
## Faculty of Electrical and Electronic Engineering Technology



# DEVELOPMENT OF G-CODE TO ROBOTSTUDIO RAPID TARGET WITH QUATERNION ORIENTATION FORMAT

**EZRASIEFRITZ JANTIN ANAK JAMES**

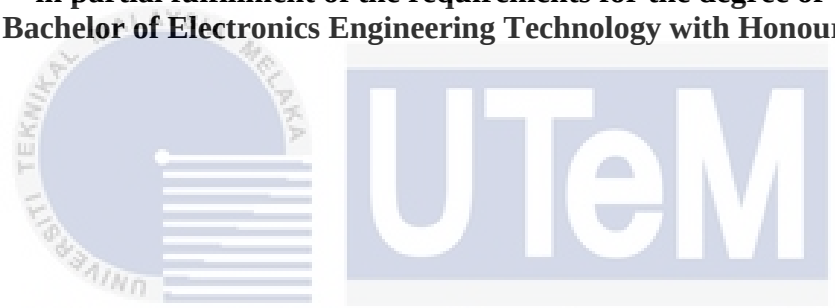
**Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours**

**2021**

# DEVELOPMENT OF G-CODE TO ROBOTSTUDIO RAPID TARGET WITH QUATERNION ORIENTATION FORMAT

**EZRASIEFRITZ JANTIN ANAK JAMES**

**A project report submitted**
**in partial fulfillment of the requirements for the degree of**
**Bachelor of Electronics Engineering Technology with Honours**



**Faculty of Electrical and Electronic Engineering Technology**

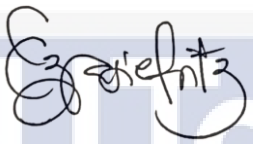


**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**2021**

**DECLARATION**

I declare that this project report entitled “Development of G-Code to RobotStudio RAPID Target with Quaternion Orientation Format” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

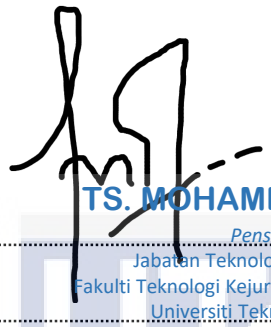
Signature : 

Student Name : EZRASIEFRITZ JANTIN ANAK JAMES

Date : 11 JANUARY 2022

**APPROVAL**

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature : 

Supervisor Name : TS. MOHAMED AZMI BIN SAID
*Pensyarah Kanan*
Jabatan Teknologi Kejuruteraan Elektrik
Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik
Universiti Teknikal Malaysia Melaka

Date : 11 JAN 2022

## *ABSTRACT*

*Every robot manufacturer has developed their own proprietary language since there are no standardizations for G-Code. Furthermore, to execute a complex milling required a competency of the operator in reading the instructions command and controlling the levers as it consists of thousand lines of command code. Hence, by develop an effective application for IRB 120 to be able to conduct milling process by using of Quaternion Orientation Format as it encodes information about an axis-angle rotation about an arbitrary axis. Based on previous version, the translation angle of Z-axis coordinates was fixed. Thus, by apply the Quaternion Orientation Format shall replacing the fixed Euler Orientation. Result show that by stimulate the Rapid Target with Quaternion Orientation Format provides a convenient mathematical notation for representing spatial orientations and rotations of elements in three-dimensional space in ABB RobotStudio.*

## *ABSTRAK*

*Setiap pengeluar robot telah membangunkan bahasa proprietari mereka sendiri kerana tiada penyeragaman untuk G-Code. Tambahan pula, untuk melaksanakan pengilangan kompleks memerlukan kecekapan pengendali dalam membaca arahan arahan dan mengawal tuas kerana ia terdiri daripada ribuan baris kod arahan. Oleh itu, dengan membangunkan aplikasi yang berkesan untuk IRB 120 untuk dapat menjalankan proses pengilangan dengan menggunakan Format Orientasi Kuaternion kerana ia mengekod maklumat tentang putaran sudut paksi tentang paksi sewenang-wenangnya. Berdasarkan versi sebelumnya, sudut terjemahan koordinat paksi-Z telah ditetapkan. Oleh itu, dengan mengaplikasikan Format Orientasi Kuaternion bagi menggantikan Orientasi Euler tetap. Keputusan menunjukkan bahawa dengan merangsang Sasaran Pantas dengan Format Orientasi Kuaternion menyediakan tatatanda matematik yang mudah untuk mewakili orientasi spatial dan putaran unsur dalam ruang tiga dimensi dalam ABB RobotStudio.*

**TABLE OF CONTENTS**

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **ABB** | - | Robot Company |
| **CAD** | - | Computer Aided Design |
| **CAM** | - | Computer Aided Manufacturing |
| **CIL** | - | Common Intermediate Language |
| **CLR** | - | Common Language Runtime |
| **CNC** | - | Computer Numerical Control |
| **C++** | - | Programming Language |
| **C#** | - | Programming Language |
| **DLL** | - | Dynamic Link Library |
| **FTP** | - | File Transfer Protocol |
| **G-Code** | - | Geometric Code |
| **GUI** | - | Graphic User Interface |
| **IRB** | - | Industrial Robot Arm |
| **JIT** | - | Just-In-Time |
| **KRL** | - | KUKA Robot Language |
| **MoveC** | - | Move Circular |
| **MoveJ** | - | Move Joint |
| **MoveL** | - | Move Link |
| **TCP** | - | Tool Center Point |

## LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Due to a lack of uniformity in converting G-Code to other robot programming languages, each industrial robot has developed its own programming language. As a result, a G-Code to ABB RobotStudio Rapid Target conversion tool is necessary to correctly execute the robot program.

The IRB 120 will be able to mill any work object in ABB RobotStudio using CAD G-Code generated by graphical design software, which will then be converted to Rapid Target code. The conversion application will solely serve as a third-party utility. To convert G-Code to Rapid Target Code, the user must first build a work object in SolidWorks software, which then generates a G-Code by using CARTIA software that can be imported into this application.



Furthermore, due to the constraints of contemporary CNC machines compared to milling robots, despite the fact that both have similar processes, milling robots have more flexibility and diversity of operations. As a result, adopting an industrial robot for the milling process would provide a lot more benefits in terms of producing higher-quality final products.

## 1.2 Problem Statement

CNC machines are commonly used in the milling process in the manufacturing business. In terms of prototype size, axis movement, and drill bit size, each CNC machine

has its unique set of limitations. For a certain CNC machine to execute a specific sort of work product, it requires a different size and type of CNC machine. The specific CNC was less appealing due to the necessity to complete a complicated operation involving several axes of orientation and a substantially greater cost. Furthermore, the more specialized the CNC Machine, the more difficult it is to set up.

An industrial robot arm can now replace a CNC machine in the production sector since industrial mechanical robots can perform numerous tasks such as picking and positioning things, painting, assembly work objects, welding, machining, and other industry needs. A machining industrial robot serves the same goal as a specialized CNC machine in that it can perform any dimension and sophisticated machining involving multiple axes of rotation.

## 1.3 Project Objective

The main aim of this project is to propose a systematic and effective third-party application to convert G-code to RobotStudio Rapid Target Code with multiple axis of orientation. Specifically, the objectives are as follows:

a) Develop an effective application for IRB 120 to perform a milling process by using mathematical algebra of Quaternion Orientation Format.

b) Apply the use of compact, efficient, and numerically stable rotation matrices of Quaternion Orientation Format by replacing fixed rotation of Euler Orientation Format for industrial robot milling's conversion software.

c) Test the application through RobotStudio Emulation by using Rapid Target Code along with Quaternion Orientation Format to perform multiple axis of orientation for milling's process.

## 1.4 Scope of Project

The scope of this project are as follows:

a) Graphical User Interface (GUI) of third-party conversion application will be developed by using Microsoft Visual Studio's software of Window Forms App (.NET Framework).

b) SolidWorks software will be used to design the work object and CARTIA will be used to generate the G-Code.

c) Only the milling procedure will be performed after converting generated G-Code to RobotStudio Rapid Target with Quaternion Orientation Format.

d) The user will manually enter the values for yaw (Rz), pitch (Ry), and roll (Rx), as well as the velocity, zone, work object, and tool, to determine the appropriate rotation angle and speed for the ABB IRB 120 to perform milling operation.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

An industrial robot arm is the most common robot in the manufacturing industry since it improves not only the speed but also the accuracy and precision of the manufacturing process. An industrial robot arm is a technical marvel since it is a combination of controller and software that resembles a human arm. Aside from that, milling is one of the most common uses for an industrial robot arm. Robotic milling's ability to manufacture high-quality complicated and unusual shapes was the cause for its replacement of CNC machines, as robot milling can do manufacturing and sculpting on a massive scale. Robot milling systems are designed with tooling and cutting system adaptability at a high level, allowing them to accommodate to various material removal processes.

Because there are no standardizations for G-Code, each robot maker has developed their own proprietary language. Furthermore, because a complicated milling comprises of thousands of lines of command code, the operator must be capable of deciphering the instructions and controlling the levers. As a result, third-party application software to translate G-Code to robot language is offered by every major robot manufacturer. As a result, creating a conversion from G-Code to RobotStudio Rapid Target Code with Quaternion Orientation Format ensures that robot milling operations and G-Code programming are as simple as possible.

This chapter will go over the existing G-Code converter solutions. As a result, the ABB RobotStudio, Microsoft Visual Studio, G-Code, Rapid Target Code, and C# programming languages used in development will be thoroughly described. Moreover, by

completing the development will aid in achieving the goal and resolving the problem that has plagued industrial manufacturing.

Aside from that, the objective of authoring this chapter is to serve as a reference or guide, as it will be based on research, articles, user manuals, theses, journals, and other types of published knowledge relevant to the project's theme.

## 2.2 Type of Programming Languages

A standardized programming language is still not used by robots today. Currently, each vendor is reliant on their own control system. The structure of the code between ABB and KUKA differs significantly. The user can specify a starting block in the KRL language for KUKA robots that determines several settings, such as the tool.

Furthermore, the tool is defined for each movement command in the Rapid Target code, thus this isn't necessary to define in the movement commands. Another distinction between these companies is that the KUKA robotics control system uses x, y, and z coordinates for translation and Roll, Pitch, and Yaw for defining rotational angles in degrees.

According to Bolmsjö, the roll, pitch, and yaw ways of representing rotation are the same movements that an airplane makes, where roll denotes a rotation along the hand's longitudinal axis, the pitch is a horizontal axis rotation that causes the end effector to rotate up and down while the yaw denotes a rotation along a vertical axis in relation to the hand longitudinal axis, causing the end effector to rotate from side to side.

Moreover, ABB robotics uses x, y, and z coordinates to determine translation and quaternions to define rotational angles[1].

### 2.2.1 G-Code Programming Language

G-Code is an operational language for CNC machining which tells numerically controlled lathes and machining centers on how to move tools in order to perform various cutting operations. It is a language that can be quite complex at times and would vary from machine to machine.

Moreover, G-Code come in group with each group can be replaced by another code in the same group. Hence, programmer be able to establish mode of operation. G-Code fits somewhere between conversational control, where an operator describes the part and how it should be machined, and CAM programs, where software develops the necessary toolpaths, feed rates that been needed for CNC machine to cut the material into end products[2].

Furthermore, G-Code is best to illustrate as a combination of geometric code and variation of the alpha numeric pattern such as, N (Line number), G(Motion), X (Horizontal position), Y (Vertical position), Z(Depth), F (Feed rate), S (Spindle speed), T (Tool selection, Miscellaneous function), I and J (Incremental center of an arc), R (Radius of an arc). While the alpha numeric codes are used for defining motion and function (G##), Declare a position (X## Y## Z##), Set a value (F## or S##), Select a tool (T##), Switching coolant on and off (M##)[3].



### 2.2.2 RobotStudio Rapid Target

Rapid Target is a robot programming language that been used to program ABB's industrial robots in RobotStudio that rely on three different types of motions[1]. First is, MoveL is a linear motion which forces the robot to move in a straight line between two points that been defined by the programmer.

Second type of motion is, MoveJ is a point-to-point motion that lets the robot move to the second point in the easiest way. Moreover, the benefit of letting the robot decide the

easiest way as it also capable to avoid collusion whenever the robot wanted to move to opposite side of the layout as it will create a circular motion around the base since it travel along a non-linear path[4]. Thus, MoveJ is often required in the beginning of the robot path in order to let the robot to configure itself without the constraint of following a specific path.

Third type of motion is MoveC where it is used to move the TCP circularly to a given destination. Furthermore, it will create an arc motion that doesn't need to be circular as the orientation normally remains unchanged relative to the circle[4].

### 2.2.3 C# Programming Language

C# is a modern object-oriented, general-purpose programming language, created and developed by Microsoft together with the .NET platform. It is simple yet powerful language primarily aimed at developer for creating various types of applications, for example web, windows, console applications, or other types of applications using Microsoft Visual Studio at a large scale[5].

Furthermore, C# is syntactically similar to Java and C++ as it's also a high-level language which consists of a set of definitions in classes that contain methods while the methods contain program logic (instructions that will be executed by computer)[6]. Moreover, C# programming language was not considered as a standalone product as is a part of Microsoft .NET framework which consists of an environment for development and execution of programs.

Microsoft .NET Framework is a software development framework for building and running an application such as website, providing a service, desktop apps on Windows. Other than that, two main components of .NET Framework are Class Library which acts as a provider to provide a set of APIs and types for common functionality such as strings, dates,

number, and Common Language Runtime (CLR) which acts as an execution engine for handling running applications[7].

In a nutshell, Microsoft .NET Framework applications are written in C# programming language code. The code been compiled into Common Intermediate Language (CIL) which then stored in assemblies file (.dll or exe file extension). When a user runs an application, Common Language Runtime CLR shall analyze the assembly's file and uses a JIT to turn into machine code to be executed on the specific architecture of the computer.

## 2.3 Grasshopper

Grasshopper is a software for Rhinoceros CAD system. Rhinoceros is a CAD software for surface modeling. Grasshopper is an add-in that enables the user to create parametric structures without the need of knowing how to write scripts. Instead, Grasshopper uses a visual scripting tool where the user can drag lines between different 8 blocks with different functions where the coding is embedded in the block so that the user doesn't need to write the code. The principle of the usage is the same as for many other visual scripting tools such as the, to engineers more commonly known software LabVIEW, and Simulink for Matlab[8].



## 2.4 HAL

HAL robot programming tool is an add-on framework for Grasshopper. HAL could be used to create robot paths with visualization directly in the CAD software. The software supports ABB IRC5 controllers which make it possible to simulate the robot motions. The premium version also has the possibility to directly create robot code in the HAL software that is executable on the robot controller. The software is also capable of importing G-code

and translate the code into paths. The software is mainly focusing on architectural applications[9].

## 2.5 ABB Machining Pac

The ABB machining PowerPac is ABB's own add-in software for ABB Robot Studio. The software makes it possible for the user to create machining paths by importing CAD geometries directly into Robot Studio. The software does support path generation from G-Code. It also features to act as a CAM software and calculate the tool path in the off-line programming software. The benefits of using ABB's own software are that the user is guaranteed that if the path is executable in the simulation, it will also be executable on the robot. A feature that other software's does not have. A drawback with the software is that it is closed source and it is therefore not possible for the user to customize the software for their specific needs[10].

## 2.6 Boot the Bot

This is a Java-based software that enables the user to import points and additional data and export valid RAPID code from Boot the Bot. Since the software is built in Java it is designed to have cross-platform support. The software is able to automatically calculate the best-suited configuration for the robot. It also allows for FTP (file transfer protocol) connection to the robot, which allows for remote upload of robot programs and remote robot control[11]. The software also gives the user a full simulation of the workflow in the program. It is designed to enable a more creative freedom for the designer and is primarily targeting users working as architectures and artists.

## 2.7 RISE from RoboMaster

A full offline programming software that is also able to import CAD geometries and create robot paths for multiple vendors of robots. The software has an in-built CAD engine that makes it possible to create and edit CAD models within the software. It has the capability to import CAM data from 2-5 axis machines and translate it into robot code. And it is able to handle configuration problems and singularities[12].

## 2.8 Types of Software

In the development for this project several software were considered to be used, due to it is a software development project. Microsoft Visual Studio serve as a major role in C# programming language for the conversion application to be achieved. Moreover, the design of the work object along with robot tool will be done in SolidWorks. Thus, the generated G-Code will be generated by using CARTIA. Furthermore, simulation will be done in ABB 10 RobotStudio along with the execution of Rapid Target code that been converted from generated G-Code.



### 2.8.1 RobotStudio 6.05

RobotStudio is an offline programming tool for the setup of ABB Robots which allow robot programmer to generate more accurate robot programs. It also capable to help the designers to visualize and determine the layouts before the robot been install. Furthermore, designers be able to perform a simulation based on a real time simulation and optimization without disturbing the production as it uses ABB Virtual Robot Technology.

Thus, the offline programming approached by ABB RobotStudio also display the best way to maximize return on investment on robot systems[1]. There are several benefits

of RobotStudio which are, Auto Reach feature that will automatically analyze reachability for user to verify and optimize the overall layout.

Next, Path Optimization which will automatically detect and warn about programs that targets can be improved to make the robot move in the most efficient way. Furthermore, the most valuable benefit using ABB RobotStudio is Virtual Flex Pendant feature which is a graphical representation of the real Flex Pendant which indicate that everything that could be done in Virtual Flex Pendant will be able to perform in real Flex Pendant[13].

## 2.9 Quaternion Orientation Format

Versors, or unit quaternions, are a useful mathematical notation for representing spatial orientations and rotations of items in three dimensions[14]. They encode information about an axis-angle rotation along an arbitrary axis in particular. Computer graphics, computer vision, robotics, navigation, molecular dynamics, flight dynamics, satellite orbital mechanics, and crystallographic texture analysis all use rotation and orientation quaternions[15].



Unit quaternions are also known as rotation quaternions because they represent the 3D rotation group when used to express rotation. They are called orientation quaternions or attitude quaternions when they are used to describe an orientation (rotation relative to a reference coordinate system). The quaternion (C,XS,YS,ZS) signifies a spatial rotation about a fixed point of theta radians along a unit axis (X,Y,Z) that denotes the Euler axis, where $C=\cos(\theta/2)$ and $S=\sin(\theta/2)$[16].

Quaternions are more compact, economical, and numerically stable than rotation matrices. They are easier to construct than Euler angles and avoid the problem of gimbal lock. They are not, however, as obvious, or as simple to comprehend as Euler angles. Furthermore, due to the periodic nature of sine and cosine, rotation angles that differ

precisely by the natural period will be encoded into identical quaternions, limiting the recoverable angles in radians to [0,2∏].

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

In order to achieve the project's objectives, flowchart method been used to illustrate the whole project's development as it also serves a purpose to monitor the project's development where it consists of three (3) stages which involved pre-development, development stage and post-development stage. These three (3) stages are essential for obtaining the desire outcome as it helps to guide the development progress. The detail of the project's process will also been cover in this chapter such as the process of generated G-Code from CARTIA, process of converting G-Code to RobotStudio Rapid Target, simulated robot's path in RobotStudio based on converted Rapid Target that been obtained from generated G-code and the development of C# in this project by using Microsoft Visual Studio.



## 3.2 Project Flowchart

The selected approach to best illustrated the execution project's development is flowchart as it is to ensure the project is well planned while act accordingly to proposed methodology. Furthermore, it also serves a purpose for a better reference for monitoring the progress of the project. Thus, the development process will be shown in detail where it consists of experimental setup for development of project structure and design, parameter for development of method, and equipment for development of the selected software.

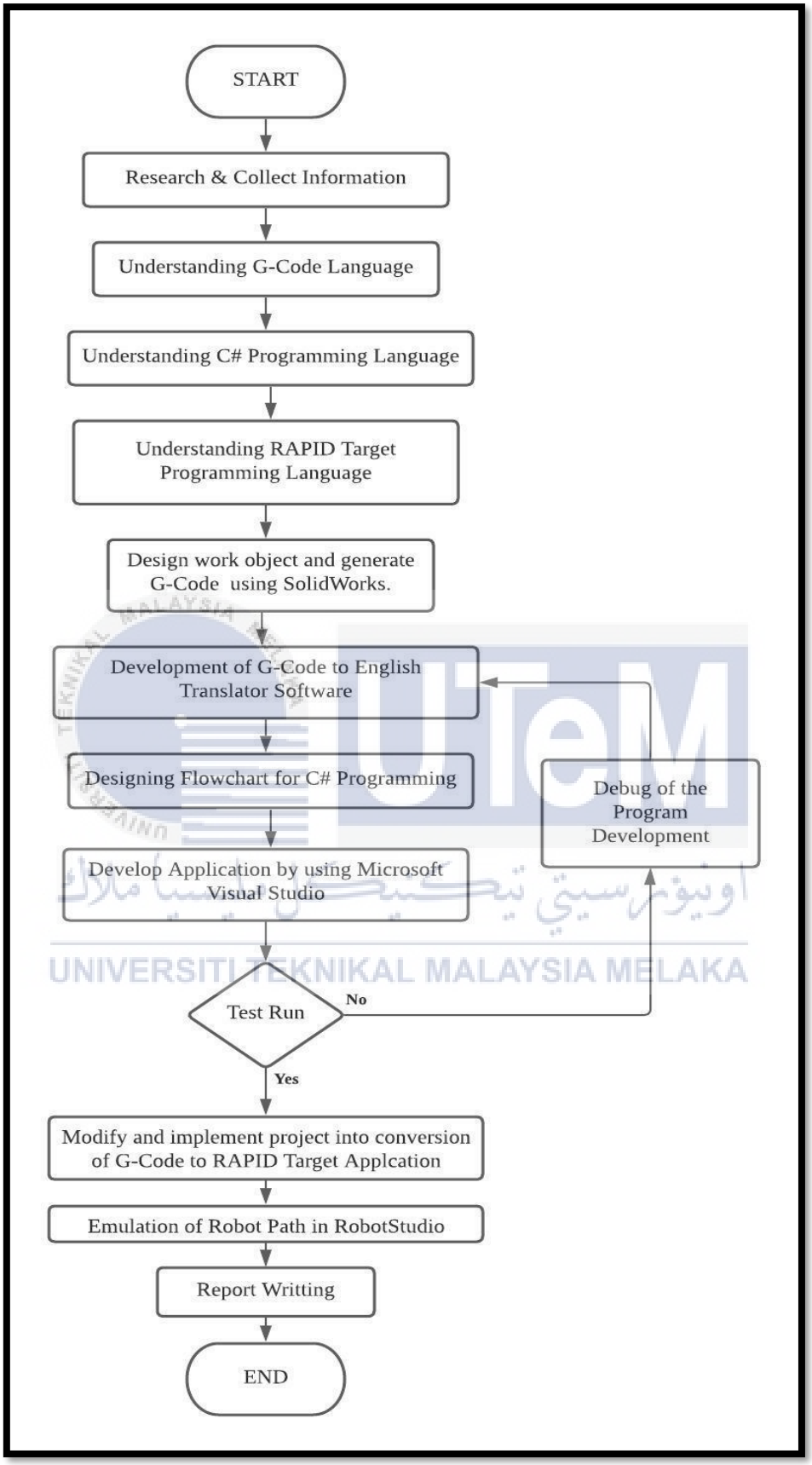



**Figure 3.1 Flowchart of project's progress**

### 3.2.1 Experimental setup

The structure and design of the project are critical because it is based on research, articles, and case studies, which need gathering material for a literature review as well as defining the problem and solution. Furthermore, brainstorming is essential to come up with the best answer to the problem and the project's functionality. As a result, creating a functional block diagram will make it easier to come up with a better solution to the problem.

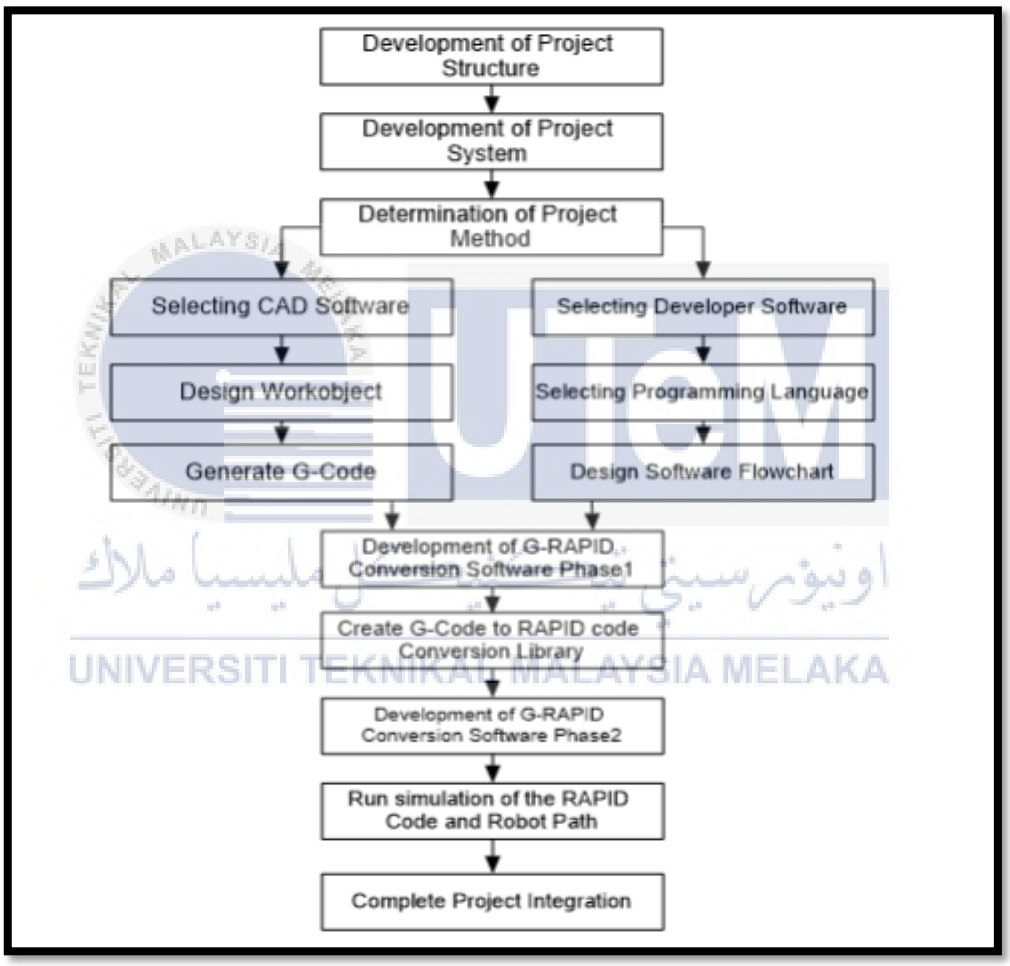



**Figure 3.2 Project Methodology's Flowchart**

## 3.3 Stage 1: Development of Project's structure

The creation of a project structure is a crucial aspect of the project. The framework of the project is based on research and data collection through a literature review, as well as

identifying the problem statement and solution. At this point, the solution on how to fix the problem with this project shall be identify. Designation of a functional block diagram for the project, as shown in Figure 3.3.

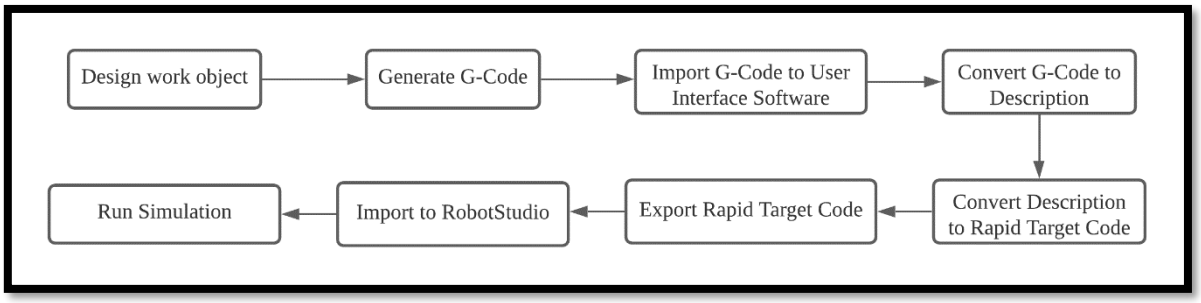



**Figure 3.3 Function Block Diagram**

## 3.4 Stage 2: Development of Project System

The development of the project system is a step in which brainstorming is required to address the problem statement and the project's function. Figure 3.3.1 depicts the software user interface design, which includes labelling and step numbers. The design, button function, labelling, and text box display are all part of the GUI user interface. A software user interface design is crucial in software development because it focuses on the appearance and style. The major goal of a software user interface design is to make it easy for users to use and pleasurable, and most importantly, it is user friendly.

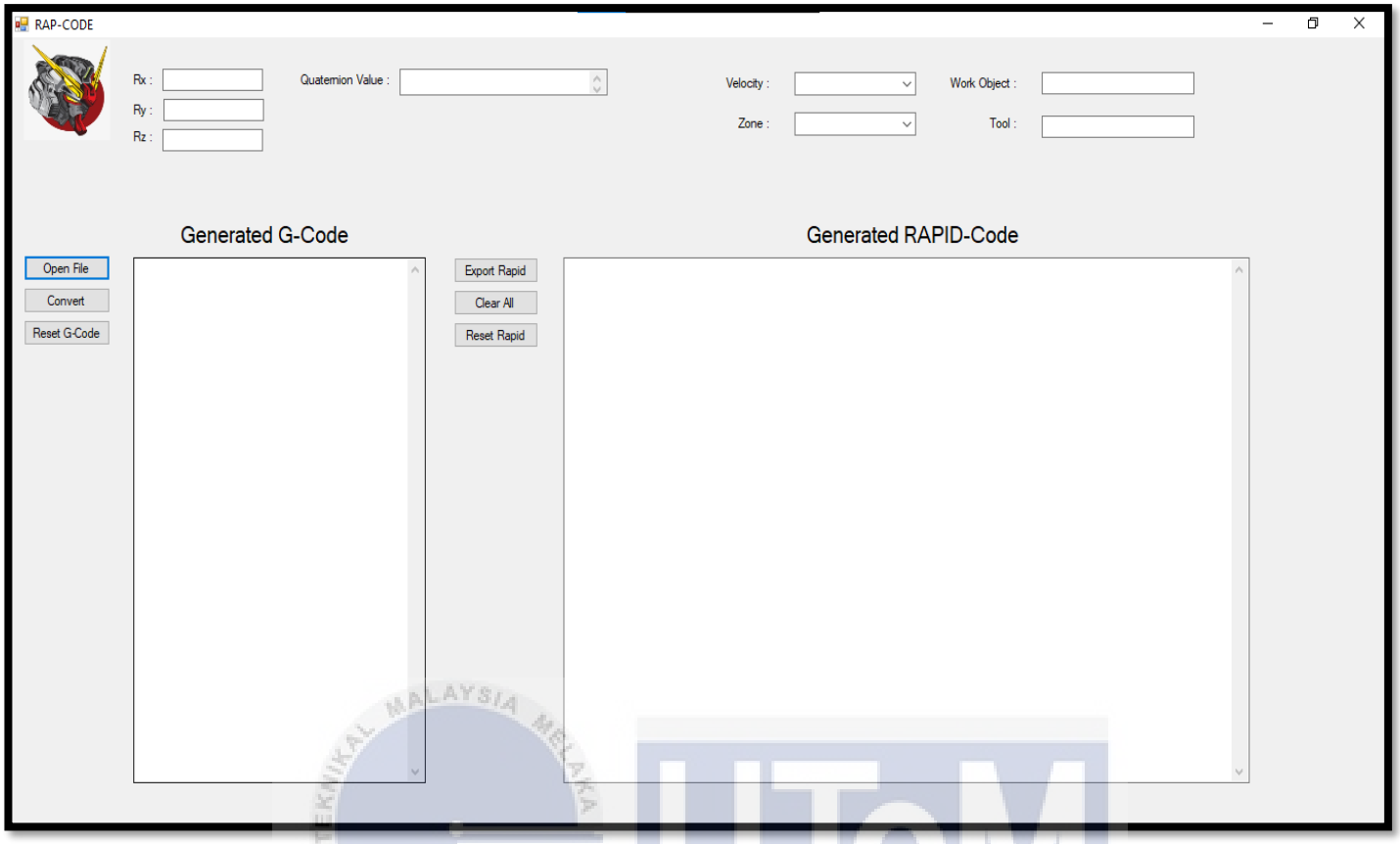



**Figure 3.4 RAP-CODE GUI Design**

## 3.5 Stage 3: Determination of Project method

The stage of determining the project method is to decide on the method that will be used to build this project. This stage will clarify what software and method will be used, as well as why it was chosen in the first place. SolidWorks and CARTIA, as well as Microsoft Visual Studio, were chosen for this project's development.

There are a few qualities to look for when picking CAD software for this project, such as efficient 3D design, user compatibility, and built-in application. SolidWorks and CARTIA have been chosen. Because it is user-friendly and has a parametric design modular function that allows users to alter and design with ease. Furthermore, CARTIA software be able to generate G-Code from CAD designs.

Work with SolidWorks to create a simple work object as the work item designed in SolidWorks while as for selecting Programming Language, the programming language utilized in this project is C# Programming Language because C# Programming requires only a rudimentary understanding of the C programming language.

With addition, C# Programming in Visual Studio supports Window Forms Applications (.NET Framework). The development for G-Code to RAPID Target Code's application is best suited as it requires only users with basic programming skills who want to create software.

Before beginning to write the programming coding, first, design the software flowchart. A software flowchart is used to reduce the number of bugs during the coding process. It's also known as a programming blueprint, and it's used to help programmers make their coding more efficient and operate as planned.

### 3.6 Stage 4: Development of Software

Microsoft Visual Studio 2021 is used to create G-code translation software, which is written in C# Programming Language by using the Window Form Application (.NET Framework). The RAP-CODE conversion software's first phase can only convert G-Code to RAPID Target Code without including Quaternion Orientation Format. As a result, at the conclusion of the development phase, it will be converted into the project's second phase, which will involve converting G-code to RAPID Target Code with Quaternion Orientation Format.

Thus, by designing the user interface in Microsoft Visual Studio's Window Form Application (.NET Framework) is the initial stage in developing the RAP-CODE conversion application. The first process that need to be fulfill first before converting G-Code to RAPID Target Code which is the process of inserting the desire information such as Rz, Ry, Rz,

Velocity, Zone, Work Object, and Tool text box. After all the required information been fulfill, then by clicking on Open file to open any file to browse G-Code in .txt. file format. Furthermore, after the desire G-Code file been chosen, the GUI shall then read the G-code and display on the first text box.

For the second text box, once the user clicks on convert button below the Open File button of the GUI, it will show the complete RAPID Target Code with Quaternion Orientation Format. Additionally, on the left side of the RapidTxtBox, there is a clear all button that acts as a reset or clear all the box, allowing the user to clear and start again with only one click.

Furthermore, the programming in C# is the second step in the development of G-code Translator software. The G-code will be converted in the future by referring to dll (Dynamic Link Library), which is a library that a program can utilize while running a function. In addition, the application makes use of the namespace "using System.Text.RegularExpressions."

Hence, the function allows the program to use the System and the.NET Framework regular expression engine of System.Text.RegularExpressions.dll. The method is called using "Regex gcode = new Regex ("[ngxyzftms] ngxyzftms ngxyzftm [+-]? [0-9] *.? [0-9]* [+-]?[0-9]* "RegexOptions.IgnoreCase, RegexOptions.IgnoreCase)".



### 3.7 Stage 5: Complete Project Integration

After the software development is complete, it will be released to a select group of beta testers. As a result, more users will be able to test this software and provide feedback on future improvements or issue detection.

## 3.8 Robot's path and RAPID Target Code for data collecting analysis

The Robot's path and RAPID Target code coordinates are the outcomes that must be analyzed. To make it easier for the developer to solve problems and predict consequences. Aside from that, the data must be transformed into a usable solution or approach for solving a problem. The major goal is to ensure that the outcome meets the goal and is capable of resolving the problem

In addition, the Robot's path will be analyzed depending on the robot's movement and behavior. By running an emulation in RobotStudio to obtain the robot path, the robot's coordinates could then be collected and compared to the coordinates in the SolidWorks simulation.

Finally, the results of the data collection and analysis will be discussed in the following chapter, Chapter 4: Results and Discussion.

## 3.9 Conclusion



The objective of methodology is to explain or expound on the methods used in the development of this project. Furthermore, methodology ensures that this project stays on track from start to finish.

The first step is the development of the project structure, the second stage is the development of the project system, the third stage is the determination of the project method, the fourth stage is the development of software, and the final stage is complete project integration. All steps of this project's development are described in detail, step by step.

The Robot Path and the RAPID code are also included in the technique. To verify that the outcome meets the goal and is capable of resolving the problem. The Robot Path and the RAPID Target code coordinate are the results that need to be analyzed in this project.

# CHAPTER 4

# RESULTS AND DISCUSSIONS

## 4.1 Introduction

The result of this project shall be shown in the result and discussion, which will go into great depth regarding the result. The robot's path and coordinates are simulated in RobotStudio with the RAPID Target Code that is converted using the conversion's application, and the result is based on the application's functionality, output, and the robot's path and coordinates that are converted using the conversion's application. The final version of RAP-CODE can convert G-Code into RAPID Target code, which translates G-Code into RobotStudio programming language, allowing the IRB 120 to perform milling operations. The user simply needs to provide Rz, Ry, Rx, velocity, zone, work object, and tool information's. Thus, the application will generate the RAPID Target Code automatically.



## 4.2 Project's functional

There are six (6) functional buttons, three (3) functional displays or text boxes, and seven (7) RAPID setting options in the RAP-CODE conversion application. To demonstrate the project's capabilities in greater depth, it will be divided into several sections. Before the conversion process, section 1 will provide the essential information as well as the result of the Open File button. Section 2 will demonstrate the results of the Convert to RAPID Target Code function, as well as some Gcodelibv1 coding. The function of the export, clear all, and exit buttons will be demonstrated, as in section 3.

### 4.2.1 Section 1

The Open File button acts as a file browser for.txt files containing G-Code from any directory on the computer. The function "`OpenFileDialog ofd = new OpenFileDialog();`" was used to create this button. The open file dialogue window appears once the Open File button is pressed. When the file selection is complete and OK is hit, the "`gcode_tbox.Text = File.ReadAllText(ofd.FileName);`" G-Code from the selected file will appear in the gcode.txt.box.

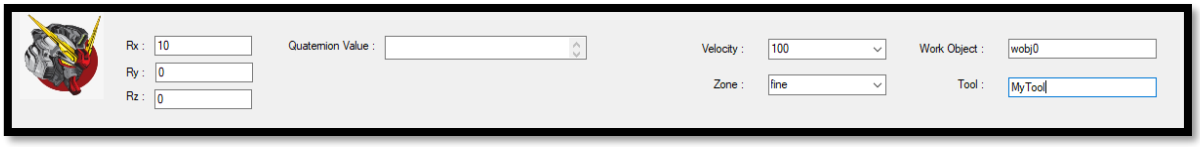



**Figure 4.1 Required Information to be fullfill**

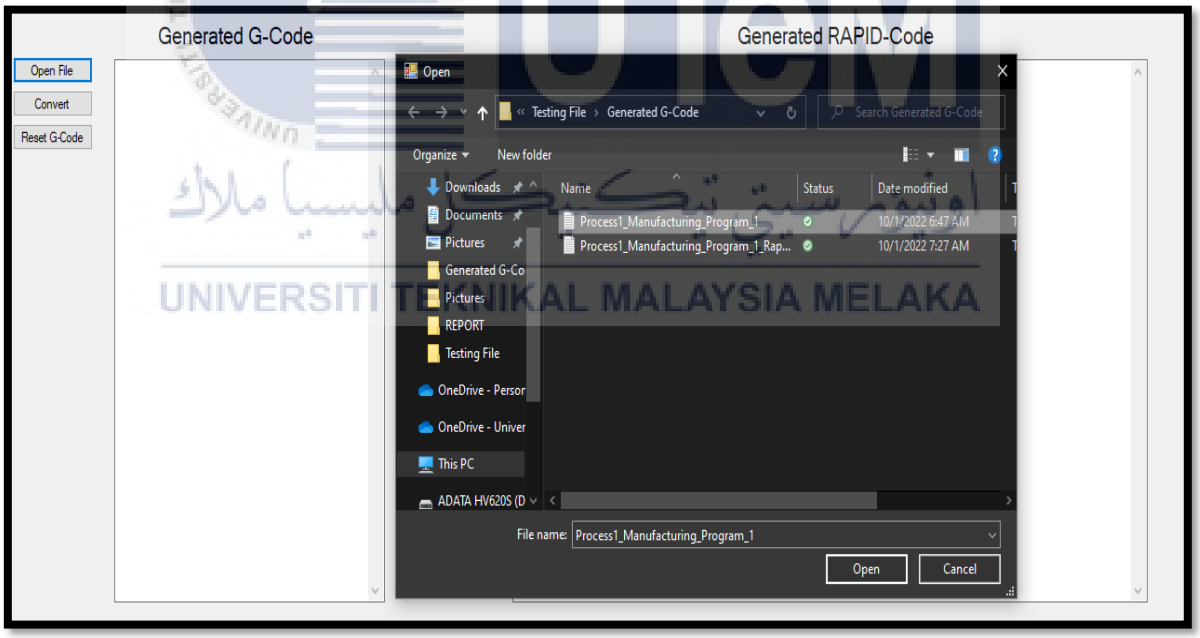



**Figure 4.2 Open File been press**

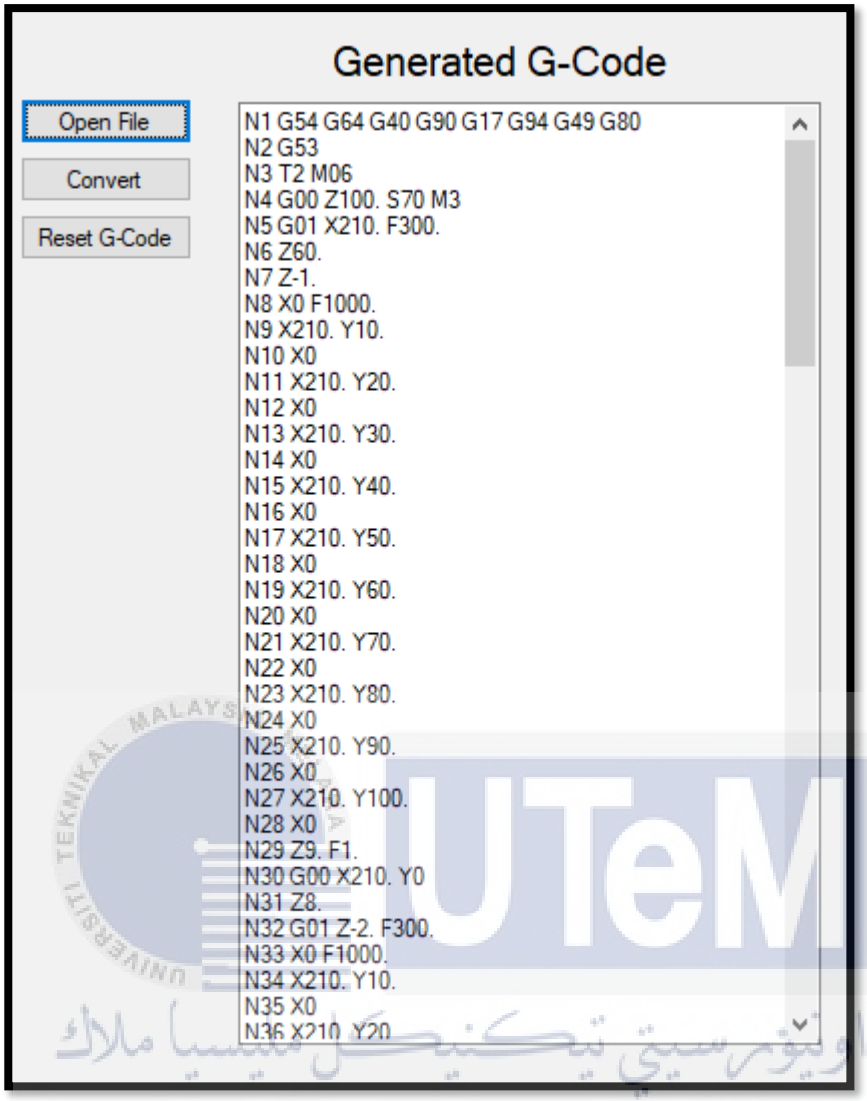




**Figure 4.3 G-Code Display in gcode.txt.box**

### 4.2.2 Section 2

Section 2 will display the outcome of the Convert button, as the application will convert G-Code to RAPID Target Code, including the Quaternion Orientation Format and a portion of the Gcodelibv1 coding. When the Convert button is clicked, the System programme function "`private void ConvertToRapid_Click(object sender, EventArgs e)`" is called "`using System.Text.RegularExpressions;`" will look for characters in every line that are mentioned in the "`string[] text_line = Regex.Split(this.gcode_tbox.Text, "\n");`". The programme will then read all of the text

and look for the value using “n” by referencing to the newly constructed GCodelibv1.dll, “`foreach (String n in text_line)`” is called. If "private void Reset G Code is clicked, the " `private void ResetG_Code_Click(object sender, EventArgs e)`" shall clear all of the text in gcode.txt.box which then allowing the user to start over.

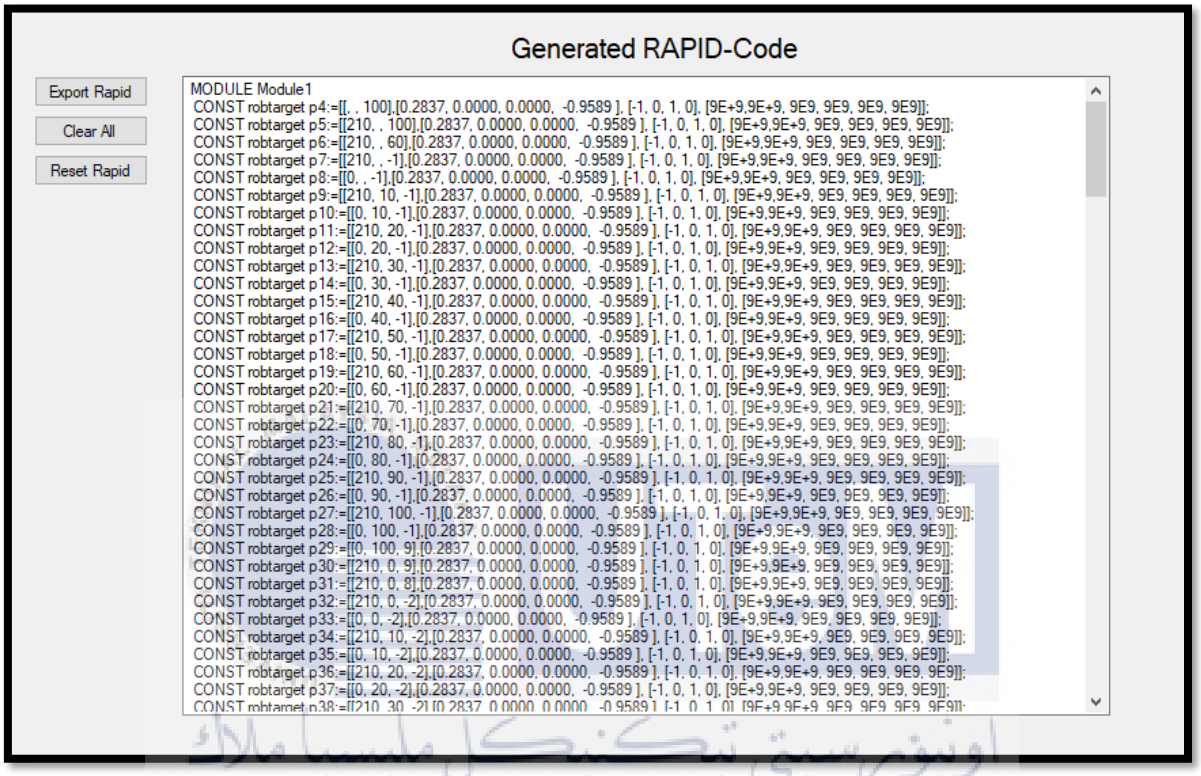





**Figure 4.4 Generated RAPID Code with Quaternion Orientation Format**

During the conversion process, after the Convert button is clicked, the application will read all lines of G-Code in the gcode.txt.box "`string[] text2_line = Regex.Split(this.gcode_tbox.Text, "\n");`" and split the line of G-code into few parts, for lines that contain G " `if (i.Contains('G')){G_cmd = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;}` ", line containing N "`if (i.Contains('N')){N = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;}`", line containing X “`if (i.Contains('X')){X = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;}`”, line containing Y "`if (i.Contains('Y')){Y = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;}`", and line

containing Z "`if (i.Contains('Z')){Z = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;}`". The software will then store the obtained information of G-Code, N, X-coordinate, Y-coordinate, and Z-coordinate into the local variable "N = Regex." [-+] Match(i, @"? [0-9]*\.?[0-9]+"). Value;".

### 4.2.3 Section 3

The operation and output of the Export Rapid, Clear All, and Reset Rapid buttons will be shown in section 3. To save the files option of the Export Rapid button, the function of "`private void ExportRapid_Click(object sender, EventArgs e){SaveFileDialog save = new SaveFileDialog(); save.Title = "Save File"; save.Filter = "Text Files (*.txt) | *.txt";`" shall be use to save once the user clicks OK. Hence, during the saving process, the function of "`if (save.ShowDialog() == System.Windows.Forms.DialogResult.OK){StreamWriter write =new StreamWriter(File.Create(save.FileName));`" to create a copy notepad format file " of RAPID Target Code from Rapid_Code_TxtBox. As for when Clear all button is pressed, the function that will be use "`private void ClearAll_Click(object sender, EventArgs e)`" as it will clear all of the text in "`{gcode_tbox.Clear();`" and "`Rapid_Code_TxtBox.Clear();}`".


**Figure 4.5 Exported .txt.file for Rapid Target Code**

## 4.3 Euler Rotation Angle into Quaternion Orientation Format.

Whenever the user clicked the Convert Button, the application will perform a mathematical calculation in order to produce the desire Quaternion Orientation value by the user. First, by the user need to insert the desire orientation value into either Rz which referring to Yaw, Ry which referring to Pitch, and Rx which referring to Roll. Once the user had entered the desire rotation value, the application will then display the Quaternion Value which then will be combine with RAPID Target Code so that the IRB 120 be able to perform orientation based on the calculated value of Quaternion.

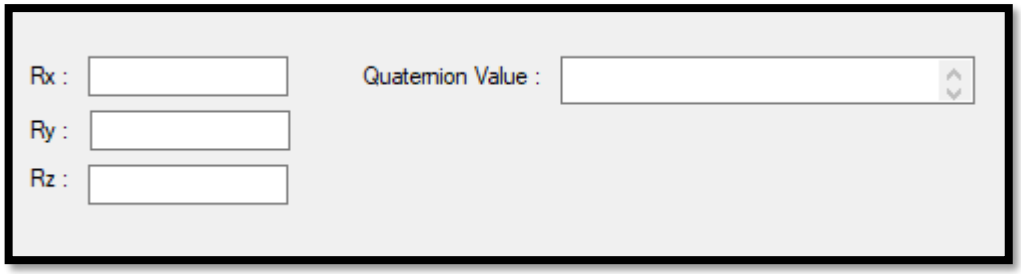



**Figure 4.6 Text Box of Rz, Ry, and Rx**

During the calculation process, the application will first convert all the Rz, Ry, and Rx value into "Double" as it used to define numeric variables holding numbers with decimal points by using the function of "`Ry = Convert.ToDouble(RyTxtBox.Text);`" for Ry (Pitch), "`Rz = Convert.ToDouble(RzTxtBox.Text);`" for Rz (Yaw), and "`Rx = Convert.ToDouble(RxTxtBox.Text);`" for Rx (Roll).

The next calculation process shall be by obtaining the value of Sine and Cosine time with 0.5 of Rz, Ry, and Rx by using the function of "`Cy = Math.Cos(Rz * 0.5);`", "`Sy = Math.Sin(Rz * 0.5);`", "`Cp = Math.Cos(Ry * 0.5);`", "`Sp = Math.Sin(Ry * 0.5);`", "`Cr = Math.Cos(Rx * 0.5);`", and "`Sr = Math.Sin(Rx * 0.5);`".

The final calculation's process involves the formula of converting Euler Rotation Value into Quaternion Orientation Format by using "`qw = Cr * Cp * Cy + Sr * Sp * Sy;`", "`qx = Sr * Cp * Cy - Cr * Sp * Sy;`", "`qy = Cr * Sp * Cy + Sr * Cp * Sy;`", and "`qz = Cr * Cp * Sy - Sr * Sp * Cy;`".



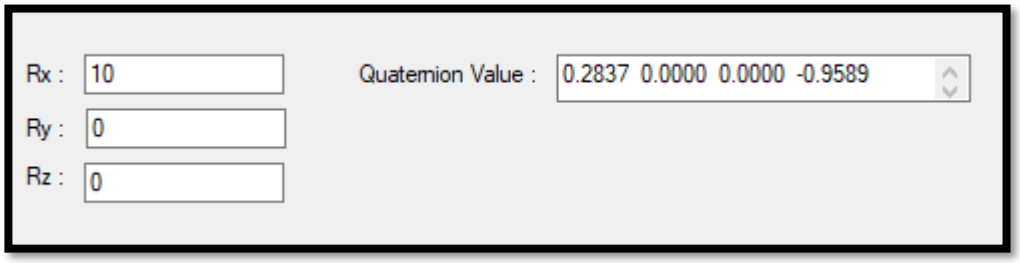



**Figure 4.7 Generated Quaternion Value**

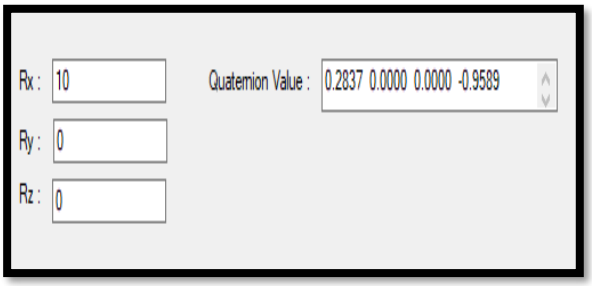
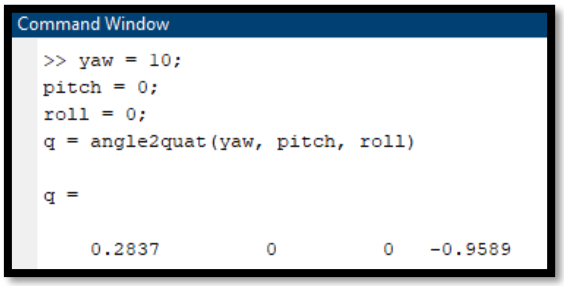
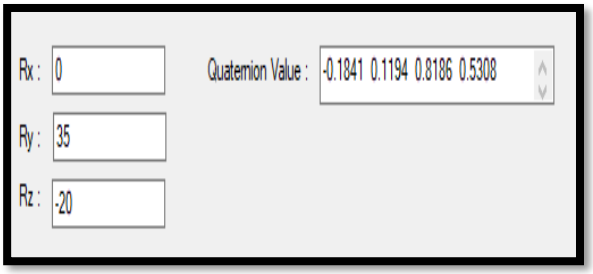
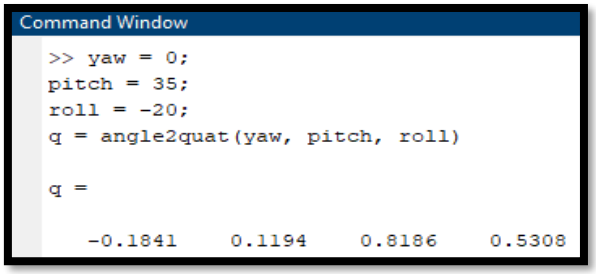
## 4.4 Results and Analysis

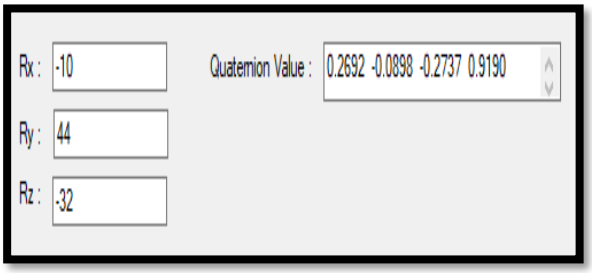
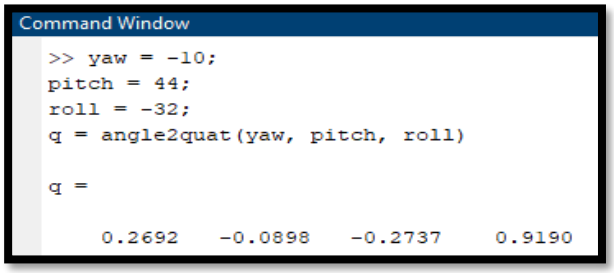
The data and information obtained from the output of this application will be presented in table and figure form and shall be discuss in this result and analysis. The G-Code description from the application output is collected by matching the G-Code with RAPID code coordinate syntax in RobotStudio emulation.

### 4.4.1 Analysis of RAP-CODE with Matlab of Quaternion Orientation Format.

The output result for the quaternion value will be compared between Matlab's software and the RAP-CODE's application in this part. The RAP-CODE application uses a formula to convert Euler Rotation to Quaternion, whereas Matlab's software uses the angle2quat function. To accomplish the conversion, both RAP-CODE and Matlab require the user to enter the values of Yaw, Pitch, and Roll. As for RAP-CODE, the Rz, Ry, and Rx will act as Yaw, Pitch, and Roll.

**Table 4.1 Comparision of Quaternion Value between RAP-CODE application with Matlab's software**

| Value | RAP-CODE Output | Matlab Output |
|---|---|---|
| Yaw = 10<br>Pitch = 0<br>Roll = 0 |  |  |
| Yaw = 0<br>Pitch = 35<br>Roll = -20 |  |  |

| | | |
|---|---|---|
| Yaw = -10<br>Pitch = 44<br>Roll = -32 |  |  |

### 4.4.2 Analysis of G-Code with RAPID Target code syntax along with the value of Quaternion Orientation Format.

The outcome is based on the converted RAPID code including Quaternion Orientation Format from the G-Code utilizing by using RAP-CODE's application as it is in this section. The G-Code coordinate value and the converted RAPID Target code coordinate will be compared in the data analysis. The const programmed contains the RAPID code coordinate. Furthermore, the N in G-Code which referring to line number will be compared to the P in RAPID Code, as the P in RAPID code will be the number of paths and will be written in the Proc Main ().

**Table 4.2 Comparison of coordinations between RAPID Target code with G-Code syntax**



| No. | Input (G-Code) | Result |
|---|---|---|
| 1. | N4 G00 Z100. S70 M3<br>N5 G01 X210. F300.<br>N6 Z60.<br>N7 Z-1.<br>N8 X0 F1000.<br>N9 X210. Y10. | Match |
| | **Output (RAPID Target code)** | |
| | MODULE Module1<br>CONST robtarget p4: = [[0,0,100], [0, -0.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p5: = [[210,0,100], [0, -0.283703561,0.958912034,0], [-1, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p6: = [[210,0,60], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]. | |

CONST robtarget p7: = [[210,0, -1], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].
CONST robtarget p8: = [[0,0, -1], [0, -0.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].
CONST robtarget p9: = [[210,10, -1], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].

```
PROC Main ()
    MoveJ p4, v100, fine, MyTool \Wobj: =wobj0.
    MoveJ p5, v100, fine, MyTool \Wobj: =wobj0.
    MoveJ p6, v100, fine, MyTool \Wobj: =wobj0.
    MoveJ p7, v100, fine, MyTool \Wobj: =wobj0.
    MoveJ p8, v100, fine, MyTool \Wobj: =wobj0.
    MoveJ p9, v100, fine, MyTool \Wobj: =wobj0.

ENDPROC
ENDMODULE
```

| 2. | **Input (G-Code)** | **Result** |
|---|---|---|
| | N10 X0<br>N11 X210. Y20.<br>N12 X0<br>N13 X210. Y30.<br>N14 X0<br>N15 X210. Y40. | **Match** |
| | **Output (RAPID Target code)** | |
| | MODULE Module1<br>CONST robtarget p10: = [[0,10, -1], [0, -.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p11: = [[210,20, -1], [0, -.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p12: = [[0,20, -1], [0, -.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p13: = [[210,30, -1], [0, -.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p14: = [[0,30, -1], [0, -.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p15: = [[210,40, -1], [0, -.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br><br>PROC Main ()<br>MoveJ p10, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p11, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p12, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p13, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p14, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p15, v100, fine, MyTool \Wobj: =wobj0.<br>ENDPROC<br>ENDMODULE | |

| 3. | Input (G-Code) | Result |
|---|---|---|
| | N16 X0<br>N17 X210. Y50.<br>N18 X0<br>N19 X210. Y60.<br>N20 X0<br>N21 X210. Y70. | Match |
| | **Output (RAPID Target code)** | |
| | MODULE Module1<br>CONST robtarget p16: = [[0,40, -1], [0, -0.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p17: = [[210,50, -1], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p18: = [[0,50, -1], [0, -0.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p19: = [[210,60, -1], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p20: = [[0,60, -1], [0, -0.283703561,0.958912034,0], [0, -1,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br>CONST robtarget p21: = [[210,70, -1], [0, -0.283703561,0.958912034,0], [-1, -2,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]].<br><br>PROC Main ()<br>MoveJ p16, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p17, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p18, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p19, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p20, v100, fine, MyTool \Wobj: =wobj0.<br>MoveJ p21, v100, fine, MyTool \Wobj: =wobj0.<br><br>ENDPROC<br>ENDMODULE | |

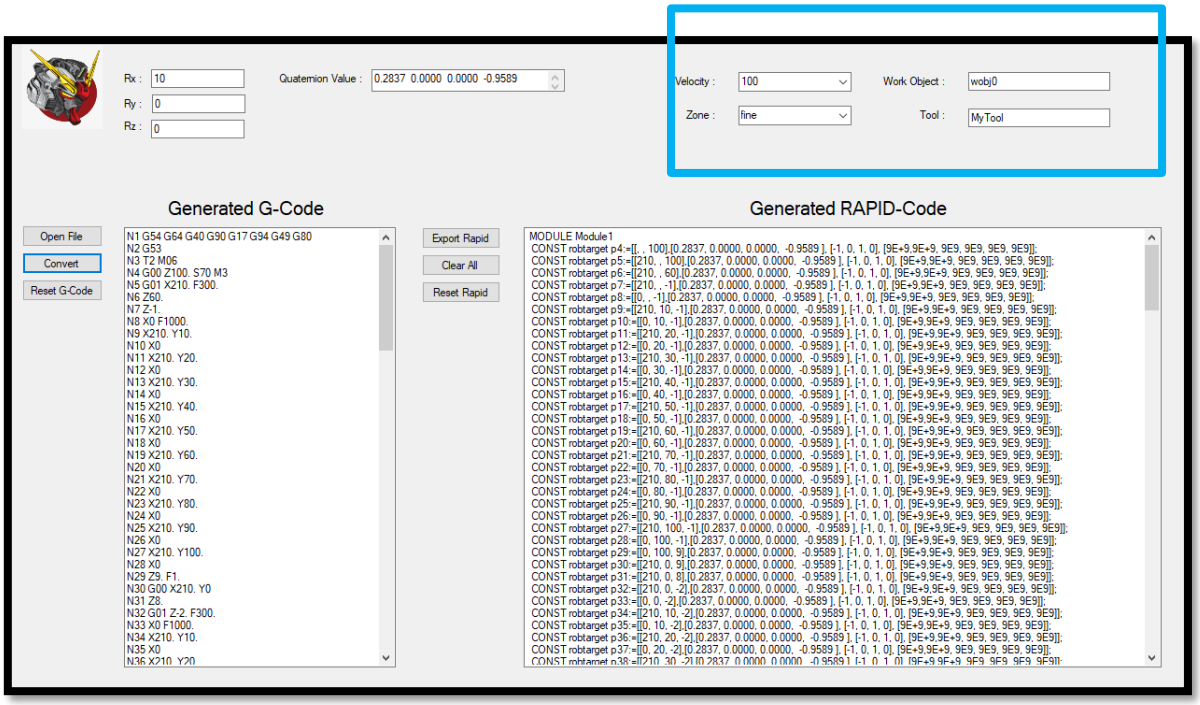



**Figure 4.8 Tool setting for RAPID Target Code in Blue Reactangle Area**


**Figure 4.9 Quaternion Value in Dark Yellow Rectangle Area**

### 4.4.3 RAP-CODE's application full process flow

The output in this section will show the entire process of converting G-Code to RAPID Target Code, as well as the tool path and simulation in RobotStudio. To begin, create a work object in SolidWorks and produce G-Code, then specify the Rz, Ry, Rx, velocity, zone, work object, and tool in RAP-CODE's application. Then convert the G-code to RAPID Target Code and save the RAPID Target code as a notepad file format. Finally, in RobotStudio, configure the robot station and tool, as well as the tool's coordinate system,

which is by default display inverted tool. After that, synchronize the station with RAPID, then import the generated RAPID Target code into RobotStudio and begin emulating it.

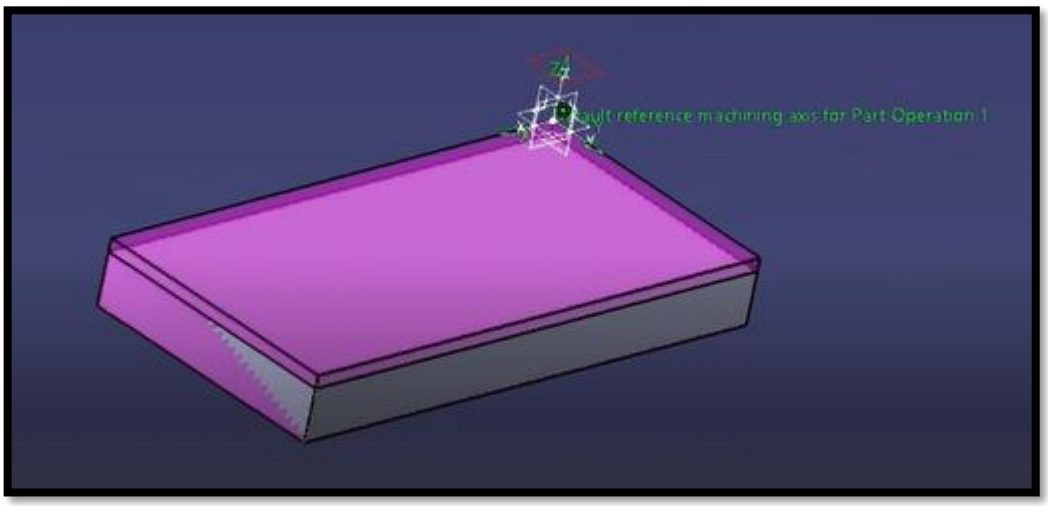



**Figure 4.10 Design Object at SolidWorks then generate G-Code by using CARTIA**

Process1_Manufacturing_Program_1 - Notepad

File Edit Format View Help

```
N1 G54 G64 G40 G90 G17 G94 G49 G80
N2 G53
N3 T2 M06
N4 G00 Z100. S70 M3
N5 G01 X210. F300.
N6 Z60.
N7 Z-1.
N8 X0 F1000.
N9 X210. Y10.
N10 X0
N11 X210. Y20.
N12 X0
N13 X210. Y30.
N14 X0
N15 X210. Y40.
N16 X0
N17 X210. Y50.
N18 X0
N19 X210. Y60.
N20 X0
N21 X210. Y70.
N22 X0
N23 X210. Y80.
N24 X0
N25 X210. Y90.
N26 X0
N27 X210. Y100.
N28 X0
N29 Z9. F1.
N30 G00 X210. Y0
N31 Z8.
N32 G01 Z-2. F300.
N33 X0 F1000.
N34 X210. Y10.
N35 X0
N36 X210. Y20.
```

**Figure 4.11 Save generated G-Code in notepad file format**

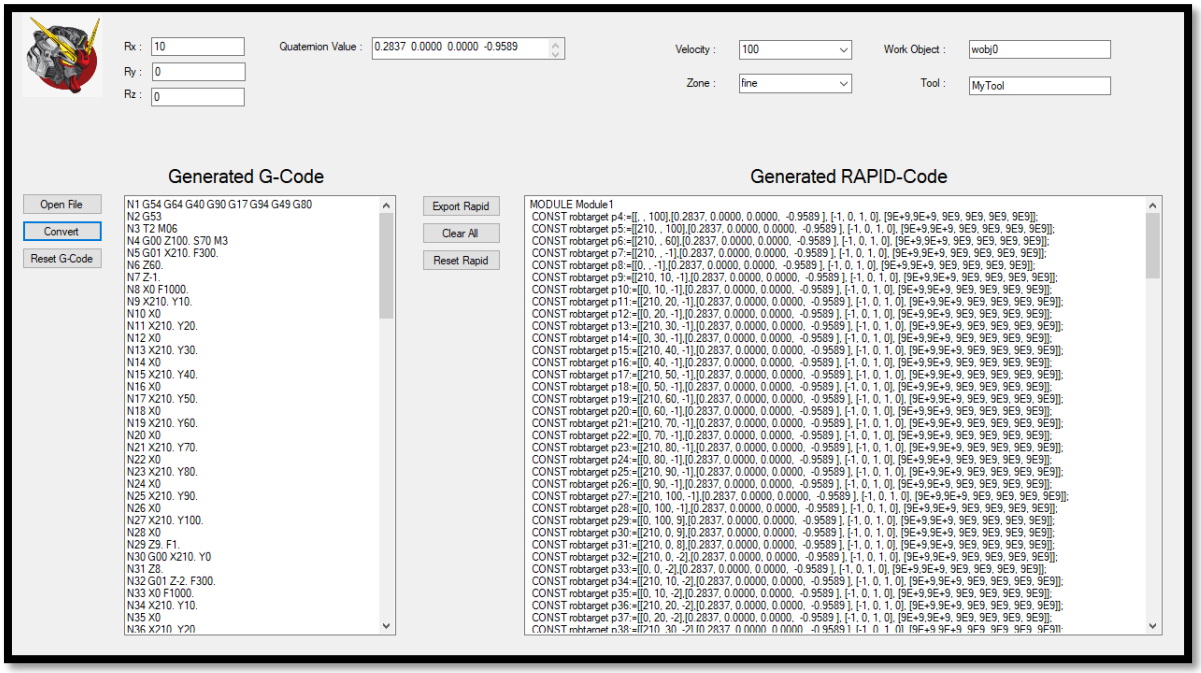



**Figure 4.12 G-Code to RAPID Code conversion by using RAP-CODE application**

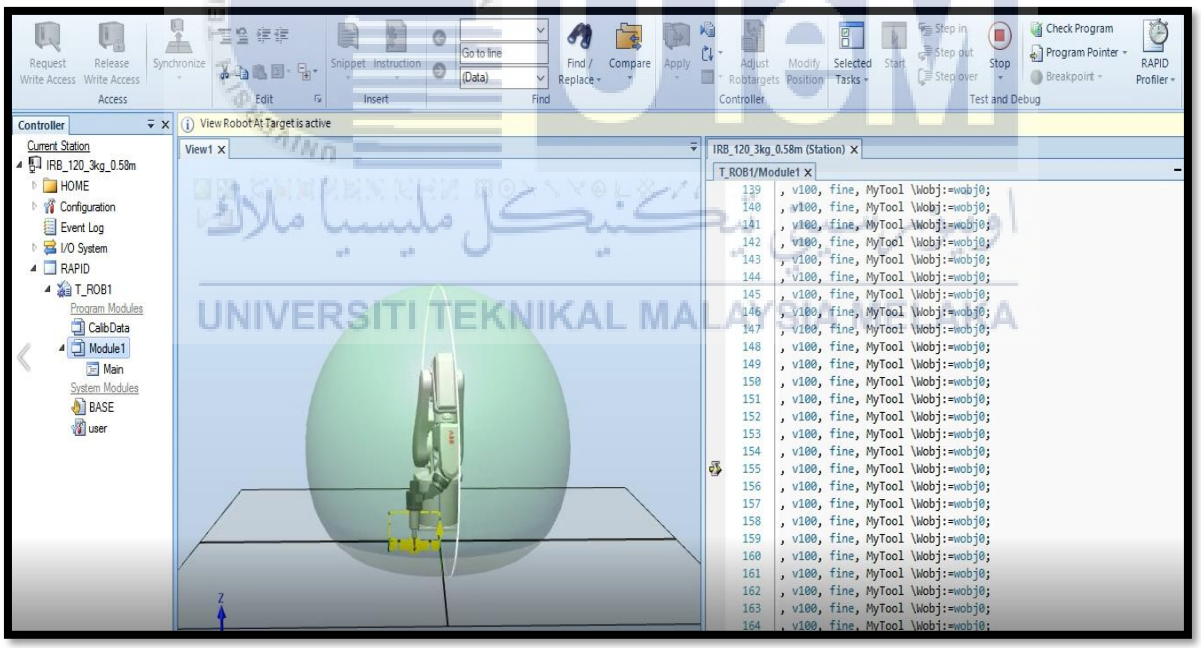



**Figure 4.13 IRB 120 perform milling's process**

## 4.5 Discussion

According to the output result, all of the buttons in the RAP-CODE conversion application are working properly, and the coordination syntax between G-Code and RAPID Target Code is checked by consulting the G-Code reference list and RAPID Target Code. Furthermore, the RAPID code that is transformed from G-code produces successful results. As shown in subtopic 4.3.1, the mathematical calculation for Quaternion value between RAP-CODE conversion application and Matlab software is same.

## 4.6 Summary

This chapter shows the output of the RAP-CODE conversion application, which includes button functionality and converted G-code. G-Code to RAPID Target Code with Quaternion Orientation Format can be successfully converted using the RAP-CODE conversion application. Also, based on a comparison of G-code and RAPID Target code syntax from the converted code, the G-code and RAPID code coordinate syntax are found to be identical. Following that, the mathematical calculation for Quaternion Value between RAP-CODE conversion application and Matlab software was successfully completed. Furthermore, the RAP-CODE conversion application functionality analysis demonstrates the whole simulation in SolidWorks to RobotStudio.

# CHAPTER 5

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

This project involves the creation of a G-code to RAPID Target code conversion application, which has been renamed RAP-CODE. The RAPID code is abbreviated as "RAP." The main goal of this application is to convert G-code to RAPID Target code to make it easier for users to convert G-code from a CAD design into RAPID Target code for usage in an ABB robot for milling because robots have several rotation axes and functions.

Multiple problems arise during software development due to a lack of C# programming language knowledge; however, as an electrical and electronic engineering student who has learned some basic C++ programming and has a background in mechatronic engineering for G-Code understanding, all of the problems can be overcome, and the project completed on time. Furthermore, the project necessitates the use of several resources as well as time. Despite the fact that this project has several limitations.



Overall, the program's development was successful in accomplishing the project's objectives, which were to design an effective application for IRB 120 to execute a milling operation using Quaternion Orientation Format mathematical algebra. IRB 120 be able to perform multiple rotation angles for milling process after using compact, efficient, and numerically stable Quaternion Orientation Format rotation matrices instead of fixed rotation of Euler Orientation Format for industrial robot milling's conversion software after testing the application through RobotStudio Emulation by using RAPID Target Code along with Quaternion Orientation Format.

## 5.2 Future Works

The development of this project application was successful, and it was able to meet the project's objectives. This software was able to convert G-Code to RAPID TARGET code successfully. Following that, more enhancements are required to improve the software's functionality as follows:

i) The MoveC should be used to convert the circular motion commands G02 and G03 to RAPID Target code.

ii) When converting is in progress, add a progress bar or indication to display the progress.

iii) More types of processes should be added to the library so that industrial robots can be used in a variety of operations.

The MoveC command, like the G-Code circular motion commands G02 and G03, is used to move the tool center point circularly to a chosen destination while keeping the orientation unchanged relative to the circle. To utilize the MoveC command in RAPID code, two positions or points of coordinate are required, thus the values of I and J in G-Code, which operate as the X and Y offset, will be transformed into the position or point for the coordinate to use in MoveC.

Furthermore, by adding a progress bar or indication be able to allow the user to be notified that the converting process is underway or if an error has occurred which resulting in the converting process failing. Last but not least, 3D printing commands, CNC turning, CNC engraving, and other CNC processes should all be added to the library so that industrial robots aren't limited to only one process.

## REFERENCES


[1] A. B. B. Robotics, “The world’s most used offline programming tool for robotics,” p. 2021, 2021.

[2] S. G-co, “An I nt r o du ction to G-Co d e an d C N C P r og r amming g ab d C u tte r r a d i us off s et acco u nt s fo r the geomet r y of the tool an d pe r mit s the p r og r amming of pa r t d imen s ion s fo r w hich the p r og r am d ete r mine s,” pp. 1–5, 2021.

[3] MachMotion, “M Code and G Code,” *Eng. Books Libr.*, pp. 66–80, 2017, [Online]. Available: https://engineeringbookslibrary.com/basic-g-and-m-codes/.

[4] ABB, “Technical reference manual, RAPID Instructions, Functions and Data types OLD,” 2018.

[5] V. Studio, *Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) by Svetlin Nakov & Co. http://www.introprogramming.info*. 2013.

[6] H. Automation, “Programming workbook,” 2000.

[7] Microsoft, “What is .NET Framework?,” *Microsoft*, pp. 6–9, 2021, [Online]. Available: https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework.

[8] “By Andrew Payne & Rajaa Issa.”

[9] “HAL_5_Brochure.pdf.” .

[10] “RobotStudio Machining PowerPac RobotStudio Machining PowerPac Introduction Features Benefits Requirements.”

[11] R. Dank, “Boot The Bot: Java-based Simulation, Code Generator and Live Controller for ABB Robots,” *J. Autom. Control Eng.*, vol. 1, no. 1, pp. 25–30, 2013,

doi: 10.12720/joace.1.1.25-30.

[12] “Robotmaster Interactive Simulation Environment ( RISE ).”

[13] L. Eitel, “RobotStudio — Software from ABB optimizes robot setups,” pp. 1–13, 2017, [Online]. Available: https://www.therobotreport.com/abb-robotstudio-software-optimizes-robot-setups-with-videos/.

[14] J. B. KUIPERS, “Quaternions and Rotation Sequences,” *Quaternions and Rotation Sequences*. 2020, doi: 10.2307/j.ctvx5wc3k.

[15] “Vehicle dynamics: theory and application,” *Choice Rev. Online*, vol. 51, no. 11, pp. 51-6199-51–6199, 2014, doi: 10.5860/choice.51-6199.

[16] M. Ben-Ari, “A Tutorial on Euler Angles and Quaternions,” pp. 1–22, 2014, [Online]. Available: http://www.weizmann.ac.il/sci-tea/benari/%0Ahttps://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf.

# APPENDICES

## Appendix A Generated G-Code

| | | |
|---|---|---|
| N1 G54 G64 G40 G90 G17 G94 G49 G80 | N45 X0 | N89 X0 |
| N2 G53 | N46 X210. Y70. | N90 X210. Y40. |
| N3 T2 M06 | N47 X0 | N91 X0 |
| N4 G00 Z100. S70 M3 | N48 X210. Y80. | N92 X210. Y50. |
| N5 G01 X210. F300. | N49 X0 | N93 X0 |
| N6 Z60. | N50 X210. Y90. | N94 X210. Y60. |
| N7 Z-1. | N51 X0 | N95 X0 |
| N8 X0 F1000. | N52 X210. Y100. | N96 X210. Y70. |
| N9 X210. Y10. | N53 X0 | N97 X0 |
| N10 X0 | N54 Z8. F1. | N98 X210. Y80. |
| N11 X210. Y20. | N55 G00 X210. Y0 | N99 X0 |
| N12 X0 | N56 Z7. | N100 X210. Y90. |
| N13 X210. Y30. | N57 G01 Z-3. F300. | N101 X0 |
| N14 X0 | N58 X0 F1000. | N102 X210. Y100. |
| N15 X210. Y40. | N59 X210. Y10. | N103 X0 |
| N16 X0 | N60 X0 | N104 Z6. F1. |
| N17 X210. Y50. | N61 X210. Y20. | N105 G00 X210. Y0 |
| N18 X0 | N62 X0 | N106 Z5. |
| N19 X210. Y60. | N63 X210. Y30. | N107 G01 Z-5. F300. |
| N20 X0 | N64 X0 | N108 X0 F1000. |
| N21 X210. Y70. | N65 X210. Y40. | N109 X210. Y10. |
| N22 X0 | N66 X0 | N110 X0 |
| N23 X210. Y80. | N67 X210. Y50. | N111 X210. Y20. |
| N24 X0 | N68 X0 | N112 X0 |
| N25 X210. Y90. | N69 X210. Y60. | N113 X210. Y30. |
| N26 X0 | N70 X0 | N114 X0 |
| N27 X210. Y100. | N71 X210. Y70. | N115 X210. Y40. |
| N28 X0 | N72 X0 | N116 X0 |
| N29 Z9. F1. | N73 X210. Y80. | N117 X210. Y50. |
| N30 G00 X210. Y0 | N74 X0 | N118 X0 |
| N31 Z8. | N75 X210. Y90. | N119 X210. Y60. |
| N32 G01 Z-2. F300. | N76 X0 | N120 X0 |
| N33 X0 F1000. | N77 X210. Y100. | N121 X210. Y70. |
| N34 X210. Y10. | N78 X0 | N122 X0 |
| N35 X0 | N79 Z7. F1. | N123 X210. Y80. |
| N36 X210. Y20. | N80 G00 X210. Y0 | N124 X0 |
| N37 X0 | N81 Z6. | N125 X210. Y90. |
| N38 X210. Y30. | N82 G1 Z-4. F300. | N126 X0 |
| N39 X0 | N83 X0 F1000. | N127 X210. Y100. |
| N40 X210. Y40. | N84 X210. Y10. | N128 X0 |
| N41 X0 | N85 X0 | N129 Z20. F1. |
| N42 X210. Y50. | N86 X210. Y20. | N130 Z100. |
| N43 X0 | N87 X0 | N131 Y0 |
| N44 X210. Y60. | N88 X210. Y30. | N132 M30 |

# Appendix B Generated RAPID Target Code

```
MODULE Module1
  CONST robtarget p4:=[[0,0,100],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p5:=[[210,0,100],[0,-0.283703561,0.958912034,0],[-1,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p6:=[[210,0,60],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p7:=[[210,0,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p8:=[[0,0,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p9:=[[210,10,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p10:=[[0,10,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p11:=[[210,20,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p12:=[[0,20,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p13:=[[210,30,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p14:=[[0,30,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p15:=[[210,40,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p16:=[[0,40,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p17:=[[210,50,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p18:=[[0,50,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p19:=[[210,60,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p20:=[[0,60,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p21:=[[210,70,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p22:=[[0,70,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p23:=[[210,80,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p24:=[[0,80,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p25:=[[210,90,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p26:=[[0,90,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p27:=[[210,100,-1],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p28:=[[0,100,-1],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p29:=[[0,100,9],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p30:=[[210,0,9],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p31:=[[210,0,8],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p32:=[[210,0,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p33:=[[0,0,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p34:=[[210,10,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p35:=[[0,10,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p36:=[[210,20,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p37:=[[0,20,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p38:=[[210,30,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p39:=[[0,30,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p40:=[[210,40,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p41:=[[0,40,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p42:=[[210,50,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p43:=[[0,50,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p44:=[[210,60,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p45:=[[0,60,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p46:=[[210,70,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p47:=[[0,70,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p48:=[[210,80,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p49:=[[0,80,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p50:=[[210,90,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p51:=[[0,90,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p52:=[[210,100,-2],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p53:=[[0,100,-2],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p54:=[[0,100,8],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p55:=[[210,0,8],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p56:=[[210,0,7],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p57:=[[210,0,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p58:=[[0,0,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p59:=[[210,10,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p60:=[[0,10,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p61:=[[210,20,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p62:=[[0,20,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p63:=[[210,30,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p64:=[[0,30,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p65:=[[210,40,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p66:=[[0,40,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p67:=[[210,50,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p68:=[[0,50,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p69:=[[210,60,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p70:=[[0,60,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget p71:=[[210,70,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget p72:=[[0,70,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p73:=[[210,80,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p74:=[[0,80,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p75:=[[210,90,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p76:=[[0,90,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p77:=[[210,100,-3],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p78:=[[0,100,-3],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p79:=[[0,100,7],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p80:=[[210,0,7],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p81:=[[210,0,6],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p105:=[[210,0,6],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p106:=[[210,0,5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p107:=[[210,0,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p108:=[[0,0,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p109:=[[210,10,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p110:=[[0,10,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p111:=[[210,20,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p112:=[[0,20,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p113:=[[210,30,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p114:=[[0,30,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p115:=[[210,40,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p116:=[[0,40,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p117:=[[210,50,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p118:=[[0,50,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p119:=[[210,60,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p120:=[[0,60,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p121:=[[210,70,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p122:=[[0,70,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p123:=[[210,80,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p124:=[[0,80,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p125:=[[210,90,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p126:=[[0,90,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p127:=[[210,100,-5],[0,-0.283703561,0.958912034,0],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p128:=[[0,100,-5],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p129:=[[0,100,20],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p130:=[[0,100,100],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p131:=[[0,0,100],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p132:=[[0,0,100],[0,-0.283703561,0.958912034,0],[0,-1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC Main()
    MoveJ p4, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p5, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p6, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p7, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p8, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p9, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p10, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p11, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p12, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p13, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p14, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p15, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p16, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p17, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p18, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p19, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p20, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p21, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p22, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p23, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p24, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p25, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p26, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p27, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p28, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p29, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p30, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p31, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p32, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p33, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p34, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p35, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p36, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p37, v100, fine, MyTool \Wobj:=wobj0;
    MoveJ p38, v100, fine, MyTool \Wobj:=wobj0;
```

```
        MoveJ p39, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p40, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p41, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p42, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p43, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p44, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p45, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p46, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p47, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p48, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p49, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p50, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p51, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p52, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p53, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p54, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p55, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p56, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p57, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p58, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p59, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p60, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p61, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p62, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p63, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p64, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p65, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p66, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p67, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p68, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p69, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p70, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p71, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p72, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p73, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p74, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p75, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p76, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p77, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p78, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p79, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p80, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p81, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p105, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p106, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p107, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p108, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p109, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p110, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p111, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p112, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p113, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p114, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p115, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p116, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p117, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p118, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p119, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p120, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p121, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p122, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p123, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p124, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p125, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p126, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p127, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p128, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p129, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p130, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p131, v100, fine, MyTool \Wobj:=wobj0;
        MoveJ p132, v100, fine, MyTool \Wobj:=wobj0;
    ENDPROC
ENDMODULE
```

## Appendix C C# Programming Coding for RAP-CODE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions;
using Gcodelibv1;
using Rcodelibv1;

namespace RAP_CODE
{
    public partial class RAP_CODE : Form
    {
        public RAP_CODE()
        {
            InitializeComponent();
        }

        OpenFileDialog ofd = new OpenFileDialog();
        private void OpenFile_Click(object sender, EventArgs e)
        {
            if (ofd.ShowDialog() == DialogResult.OK)
            {
                gcode_tbox.Text = File.ReadAllText(ofd.FileName);
            }
        }

        private void listView1_SelectedIndexChanged(object sender, EventArgs
e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void ResetG_Code_Click(object sender, EventArgs e)
        {
            gcode_tbox.Clear();
        }

        private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)
        {

        }

        private void label5_Click(object sender, EventArgs e)
        {

        }
```

```
        private void ConvertToRapid_Click(object sender, EventArgs e)
        {
            Rapid_Code_TxtBox.Clear();
            Rapid_Code_TxtBox.Text += " MODULE Module1 ";

            string velo = comboBox1.Text;
            string zone = comboBox2.Text;
            string tool = textBox3.Text;
            string wobj = textBox2.Text;

            string[] text_line = Regex.Split(this.gcode_tbox.Text, "\n");

            String G_cmd = "";
            String N = "";
            String X = "";
            String Y = "";
            String Z = "";

            double Rz, Ry, Rx, Cy, Sy, Cp, Sp, Cr, Sr, qw, qx, qy, qz;

            Ry = Convert.ToDouble(RyTxtBox.Text);
            Rz = Convert.ToDouble(RzTxtBox.Text);
            Rx = Convert.ToDouble(RxTxtBox.Text);

            Cy = Math.Cos(Rz * 0.5);
            Sy = Math.Sin(Rz * 0.5);
            Cp = Math.Cos(Ry * 0.5);
            Sp = Math.Sin(Ry * 0.5);
            Cr = Math.Cos(Rx * 0.5);
            Sr = Math.Sin(Rx * 0.5);


            qw = Cr * Cp * Cy + Sr * Sp * Sy;
            qx = Sr * Cp * Cy - Cr * Sp * Sy;
            qy = Cr * Sp * Cy + Sr * Cp * Sy;
            qz = Cr * Cp * Sy - Sr * Sp * Cy;

            QuaternionValueTxtBox.Text = qw.ToString("0.0000") + "  " +
qz.ToString("0.0000") + "  " + qy.ToString("0.0000") + "  " +
qx.ToString("0.0000");

            foreach (String n in text_line)
            {
                N = "";

                string[] param = Regex.Split(n, " ");
                foreach (String i in param)
                {
                    if (i.Contains('G'))
                    {
                        G_cmd = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('N'))
                    {
                        N = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('X'))
                    {
                        X = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
```

```
                    }
                    if (i.Contains('Y'))
                    {
                        Y = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('Z'))
                    {
                        Z = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }
                }

                if ((G_cmd == "00" | G_cmd == "02" | G_cmd == "43" | G_cmd ==
"01" | G_cmd == "17" | G_cmd == "91" | G_cmd == "28" | G_cmd == "54") & N !=
"")
                {
                    Rapid_Code_TxtBox.Text += rcode.NewLine() + "  CONST
robtarget";
                    Rapid_Code_TxtBox.Text += " p" + N + ":=";
                    Rapid_Code_TxtBox.Text += "[[" + X + ", " + Y + ", " + Z +
"],[" + qw.ToString("0.0000") + ", " + qz.ToString("0.0000") + ", " +
qy.ToString("0.0000") + ",  " + qx.ToString("0.0000") + " ], [-1, 0, 1, 0],
[9E+9,9E+9, 9E9, 9E9, 9E9, 9E9]]; ";
                }
            }

            Rapid_Code_TxtBox.Text += rcode.NewLine() + "      PROC Main() ";

            string[] text2_line = Regex.Split(this.gcode_tbox.Text, "\n");
            foreach (String n in text2_line)
            {
                N = "";

                string[] param = Regex.Split(n, " ");
                foreach (String i in param)
                {
                    if (i.Contains('G'))
                    {
                        G_cmd = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('N'))
                    {
                        N = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('X'))
                    {
                        X = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }
                    if (i.Contains('Y'))
                    {
                        Y = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }

                    if (i.Contains('Z'))
                    {
                        Z = Regex.Match(i, @"[-+]?[0-9]*\.?[0-9]+").Value;
                    }
                }
```

```
                if ((G_cmd == "00" | G_cmd == "02" | G_cmd == "43" | G_cmd ==
"01" | G_cmd == "17" | G_cmd == "91" | G_cmd == "28" | G_cmd == "54") & N !=
"")
                {
                    Rapid_Code_TxtBox.Text += rcode.NewLine() + "
MoveJ p" + N + ", v" + velo + ", " + zone + ", " + tool + @" \Wobj:=" + wobj +
";";
                }
            }
            Rapid_Code_TxtBox.Text += rcode.NewLine() + "      ENDPROC" +
rcode.NewLine();
            Rapid_Code_TxtBox.Text += "ENDMODULE " + rcode.NewLine();
        }

        private void ExportRapid_Click(object sender, EventArgs e)
        {
            SaveFileDialog save = new SaveFileDialog();
            save.Title = "Save File";
            save.Filter = "Text Files (*.txt) | *.txt";

            if (save.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                StreamWriter write = new
StreamWriter(File.Create(save.FileName));
                write.WriteLine(Rapid_Code_TxtBox.Text);
                write.Dispose();
            }
        }

        private void ClearAll_Click(object sender, EventArgs e)
        {
            gcode_tbox.Clear();
            Rapid_Code_TxtBox.Clear();
        }

        private void ResetRapid_Click(object sender, EventArgs e)
        {
            Rapid_Code_TxtBox.Clear();
        }

        private void Rapid_Code_TxtBox_Click(object sender, EventArgs e)
        {

        }

        private void gcode_tbox_TextChanged(object sender, EventArgs e)
        {

        }

        private void label7_Click(object sender, EventArgs e)
        {

        }
    }
}
```

## Appendix D C# Programming Coding for G-Code & M-Code

```
using System;
using System.Collections.Generic;
using System.Dynamic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Gcodelibv1
{
    public class gcode
    {
        public static string dot2()
        {
            return ": ";
        }
        public static string NewLine()
        {
            return Environment.NewLine;
        }
        public static string N()
        {
            return Environment.NewLine + "Line";
        }

        public static string S()
        {
            return Environment.NewLine + "Set Spindle Speed: ";
        }
        public static string X()
        {
            return Environment.NewLine + "Move to Position: ";
        }
        public static string Y()
        {
            return Environment.NewLine + "Move to Position: ";
        }
        public static string Z()
        {
            return Environment.NewLine + "Move to Position: ";
        }
        public static string F()
        {
            return Environment.NewLine + "Set Feed Rate: ";
        }
        public static string T()
        {
            return Environment.NewLine + "Select Tool: ";
        }
        public static string G(String n)
        {
            switch (n)
            {
                case "G00":
                    return "G00: Move in a straight line at rapids speed.";
                case "G01":
                    return "G01: Move in a straight line with Feedrate
speed.";
```

```
                case "G02":
                    return "G02: Clockwise circular arc at Feedrate speed.";
                case "G03":
                    return "G03: Counter-Clockwise circular arc at Feedrate
speed.";
                case "G04":
                    return "G04: Dwell: Stop for a specified time.";
                case "G05":
                    return "G05: FADAL Non-Modal Rapids.";
                case "G09":
                    return "G09: Exact stop check.";
                case "G10":
                    return "G10: Programmable parameter input.";
                case "G15":
                    return "G15: Turn Polar Coordinates OFF, return to
Cartesian Coordinates.";
                case "G16":
                    return "G16: Turn Polar Coordinates ON.";
                case "G17":
                    return "G17: Select x-y plane.";
                case "G18":
                    return "G18: Select x-z plane.";
                case "G19":
                    return "G19: Select y-z plane.";
                case "G20":
                    return "G20: Program coordinates are inches.";
                case "G21":
                    return "G21: Program coordinates are mm";
                case "G27":
                    return "G27: Reference point return check";
                case "G28":
                    return "G28: Return to home position";
                case "G29":
                    return "G29: Return from the reference position";
                case "G30":
                    return "G30: Return to the 2nd, 3rd, and 4th reference
point";
                case "G32":
                    return "G32: Constant lead threading (like G01
synchronized with spindle)";
                case "G40":
                    return "G40: Tool cutter compensation off (radius comp.)";
                case "G41":
                    return "G41: Tool cutter compensation left (radius
comp.)";
                case "G42":
                    return "G42: Tool cutter compensation right (radius
comp.)";
                case "G43":
                    return "G43: Apply tool length compensation (plus)";
                case "G44":
                    return "G44: Apply tool length compensation (minus)";
                case "G49":
                    return "G49: Tool length compensation cancel";
                case "G50":
                    return "G50: Reset all scale factors to 1.0";
                case "G51":
                    return "G51: Turn on scale factors";
                case "G52":
                    return "G52: Local workshift for all coordinate systems:
add xyz offsets";
                case "G53":
```

```
                return "G53: Machine coordinate system (cancel work
offsets)";
            case "G54":
                return "G54: Work coordinate system (1st Workpiece)";
            case "G55":
                return "G55: Work coordinate system (2nd Workpiece)";
            case "G56":
                return "G56: Work coordinate system (3rd Workpiece)";
            case "G57":
                return "G57: Work coordinate system (4th Workpiece)";
            case "G58":
                return "G58: Work coordinate system (5th Workpiece)";
            case "G59":
                return "G59: Work coordinate system (6th Workpiece)";
            case "G61":
                return "G61: Exact stop check mode";
            case "G62":
                return "G62: Automatic corner override";
            case "G63":
                return "G63: Tapping mode";
            case "G64":
                return "G64: Best speed path";
            case "G65":
                return "G65: Custom macro simple call";
            case "G68":
                return "G68: Coordinate System Rotation";
            case "G69":
                return "G69: Cancel Coordinate System Rotation";
            case "G73":
                return "G73: High speed drilling cycle (small retract)";
            case "G74":
                return "G74: Left hand tapping cycle";
            case "G76":
                return "G76: Fine boring cyle";
            case "G80":
                return "G80: Cancel canned cycle";
            case "G81":
                return "G81: Simple drilling cycle";
            case "G82":
                return "G82: Drilling cycle with dwell.";
            case "G83":
                return "G83: Peck drilling cycle (full retract)";
            case "G84":
                return "G84: Tapping cycle";
            case "G85":
                return "G85: Boring canned cycle, no dwell, feed out";
            case "G86":
                return "G86: Boring canned cycle, spindle stop, rapid
out";
            case "G87":
                return "G87: Back boring canned cycle";
            case "G88":
                return "G88: Boring canned cycle, spindle stop, manual
out";
            case "G89":
                return "G89: Boring canned cycle, dwell, feed out";
            case "G90":
                return "G90: Absolute programming of xyz.";
            case "G90.1":
                return "G90.1: Absolute programming IJK.";
            case "G91":
                return "G91: Incremental programming of xyz.";
```

```
                case "G91.1":
                    return "G91.1: Incremental programming IJK.";
                case "G92":
                    return "G92: Offset coordinate system and save
parameters";
                case "G92.1":
                    return "G92.1: Cancel offset and zero parameters";
                case "G92.2":
                    return "G92.2: Cancel offset and retain parameters";
                case "G92.3":
                    return "G92.3: Offset coordinate system with saved
parameters";
                case "G94":
                    return "G94: Units per minute feed mode. Units in inches
or mm.";
                case "G95":
                    return "G95: Units per revolution feed mode. Units in
inches or mm.";
                case "G96":
                    return "G96: Constant surface speed";
                case "G97":
                    return "G97: Cancel constant surface speed";
                case "G98":
                    return "G98: Return to initial Z plane after canned
cycle";
                case "G99":
                    return "G99: Return to initial R plane after canned
cycle";

            }
            return "";
        }

        public static string M(String n)
        {
            switch (n)
            {
                case "M00":
                    return "M00: Program Stop.";
                case "M01":
                    return "M01: Optional Stop: Operator Selected to Enable.";
                case "M02":
                    return "M02: End of Program.";
                case "M03":
                    return "M03: Spindle ON Clock Wise Rotation.";
                case "M04":
                    return "M04: Spindle ON Counter Clock Wise Rotation.";
                case "M05":
                    return "M05: Spindle Stop.";
                case "M06":
                    return "M06: Tool Change.";
                case "M07":
                    return "M07: Mist Coolant ON.";
                case "M08":
                    return "M08: Flood Coolant ON.";
                case "M09":
                    return "M09: Coolant OFF.";
                case "M17":
                    return "M17: FADAL Subroutine Return.";
                case "M29":
                    return "M29: Rigid Tapping Mode On Fanuc Controls.";
                case "M30":
```

```
                    return "M30: End of program, Rewind and Reset Modes.";
                case "M97":
                    return "M97: Haas-Style Subprogram Call.";
                case "M98":
                    return "M98: Subprogram Call.";
                case "M99":
                    return "M99: Return from Subprogram.";
            }
            return "";
        }

    }
}
```

## Appendix E C# Programming Coding for Quaternion Formula

```
            Ry = Convert.ToDouble(RyTxtBox.Text);
            Rz = Convert.ToDouble(RzTxtBox.Text);
            Rx = Convert.ToDouble(RxTxtBox.Text);

            Cy = Math.Cos(Rz * 0.5);
            Sy = Math.Sin(Rz * 0.5);
            Cp = Math.Cos(Ry * 0.5);
            Sp = Math.Sin(Ry * 0.5);
            Cr = Math.Cos(Rx * 0.5);
            Sr = Math.Sin(Rx * 0.5);


            qw = Cr * Cp * Cy + Sr * Sp * Sy;
            qx = Sr * Cp * Cy - Cr * Sp * Sy;
            qy = Cr * Sp * Cy + Sr * Cp * Sy;
            qz = Cr * Cp * Sy - Sr * Sp * Cy;

            QuaternionValueTxtBox.Text = qw.ToString("0.0000") + "  " +
qz.ToString("0.0000") + "  " + qy.ToString("0.0000") + "  " +
qx.ToString("0.0000");
```