registers the Home position for that specific assembly axis. By completing this for other assembly's axes, the machine is considered as "Homed". The coordinates system of the module shelves is defined in Fig. 13. Each cell has its coordinate of (z, y) to ease the localization process during storing and retrieving activities. The system structures included are the location of the initial position of the linear motion of assembly, location of stacking and receiving are taken from shelf 8 which in future possibly be linked to a conveyor machinery.

Shelf 1 (3, 0)		
Shelf 2 (2, 0)	**Shelf 5** (2, 1)	
Shelf 3 (1, 0)	**Shelf 6** (1, 1)	**Shelf 8** (1, 2)
Shelf 4 (0, 0)	**Shelf 7** (0, 1)	**Shelf 9** (0, 2)
Crane Assembly		

Fig. 13. Coordinates system of shelfs

2.3.6 Flow chart and coding description

The program has following functions which interconnect with each other to perform the required tasks. They are:

//Assignment of pin numbers to direction and pulse of X-Y-Z axes:

```
const int stepPinX = 12;
const int dirPinX  = 13;
const int stepPinY = 10;
const int dirPinY  = 11;
const int stepPinZ = 2;
const int dirPinZ  = 3;
const int enPin    = A0;
```

//Setting up the Global variable to which the pulse pin number is assigned:

```
int stepPin;
```

//Assignment of pin numbers to switches:

```
const int Z_home   = 9;
const int Y_home   = 8;
const int X_home   = 7;
const int X_safety = 6;
```

```
const int Y_safety = 5;
const int Z_safety = 4;
```

//Assignment of fixed number of steps representing:

```
const int dist_col = 15500; //distance between columns
const int dist_row = 24000; //distance between rows
const int dist_x   = 18000; //distance for X-axis to move
const int offset   = 3000;  //offset value for retrieve and store operation
const int offset_z = 3800;  //offset value for Z-axis initial position
```

//Integer values to represent coordinate positions for Z and Y axes:

```
int coordinate_Z;
int coordinate_Y;
```

//Integer values to store current coordinate positions:

```
int temp_coordinate_Z;
int temp_coordinate_Y;
```

//Integer values representing the difference between coordinates -> how far one shelf from another in ZY plane:

```
long int result_Y;
long int result_Z;
```

//Integers for calculating the number of steps axes should move from one shelf to another:

```
long int steps_Y;
long int steps_Z;
```

//Integers for counting steps of Z and Y axes in while loop:

```
unsigned long int b;
unsigned long int c;
```

//Integers that representing shelf to go to and operation (0-retrive;1-store) to perform:

```
int shelfi;
int operationi;
```

//Integers for for/while loops & if/else statements:

```
int i=0,y=0,d=1,k=0;
```

Void setup, Void loop, Void move-self, Void retrieve, Void store, Void movement, Void movement_YZ, Void operation_movement_YZ, String getValue and Void initialize.

Please refer Appendix E for the functions descriptions of the program.

2.3.7 Graphical user interface

For this module the Graphical User Interface (GUI) was created using a Microsoft Visual Studio. The application serially communicates with the module through an Arduino Board via 9600 baud rate. Fig. 14 shows the interface of the application. The Application has three main panels, i.e., Port Properties, Manual Operation, Reset, Storage and Retrieval. In Port Properties the COM port is designed as drop select function button, where it detects the connected port to the PC, the Baud rate is assigned in the code as 9600 which has to be selected after pressing the Open Port button and thus the module is successfully connected with the PC. The Manual

Operation includes shelf selection from 1 to 9, the operation is either retrieval or Storage by entering “0” and “1” and the reset is to initialize the whole system to its “Home Position”. The Move button is to order the machine to perform the given tasks; the Store is the option to save the current data in a text file to keep the machine and app up to date to not perform the operation twice for the same shelf. In the Manual Operation Panel, it has nested panel of Reset, where the two buttons Reset has the value “1” and Don’t has the value “0”, by pressing each of the buttons it writes the value in the Reset Box mentioned in the Manual Operations. The Store Panel has all the shelves from 1 to 9 among which one is Pick and Place shelf for the item to be stored. It does the same as Reset Panel by writing the Values in Manual Operation panel under the Operation Box to write “1” and Shelf Box “1” as storage, After the user press the Shelf 1, it will write the value as described “1” and changing its colour to Red from Green, where Red indicates the shelf is stored or full and Green means its vacant, which simultaneously change the colour of Shelf 1 button in Retrieve Panel, to Green. So once the buttons are Red in colour it acts as disabled. The same operation goes to all other shelves, resulting in good communication way between the User and the Machine.

2.3.8 Program function

The GUI constructed by using the Microsoft Visual Studio controls the sorting machine by sending the appropriate commands through serial port communication. It is constructed in a friendly way with buttons and labels to ease the process for the users. On the left side, there is the serial port communication. That opens and closes the serial port connected with the circuit board. If the user wishes to control the machine from the computer, the serial port needs to be open and the baud rate, which represents the speed of transforming information needs to be set as well. The GUI includes two main parts where one for manually inserting the shelf number and operation, and the other one by clicking on the respective buttons, which makes it more convenient for users. The second part of the GUI offers control for both storage and retrieval. The user may easily press a button for a respective shelf under the “store” section to store a container in the shelf or press a button for a respective shelf under the “retrieve” section to retrieve a container from that shelf. Once the store button is pressed, it will turn red to notify the user there is already a container stored in that shelf. Similarly, once the item is retrieved, the button will turn green to notify the user that the shelf is empty and available for storage. All the commands and operations are assembled in the form: shelf number: operation: reset status; where the shelf number represents the respective shelf number to store or retrieve and it is given values from 1 to 8. The operation represents whether to store or retrieve, and is represented by a 1 or 0, for storage and retrieval, respectively. The final subcommand is the reset status, where if set to 0, the machine will reset. So if the user wanted to store on shelf 1, by pressing the “shelf 1” button

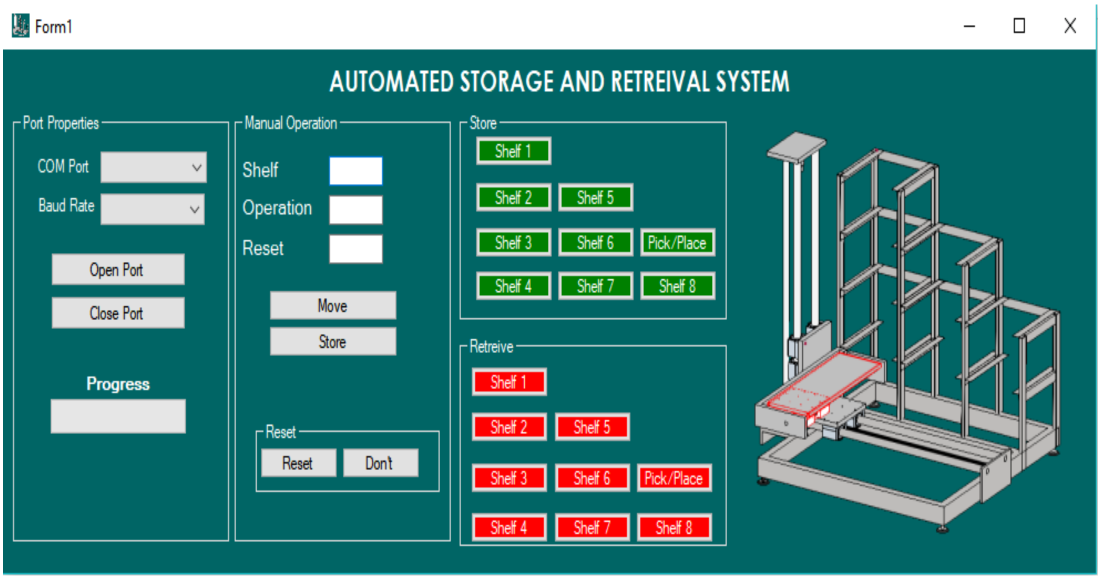


Fig. 14. Proposed GUI for the current work

in the storage section of the GUI, the command would automatically be constructed as: 8:1:0 Alternatively, the user may manually enter the command under the "manual" section in the respective fields for shelf number, operation and reset. The command is sent using serial WriteLine () which uses the Serial Port Library by C# to send the command to the Arduino board. The button may also be easily switched colour using the button. Color = "red" and button. Color = "green". The string is also constructed using a string concatenation of the shelf number, operation and reset.

3. OUTCOME OF THE FABRICATED ASRS MACHINE

3.1 Experimental Operation Result of Storage and Retrieval Machine

The experimental operation result is tested on the efficiency of the ASRS machine to perform a given task in the aspect of time taken and the maximum handling load. In the Storage and Retrieval process, there are three processes in total which extend platforms, lifting or depositing of platforms and retraction of platforms. The summation of the time taken in these three processes will be the total time taken for Storage and Retrieval machine performs a Storage and Retrieval task. Both the extension and retraction of the platform takes about 25 seconds each with lifting and depositing taking 2 seconds and retraction of item loads takes same as 25 seconds. Therefore, the total time taken is 52 seconds for completing the Storage and Retrieval task. This speed can be improved by supplying higher voltage so the power in total would be much enough to boost the process in efficiency and time. Overall the movement achieved is smooth just the linear screw which has to be adjusted to eliminate the squeaky noise during machine operation. The trial activity result is tested on the effectiveness of the Storage and Retrieval machine to carry out the Storage and Retrieval activity in the part of the time taken to carry out the assignment and the most max load of Storage and Retrieval machine ready to deal with. In the Storage and Retrieval process, there are three processes in total which extend item, lifting or depositing of item and retraction of item. The summation of the time taken in these three procedures will be the complete time taken for Storage and Retrieval machine play out a Storage and Retrieval job. Both the expansion and withdrawal of the stage takes 25 seconds each with lifting and depositing taking 2 seconds and retraction of item loads takes same as 25 seconds. Then the all-out time taken is 52 seconds for finishing the Storage and Retrieval undertaking. This speed can be improved by supplying higher voltage so the power in total would be much enough to boost the process in efficiency and time. Overall the movement achieved is smooth just the linear screw which has to be adjusted to eliminate the squeaky noise during machine operation.

3.2 ASRS Machine View

The different views of the fabricated machine is presented in Fig. 15 respectively. The final dimensions of the machine in terms of its length by width by height is: 80cm by 73cm by 94cm respectively. Fig. 16 depicts the actual image of the developed ASRS machine showing its important segments.

3.3 Weight Lifting Calculation Using

3.3.1 NEMA 34

Maximum weight a 8.5 Nm Stepper Motor torque can lift by the NEMA 34 is computed using Equation 1 where:

Torque (T) = 8.5Nm; Radius = 0.007m; Diameter = 0.014m

$$T = \frac{1}{(2*pi)*(F+W*g)} \qquad \text{Equation 1}$$

$$T = \frac{1}{(2*pi)*0.00005}$$

$$T = 1*9.81*0.007$$

$$8.5 = 1*9.81*0.007$$

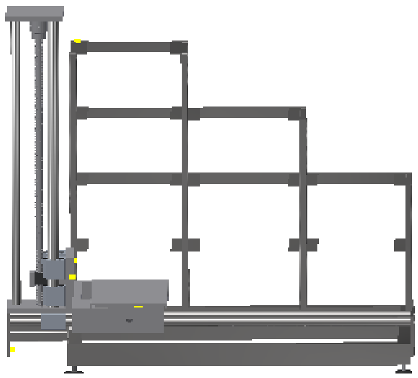
$$\frac{8.5}{9.81}*0.007 = 123.78\,kg$$

3.3.2 NEMA 23

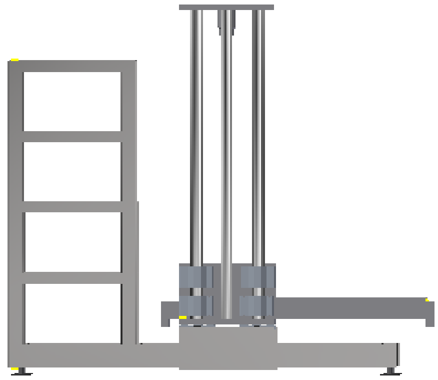
Maximum weight a 1.89 Nm Stepper Motor torque can lift NEMA 23 is computed using Equation 1 where:

Torque = 1.89Nm; Radius = 0.007m; Diameter = 0.014m

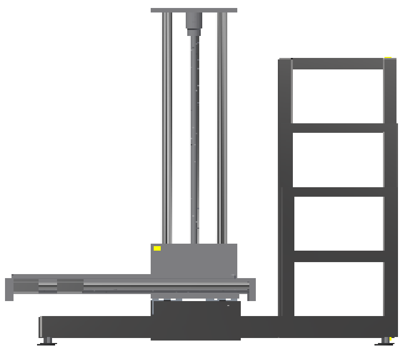
$$\frac{1.89}{9.81}*0.007 = 27.66\,kg$$



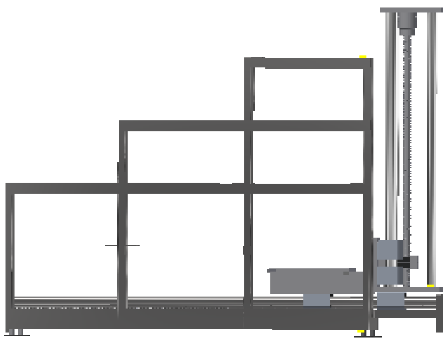
a) Front view



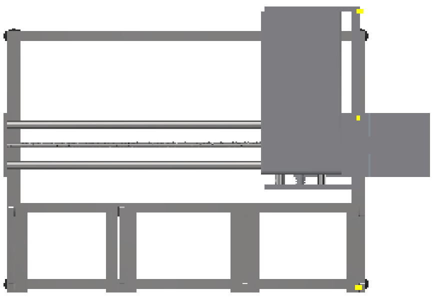
b) Left view



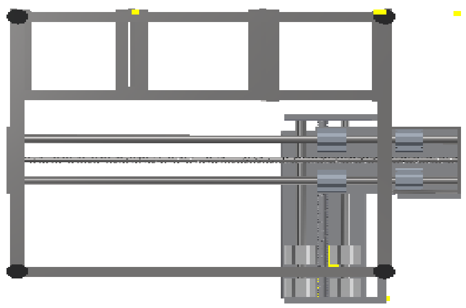
c) Right View



d) Back View



e) Top view



f) Bottom view

Fig. 15. Different views of the fabricated ASRS machine

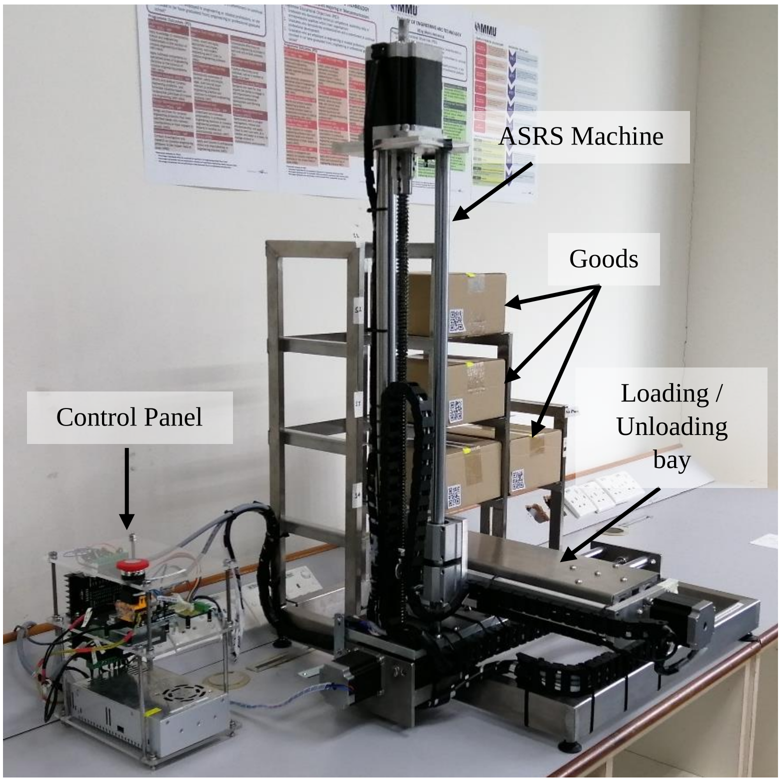


Fig. 16. Actual image of the fabricated ASRS machine showing its important segments

3.4. Proposed Future Works

The following may be carried out in the future to further enhance the developed ASRS machine. They are but not limited to:

- Stability of Storage & retrieval assembly can be further improved by adding an additional column.
- Wifi based Arduino to enhance user-friendliness. Personnel can operate the machine using their cellular devices or tablets.
- Robotic Arm can be installed based on machine vision to carry out sorting and synchronizing tasks.
- Power specifications: Currently the module is rated as P = I x E (12 x 4.5 = 54 Watt). To further improve it, we can supply up to 42 Volt to the current system which can derive up to 189 Watt.
- Speed can further be improved by replacing linear-screw mechanism by gear-chain or linear-gear mechanism.
- Following the same design, ASRS can be attached to the end of an industrial conveyor. Incoming packages from the conveyor will be placed in the 'General Bay'. Later, that package can be moved to its respective bay. Vice versa, outgoing packages can also be placed on the conveyor using the same mechanism.

4. CONCLUSIONS

The flat bat of the 'x' assembly permits the Storage and Retrieval machine to manage the load without moving in the whole Storage and Retrieval mechanism assembly underneath the load. This mechanism simplifies the necessities of the storage racks to be altered for having additional room under the loads in which spares the quantity of storage unit in vertical height. Also, the little estimated Storage and Retrieval assembly permits the storage room ready to be considerably smaller which could spare the floor space by trading off the space of the storage unit. In programming, the whole assembly is ready to perceive the present status of the machine and can perform a task intelligently by reinitializing to read the correct amount of movement.

ASRS solutions inventory within a fully enclosed system; providing safe, secure and controlled inventory management. For an instance, the system can be configured to permit only authorized personnel access to store inventory through a personal login password and the GUI created. This allows missing or misplaced goods to be tracked back to a specific individual. This enhanced level of accountability and security eliminates inventory shrink and its negative impact on the bottom line.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Yong Yik Seng and Dr. Lim Wee Kuan for their valuable comments and efforts contributed to the realization of this research.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Boysen N, Emde S, Hoeck M, Kauderer M. A survey on part logistics in the automotive industry. European Journal of Operational Research. 2015;242:107-120.
   Bezel YA, White JA. Travel-time models for automated storage/retrieval systems. 11E Transactions. 1984;16:329-338.
2. Wauters T, Villa F, Christiaens J, Alvarez-Valdes R, Vanden Berghe G. A decomposition approach to dual shuttle automated storage and retrieval systems. Comput. Ind. Eng. 2016;101:325–337.
3. Chu WS et al. From design for manufacturing (DFM) to manufacturing for design (MFD) via hybrid manufacturing and smart factory: A review and perspective of paradigm shift. Int. J. Precis. Eng. Manuf. Technol. 2016;3(2):209–222.
4. Gharehgozli AH, Vernooij FG, Zaerpour N. A simulation study of the performance of twin automated stacking cranes at a seaport container terminal. Eur. J. Oper. Res. 2017;261(1):108–128.
5. Material handling industry of America. Automated storage systems make a play for sustainability. AS/RS industry group quarterly report fall 2009. Material Handling Industry of America; 2009.
6. Herrmann C, Schmidt C, Kurle D, Blume S, Thiede S. Sustainability in manufacturing and factories of the future. Int. J. Precis. Eng. Manuf. Technol. 2014;1(4):283–292.
7. Boysen N, Emde S, Hoeck M, Kauderer M. Part logistics in the automotive industry: Decision problems, literature review and research agenda. Eur. J. Oper. Res. 2015;242(1):107–120.
8. Karimi Eskandary P, Angeles J. The virtual screw: Concept, design and applications. Mech. Mach. Theory. 2018;128:349–358.
9. Grosse EH, Glock CH, Jaber MY, Neumann WP. Incorporating human factors in order to pick planning models: Framework and research opportunities. International Journal of Production Research. 2015;53:695-717.
10. Chu WS et al. From design for manufacturing (DFM) to manufacturing for design (MFD) via hybrid manufacturing and smart factory: A review and perspective of paradigm shift. Int. J. Precis. Eng. Manuf. Technol. 2016;3(2):209–222.
11. Herrmann C, Schmidt C, Kurle D, Blume S, Thiede S. Sustainability in manufacturing and factories of the future. Int. J. Precis. Eng. Manuf. Technol. 2014;1(4):283–292.
12. Karimi Eskandary P, Angeles J. The virtual screw: Concept, design and applications. Mech. Mach. Theory. 2018;128:349–358.
13. McGuire AD et al. Analysing the effect of screw configuration using a stochastic twin-screw granulation model. Chem. Eng. Sci. 2019;203:358–379.
14. Tandler R, Bohn N, Gabbert U, Woschke E. Experimental investigations of the internal friction in automotive bush chain drive systems. Tribol. Int. 2019;140:105871.
15. McGuire AD et al. Analysing the effect of screw configuration using a stochastic twin-screw granulation model. Chem. Eng. Sci. 2019;203:358–379.
16. Tandler R, Bohn N, Gabbert U, Woschke E. Experimental investigations of the internal friction in automotive bush chain drive systems. Tribol. Int. 2019;140:105871.
17. Candelas FA et al. Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. IFAC-PapersOnLine. 2015;48(29): 105–110.
18. Nickols F, Lin YJ. Chapter 4 - Theory III: The stepper motor and Its control. Nickols F, Y-J B, T-C P, Lin R. Eds. Butterworth-Heinemann. 2019;47–86.
19. Jo H, Do Noh S, Cho Y. An agile operations management system for green factory. Int. J. Precis. Eng. Manuf. Technol. 2014;12:131–143.

APPENDIX A: Information on the types of bolds used

Allen key

							
Button Head Cap		**Counter Suck Head**		**Socket Head Cap**		**Double Ended**	
Description	**Pcs**	**Description**	**Pcs**	**Description**	**Pcs**	**Description**	**Pcs**
M4 x 20mm	2	M4 x 15mm	3	M4 x 15mm	4	M2.5 x 5mm	8
M4 x 30mm	32	M4 x 20mm	16	M4 x 20mm	12		
M4 x 35mm	4	M4 x 25mm	2				
M3 x 10mm	5						

Hex Head

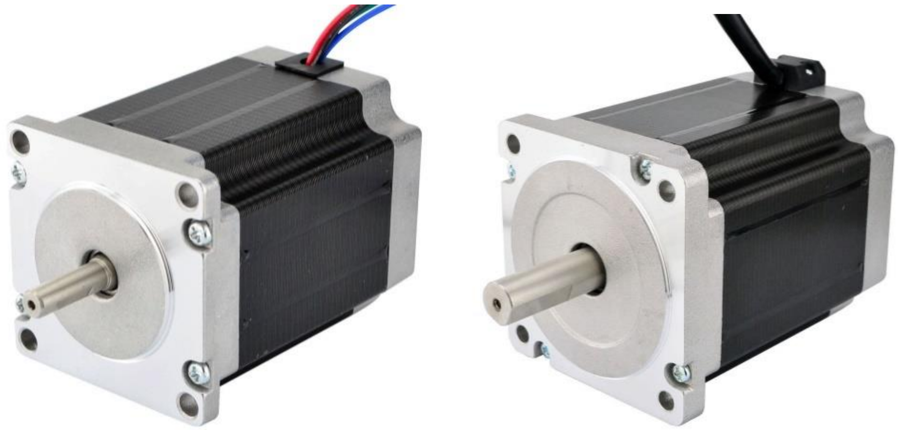
Hex Head Cap

Description	Pcs
M14 x 15mm	4

APPENDIX B: Information on the types of drag chains used

		
	Sizes	
7 x 7 x 450 mm	**10 x 15 x 730 mm**	**15 x 30 x 990 mm**

APPENDIX C: Information on the Stepper Motor - NEMA 23 on left and NEMA 34 on right



APPENDIX D: Information on the Features and Specification of Motor Driver TB6600

Features	Specification
Support eight classes of current controller	Input Current: 0 to 5 Ampere
Support seven classes of micro-steps changeable	Output Current: 0.5 to 4.5 Ampere
The interface adopts high-speed	Control Signal: 3.3 to 48 Voltage
Automatic semi-flow to decrease heat	Maximum Power: 160 Watt
Outsized space heat sink	Micro-Step: 1, 2-A, 2-B, 4, 8, 16, 32
Anti-high-frequency interference ability	Temperature: - 10 to 45 Celsius
Input anti-inverse protection	Humidity: No Condensation
Excess heat, Excess current and short circuit protection	Weight: 0.2 kg
	Dimension: 97 x 70 x 36.5 mm

APPENDIX E: Functions Descriptions of the program

Void setup

In the Setup function, the direct, pulse (step) and enable pins are initializes as High or Low However the pins are categorized as input and output for their respective operation to perform. Limit Switches are initialized as Input and Motors are initialized as Output. After Initialization the stored coordinated are initialized as coordinates Z=0, Y=0. Lastly, the Serial port is open as 9600 for data transmission.

Void loop

The loop function coordinates the movement of assembly as designated positions, by calling the movement function in switch cases. Initially, the initialisation occurs to maintain the Home Position for the assemblies. The Serial String reads the operation by receiving the value from the user as "0" means retrieve, "1" means store and "2" means reset the X, Y and Z axes assemblies. There are nine Switch Cases, Case =1 is assigned for Shelf 1 where the coordinates are z = 3. y = 0. Case 2 is assigned for Shelf 2 and the coordinates are z = 2, y = 0. In Case = 3, coordinates are z = 1, y = 0 and is assigned for Shelf 4. Case = 4, coordinates are z = 0, y = 0 assigning to Shelf 4. The Shelf 5 have coordinates z = 2, y = 1 in Case 5. Case 6, represent Shelf 6 with the coordinates z = 1, y = 1. Case 7 of switch function is assigned to Shelf 7, with the coordinates z = 0, y = 1. Shelf no 8 which is Pick and Place bay, have the coordinates z = 1, y = 2 under Case 8. For Lastly Case 9 is assigned for Shelf no 9 with the coordinates z = 0, y = 2. Else it will print "Wrong entry" if the numbers are other than 0, 1 and 2 for retrieving, storing and resetting the X, Y, and Z assembly. The assembly moves accordingly by specified steps in movement function given the command by the user.

Void move-self

This function performs the movement operation of Z and Y axes. When the Operation reads “0” which means retrieval, the assigned steps are performed with a reference coordinates as when z and y are at (0, 0) respectively. Distance Column and Distance Row are specified initially which are called by dist_col, dist_row and offset. During the movement forward and backward whenever the result_Y or result_Z values are in negative integer, it multiplies them with positive one value to get the right calculated movement. After reading the operation of the Digital Write pin assign the Direct (step) pin as either “High or Low”. In the function, the Z and Y axes are coded as to move simultaneously to directed shelf. Lastly, if operation entered is “0”, means retrieve which will call the retrieve function to perform the task. Else if operation is “1”, means store by calling store function.

Void retrieve

In the retrieve function, the responsible assemblies are Z and X axes, as to retrieve the Z-axis have to be 1500 steps lower than the rack levelling position. After which the X-axis direct pin is assigned as “High”, in case if it hit the switch either Safety or Home and the system reads the X_safety or X_Home as “High” it will call the movement function to Stop. However the Z-axis follows the same criteria for the movement where the steps in Z-axis are mentioned by offset which is 3000 Steps, For the retrieval purpose we need to move the Assembly exactly by 3000 steps as it goes below the rack with -1500 and then moving up by 3000 steps lifting the item after which Digital writes the X step pin as “Low” moving the bay backward and the offset is divided by 2 to make the level same as the shelf rail by calling the movement function. The retrieve operations perform successfully.

Void store

The Store function act the same as retrieve function also responsible for Z and X axes assemblies, to store, the Z-axis must be 1500 steps higher than the rack rail position. After which the X-axis direct pin is assigned as “High”, in case if it hit the switch either Safety or Home and the system reads the X_safety or X_Home as “High” it will call the movement function and Stop. However the Z-axis follows the same criteria for the movement where the steps in Z-axis are mentioned by offset which is 3000 Steps, For the storage purpose we must move the Assembly exactly by 3000 steps as it goes above the rack with +1500 and then moving down by 3000 steps putting the item after which Digital writes the X step pin as “Low” moving the bay backward and the offset is divided by 2 to make the level same as the shelf rail by calling the movement function. The storage operations perform successfully.

Void movement

In this function, the Digital Write stepPin, “HIGH” with delay in 2 microseconds and for the Digital Write stepPin, “LOW” the delay of microseconds 740. These are the delays that need to be assigned for stepper motors between the steps movement as how precisely it will move.

Void movement_YZ

This function has the exact same behaviour as the movement function above, adding which just for the simultaneous movement of Z and Y axes assemblies the step pin Z and step Pin Y are instructed together with same microsecond delays for High as 2 and Low as 740. It is called when there is simultaneous movement of Z and Y axes assemblies’ movement.

Void operation_movement_YZ

Function operation movement YZ was created for Y and Z axes to move simultaneously. The function is called inside a while loop which condition will be only satisfied when both Y and Z step counters will be equal to the previously calculated number of steps. Inside function, there are few conditions that must be satisfied on order the set the pulse signal high or low for both Y and Z. The first condition is checking whether the Limit Switch is pressed or not, although the number of steps is predefined and the machine should not exceed its operation limit, this condition is implemented for the safety purposes. Second condition is implemented to verify that either direction pin Y or Z is activated and lastly the third condition is the most important one, as it checks whether the steps counter is equal to the calculated steps for the axis to move to another shelf. Let's say we must move 24 thousand steps, this means that the pulse signal will be sent to the controlling driver 24 thousand times, providing that the previous two conditions are satisfied. At last, how both axes move at the same time, this is by sending pulse high to both Y and Z and only after short delay low. Otherwise, it would be the same as i initialize function where firstly we moved X-axis and only after that Y and Z.

Void initialize

In initialization function, we created with a dummy value, i.e., d= 0. It is an infinite loop with no body. The purpose would be to make sure the program doesn't end, yet it does nothing. It is also known as an idle loop. While the System boot up the serial print will print "Doing the Initialization Sequence" Writing the digital stepPin X as Low which will move backward till it hit the switch. By infinite steps where it reads stepPin=stepPin, When d=1, digital read home pin in High, if it is true it will keep moving by calling movement function, if the value is False it reads d=2 which is true value for home position, and the mechanism stops at home position by Printing the "Alignment by x-axis done". And the While loop for Z-axis and Y-axis follows the same as X-axis by writing the digital stepPin Z and digital stepPin Y as Low. If it reads d=3, it repeats until the assemblies are in Home position when the d=2 after hitting the Limit Switches respectively so the offset is True and it stops if False it repeat.

© 2020 Iqbal et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/63733