



Master Thesis

Control system for three-axis CNC milling machine with automatic tool change

Study programme: N0715A270018 Machines and Equipment Design

Author: **Pradeep Appunu**

Thesis Supervisors: Ing. Petr Keller, Ph.D.
Department of Manufacturing Systems and
Automation

Liberec 2023



Master Thesis Assignment Form

Control system for three-axis CNC milling machine with automatic tool change

Name and surname: **Pradeep Appunu**
Identification number: S21000270
Study programme: N0715A270018 Machines and Equipment Design
Assigning department: Department of Manufacturing Systems and Automation
Academic year: 2022/2023

Rules for Elaboration:

The aim of the work is to design a control system for a three-axis milling machine with automatic tool change. For the control of movements in individual axes, it is possible to use an existing solution (e.g. GRBL for Arduino), but it is necessary to add at least tool change control sequences. It is also advisable that it should be possible to connect measuring probes to the designed control system in the future to determine the reference point on the workpiece and the length of the tools, etc.

Recommended processing methods:

1. Perform a research of existing available open-platform control system solutions for CNC milling machines. Furthermore, familiarize yourself with the design of a small CNC milling machine on KSA and its control requirements.
2. Based on the research, select a suitable basis for the control system and specify the necessary modifications to control the movements of the milling machine, including automatic tool change.
3. Select a suitable electronic platform and implement the control system. Implement the necessary software and verify the correctness and functionality of the solution.
4. Evaluate your work.
5. Prepare materials for creating a publication on given topic in a technical journal or at a conference.

Scope of Graphic Work: according to need
Scope of Report: approx. 55 pages
Thesis Form: printed/electronic
Thesis Language: english

List of Specialised Literature:

- [1] MAREK, J. *Design of CNC machine tools*. Translated by I. KLEPSOVA, I. SIMAN, Z. ZIDLICKA. Prague: MM publishing, 2015. MM special. ISBN 9788026086376.
- [2] SUNGEUN, J. *GitHub – gnea/grbl-Mega: An open source, embedded, high performance G-code-parser and CNC milling controller written in optimized C that will run on an Arduino Mega2560*. [online soft.]. 2018. [cit. 2022/09/29]. Available form: <https://github.com/gnea/grbl-Mega>
- [3] KIEF, Hans B. *CNC for industry: by Hans B. Keif*. Cincinnati: Hanser Gardner Publications, 1999. ISBN 1-56990-296-8.
- [4] KUMAR, Kaushik, Chikesh RANJAN a J. P. DAVIM. *CNC programming for machining*. Cham, Switzerland: Springer, [2020]. Materials forming, machining and tribology. ISBN 978-3-030-41281-4.

Thesis Supervisors: Ing. Petr Keller, Ph.D.
Department of Manufacturing Systems and
Automation

Date of Thesis Assignment: November 20, 2022

Date of Thesis Submission: May 20, 2024

doc. Ing. Jaromír Moravec, Ph.D.
Dean

L.S.

Ing. Petr Zelený, Ph.D.
Head of Department

Declaration

I hereby certify, I, myself, have written my master thesis as an original and primary work using the literature listed below and consulting it with my thesis supervisor and my thesis counsellor.

I acknowledge that my master thesis is fully governed by Act No. 121/2000 Coll., the Copyright Act, in particular Article 60 – School Work.

I acknowledge that the Technical University of Liberec does not infringe my copyrights by using my master thesis for internal purposes of the Technical University of Liberec.

I am aware of my obligation to inform the Technical University of Liberec on having used or granted license to use the results of my master thesis; in such a case the Technical University of Liberec may require reimbursement of the costs incurred for creating the result up to their actual amount.

At the same time, I honestly declare that the text of the printed version of my master thesis is identical with the text of the electronic version uploaded into the IS/STAG.

I acknowledge that the Technical University of Liberec will make my master thesis public in accordance with paragraph 47b of Act No. 111/1998 Coll., on Higher Education Institutions and on Amendment to Other Acts (the Higher Education Act), as amended.

I am aware of the consequences which may under the Higher Education Act result from a breach of this declaration.

August 17, 2023

Pradeep Appunu

ACKNOWLEDGEMENT

I wish to express my heartfelt thanks to my university for helping me get this fantastic opportunity to advance my engineering skills and appreciate everyone whose assistance was valuable in completing my diploma thesis. I sincerely appreciate every expert advice, which was key to my study success.

First and foremost, I would like to express my heartfelt gratitude to my diploma thesis supervisor **Ing. Petr Keller, Ph.D.**, - Department of Manufacturing System and Automation, for his invaluable guidance, unwavering support, and continuous encouragement throughout the diploma thesis. His expertise, insights, and helpful feedback have been instrumental in shaping the direction of this work

I would like to extend my sincere gratitude to **Ing. Petr Zeleny, Ph.D.**, Head of the Department, Department of Manufacturing System and Automation for his commitment to excellence in education and for providing a stimulating academic environment that fostered my intellectual growth.

Finally, My gratitude extends to my family and friends, whose unwavering belief in me and constant encouragement have been a source of motivation and strength. Their willingness to listen, discuss ideas, and offer perspectives has been truly invaluable.

Lastly, I would like to acknowledge all those whose names might not be mentioned here but have nonetheless played a crucial role in shaping my academic journey.

ABSTRAKT

Tématem této práce je návrh řídicího systému pro tříosou frézku s automatickou výměnou nástrojů. Cílem je zaměřit se na výběr vhodného firmwaru a elektronické platformy a na vývoj a testování řídicího systému pro tento stroj. Bylo třeba pochopit princip fungování CNC řídicí jednotky v součinnosti s funkcemi tříosé frézky a vypracovat návrh automatické výměny nástrojů. Dále je provedena optimalizace a testování vyvinutého algoritmu řídicího systému stroje.

Klíčová slova

Tříosá frézka, CNC řídicí systém, OSEC, Repetier Firmware, Arduino, C++, automatická výměna nástrojů, seřizování nástrojů.

ABSTRACT

The topic presented in this thesis is a Control system for a three-axis milling machine with an automatic tool changer. The goal is to focus on the development and testing of a control system for a three-axis milling machine with a selection of suitable firmware and electronic platform. Understanding the working principle of the CNC controller and the functioning of a three-axis milling machine, developing a design of automatic tool changer. Furthermore, improvising the control system by optimising the machine control system by testing the developed algorithm.

Keywords

Three-axis milling machine, CNC control system, OSEC, Repetier Firmware, Arduino, C++, Automatic Tool Changer, Tool setter.

Table of Contents

1	Introduction	12
2	Aim and Objectives of Thesis	13
3	Literature Review	14
3.1	Industrial Three-Axis CNC Machine.....	14
3.2	The Architecture of the CNC System	14
3.2.1	Open Architecture System	14
3.2.2	Specification of Open Architecture	15
3.3	Various Types of Architecture.....	15
3.3.1	OSACA Architecture	15
3.3.2	OMAC Architecture.....	16
3.3.3	HOAM-CNC Architecture.....	17
3.3.4	OSEC Architecture	18
3.4	Three-Axis Milling Machine	20
3.4.1	Review of Existing CNC Controller	20
3.4.2	Firmware	24
4	System Design Methodology.....	27
4.1	Hardware Configuration	27
4.1.1	Arduino Mega 2560	27
4.1.2	RAMP 1.4 Shield	29
4.1.3	Bipolar Stepper Motor	29
4.1.4	DRV8825 Stepper Motor Drivers.....	31
4.1.5	DC Spindle Motor.....	32
4.1.6	OLED Display	32
4.1.7	Micro SD Card Socket.....	34
4.1.8	Power Supply	35
4.1.9	Mechanical Base Frame.....	35
4.1.10	Automatic Tool Changer (ATC).....	36
4.1.11	Tool Setting.....	39
5	Experimental Work.....	42
5.1	Architecture of Hardware Connection	42
5.1.1	Hardware Connection of Arduino Mega 2560 with PC Screen.....	42
5.1.2	Hardware Connection of Arduino Mega 2560 with LCD Screen.....	44
5.1.3	Conclusion of Comparison	46
5.2	Three-Axis Milling Machine	46
5.2.1	Machine Configuration	46
5.2.2	Three-Axis milling machine setup.....	47

FACULTY OF MECHANICAL ENGINEERING TUL

5.3	Testing of Tool changer mechanism.....	50
5.3.1	Overview of Tool Changing Mechanism.....	50
5.4	Tool Setter.....	60
5.4.1	Hardware Selection.....	60
5.4.2	Measurement of Tool Length.....	61
5.4.3	Measurement of Radius/ Diameter	64
6	Conclusion.....	66
7	Reference.....	67
	APPENDIX.....	69

List of Figure

Figure 1: OSACA Architecture [4]..... 16

Figure 2: OMAC [5] 17

Figure 3: HOAM-CNC Architecture [3]..... 18

Figure 4: OSEC Architecture [3] 19

Figure 5: Block Diagram of Linux CNC controller 22

Figure 6: Arduino Mega 2560 [17] 28

Figure 7: RAMPS 1.4 Module [18] 29

Figure 8: Stepper motor SM 15DD..... 30

Figure 9: 3D model of Stepper motor with lead screw for axes [Source: Own]..... 30

Figure 10: 3D model of a stepper motor for ATC [Source: Own]..... 31

Figure 11: DRV 8825 Driver [19] 31

Figure 12: SSH 1106 OLED Display Front View [21]..... 33

Figure 13: SSH 1106 OLED Display Bottom View[21] 33

Figure 14: MicroSD Card Module [22] 35

Figure 15: Virtual 3D image of projected Three-axis machine [Source: Own]..... 36

Figure 16: Design of Tool Magazine [Source: own] 38

Figure 17: Assembly of Automatic Tool Changer Magazine [Source: Own] 39

Figure 18: Industrial Laser Tool Setter 41

Figure 19: Schematic representation of hardware connection with PC screen [Source: Own] 43

Figure 20: Schematic representation of hardware connection with LCD screen [Image edited from [25]]..... 44

Figure 21: a) 3D model of a three-axis milling machine b) Real machine Image [Source: Own] 47

Figure 22: Window of Arduino IDE[Source: Own] 48

Figure 23: Flow chart for initial tool pick, with empty spindle [Source: Own]..... 51

Figure 24: Flow chart for new tool change, with a tool in spindle [Source: Own]..... 55

Figure 25: Flow chart of tool change with no tool in spindle [Source: Own] 58

Figure 26: Optical End Switch Module [26]..... 61

Figure 27: Tool setter Geometric parameter for length measurement [Source: Own]..... 62

Figure 28: Flow chart of tool setting of tool length [Source: Own] 63

Figure 29: Tool Setter Geometric parameter for Radius/ Diameter of tool [Source: Own].... 64

Figure 30: Flow chart for Radius/Diameter measurement in tool setter [Source: Own] 65

List of Tables

Table 1: Comparison between GRBL, Marlin and Repetier Firmware [Source:]26
Table 2: Technical Specification of Arduino Mega 2560[17]28
Table 3: Technical Specification of DRV 8825 Driver [19].....32
Table 4: Technical Specification of SSH1106 OLED Display34
Table 5: Technical Specification of Micro SD Card Module [22]35
Table 6: Comparison of PC screen and LCD screen configuration for Arduino45
Table 7: Pinout configuration of RAMPS 1.4 for Axes.....49
Table 8: ATC String name with axis coordinates49

List of Abbreviations

NC- Numerical Control

CNC- Computer numerical control

CAD- Computer-Aided Design

CAM- Computer-Aided Manufacturing

OSACA- Open System Architecture for Controls within Automation

OMAC- Open Module Architecture Controllers

HOAM-CNC – Hierarchical Open Architecture Multiprocessor CNC

OSEC- Open System Environment for Controller

OEM- Original Equipment Manufacturers

EMC2- Enhanced Machine Controller

I/O - Input and output

KB - Kilobyte

MHz - Mega Hertz

RAMPS- RepRap Arduino Mega Pololu Shield

OEM – Original Equipment Manufacturer

I2C - Inter-Integrated Circuit

SPI - Serial Peripheral Interface

RAMP- RepRap Arduino Mega Pololu

OLED- Organic light-emitting diode

PC- Personal Computer

ATC- Automatic Tool Changer

RF- Radio Frequency

IDE- Integrated Developing Environment

1 Introduction

In this thesis, the "Development of a Control System for Three-Axis Milling Machine with Automatic Tool Changer" is discussed. For the past few decades, the advancement in the manufacturing sector has led to the development of CNC controllers. The evolution of the NC machine controller has paved the way for flexibility in programming and controlling the machines. CNC Milling machines, revered for their versatility and capacity to shape an array of materials, have become indispensable tools across industries ranging from aerospace to consumer electronics. At present the traditional CNC milling machine has a complex structure, closed control system with the inability to add extra features and further it restricts the development of many features in CNC.

To overcome this problem addressed above, it urges to design and develop a low-cost milling machine with a simple structure at an affordable price, a milling machine with automatic tool changers controlled by an electronic microcontroller platform. The development of low-cost CNC machines has been a trend to overcome the initial cost of investment and the possibility of adding new features to the machine.

In this thesis, a control system is developed for the three-axis milling machine with an automatic tool changer. Initially, studies on the various type of architecture systems used in CNC machine controllers is discussed in the following chapters. Throughout this thesis, we will navigate the theoretical underpinnings of machine control. By using an extensive understanding of these core ideas behind a three-axis milling machine, we will embark on the practical journey of designing and constructing a control system. The development process will encompass the selection and integration of appropriate electronic control units, stepper motors, and various electronic components, all while ensuring seamless communication between the Arduino microcontroller and the milling machine's hardware components.

The software architecture that supports the control system is deeply integrated into this work. Through the skillful development of code, we will create a responsive and intuitive interface that empowers users to wield unparalleled command over the milling process.

The goal of this thesis is to visualize, develop, construct, and optimize a control system for a three-axis milling machine with an automatic tool changer that uses the power of Arduino to coordinate the motions and operations of a three-axis milling machine. The study and development of the combination will be done in-depth.

2 Aim and Objectives of Thesis

The aim of the thesis is to develop a control system for a three-axis milling machine with an automatic tool changer, by programming an algorithm for the Arduino to perform the movement of the axis and an automated tool-changing sequence. Furthermore, to evaluate the effectiveness of the functioning of the machine.

Objectives of the thesis

The following objectives were addressed in the project:

1. To conduct an extensive literature review on the existing control systems for a three-axis milling machine has to be done for performing the task.
2. Based upon the study, to select an optimal solution for the task of creating the control system for a three-axis milling machine.
3. To select a suitable electronic platform as a processor for a three-axis milling machine.
4. To develop an algorithm for the Arduino to perform the movement of the axis and an automated tool-changing sequence for a three-axis milling machine.
5. To evaluate the functionality of the algorithm for the desired movement of the axis and automated tool-changing sequence.

3 Literature Review

In this chapter, we have discussed the various type of architecture for CNC control systems that currently exist in the market and the type of CNC control architecture varied based on their nature.

3.1 Industrial Three-Axis CNC Machine

CNC utilizes a dedicated microcomputer to carry out all numerical control functions. The capabilities of machine programming and control have increased significantly, holding with the advancement of NC machine technology. [1]

Manufacturers and machinists in the manufacturing and industrial sector employ computer numerically controlled machining as a high-precision manufacturing technology. A machine's axis movement and function are managed by CAD/CAM software code.

Three-axis CNC mills are machines that move a tool or workpiece in space using three linear axes. The X, Y, and Z axes are the usual names for the three axes. The tool or workpiece is manipulated horizontally along the X-axis, vertically along the Y-axis, and toward or away from the spindle along the Z-axis [2].

3.2 The Architecture of the CNC System

The advancement of digital circuit technology has led to the development of the control system for the new production system. However, most of the control systems developed are by correspondence with the machine tool manufacturer. It is difficult to make any modifications or adjustments to their components. The hardware and software limitations that the machine designers should operate with include things like the inability to add new functions and the lack of interaction between the system's components. This is caused by incompatible hardware and software that prevents alterations to the source code [3].

3.2.1 Open Architecture System

A new generation of numerical controllers emerged as a result of the implementation of the open architectural principle. This enabled the basic IBM - PC microcomputer to use normal CNC technology. As a result, the CNC's positioning capabilities and the microcomputer's programming simplicity, system configuration options, network connectivity, etc. are merged. A flexible software and hardware architecture that permits changes to the

device's fundamental configuration and all levels of control software has been discussed by several academics [3].

3.2.2 Specification of Open Architecture

Any engineer or technician should be able to develop hardware and software for the Open Architecture Controller and incorporate it with other controllers [3].

The open controller for manufacturing equipment should enable the incorporation of independent application program modules, control algorithms, sensors and computer hardware developed by different producers.

An Open System enables programming across multiple platforms that communicate with other application systems.

Some other specifications of an open architecture system by (Miles,1998) (Oshiro,1998):

- **Interaction:** communication of standards
- **Interoperability:** Same component function by different manufactures
- **Portability:** Easy with application software transferred from one environment to another
- **Scalability:** System ability to increase or decrease according to the demand

3.3 Various Types of Architecture

Architecture refers to the control system used to operate and control the CNC machine. There are different types of architecture on which the control system is developed on. Here are a few common types of architecture

- Open System Architecture for Automatic System
- Open Modular Architecture Controllers
- Open System Architecture Hierarchical Multi-processor for CNC machines
- Open system Environment for controller

3.3.1 OSACA Architecture

A Control Open System Architecture for Automatic Systems is the OSACA (1998) architecture. The OSACA project's primary objectives were the definition of independent hardware and modularity, i.e., the ability to work in modules to add or remove numeric control, robot control, Programmable Logical Controllers (PLCs), cell control, etc.

The OSACA design allows the assembly of machine tool controls via a user interface, eliminating the need to review the entire software. To accomplish this goal, you must first understand the concept of a platform. [3]

A platform is made up of hardware and software groups (operating system, communication system) that provide a consistent service for functional unit (FU) control. The application programming interface (API) with the FU is built around a specific task. [4]

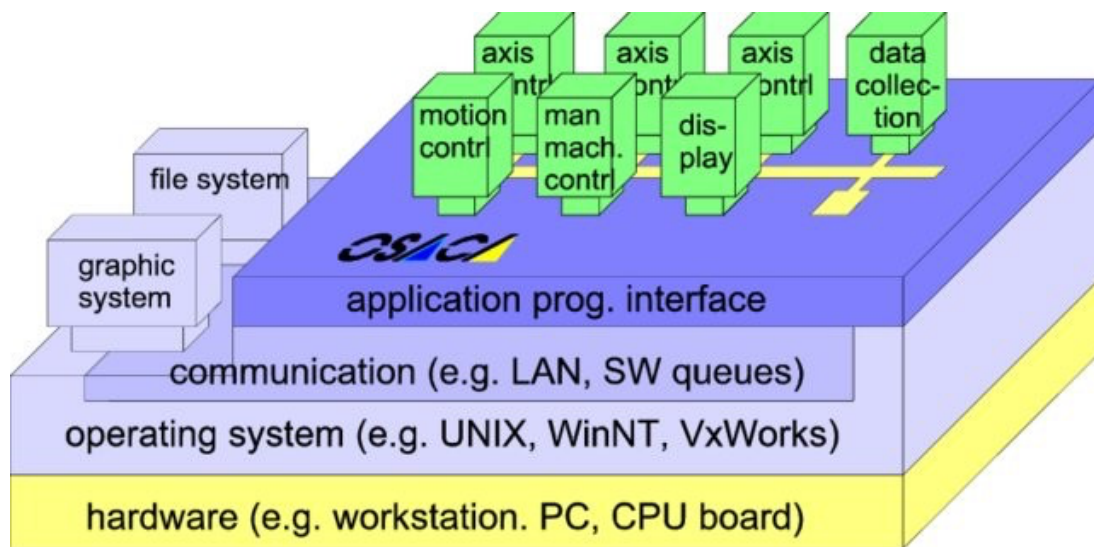


Figure 1: OSACA Architecture [4]

3.3.2 OMAC Architecture

The OMAC architecture got its start in December 1994 when Chrysler, Ford, and General Motors published "Requirements of Open, Modulate Architecture Controllers for Applications in the Automotive Industry."

The OMAC organization is comprised of users of open architecture systems, and their goal is to collaborate to bring about several advantages including:

- To develop an open architecture controller position based on the knowledge of software users and hardware providers.
- To utilize APIs to speed the adoption of open control inside industries.
- To encourage controller manufacturers to use open control.
- To establish social solutions for the creation, marketing, and application of open architecture controller technology.[3] [5]

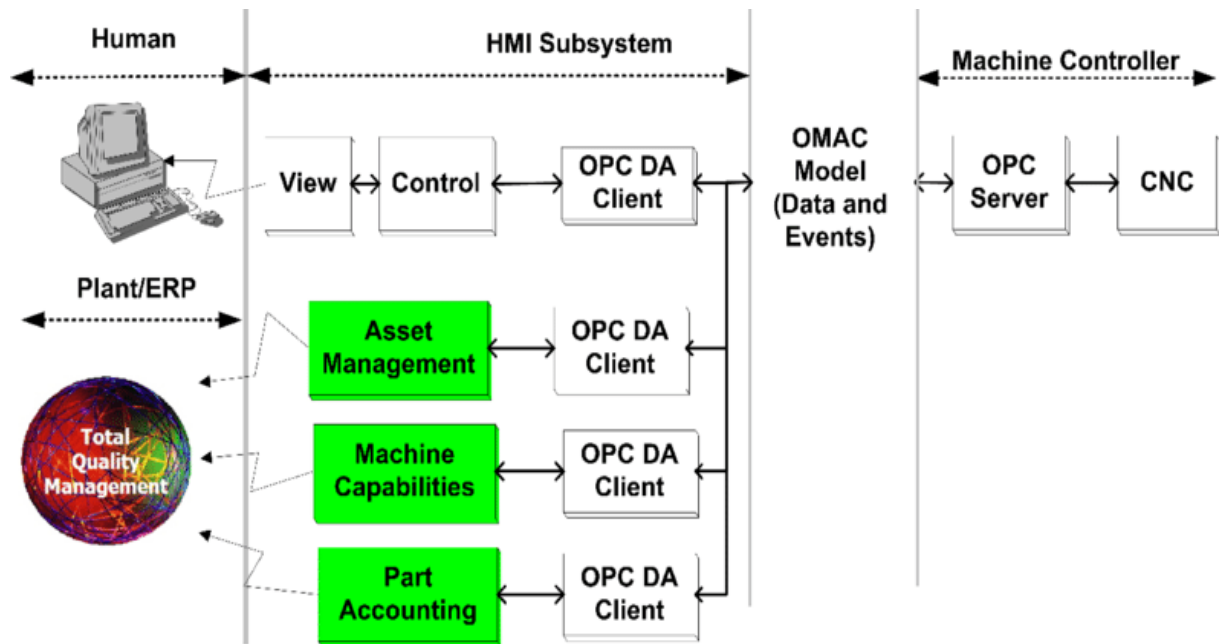


Figure 2: OMAC [5]

3.3.3 HOAM-CNC Architecture

The Open System Architecture Hierarchical Multi-processor for CNC machines, or HOAM-CNC architecture was proposed as an open platform for implementing machine algorithms. [6] HOAM-CNC specifically targets the hardware of the machine and gives the benefits of having two buses: a CNC control bus and another bus for adding new components. [3]

In Figure 3, Two different buses are indicated by the hardware architecture. The operating system is located on the main bus, which also handles data processing, HMI, and adaptive control as well as direct continuous monitoring. The servo drives and axes' real-time position and speed controls are controlled by a secondary CNC bus. The addition of two distinct control buses enables the system to expand modularly. To increase the system's operational capabilities, additional processing modules can be added to the main bus and axis control can be added to the CNC bus. [7]

From Figure 3, the modular structure of this architecture's design determines how efficiently it works. Tasks for controlling toolpaths and monitoring and controlling machine processes are divided into two categories in this design.

The system master control's CNC interface module communicates with and schedules several functional jobs that make up the toolpath control tasks.

A specially created real-time micro-kernel that manages scheduling and orchestration tasks is located within the system master control.

The machine process control tasks are integrated into the CNC master control and communicate with the system master control by sending data and requests via the CNC interface module.

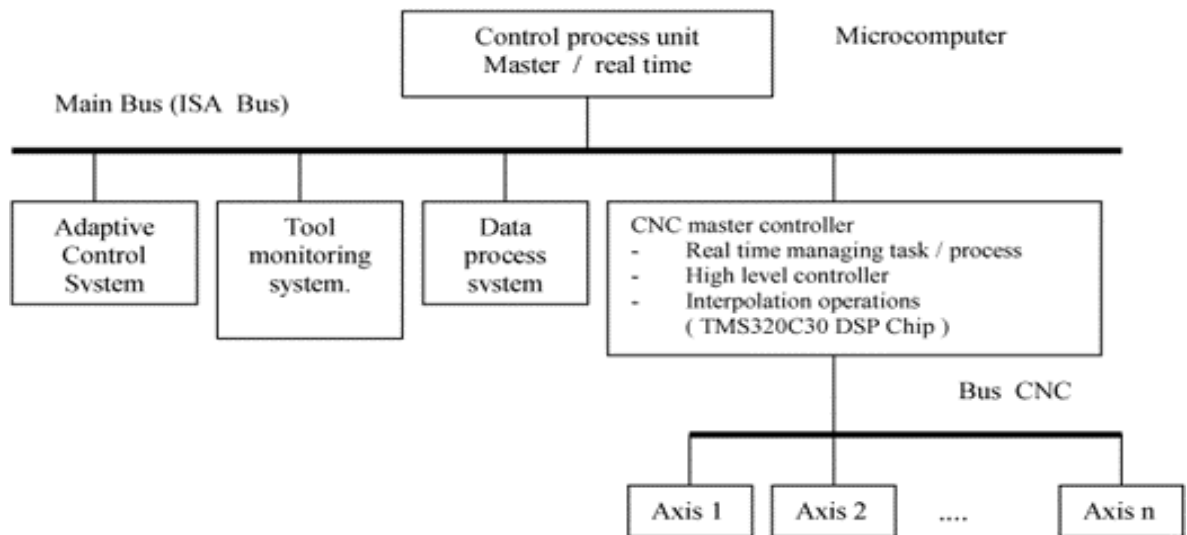


Figure 3: HOAM-CNC Architecture [3]

3.3.4 OSEC Architecture

The open system Environment for controller concept was begun in the year of 1994 by many Japanese companies for Industrial automation in many sectors. They developed the Open System Environment for Controllers (OSEC). Controlling industrial equipment while enhancing performance and simplifying maintenance were the key objectives. The architecture was designed to offer a common industrial platform on which common industrial controllers might be based to the eventual users, machine makers, control and software vendors. Then, these controllers would be altered to accommodate the various end-users, fostering the technological and commercial advancement of industrial machines [7].

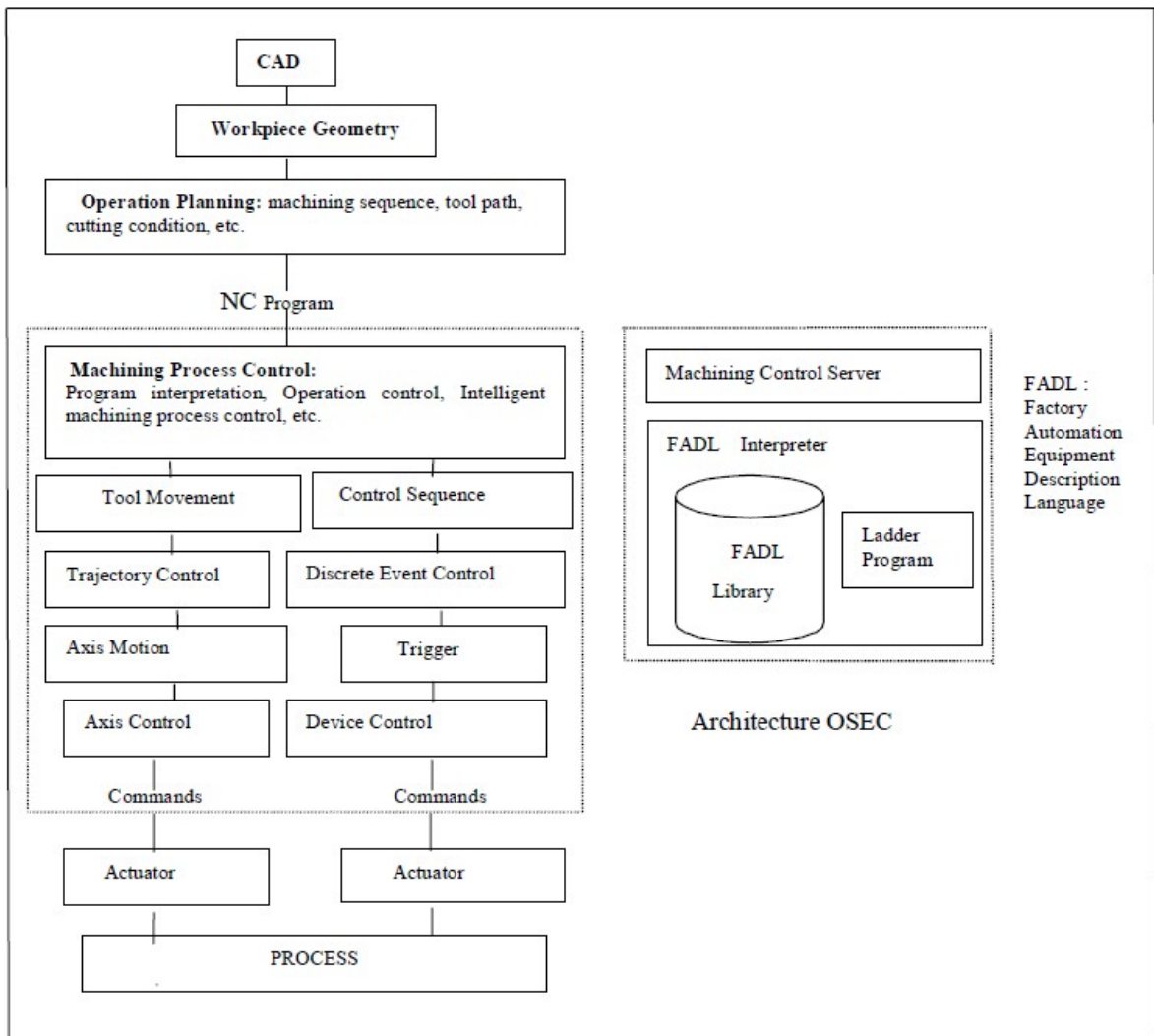


Figure 4: OSEC Architecture [3]

This architecture is focused on five levels of [7]:

1. Workstation planning
2. Workstation managing
3. Plan interpretation
4. Trajectory generation/discrete input and output
5. Servo controls

3.4 Three-Axis Milling Machine

Recently, the industrial sector has downsized more quickly and adopted cutting-edge technology to create items that are more compact, portable, and thin. Furthermore, there are now microcontrollers available at NANO and micro sizes in the engineering and technological world. This also applies to Three - axis CNC machines, which currently come in a variety of sizes on the open market. Regardless of size, every sort of machine serves a certain purpose.

Industrial technology has changed many aspects of daily living over many years. For the creation of such a (CNC) machine on smaller, thinner, lighter weighted, and affordable costs, numerous studies have been conducted.

When compared to conventional machines, the advanced ability and precision of CNC tool control have significantly influenced the development of function components, frame bodies, stepper motors, and control circuits. Building and testing a cheap table CNC milling machine Due to a low voltage supply of the main cutting forces, it is possible to employ smaller-sized tools to machine materials like wood, aluminium, and plastic by using a low-cost milling cutter for the main spindle. [8]

3.4.1 Review of Existing CNC Controller

In this chapter, we have discussed the various type of CNC controllers currently used on CNC machine and their nature. Here are a few types of CNC controller

- Microcontroller-based CNC controller
- PC-based CNC controller
- Industrial OEM CNC Controller

3.4.1.1 Microcontroller-based CNC Controller

A microcontroller-based CNC Controller is a device that uses a microcontroller such as Arduino or a Raspberry Pi to control a CNC Machine. The microcontroller is programmed to perform various tasks such as reading G codes, converting them into motion commands, and sending those commands to the CNC machine motor drivers.

The advantage of using a microcontroller-based CNC controller is that low-cost alternative to a traditional CNC controller which is very expensive and closed in nature, difficult to configure additional hardware.

To build a microcontroller-based CNC controller one needs to know electronics, microcontroller programming, and CNC technology.

The most widely used microcontroller platforms are

- Arduino
- Raspberry pi

Arduino

Arduino is an open-source electronic platform based on simple software and hardware. It was designed to make it easy for anyone to create interactive electronic projects. The hardware consists of a simple circuit board with a microcontroller, which is a programmable chip and various input/output (I/O) components such as sensors, actuators, and communication interfaces.

The software, the Arduino Integrated Development Environment (IDE), provides a programming environment where users can write, upload, and run code on the microcontroller. Arduino uses a variant of the C++ language [9].

Raspberry Pi

Raspberry Pi is a low-cost, credit card-sized computer that is designed to promote the teaching of basic computer science in schools. It was developed to provide a low-cost and accessible way for people to learn computer programming and experiment with various hardware projects. The Raspberry Pi can be used for a variety of purposes such as media centres, Retro gaming, Robotics, Home automation, and Web server.

The Raspberry Pi comes in several models. It supports a variety of programming languages including Python, and C++ [10].

3.4.1.2 PC-Based CNC Controller

A PC-based CNC controller refers to a system that uses a personal computer to control and operate a CNC machine. In this type of setup, the computer acts as the central processing unit to communicate with the various components of the CNC machine, such as the drive system, motors, and actuators to carry out the desired action.

The software used in PC-based CNC controllers typically involves CNC software, which can vary greatly in terms of functionality and complexity. Some software programs are

designed for basic operations, while others offer more advanced features such as 3D simulation and toolpath generation.

One of the benefits of using a PC-based CNC controller is that offers more flexibility and customization compared to traditional standalone CNC controllers.

However, there are also some drawbacks to using a PC-based CNC controller, including the need for a reliable computer system, compatibility issues between the software and hardware components, and the risk of data loss, delayed data transfer or software crash.

Widely used PC-based CNC controllers are Linux CNC, Flash Cut CNC, and Mach3 CNC controllers.

Linux CNC Controller

Linux CNC controller is usually called EMC2. It is an open-source software designed to control computer numerical control (CNC) machines. It provides a platform for converting a PC running on a Linux operating system into a powerful CNC controller that can drive various types of CNC machines like milling machines, lathes, routers, and plasma cutters.

Linux controller typically operates in a hardware configuration where it communicates with the CNC machine motors, drives, and sensors. It receives commands and instructions from a user-defined CNC program and translates them into precise movements and actions of the machine. These commands can be generated using various methods, such as manual input, G-code programming, or CAD/CAM software[11].

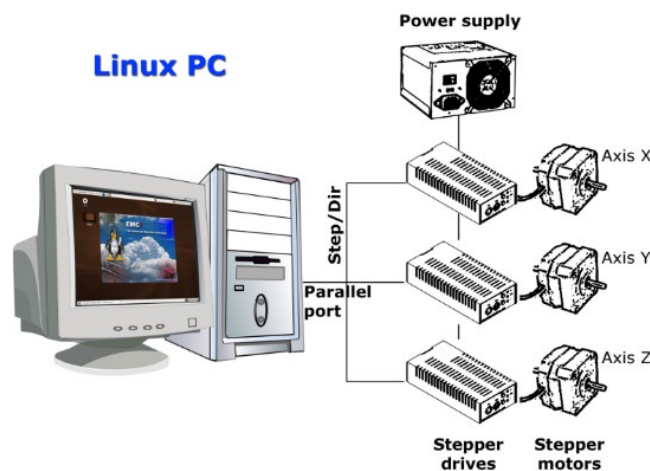


Figure 5: Block Diagram of Linux CNC controller[11]

Mach3 CNC Controller

Mach3 is a popular CNC controller that is used to control and operate CNC machines, such as mills, lathes, and routers. Mach3 was developed for use with Windows operating systems and provides a user-friendly interface and can be used for both small hobby projects and large-scale industrial productions.

The Mach3 software allows you to create and edit CNC programs, generate tool paths, control the movements of the machine, and monitor the status of the cutting process. It supports a variety of input devices including conventional keyboards and mice as well as specialised CNC controllers. It is known for its reliability and stability, making it a popular choice among CNC machine operators and hobbyists. [12]

3.4.1.3 Industrial OEM CNC Controller

Industrial OEM CNC controllers are devices used to control the operation of industrial machinery like lathes, mills, and routers. These controllers interrupt user input and convert it into specific machine actions, such as cutting or drilling. They are designed to work with a wide range of industrial machinery and are customised as per the needs of individual manufacturers.

An OEM CNC controller typically includes a control panel, an operator interface and control software. The control panel houses the electrical components of the system, including the microprocessor and input/output (I/O) interface.

OEM CNC controllers are designed to be integrated into a specific machine and are not typically sold as standalone products. Instead, they are custom-built by the machine manufacturers. This allows the machine manufacturer to optimize the controller for specific capabilities and requirements of their machinery, leading to improve performance and greater efficiency.

There are many different types of Industrial OEM CNC controllers on the market, each with its unique features and capabilities. Some of the common types include:

1. PLC-based CNC controllers
2. PC-based CNC controllers
3. Standalone CNC controller
4. Servo-based CNC controller
5. Motion control CNC controller

Discussion of Result

Based on the above-discussed CNC controllers each of them has its kind of working principle and setup procedures. We have chosen the microcontroller-based CNC controller for this project. Arduino is used as the electronic platform as it is widely used in the three-axis milling machine quite cheap and the source of code is available in many open-source platforms. Various firmware for Arduino-based CNC machines is available in open-source platforms which are used for the controlling machine. Mostly it can be modified as per our custom need when it is required as it was developed on the programming language C++.

3.4.2 Firmware

The base of the software stack that computer hardware utilizes to perform fundamental functions and run applications is called firmware. To control the operations of numerous hardware components, hardware manufacturers utilize embedded firmware.

In this project, the Arduino microcontroller is used as an electronic platform which requires the program to control the other hardware. There is several firmware in the open-source platform which can be used as the firmware performing the functionality of a three-axis milling machine.

1. GRBL

GRBL is one of the open-source firmware for the motion control of CNC machines. It runs on a microcontroller board such as Arduino. It is written in the C language, which is highly configurable.

The firmware is responsible to interpret the G-code and instructs the CNC machine to move to a specific position, at a specific speed. It also includes several features such as G-code interpretation, Acceleration and Deacceleration control, homing and limit switches, and real-time override. But PC is required to run the universal G code, to send the commands[13].

2. Marlin

Marlin is one the most widely used open-source firmware for 3D printers and even supports three-axis milling machines and CNC Laser machines. It is written in the C, C++ language. It is highly customisable and allows users to modify the firmware to suit their specific needs. It comes with nice features like bed levelling with a probe, AutoStart, multiple stepper motor compatibility, temperature control of motors, and an LCD menu tree. Since it's an open-source platform, firmware can be easily edited or

changed as per to meet our needs like support for additional stepper motors, sensors, or controllers. Marlin is compatible with more than 150 standard G codes.

It can support SD cards. It is compatible with Graphical User Interface (GUI). Possibility to transfer the information from the SD card and work as a standalone unit [14].

3. Repetier

Repetier is a firmware developed for 3D printers and CNC machines. Repetier provides a host and server to communicate with CNC. It can configure firmware with multiple unique features for CNC machines including support for controllers and stepper motors, configuring multiple tools allowing for the simultaneous use of different types of tools. It can support a wide range of standard G codes in CNC machines. It has a user-friendly interface that makes it configure easily with a wide range of options.

It is compatible with Graphical User Interface (GUI). Possibility to transfer the information from an SD card and work as a standalone unit [15].

4. Maslow

Maslow is a complete open-source CNC design, but a bit odd firmware for the CNC routers that rely on Gravity to position and maintain their positional control. It hangs the router from two drive chains at an angle resting the router on the stock. The Maslow design allows it to be a large and affordable CNC router.

It is also considered because it was developed to work on the Arduino as its controller. It has its own Custom Shield, while it may not support any other machines [16].

5. Klipper

Klipper is a firmware that uses Arduino and a computer to control CNC. Klipper takes a different approach to CNC control than most. In Klipper, the computer (in this case, a Raspberry Pi) takes care of the timing, physics, and G-code interpretation while the MCU follows the instructions on time. We can have multiple Arduinos, or other boards, controlling different parts of your CNC. No longer would there be an issue with having a single board with enough stepper drivers or power. Even better, all of the configuration is done quickly and easily on the Raspberry Pi. Feature of the Klipper Multiple microcontrollers Simultaneous operation and synchronization,

Precise stepper movement, Exclusive to 3D printers of type Cartesian, delta, and core XY printers [17].

Table 1: Comparison between GRBL, Marlin and Repetier Firmware [Source:]

Features	GRBL	Marlin	Repetier
CNC Feature Support	Yes	Limited	Yes
Motion Control (Interpolation)	Yes	Yes	Yes
CNC Specific Features	Yes	Yes	Yes
G code support	Yes	Yes	Yes
Bed Levelling	No	Yes	Yes
Support for multiple tool	No	Yes	Yes
Real time status update	Yes	No	Yes
User friendly Interface	No	Limited	Yes
Customizability	Limited	Yes	Yes
Modular Architecture	No	Limited	Yes

Discussion of Result

Based on the above-discussed CNC controlling Firmware most of them are developed for 3D printers and CNC Controllers. We have chosen Repetier Firmware for our hardware for CNC controlling, as compared with other firmware for CNC controlling Repetier has unique features like host and server for communication, multiple stepper motor compatibility, supports most of the standard G codes, Graphical User Interface. As Repetier can be sourced from the open-source platform. We could have regular updates as it is used by many around the world.

4 System Design Methodology

The purpose of this thesis is to build a low-cost milling machine employing a three-axis mill with an automatic tool changer. This is accomplished by employing an electronic platform based on Arduino to control the motion of the machine's axis. The key principles upon which the machine structural design is

- Hardware Configuration
- Software Configuration

4.1 Hardware Configuration

Hardware configuration is the process of configuring and personalizing hardware elements and settings to enhance effectiveness, performance, and compatibility within a computer system or other electronic device.

The hardware configuration of this machine includes the main control board, electronic devices, supporting drives, electrical components, mechanical frame, and automatic tool changer.

4.1.1 Arduino Mega 2560

Arduino Mega 2560 is a microcontroller board that is widely used in various electronic projects, particularly in the field of hobby electronic projects, robotics and the Internet of Things (IoT). This board is based on an ATmega2560 microcontroller and provides a large number of input/outputs, and serial communication interfaces, making it well-suited for a wide range of applications.

The Mega2560 has 54 digital input/output pins, 16 analogue inputs, 4 UART (hardware serial ports) a 16MHz crystal oscillator, A USB connection for programming and power and power jack for an external power source. These features allow for the development of complex projects, such as remote-controlled robots, wireless data collection and transmission systems.

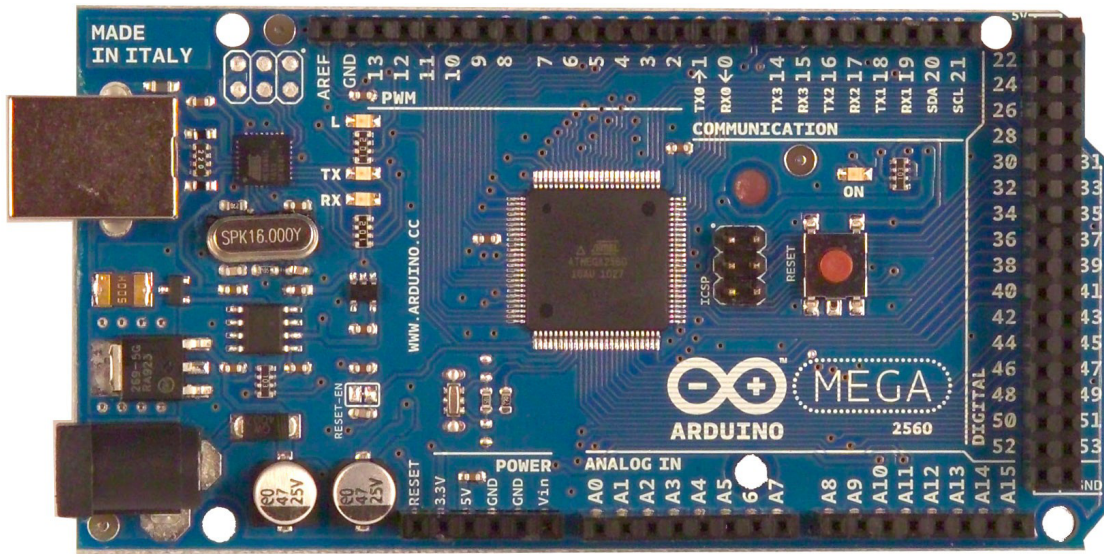


Figure 6: Arduino Mega 2560 [18]

Additionally, the Mega 2560 is compatible with a vast library of existing software and hardware tools, including the Arduino integrated development environment (IDE), making it relatively easy to use even with limited programming knowledge [18].

Table 2: Technical Specification of Arduino Mega 2560[18]

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (Limited)	20V
Digital I/O Pins	54
Analog Input Pins	16
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EPROM	4 KB
Clock Speed	16 MHz

4.1.2 RAMP 1.4 Shield

RAMP 1.4 (RepRap Arduino Mega Pololu Shield) shield refers to a hardware device that is designed to be used with the RAMP 1.4 framework. A shield is an add-on device for microcontrollers, such as Arduino that provides additional functionality and capabilities.

The shield provides additional output, communication interfaces, sensors, or other hardware components that are needed for the experiments being performed.

It is beneficial because it can simplify the process of performing experiments and collecting data. By having all these hardware components integrated into a single device, this shield can save time and reduce the complexity of setting up [19]

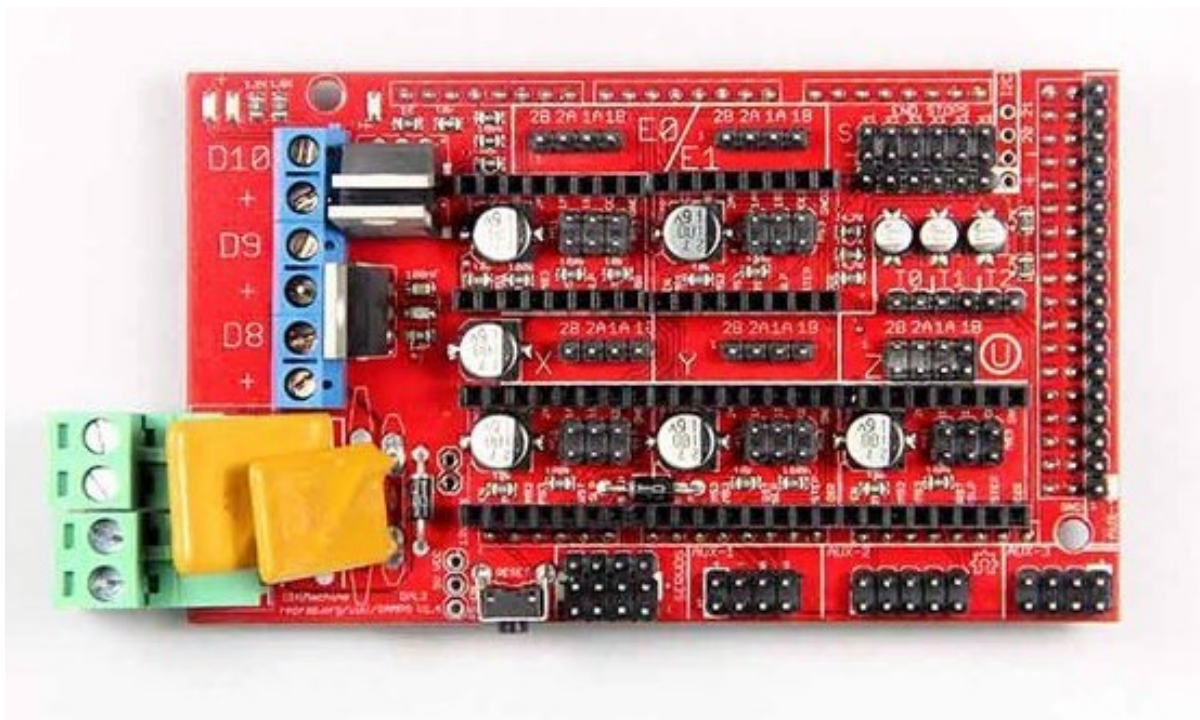


Figure 7: RAMPS 1.4 Module [19]

4.1.3 Bipolar Stepper Motor

A bipolar stepper motor can be described as a type of electric motor that operates based on the interaction between the magnetic field in the stator and rotor. The stator is made up of two pairs of magnetic poles and the rotor has two pair of windings. By controlling the energization of these windings, the magnetic field can be made to interact in a specific sequence, causing the rotor to rotate step by step. This results in precise control over the position and speed of the motor.

The motor's unique step-by-step operation makes it well-suited for this application like robotics, automated machinery, and other technical devices. However, bipolar stepper motors have limitations such as lower torque compared to other types of motors and the need for specialized control electronics drives.

Bipolar Stepper Motor (SM15DD) is used for the axis of the machine.



Figure 8: Stepper motor SM 15DD

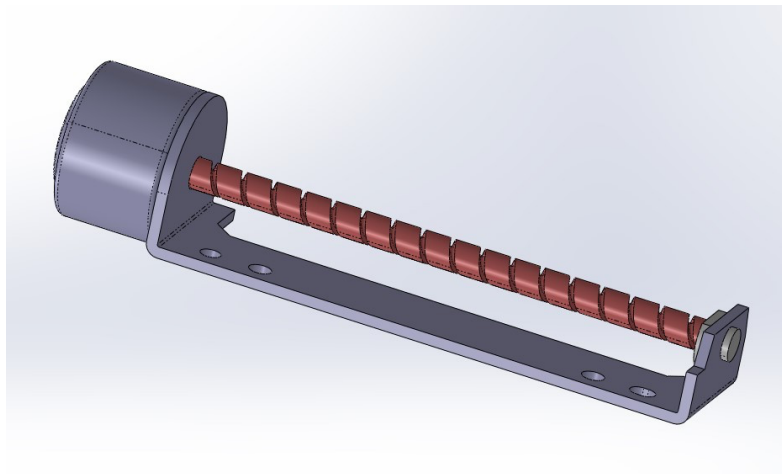


Figure 9: 3D model of Stepper motor with lead screw for axes [Source: Own]

For the Automatic tool changer stepper motor is used.

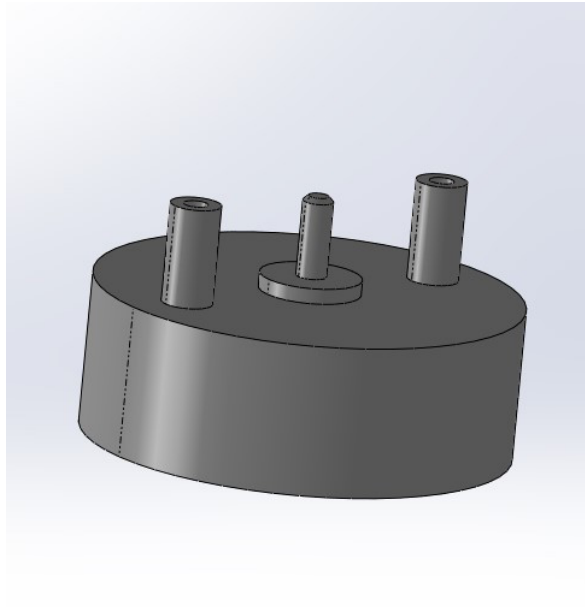


Figure 10: 3D model of a stepper motor for ATC [Source: Own]

4.1.4 DRV8825 Stepper Motor Drivers

Stepper motor drivers are an electronic device that controls the motion of stepper motors, which are capable of continuous rotation with precise position control even without a feedback system. They convert electrical pulses into precise mechanical movements allowing stepper motors to move in the desired direction with precise steps.

Stepper motor drivers can be either micro-stepping or full-step. Micro-stepping drivers provide finer control over the motor's movement by dividing each full step into small steps, resulting in smoother, more precise movement. Full-step drivers control the motor's movement by only energizing the motor in full steps, which is a more basic and jerky movement.

Stepper motor drivers also have different types of control methods, including pulse width modulation, chopper control and constant current control [20].

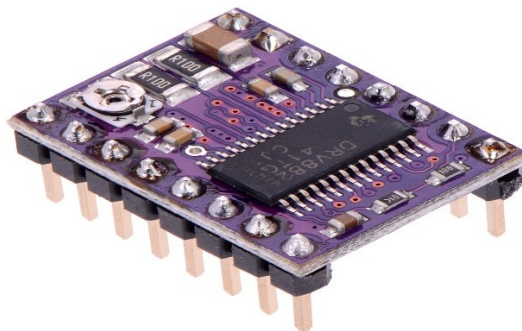


Figure 11: DRV 8825 Driver [20]

In this project, we have chosen the DRV8825 micro stepping bipolar stepper driver which is used to control the movement of stepper motors with feature adjustable current limiting, over current, and over temperature protection and six micro step resolution. It functions as a master device, it takes the commands from a controller and converts them into electrical signals to drive the stepper motors. It is capable of producing high current, high-resolution output making it suitable for use in the 3D printer, CNC machine, and robotics.

Table 3: Technical Specification of DRV 8825 Driver [20]

Minimum operating voltage	8.2 V
Maximum operating voltage	45 V
Continuous current per phase	1.5 A
Maximum current per phase	2.2 A
Minimum Logic Voltage	2.5 V
Maximum Logic Voltage	5.25 V
Micro step Resolution	Full, 1/2, 1/4, 1/8, 1/16, 1/32
Weight	1.6 g

4.1.5 DC Spindle Motor

The brushed DC motor operates on the fundamental principle of electromagnetism, transforming electrical energy into mechanical motion through a well-defined process. A brushed DC motor comprises two primary components: the stator and the rotor. The stator remains stationary, housing field windings that generate a magnetic field when an electrical current flows through them. In contrast, the rotor, connected to the motor's output shaft, is designed to rotate which coil has electricity flowing through it, the rotor is propelled. DC motors offer a variety of advantages and are affordable and simple to use [21].

A 5V DC motor is used as the spindle motor of the machine. This DC motor has a working range of 4-10V. Under no load condition at 7V, it has a speed of 10300 rev/min.

4.1.6 OLED Display

OLED stands for “Organic light emitting diode”. OLED displays are made up of thin organic film that emits light when an electrical current is applied to them, which allows for the creation of highly efficient displays that can produce deep black and vibrant colours.

FACULTY OF MECHANICAL ENGINEERING TUL

The SSH1106 is a type of OLED display that is commonly used in small, low-power devices. It is a monochrome display, which means that it can display only one colour white. And cannot display images in full colour.

The SSH1106 uses a communication interface called I2C (Inter-Integrated Circuit) to receive commands and data from a controlling device such as a microcontroller Field[22].



Figure 12: SSH 1106 OLED Display Front View [22]

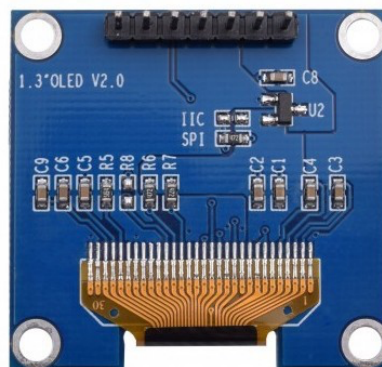


Figure 13: SSH 1106 OLED Display Bottom View[22]

Table 4: Technical Specification of SSH1106 OLED Display

Display mode	Passive Matrix
Display Colour	Monochrome
Number of Pixels	128 x 64
Display Size	1.3 inch
Active Area	29.42 x 14.7 mm
Operating Temperature	-40 to +70°C
Operating Voltage	3.3 V
Weight	5.7 g

4.1.7 Micro SD Card Socket

A micro-SD module SPI (Serial Peripheral Interface) is a type of device that allows you to connect a micro-SD card to a microcontroller or computer using the SPI communication interface. The micro-SD card is a type of small portable memory device. By connecting a micro-SD card to a microcontroller via an SPI interface you can store large amounts of data and easily retrieve it as needed.

An SPI-based micro-SD module typically includes a micro-SD card slot and a small microcontroller that provides the necessary interface between the micro-SD card and the SPI bus for controlling the device.

This allows the controlling device to access the stored data on the micro-SD card as if it were part of its own. The micro-SD card is used to store the G codes, which are generated from the CAM software to run the axis motors.

The advantage of using micro-SD module SPI is that allows you to easily expand the storage capacity of a microcontroller without requiring significantly changing the hardware or software. Then micro-SD card is relatively inexpensive, they provide a cost-effective way to add a large amount of storage field[23].

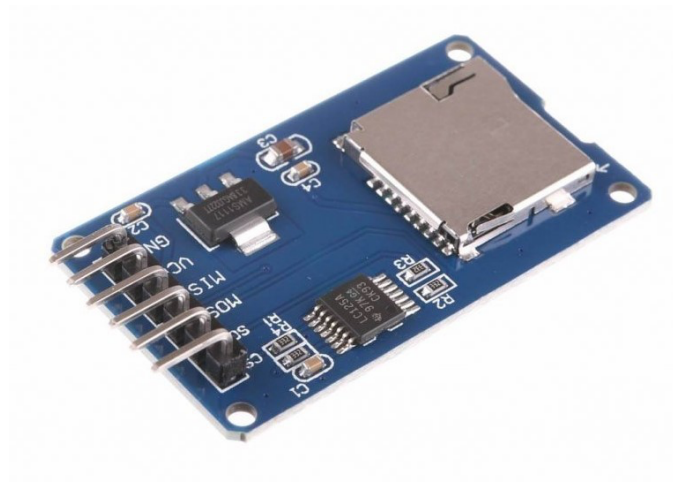


Figure 14: MicroSD Card Module [23]

Table 5: Technical Specification of Micro SD Card Module [23]

Power Supply	5V
Support	Micro SD card
Interface	Standard SPI Interface
Weight	5 g

4.1.8 Power Supply

The power supply provides the necessary electrical power to the Arduino board, motor drivers, and other components of the machine. It typically includes a transformer or AC to DC adapter to convert the main supply to the desired voltage or power output.

4.1.9 Mechanical Base Frame

Machine frame or Machine base usually refers to the foundation or structure that supports and provides stability to the machine. The base includes the provisions for mounting the machine components such as the linear guideways, spindle and axis motors. The design of the base must also allow for accurate positioning and alignment of the various machine components to ensure precious motion and high performance.

A well-designed machine base can also reduce vibration and noise, and provide overall machine performance and accuracy. The design and construction of the machine base will vary

depending on the specific requirements of the machine, such as the size, weight, and type of the workpiece, and desired level of precision.

In this project, the machine frame is constructed using a 3D printing. Each component is 3D printed and assembled individually as per the functions and requirements.

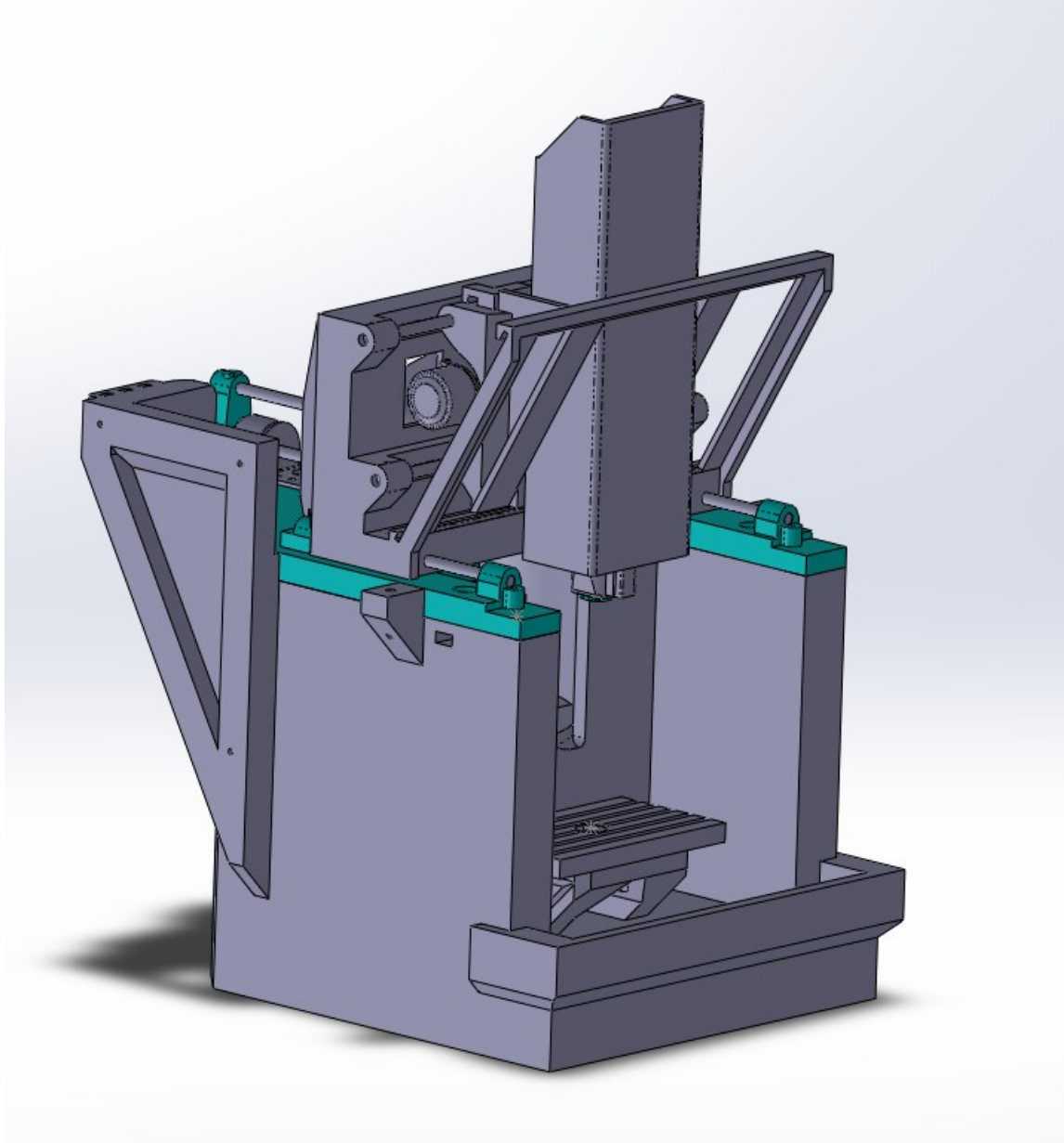


Figure 15: Virtual 3D image of projected Three-axis machine [Source: Own]

4.1.10 Automatic Tool Changer (ATC)

Automatic tool changers (ATCs) are used in computerized numerical control (CNC) machine machines to increase production and tool-carrying capacity. ATCs switch between tools quickly, cutting down on excess time. They are typically utilized to increase the machines'

ability to work with a variety of tools. They are also used to replace tools that are damaged or worn out.

To ensure the smooth run of the Automatic tool changer (ATC) through the complete automation of the tool change operation sequence the path followed by the tool from the magazine to the spindle [24].

The following are the basic building blocks that make up the ATC systems:

- Tool Magazine
- Tool Transfer Mechanism
- Parking Station
- ATC Arm
- Tool clamp Mechanism (Spindle)

4.1.10.1 Selection of Tool Changer Mechanism

A carousel-type tool-changing mechanism is selected as the tool-changer mechanism. The mechanism typically consists of a circular or semi-circular tool magazine or carousel that holds multiple tool holders or tool pockets. Each tool holder is designed to securely hold a specific tool.

Carousel-type tool-changing mechanisms provide advantages such as reduced setup time, increased machine uptime, and the ability to accommodate a larger number of tools without requiring manual intervention.

Description of carousel-type tool changing mechanism

1. **Tool Magazine/Carousel:** The multiple tool holders are held in place on a rotating component called the tool magazine. Usually, it's integrated into the design of the machine or placed close to the machining area.
2. **Tool Holders:** A specific tool is intended to fit in each tool holder. Assuring stability when performing machining operations, it firmly keeps the tool in position.
3. **Tool Selection:** The CNC machine's control system rotates the tool magazine when a tool change is necessary to align the appropriate tool holder with the tool changing mechanism.
4. **Tool Changing Mechanism:** The tool holders are grabbed and released by the tool changeover mechanism. Depending on the machine design, there are several kinds of

mechanisms, but typical ones include pneumatic clamping, hydraulic clamping, or mechanical locking systems.

5. **Tool Exchange:** The tool holder is released from the tool magazine once the tool changeover mechanism has firmly grasped it. The current tool is then unclamped or released as the mechanism moves the tool holder to the desired tool change position.
6. **Tool Installation:** The mechanism feeds the new tool holder into the spindle or tool holder interface when the old tool has been removed to ensure appropriate alignment and engagement. The tool has been locked or held in position.

4.1.10.2 Proposal on Design of Automatic Tool Changer Magazine

The design proposed for this project is a design of machining centres having magazines of the circular type, where the tool is stored in a position parallel to the machine spindle's axis. The tool changing is based on the carousel-type tool-changing mechanism. The transfer mechanism and waiting post are replaced by the automatic tool changer mechanism.

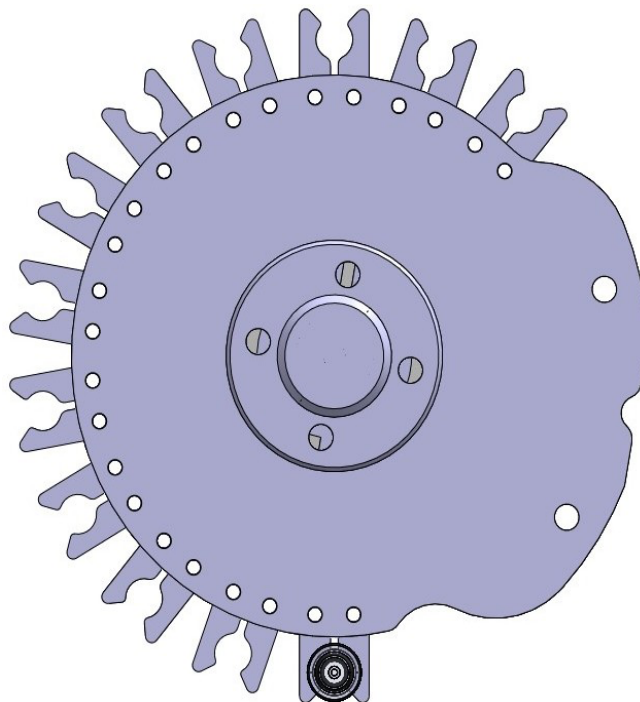


Figure 16: Design of Tool Magazine [Source: own]

From Figure 16, the disc-type magazine is designed in a way of having the capacity to store 12 tools. The first tool is positioned at 80° , then each tool is positioned at 20° intervals. The 0° is considered the home position of ATC magazine.

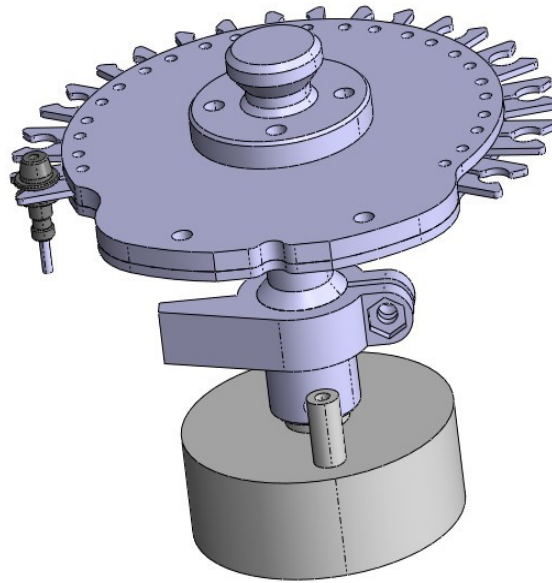


Figure 17: Assembly of Automatic Tool Changer Magazine [Source: Own]

From Figure 17, the end shaft is coupled to the stepper motor which is responsible for positioning the position of the tool in the tool magazine. The position of the tool in the magazine is controlled by controlling the rotation of the stepper motor. A limit switch is used to determine the homing position when the disc is at 0° and limit the rotation of the shaft.

4.1.11 Tool Setting

Tool setting is a technique used in Computer Numerical Control (CNC) machines to precisely measure and set the position of cutting tools. It involves using a laser sensor to detect the exact location of a tool mounted on the CNC machine's spindle. This information is then used to accurately establish the tool's dimensions relative to the machine's reference point.

In a typical CNC milling machine, a laser tool setter is attached to the machine's table. When the tool setting process is initiated, the CNC machine moves the tool close to the laser sensor. The laser beam is then projected onto the tool surface, and the reflected or interrupted light is detected by the sensor. The sensor can accurately determine the distance between the machine's reference point and the tool surface based on the reflected light's time of flight.

From the process of tool offsetting it is used to determine geometric information like length, radius or diameter of a cutting tool. Non-contact tool setting systems use an optical (laser) beam to detect tool presence.

4.1.11.1 Types of Tool Setters

Tool setters for CNC machines are devices used to accurately measure and set the position of cutting tools and workpieces in the machine. They play a crucial role in ensuring precision and efficiency in CNC machining operations. There are various types of tool setters, each with its specific use and advantages. Here are some common types and categories of tool setters for CNC machines:

1. **Touch Probes:** They consist of a stylus or probe that makes contact with the tool to measure its position. The probe sends signals to the CNC control unit, allowing it to determine the exact position of the tool.
 - **Tool Touch Probes:** Used for measuring and setting the position of cutting tools, such as end mills, drills, or inserts. They help with tool length and diameter measurement, as well as tool offset compensation.
2. **Optical Tool Setters:** Optical tool setters use laser technology to measure tool length and diameter without making physical contact with the tool. They are fast and accurate, allowing for rapid tool setup.
3. **Vision-based Tool Setters:** These tool setters use a camera system to visually inspect and measure cutting tools. They can be effective for complex geometries or non-standard tools.
4. **Electromagnetic Tool Setters:** This type of tool setter uses electromagnetic sensors to measure the tool length or workpiece position. They are useful for non-conductive or non-metallic tools and workpieces.
5. **Renishaw Tool Setters:** Renishaw is a well-known brand that manufactures tool setting systems. Their tool setters often use radio frequency (RF) communication to transmit measurement data to the CNC control.
6. **3D Tool Setters:** These advanced tool setters can measure not only the length and diameter of the tool but also its radial and axial runout, as well as other geometric parameters. They provide more comprehensive data for tool setup and diagnostics.

4.1.11.2 Optical Tool Setter Benefits

Optical tool setter has several benefits that make the popular choice of using it in CNC machines. Here are some of the advantages of the optical tool setter. [25]

- Non-contact measurement
- High precision
- Higher measurement
- Reduced setup time
- Integration with CNC controller
- Versatility

When the tool passes through the beam of light of the sensor, the tool breaks the light. Where the dimension of the tool like length and Radius/Diameter of the tool is calculated. When the tool moves in the vertical direction of the light, the length of the tool is measured. If the tool moves in the horizontal direction the radius and diameter of the tool can be determined.

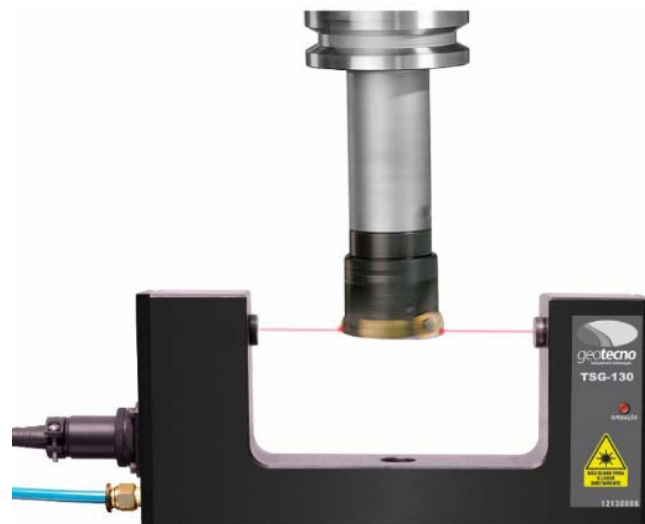


Figure 18: Industrial Laser Tool Setter

5 Experimental Work

The firmware for the three-axis milling machine with an automatic tool changer has been developed and tested by the requirement. The test results are concluded and discussed.

For the functioning of a three-axis milling machine, an algorithm from the Repetier firmware is installed in the Arduino microcontroller.

The algorithm codes are developed and verified using the Arduino IDE (Integrated Development Environment) software. The Arduino IDE platform is used to send the algorithm from the computer to the Arduino Mega 2560 microcontroller.

5.1 Architecture of Hardware Connection

The architecture of hardware connections forms the intricate framework upon which the digital world is built. It encompasses the design, arrangement, and interplay of physical components that enable devices to communicate, compute, and function seamlessly.

5.1.1 Hardware Connection of Arduino Mega 2560 with PC Screen

The structure and organization by which various hardware components are connected to a computer system. From Figure 19, the design is proposed for the Arduino with PC screen configuration. This design is enabling the data transfer and communication between different hardware components assuring the system's efficient operation.

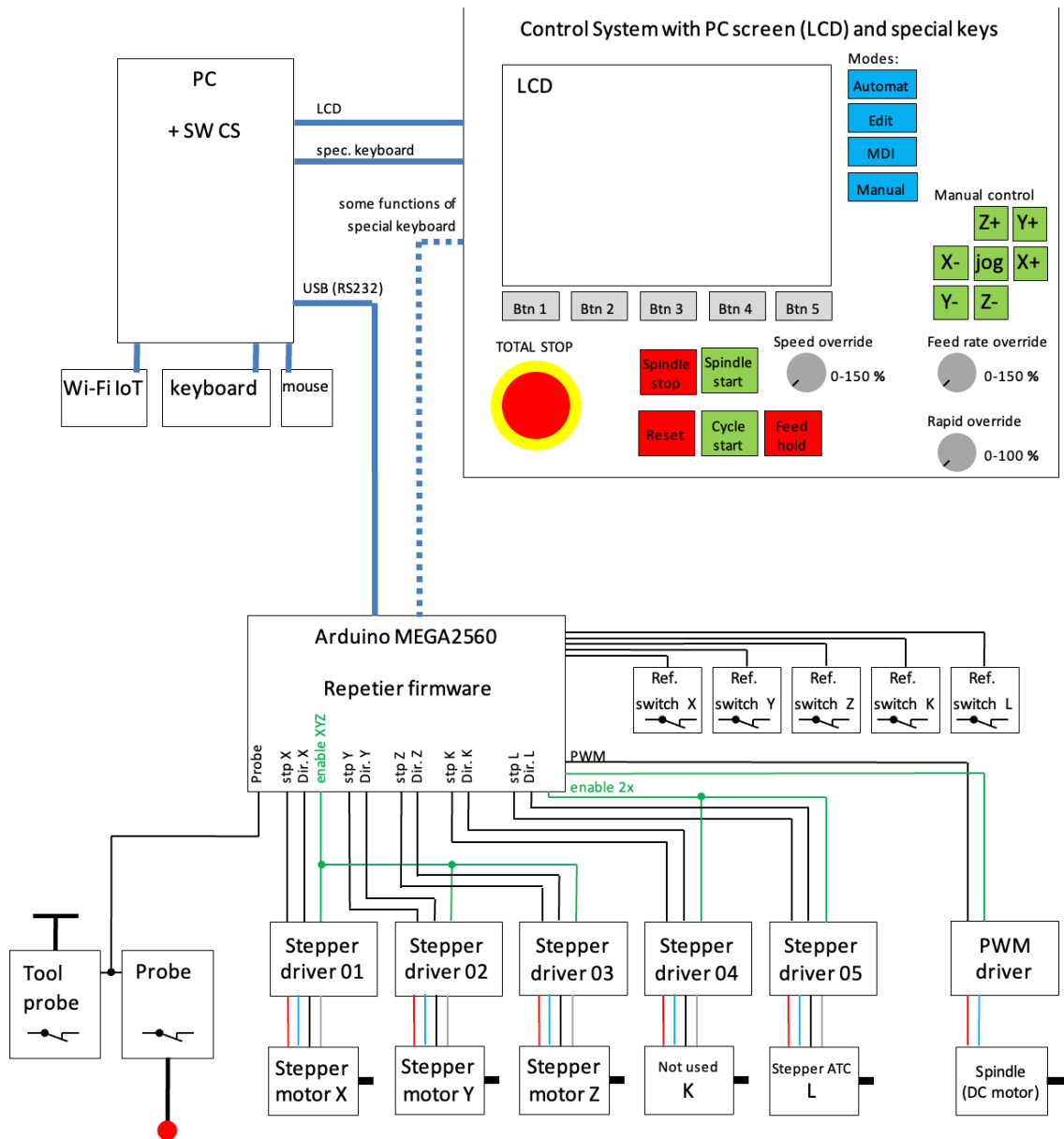


Figure 19: Schematic representation of hardware connection with PC screen [Source: Own]

From Figure 19, the hardware connection for the three-axis milling machine with PC is shown as a schematic representation between the various components. The Arduino board serves as the microcontroller configured with Repetier firmware that controls the machine's operation. Various inputs are given to the Arduino from PC. Then microcontroller instructs the motors by sending electrical signals from it. Motor drivers are attached to the RAMP 1.4 shield as an add-on to the Arduino board with many outputs to which the stepper motors are attached with it. The stepper motor drivers are for precise control of motor movement. The spindle motor is connected to the microcontroller by a Pulse width Modulation driver.

The hardware connection of various components is represented in Figure 16 with the Arduino microcontroller. The microcontroller is connected to the four stepper motors with DRV8825 drivers. DC spindle motor is connected to the Pulse Width Modulation driver for controlling the rotation of the spindle.

5.1.2 Hardware Connection of Arduino Mega 2560 with LCD Screen

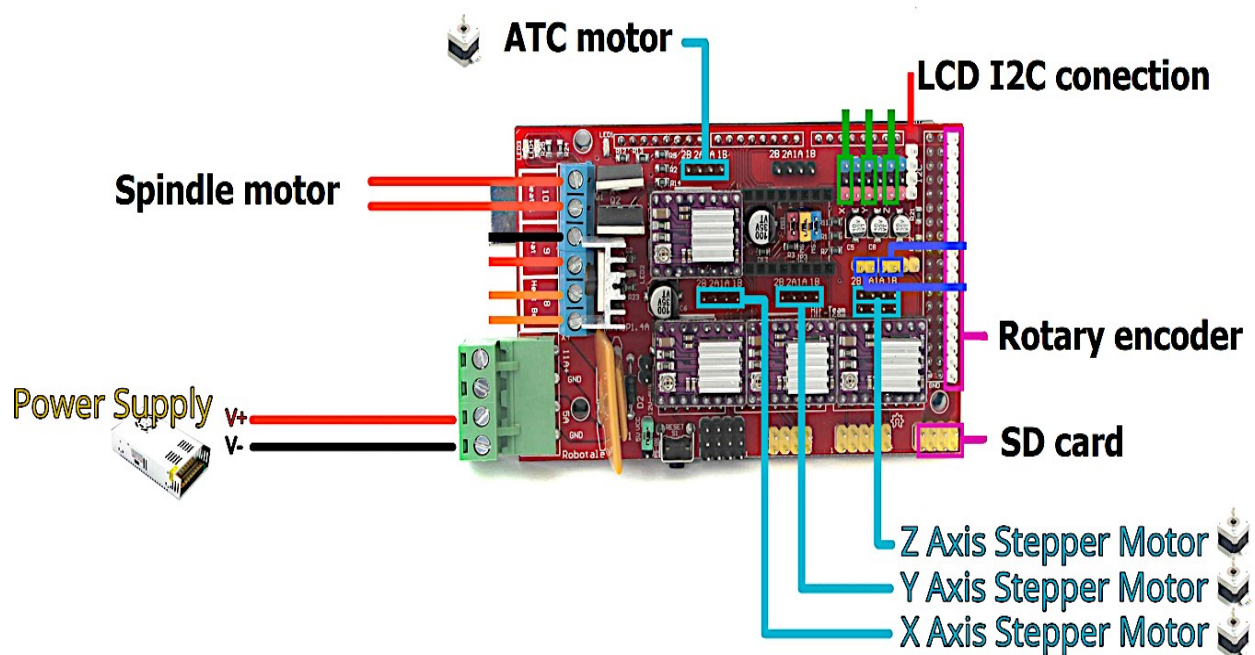


Figure 20: Schematic representation of hardware connection with LCD screen [Image edited from [26]]

From Figure 20, it is shown the connection of the three-axis milling machine with the hardware components. The Ramp 1.4 shield is connected to the Arduino Mega 2560 microcontroller.

The power supply to the board is connected to the 5A pin.

The spindle motor is connected to the D10 pin. X, Y, and Z axes stepper motors are connected to the represented X, Y, and Z pinouts on the Ramps shield with the stepper motor drivers.

ATC stepper motor is connected to the E0 pinout with the stepper motor driver.

FACULTY OF MECHANICAL ENGINEERING TUL

The rotary encoder is connected to the D31, D33, and D35 in the Aux-4 section, as it is used to navigate between the options menu list on the LCD screen. The LCD screen is connected to the I2C pinout on the board. Inputs are given to the Arduino microcontroller using the rotary encoder.

Table 6: Comparison of PC screen and LCD screen configuration for Arduino

Aspects	Arduino Mega 2560 with PC screen	Arduino Mega 2560 with LCD screen
Display Quality	PC screen with higher resolution and larger in size	Compact and suitable for embedded systems. Limited graphics capability
User Interaction	Graphical User Interface based	Limited user interaction.
Compatibility	Compatible with any PC with Compatible ports	Cost effective especially for small screen
Software Compatibility	Various software options available (Ex: Repetier Host, Pronterface, etc)	Firmware based standalone
Setup complexity	More involved setup process	Simplified setup process
Connectivity	Requires continuous connection to a PC	Standalone operation after firmware flashed
Flexibility	Versatility and extensive features	Provides a simpler, dedicated display solution.
User Experience	Enhanced visual experience and control option	Basic display or relevant information
Power supply	External power supply	Powered via USB connection
Cost	Expensive	Cost effective
Space	Space consuming requiring support components like mouse , keyboards.	Integrated within a board

5.1.3 Conclusion of Comparison

From the above comparison review, the choice between the PC screen and LCD screen depends on the project's needs and constraints. While LCD screen configurations have many advantages suitable for embedded systems, cost-effective, lower power consumption, simplified setup, and versatility. So the system with Arduino configured with LCD screen is chosen for this project.

5.2 Three-Axis Milling Machine

This three-axis milling machine is a prototype based on the real-time machine to develop a control system. It is an Arduino-based three-axis milling machine configured with the Repetier CNC firmware for the functioning of axis and spindle control. Repetier CNC firmware is an open-source platform firmware.

5.2.1 Machine Configuration

It is a three-axis gantry milling machine with a movable column and fixed table. This three-axis milling device is powered by an Arduino and equipped with a Ramp 1.4 shield. The coordinate axes are configured with dimensions of 45 x 45 x 45 mm. To move each axis, a stepper motor with a 3mm guide bar is used. The complete machine frame is produced through 3D printing using Acrylonitrile Butadiene Styrene (ABS), a polymer with exceptional heat, chemical, and abrasion resistance, as well as surface hardness and rigidity [19].

This machine has a function called the fourth axis, which is activated and utilized to configure an automatic tool changeover. The axis drive is provided by a stepper motor that is connected to the ATC magazine by a connecting rod.

The machine's spindle is powered by a DC motor, and magnetic clamping is used to secure the tool to the spindle.

It includes an LCD monitor that shows the machine's current coordinates and state of milling. For display navigation, this machine is connected to a rotary encode.

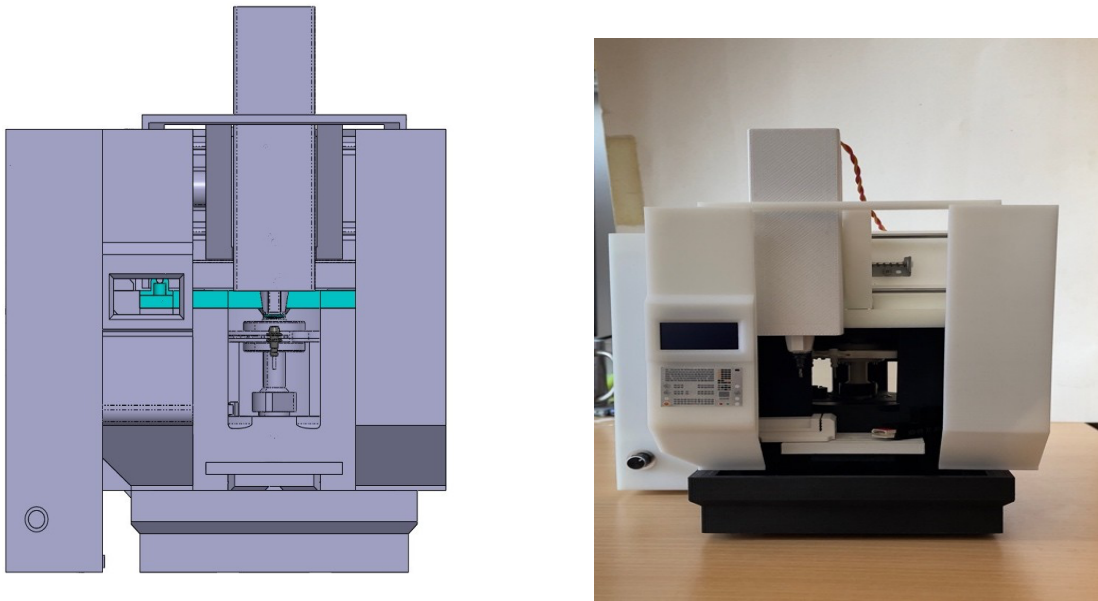


Figure 21: a) 3D model of a three-axis milling machine b) Real machine Image [Source: Own]

5.2.2 Three-Axis milling machine setup

This machine is based on the Repetier firmware, it uses configuration.h file for the machine setup. A Repetier host software is used to set up the machine configuration. Initially, the machine is configured with the required aspects and firmware is generated.

The firmware is then uploaded to the Arduino Mega 2560 microcontroller board, using an Arduino Integrated Developing Environment (IDE) platform which is used to upload, edit, and verify the correctness of the firmware.

The Arduino Mega 2560 microcontroller is connected to the computer through a micro USB cable which provides the power supply and data transfer through it



Figure 22: Window of Arduino IDE [Source: Own]

From the configuration.h file the aspects and connection of the machine with stepper motors are known. The machine has a configuration of X, Y, Z and E0. E0 represents the ATC configuration.

The machine has a max feed rate of 50 mm/s for all three axes, max acceleration of all three axes is 1000 mm/s².

Homing order of the machine is ZYX. Initially, the Z-axis reaches its homing position followed by the Y-axis and X-axis. This order homing is configured for the purpose of the safety of the machine to avoid any collision with the tool machine.

The Pinout Configuration of axes for stepper motors is presented in the following table 7. The pinout corresponds to the RAMPS 1.4 shield connection.

Table 7: Pinout configuration of RAMPS 1.4 for Axes

AXIS	ASPECTS	PINOUT
X AXIS	STEP PIN	54
	DIRECTION PIN	55
	ENABLE PIN	38
Y AXIS	STEP PIN	60
	DIRECTION PIN	61
	ENABLE PIN	56
Z AXIS	STEP PIN	46
	DIRECTION PIN	48
	ENABLE PIN	62
E0 AXIS	STEP PIN	26
	DIRECTION PIN	28
	ENABLE PIN	24

The constant coordinates for the ATC are configured in the configuration.h file as a macro. The macro corresponds during the tool-changing sequence. The macro used in Automatic Tool Changer is explained in the following Table 8 with corresponding coordinates.

Table 8: ATC String name with axis coordinates

String Name	Name	Axis	Coordinates
S1	X1	X	23.0
S2	Y_POINT_A	Y	28.0
S3	Y_POINT_B	Y	44.0
S4	Z_POINT_1	Z	40.5
S5	Z_POINT_1	Z	45.0
S6	ATC_OFFSET	E0	3.2
S7	ATC_HOME	E0	0.0

5.3 Testing of Tool changer mechanism

Carousel-type tool changer mechanism is the basis for the tool changer in this project. The firmware was developed using the C++ programming language. The tool changer mechanism is focused on three conditions.

1. Selection of new tool.
2. Change of tool, when there is another tool in the spindle.
3. The spindle has to be Empty

5.3.1 Overview of Tool Changing Mechanism:

Step 1: The spindle approaches the magazine to insert the old tool into the empty pocket.

Step 2: The tool is placed into the empty pocket and released in that position.

Step 3: The spindle lifts to a safety limit distance.

Step 4: The disc rotates a specific tool number which is positioned normally to the spindle axis.

Step 5: Spindle lower and clamps the tool and move forward from the magazine.

Step 6: Tool magazine door closes.

Condition 1: Selection of new tool

The condition is met when the spindle is empty and a new tool is selected from the magazine. Upon issuing the tool command, a systematic sequence of tool change operations commences, providing the user with the option to select a new tool.

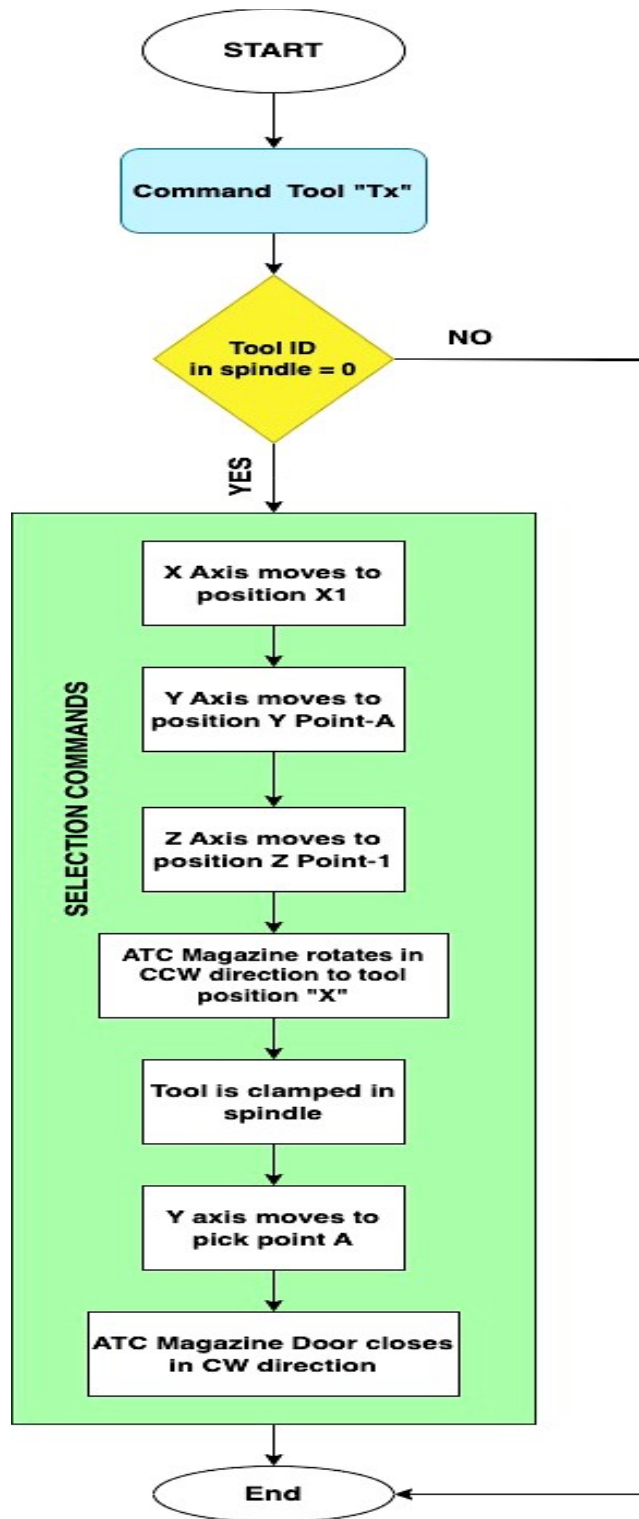


Figure 23: Flow chart for initial tool pick, with empty spindle [Source: Own]

Figure 23, the process flow diagram for changing tools while the spindle is initially empty. The sequence demonstrates what takes place when a tool gets the "Tx" command. X is

the Tool Id, which is in the range of 1 to 12. The Y axis travels to the pick point- A position while the X axis moves from the current position to the specified position-X1, which is in the "X23" coordinate. This location is in front of the ATC magazine. The Z axis then moves to position 1, which is the tool pick position in the ATC magazine above the tool with the safe limit. When the tool position "x" has been set, the ATC magazine turns counterclockwise. Moving the Y axis to the pick point-B, which is parallel to the tool axis in ATC magazine Z axis descends to position 2, which is the pick-up or spindle-clamping position. The spindle has the tool secured within. Then The Y axis reaches the pick point A. The ATC magazine door then closes clockwise.

```
Tool* current = tool->current;  
Tool* next = &tool[toolId];
```

Here, a pointer named current of type Tool* is declared and assigned the value of tool->current. It seems that the tool is a pointer to some data structure or array that holds the tool object and current is assigned the value of the pointer tool-> current, which points to one of those tool objects.

Another pointer named next of type Tool* is declared and assigned the address of a specific tool object in the array. The address is obtained using the & operator followed by tool[toolId]. This line suggests the tool is an array of tool objects, and toolId is an index used to access a specific element in that array.

Here, Several String objects are created, and they are initialized with the string representations of different numerical values. The provided code uses String() constructor, which is often used in Arduino programming to convert numerical values to strings.

```
String S1 = String(X1);  
String S2 = String(Y_point_A);  
String S3 = String(Y_point_B);  
String S4 = String(Z_point_1);  
String S5 = String(Z_point_2);  
String S6 = String(ATC_OFFSET + current->id);  
String S7 = String(ATC_OFFSET + toolId);
```



```
bool execute select = toolId != current->id;
Com::printFLN(PSTR("SelectTool:"), static_cast<int>(tooled));
```

Here, a Boolean variable executeSelect is declared and initialized. It is assigned the value of the result of comparison toolId! = current toolId. This comparison checks if the value of variable toolId is not equal to the value pointed to by the pointer current toolId. This result is true if they are not equal and false if they are equal.

```
if (executeSelect) { // Deselection of current tool, if any
if (toolId == 0) { // IF No new toolID is given
command += ("\nG1 Y" + S2 + " F1000\nM400\nG205 P0 S0 E0\nG204 P0 S0");
```

This is a conditional statement that checks the value of the boolean variable execute select. If this value is true, the code block enclosed within the curly braces {} will be executed. This implies that the code inside the block will be executed only when the tool ID is different from the current tool ID.

Within the first conditional block, there's another nested conditional statement. This one checks if toolId is equal to 0. If this condition is true, the code block within the curly braces will be executed.

There is a sequence of G-code commands used to act deselecting the tool from the spindle.

It seems to be a G-code command starting with "\n" (which is used to designate a new line). This G-code commands to move the machine to the Y-coordinate specified by the S2 (Y_point_A) variable at a feed rate of 1000mm/min. M400 G code command that waits for all previous movements to complete before proceedings.

G-code command with parameters "P0 S0 E0."commands the "P0" ATC magazine stepper motor to "S0" return to its homing position and "E0" home its position if it meets the end stop.

```
Tool::current = next; // now new tool is active
Com::printFLN(PSTR("NowSelectedTool:"), static_cast<int>(next->id));
```

In C++, when we use the scope resolution operator ::, it allows you to access static members of a class without needing an instance of the class. By doing Tool::current = next;, the static member variable current is updated to point to the same Tool object that next is pointing to. This means the current static member refers to the new active tool. As a static

member, its value is shared across all instances of the Tool class, so any changes made to it will be visible to all instances of the class. It's a way of designating the currently active tool that can be accessed globally within the class.

```
if (executeSelect) { // selection of new tool if toolId > 0
if (toolId > 0) { // put new tool to the spindle
strCommand = ("nG201 P0 X" + S7 + "\nM400\nG1 Y" + S3 + " F800\nG1 Z" + S4 + "
F200\nM400\nG1 Y" + S2 + " F1000\nM400\nG205 P0 S0 E0\nG204 P0 S0x0");
```

The code inside the curly braces {} will only be executed if the variable executeSelect is true. Inside the conditional block, there's a comment: // selection of new tool if toolId > 0. As mentioned earlier, executeSelect was initialized based on the comparison toolId != current->id, which means it will be true if toolId is different from the value pointed to by current->id.

There is a sequence of G-code commands used to act for selecting the new tool into the spindle. The G-code command appears to be "G201 P0 X" followed by the value of the S7 variable, which represents ATC_OFFSET + toolId. This commands the ATC magazine stepper motor to rotate to the specific tool ID position. Then it commands to move the machine to the Y-coordinate specified by the S3 (Y_point_B) variable at a feed rate of 800mm/min.

The machine moves to the Z-coordinate specified by the S4 (Z_point_1) variable at a feed rate of 200mm/min. Another Y-axis movement command, this time to the Y-coordinate specified by the S2 (Y_point_A) variable at a feed rate of 1000mm/min.

This G-code command with parameters "P0 S0 E0." commands the "P0" ATC magazine stepper motor to "S0" return to its homing position and "E0" home its position if it meets the end stop.

Condition 2: Change of tool, when there is another tool in the spindle

These conditions will be satisfied when there is a tool in the spindle and the selection of a new tool from the tool magazine. Tool deselection from the spindle would come first in the tool-changing cycle, and new tool selection from a tool magazine would come next.

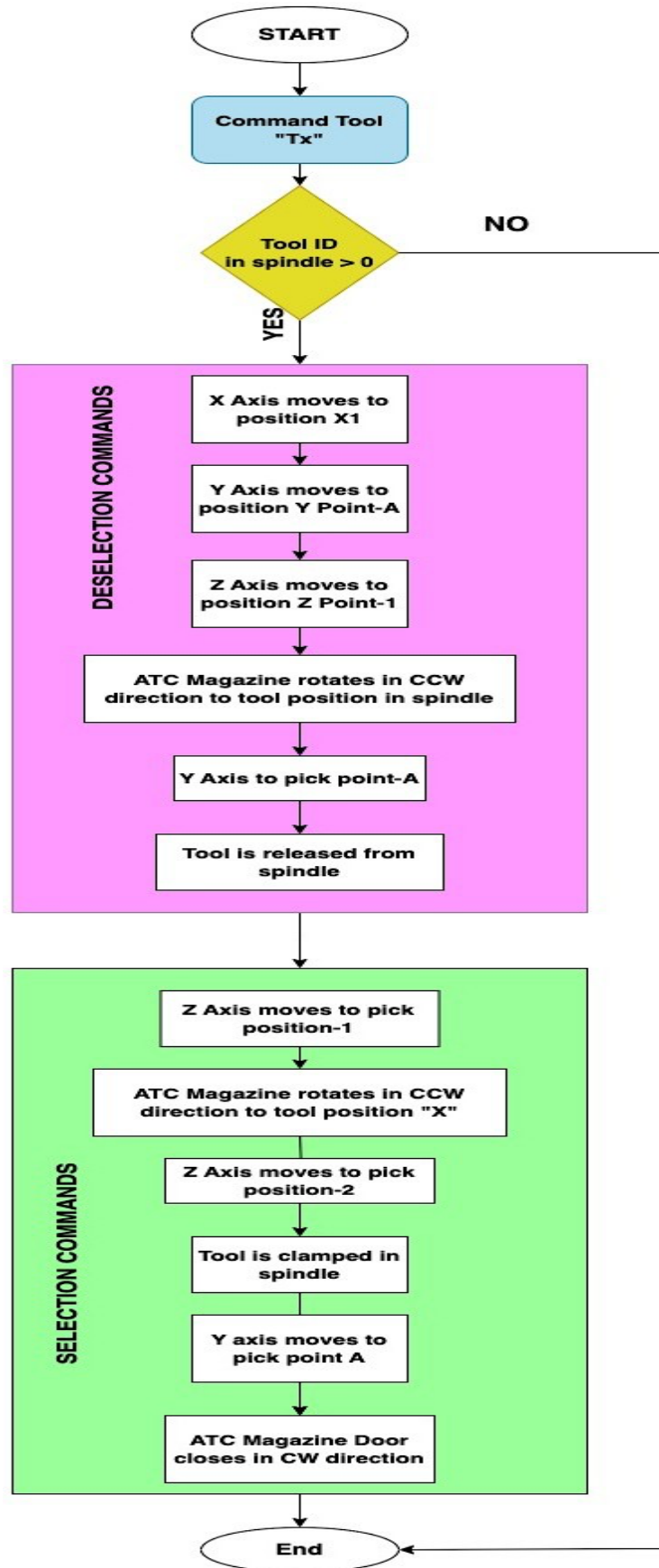


Figure 24: Flow chart for new tool change, with a tool in spindle [Source: Own]

From Figure 24, the process of changing a current tool in a spindle with a commanded tool is shown in the flow chart. Initially, "Tx" commands the tool. x is the Tool Id, which goes from 1 to 12. The Y axis travels to the pick point -A while the X axis moves to position -1, the Y axis parallel to the tool's axis in the ATC magazine. The tool clamping or releasing position is followed by the Z axis moving to position -2. The ATC magazine rotates in response to the present tool position in the spindle. To release the tool, the Y axis travels to the pick point-B position. The tool is released from the spindle to the ATC magazine. Z axis rises to position 1 as the position with a safe limit above the tool. The new commanded tool position "x" can be attained by rotating the ATC Magazine counterclockwise. To pick up the tool and clamp it on the spindle, the Z axis moves down to position -2. The Y axis returns to the pick point -A position when the tool is attached to the spindle. The ATC magazine door then closes counterclockwise.

```
bool executeSelect = toolId != current->id;
Com::printFLN(PSTR("SelectTool:"), static_cast<int>(toolId));
```

A Boolean variable executeSelect is declared and initialized. The comparison between the toolId! and current->id checks, if the value of variable toolId is not equal to the value, pointed to by the pointer current toolId. This result is true if they are not equal and false if they are equal.

```
if (executeSelect) { // Deselection of current tool, if any
if (current->id > 0) { // some tool is in the spindle
strCommand = ("M5\nG00 X" + S1 + " Y" + S2 + " Z" + S4 + "\nG205 P0 S0\nG201
P0 X" + S6 + "\nM400\nG1 Y" + S3 + " F800\nG1 Z" + S5 + " F1000");
```

if class conditional statement that checks the value of boolean variable execute select. If this value is true, the code inside the block will be executed only when the tool ID is different from the current tool ID.

current->id > 0 is a condition to check if the value stored in the "id" variable of the "current" structure is greater than 0. If this condition is true, it specified that there is some tool in the spindle, and the code inside the curly braces {} will be executed.

Then the deselection commands will be executed by the following G-code. The G-code commands with "G00 X" + S1 + " Y" + S2 + " Z" + S4 +" moves the machine to the X

FACULTY OF MECHANICAL ENGINEERING TUL

coordinate specified by S1 (X1 position), Y coordinates specified by the S2 (Y_point_A), Z coordinates with the S4 (Z_point_1) in rapid movement.

G-code with the parameters "G205 P0 S0" states that starting of the new line. G205 specifies the ATC stepper motor is selected and checks whether it is in its home position. Followed by the command "G201 P0 X" + S6 + " commands the ATC stepper motor to move to the "ATC_OFFSET + current->id" which is the current toolId in the spindle for the deselection of the tool.

Further, the Y axis moves to the coordinate S3 (Y_point_B) with a feed rate of 800 mm/min and the Z axis moves to the coordinate S5 (Z_point_2) which is specified for the deselection of the tool from the spindle with the feed rate of 1000mm/min.

```
Tool::current = next; // now new tool is active
Com::printFLN(PSTR("NowSelectedTool:"), static_cast<int>(next->id));
if (executeSelect) { // selection of new tool if toolId > 0
if (toolId > 0) { // put new tool to the spindle
strCommand = ("\nG201 P0 X" + S7 + "\nM400\nG1 Y" + S3 + " F800\nG1 Z" + S4
+ " F200\nM400\nG1 Y" + S2 + " F1000\nM400\nG205 P0 S0 E0\nG204 P0 S0\x0");
```

if class conditional statement that checks the value of boolean variable execute select.

toolId > 0 is a condition to check if the value stored in the "toolId" variable is greater than 0. If this condition is true, further G-code is executed for the selection of a new tool to the spindle with the tool.

The G-code commands the machine for the selection of a new tool.

Condition 3: The spindle has to be empty

This condition will be met when there is a tool in the spindle and release the tool to the tool magazine using the special command T0.

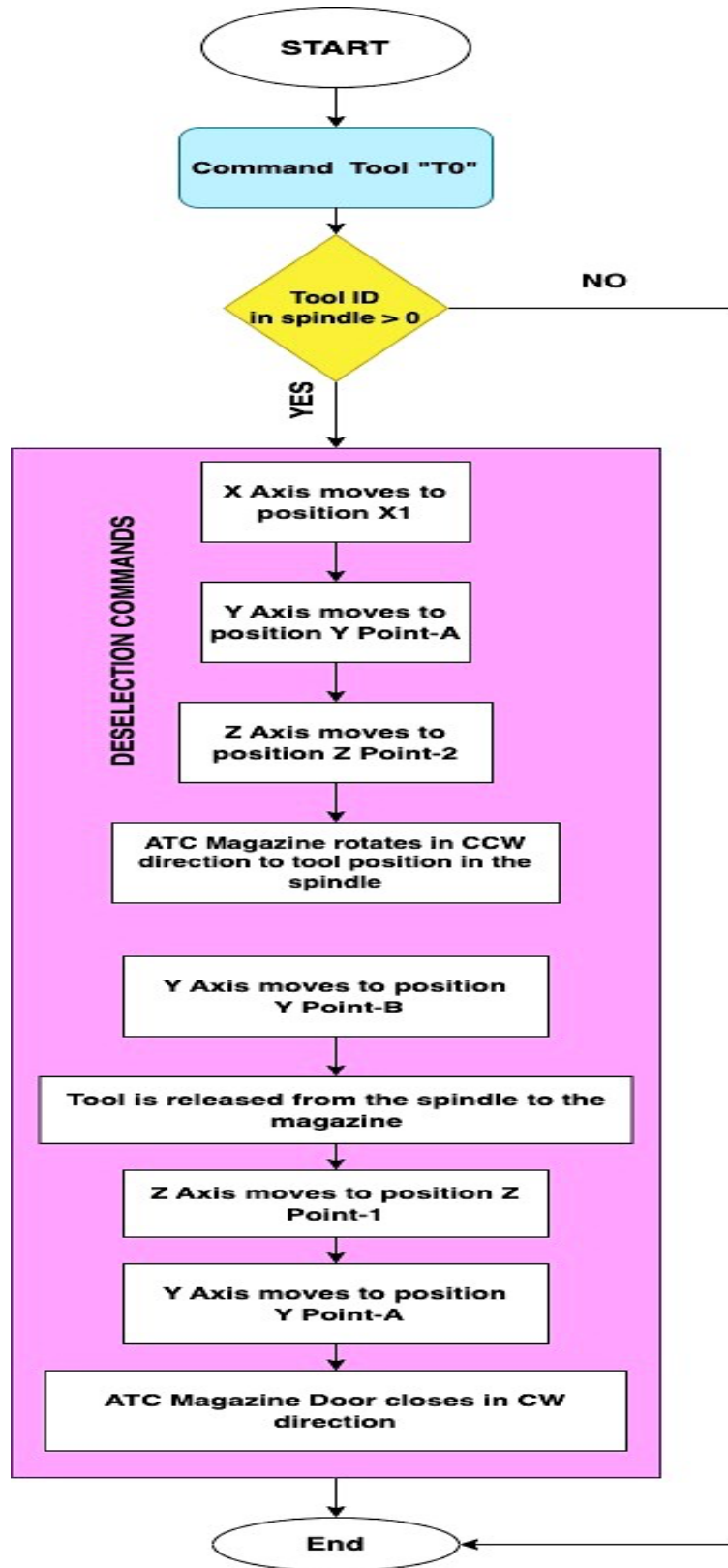


Figure 25: Flow chart of tool change with no tool in spindle [Source: Own]

From Figure 21, the process flow of the tool changing with the special command “T0” is shown. The tool changing command "T0" initiates the flow; it tells the spindle to release the tool and send it back with an empty spindle. The X-axis moves to position 1 when command T0 is provided, followed by the Y-axis moving to position A for the pick point, and the Z-axis moving to position 2 for clamping the tool position. To release the tool from the spindle, the ATC magazine rotates to the location of the currently-mounted tool. The tool is secured in the tool magazine and the Z axis travels up to position 1 when the spindle is directed to the position of the tool. As the Y axis approaches pick point 1, it leaves the magazine. ATC magazine doors close in a clockwise motion after Y axis movement.

```
bool executeSelect = toolId != current->id;
Com::printFLN(PSTR("SelectTool:"), static_cast<int>(toolId));
```

A Boolean variable executeSelect is declared and initialized. The comparison between the toolId! And current->id checks the value of variable toolId is greater than the value pointed to by the pointer current toolId. This result will be true if they are greater.

```
if (executeSelect) { // Deselection of the current tool, if any
if (current->id > 0) { // some tool is in the spindle
strCommand = ("M5\nG00 X" + S1 + " Y" + S2 + " Z" + S4 + "\nG205 P0 S0\nG201
P0 X" + S6 + "\nM400\nG1 Y" + S3 + " F800\nG1 Z" + S5 + " F1000");
```

if class conditional statement checks the value of the boolean variable executes select. If the value is true, the code inside the curly braces will execute only if there is a toolId is 0 which is different from the current toolId.

current->id > 0 is a condition to check if the value stored in the "id" variable of the "current" structure is greater than 0. If this condition is true, it specifies that there is some tool in the spindle, and the code inside the curly braces {} will be executed.

After the conditional check, the G-code command will be executed for the deselection of the tool from the spindle. The G-code with “G00 X” + S1 + “ Y” + S2 + “ Z” + S4 + “states that the machine moves in rapid movement to the X axis to the coordinate specified by the S1(X1 position), Y axis to the coordinate specified by the S2 (Y_point_A), Z axis to the coordinate specified by the S4 (Z_point_1).

G-code with parameter” G205 P0 S0” states that the ATC stepper motor is selected and checked for its homing position by the end stop limit switch. The ATC stepper motor is

commanded to move its position specified by S6 (“ATC_OFFSET + current->id”) position of the tool in the spindle. The machine waits for the commands to be executed. Further, the Y axis moves out of the ATC magazine to the position specified by the S3 (Y_point_B) with a feed rate of 800mm/min. Z axis moves to the position specified by S5 (Z_point_2) with a feed rate of 1000mm/min.

```
else { // spindle is empty
    strCommand = ("M5\nG00 X" + S1 + " Y" + S2 + " Z" + S5);
```

Further on completing the deselection commands of the tool from the spindle a conditional check is executed checking whether any new tool Id greater than 0 is given. If not the spindle will be empty. G-code with the parameter “G00 X” + S1 + “ Y” + S2 + “ Z” + S5” states that all three axes will be in a specified position.

5.4 Tool Setter

Tool setter is used to determine the length, radius/diameter of various tools used in the machine. This process reduces the downtime of tool settings during the machine operation. To establish the tool setter an optical-based end stop switch is fixed in the table of the machine. It's configured with the Ramp 1.4 shield as an end switch.

5.4.1 Hardware Selection

An optical end-stop switch can be employed to complete the tool setting operation. The infrared sensor is a U-shaped device that serves as the optical end switch. An IR emitter is located on one arm, and a light sensor is located on the other arm. The optical end switch operates by detecting the emitter's infrared light. The output remains high until an object passes through the laser beam of the sensor [19].

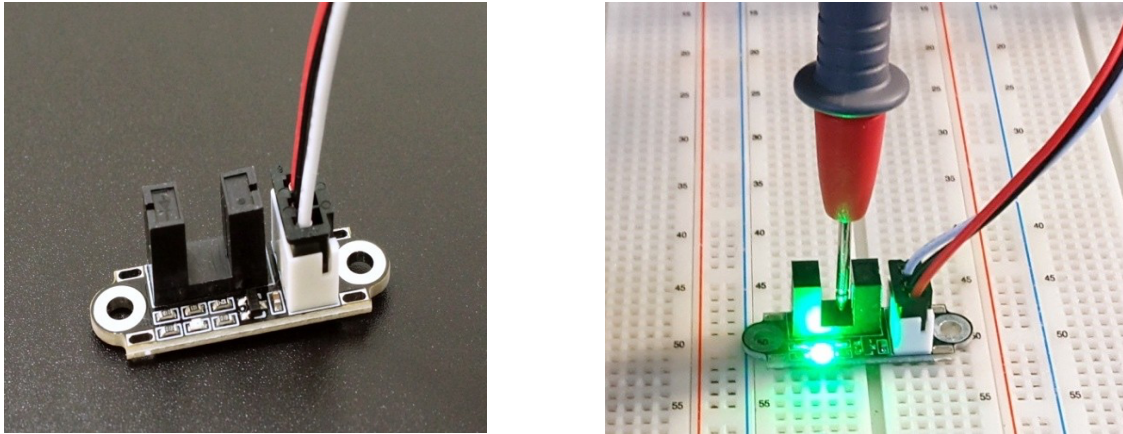


Figure 26: Optical End Switch Module [27]

5.4.2 Measurement of Tool Length

Initially, a pre-programmed tool reference program is executed to ensure the position of the laser with the reference tool with known length, radius/ diameter.

Figure 27, represents the tool length measurement with known dimension of the tool (h_1), tool holder length (h_2), and diameter of the tool (d) for referencing the position of the laser tool setter. The optical end switch is fixed to the table of the machine in predefined coordinates.

At a safe distance from the tool setter, the machine spindle carrying the reference tool approaches. The tool would then travel downward at the designated feed rate to prevent a tool-machine collision. The machine stops operating all axes when the tooltip breaks the laser beam. To ensure that the tool setter location and tool length is accurate, the machine's absolute coordinates are checked after being stored and cross-checked with the prior values. Once the tool's length and sensor's location have been checked.

The process of measuring the length of additional tools is continued.

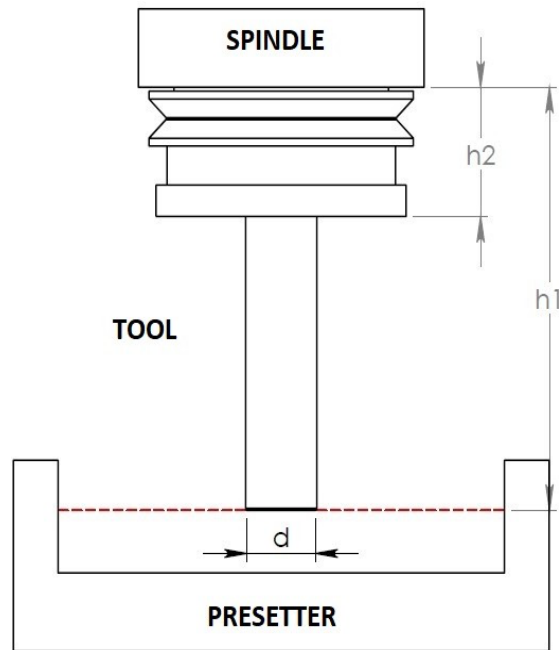


Figure 27: Tool setter Geometric parameter for length measurement [Source: Own]

when the procedure of tool referencing is performed. The machine then switches to a tool with a different length, radius, or diameter. Again, the same procedure is employed to measure tool length as for tool referencing.

From Figure 28, The functional part of a tool's tool length setting is represented by the following flow chart. When the tool length setting command is initialized the tool selection is executed. X, Y, and Z axes are moved to position A1 which is above the laser beam at a safety distance. Then the Z axis moves and descends towards the laser beam until it breaks the laser beam continuity. When the beam breaks all axes stop its action, and the position of the Z axis is stored. Then it is compared with the machine's absolute coordinate to know the length of the tool. When the tool length is known it is stored in the tool offset of the respective tool number.

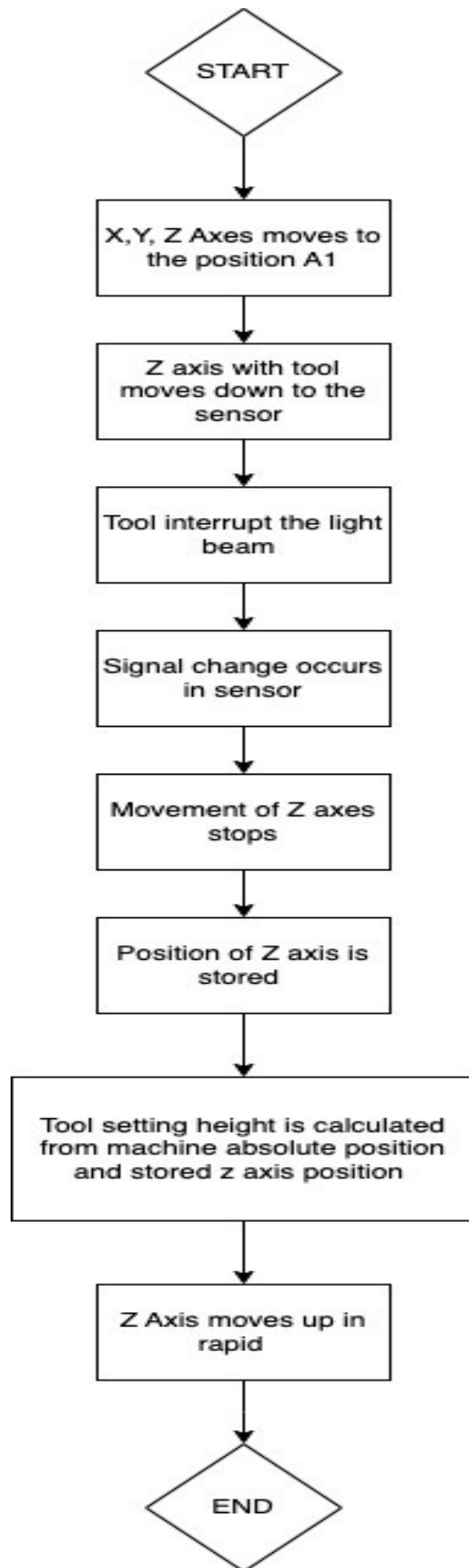


Figure 28: Flow chart of tool setting of tool length [Source: Own]

5.4.3 Measurement of Radius/ Diameter

The measurement of the Radius/ Diameter is similar to the length measurement of the tool. From Figure 29, Y1 and Y2 represent the coordinate machine where the tool where breaks the laser beam. From the coordinates, the Y1 and Y2 Radius or Diameter of the tool can be calculated with the reference of the laser beam reference position.

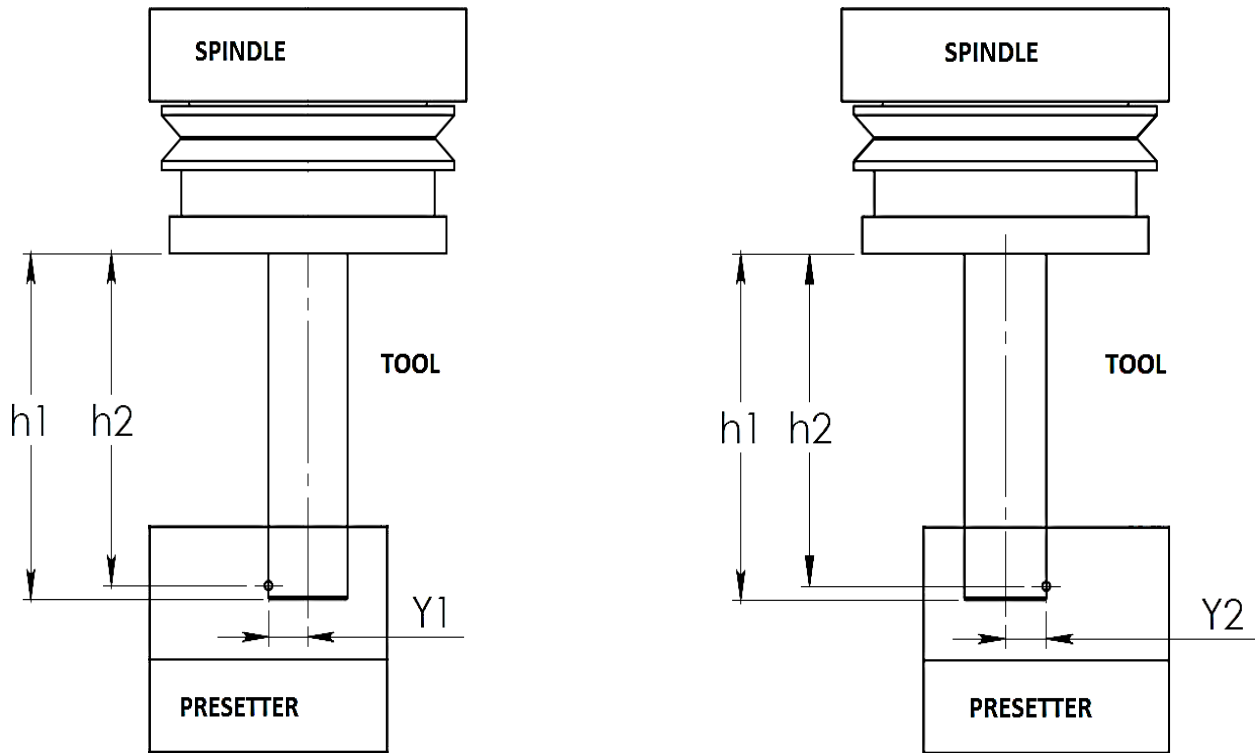


Figure 29: Tool Setter Geometric parameter for Radius/ Diameter of tool [Source: Own]

From Figure 30, The flow chart would be helpful to understand the functional flow of the radius/ Diameter measurement. When the tool Radius/ Diameter measurement program is initialized the X, Y and Z axes would move to position A2 which is 1mm below the laser beam with a safe limit on the Y axis. To measure the Y1 coordinate the tool would move from left to right direction with a prescribed feed rate until it breaks the laser beam to avoid collision with the tool-machine. When the laser beam breaks Y coordinate is stored. Then Z axis retards upwards in rapid movement. Then Y axis moves to the other side of the laser beam to measure the Y2 coordinate. When the Y1 and Y2 coordinates are measured, the dimension of Radius/ Diameter is computed and stored in the tool offset.

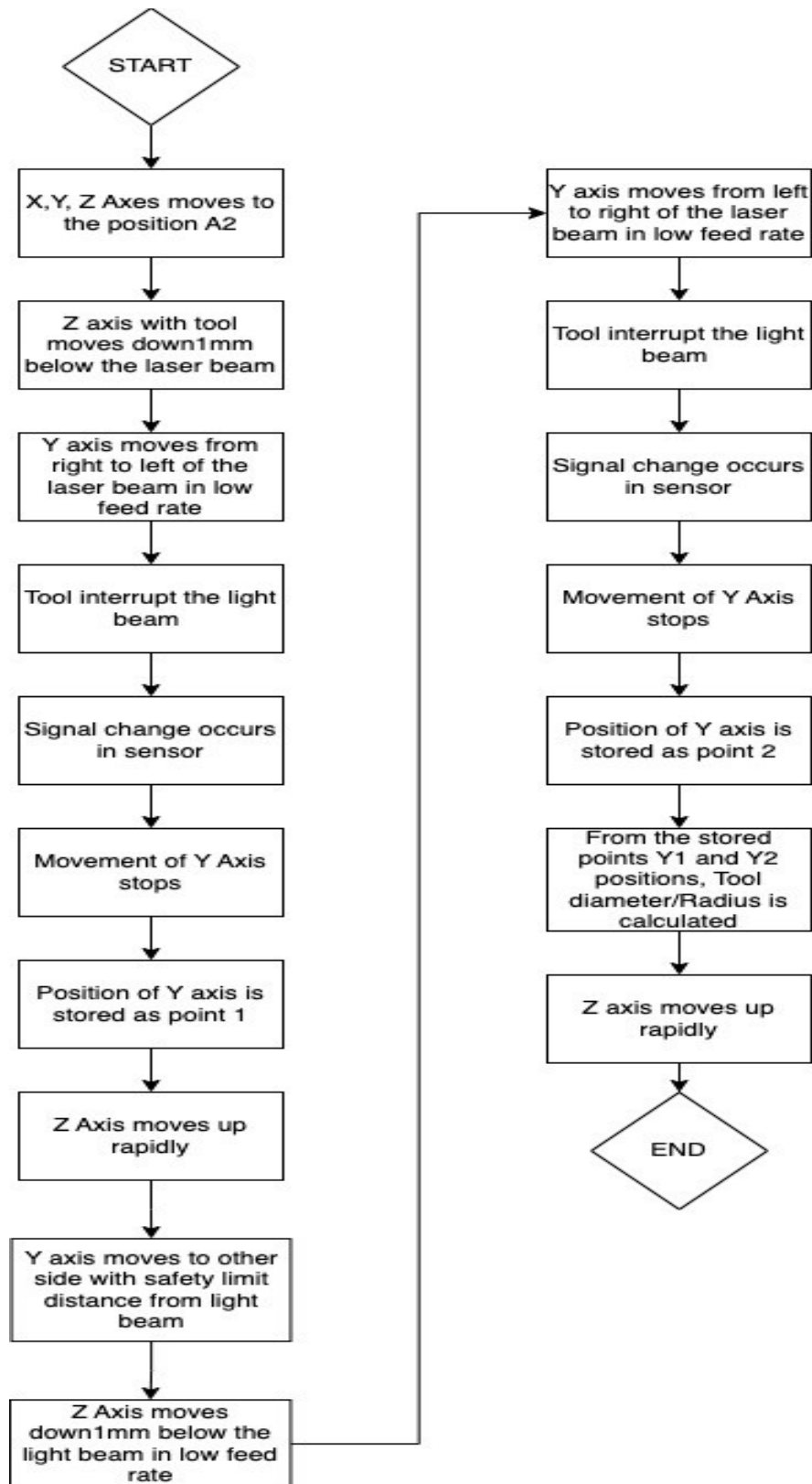


Figure 30: Flow chart for Radius/Diameter measurement in tool setter [Source: Own]

6 Conclusion

The development of a control system for a three-axis milling machine with an automatic tool changer is the primary objective of the thesis paper. Literature review, firmware selection, electronic platform selection, firmware development for axis movement, and an automated tool-changing mechanism are our primary areas of concentration here.

The difficulties and complexity involved in creating a control system for a three-axis milling machine with an automatic tool change have been discussed in this thesis.

The first section of the thesis begins with an extensive study of the architecture of CNC controllers that are currently available on the market. The OSEC architecture has been chosen as the best solution for the control system's development.

The study and analysis of the best available firmware for the machine will come after the first section. Repetier firmware is selected as the best option for this machine under investigation and review since it is an open-source platform, and source code could be downloaded and modified for our purposes. When compared to alternative firmware currently available on the market, Repetier firmware shows a variety of advantages and compatibility for this machine setup.

The Arduino Mega 2560 is selected as the machine's electronic microcontroller, and it is assembled with other electronic parts to create the machine model.

By using Repetier, which is configured and optimized for the needs of the machine, the algorithm for the machine's axes movement is developed. It was developed and tested to use the tool changer sequence. The integration of the algorithm with the Repetier firmware, the proper operation of the ATC mechanism, and the Design of the ATC magazine were some of the problems we encountered when developing and testing the algorithm of the ATC mechanism. An improved solution is provided for the ATC mechanism after addressing the problems encountered during the earlier stages of development.

Furthermore, the tool setter's implementation is idealized, and the process of developing the algorithm is being studied for further testing.

In future, when the firmware has been developed completely, with a tool setter, tool detection, and improvisation of the machine's memory of the tool on the spindle when the machine is turned on or off will be tested on the prototype of the machine model. An actual machine would use this firmware after being tested and improvised on the prototype.

7 Reference

- [1] TSENG, A A, S P KOLLURI and P RADHAKRISHNAN. A CNC Machining System for Education. *Journal of Manufacturing Systems*. no date, **8**(3).
- [2] *A Comprehensive Guide To 3 Axis CNC: Machine Specifications And Comparisons - Xin Tian Jian Precision Mfg Ltd 2023 January* [online]. [accessed. 2023-01-21]. Available at: <https://cncpartsxtj.com/cnc-machining/cnc-axis/3-axis-cnc/>
- [3] ASATO, O. L., E. R. R. KATO, R. Y. INAMASU and A. J. V. PORTO. Analysis of open CNC architecture for machine tools. *Journal of the Brazilian Society of Mechanical Sciences* [online]. 2002, **24**(3), 208–212. ISSN 0100-7386. Available at: doi:10.1590/S0100-73862002000300009
- [4] NACSA, János and Géza HAIDEGGER. BUILT-IN INTELLIGENT CONTROL APPLICATIONS OF OPEN CNCs. no date.
- [5] VENKATESH, S., B. SIDES, J. MICHALOSKI and F. PROCTOR. Case Study in the Challenges of Integrating CNC Production and Enterprise Systems. In: *ASME 2007 International Mechanical Engineering Congress and Exposition: Volume 3: Design and Manufacturing* [online]. Seattle, Washington, USA: ASMEDC, 2007, p. 207–213 [accessed. 2023-02-05]. ISBN 978-0-7918-4297-3. Available at: doi:10.1115/IMECE2007-43009
- [6] ALTINTAS, Y. and W.K. MUNASINGHE. A Hierarchical Open-Architecture CNC System for Machine Tools. *CIRP Annals* [online]. 1994, **43**(1), 349–354. ISSN 00078506. Available at: doi:10.1016/S0007-8506(07)62229-X
- [7] HASCOET, Jean-Yves and Matthieu RAUCH. Enabling Advanced CNC Programming with openNC Controllers for HSM Machines Tools. *High Speed Machining* [online]. 2016, **2**(1) [accessed. 2023-02-05]. ISSN 2299-3975. Available at: doi:10.1515/hsm-2016-0001
- [8] KHAN, Mr Amir, Aditya Kumar SHUKLA and Aashish SINGH. Design and Fabrication of 3-Axis Computer Numerical Control (CNC)Milling Machine. 2018, **6**(2).
- [9] *About Arduino | Arduino* [online]. [accessed. 2023-02-10]. Available at: <https://www.arduino.cc/en/about>
- [10] *Raspberry Pi - Wikipedia* [online]. [accessed. 2023-02-10]. Available at: https://en.wikipedia.org/wiki/Raspberry_Pi
- [11] *LinuxCNC User Introduction* [online]. [accessed. 2023-08-17]. Available at: <https://linuxcnc.org/docs/html/user/user-intro.html>
- [12] *Newfangled Solutions Mach4* [online]. [accessed. 2023-02-10]. Available at: <https://www.machsupport.com/software/mach4/>
- [13] *grbl/grbl* [online]. C. B.m.: Grbl CNC controller. 14. August 2023 [accessed. 2023-08-14]. Available at: <https://github.com/grbl/grbl>

- [14] MARLINFIRMWARE. Home. *Marlin Firmware* [online]. [accessed. 2023-08-14]. Available at: <https://marlinfw.org/>
- [15] <https://www.repetier.com/>. *Repetier Software* [online]. [accessed. 2023-08-14]. Available at: <https://www.repetier.com/>
- [16] *Maslow Firmware* [online]. [accessed. 2023-08-14]. Available at: <https://maslowcommunitygarden.org/Firmware.html>
- [17] *Overview - Klipper documentation* [online]. [accessed. 2023-08-14]. Available at: <https://www.klipper3d.org/Overview.html>
- [18] *File VIII (Lampiran).pdf* [online]. [accessed. 2023-02-06]. Available at: <http://eprints.polsri.ac.id/4598/8/File%20VIII%20%28Lampiran%29.pdf>
- [19] *Ramps1.4 - Geeetech Wiki* [online]. [accessed. 2023-02-17]. Available at: <https://www.geeetech.com/wiki/index.php/Ramps1.4>
- [20] *Pololu - DRV8825 Stepper Motor Driver Carrier, High Current* [online]. [accessed. 2023-02-08]. Available at: <https://www.pololu.com/product/2133/specs#note1>
- [21] How do brushed DC motors work? The need for regular maintenance explained | ASPINA. *ASPINA | Corporate brand of Shinano Kenshi* [online]. [accessed. 2023-08-14]. Available at: <https://www.aspina-group.com/en/learning-zone/columns/what-is/013/>
- [22] 1.3 inch OLED Display (SH1106, SPI/I2C, 128x64). *Smart Prototyping* [online]. [accessed. 2023-07-30]. Available at: https://www.smart-prototyping.com/1_3-inch-OLED-Display-SH1106-SPI-I2C-128-64
- [23] *Micro SD Card Adapter Module Pinout, Specifications, Datasheet, Working, Applications, Alternatives* [online]. [accessed. 2023-02-17]. Available at: <https://components101.com/modules/micro-sd-card-module-pinout-features-datasheet-alternatives>
- [24] OBREJA, Claudiu, Gheorghe STAN, Dragoş ANDRIOAIA and Marian FUNARU. Design of an Automatic Tool Changer System for Milling Machining Centers. *Applied Mechanics and Materials* [online]. 2013, **371**, 69–73. ISSN 1662-7482. Available at: doi:10.4028/www.scientific.net/AMM.371.69
- [25] *Brochure_Powerful_and_intuitive_machine_tool_probing_software.pdf*
- [26] Pin 9 - Arduino Mega 2560 - Using Arduino / Project Guidance. *Arduino Forum* [online]. 17. January 2019 [accessed. 2023-08-15]. Available at: <https://forum.arduino.cc/t/pin-9-arduino-mega-2560/568319>
- [27] Optical Endstop Module. *ProtoSupplies* [online]. [accessed. 2023-08-13]. Available at: <https://protosupplies.com/product/endstop-roller-lever-switch-module/>
- [28] JUNIOR, Milton Vieira, Elesandro Antonio BAPTISTA, Luciana ARAKI, Scott SMITH and Tony SCHMITZ. The role of tool presetting in milling stability

uncertainty. *Procedia Manufacturing* [online]. 2018, **26**, 164–172. ISSN 23519789.
Available at: doi:10.1016/j.promfg.2018.07.023

APPENDIX

Repetier Firmware – Can be downloaded from the website www.stag.tul.cz