

Design and Programming of 5 Axis Manipulator Robot with GrblGru Open Source Software on Preparing Vocational Students' Robotic Skills

Marsono^{1*}, Yoto², Agus Suyetno³, Riana Nurmalasari⁴

^{1, 2, 3, 4} Department of Mechanical Engineering, Universitas Negeri Malang, Malang, Indonesia

Email: marsono.ft@um.ac.id

*Corresponding Author

Abstract— The rapid development of technology at the end of this decade, the industrial world is changing the human resources they have using the help of robot technology. The use of a robot was chosen because of the accuracy and precision that it can do in doing the job. In addition, robotic technology requires only one programming to do many things. Thus, there are many advantages for companies by using robotic technology in their production processes. This research aims to designing and building arm robot prototype to teach the robotics programming language. The result of this research is an arm robot prototype with Arduino Mega 2560 based programming. In addition, the robot movement programming uses the open source GrblGru software Grbl based. The GrblGru used for interpreting G-code and to convert pulse and direction information to control the stepper motor. This research concludes that there are 5 importance steps on develop the arm robot prototype, 1. Analyze; 2. Design; 3. Develop; 4. Evaluate; and 5. Implementation. With the GrblGru software the movements of arm robot prototype can work on simulation mode and control mode, there for students easier to study robotics programming language.

Keywords—robot, manipulator, GrblGru, vocational student, robotic skills

I. INTRODUCTION

In the digital era, everyone is required to have broad insight, the ability to think systematically and creatively to adapt to the rapid development of information technology [1]. There for, schools should not only focus on developing scientific & theoretical knowledge, but also must pay attention to the process of looking for systematic thinking and behavior, both a person and the surrounding environment [2]. These conditions require education actors, especially vocational education as a labor supplier, not only to focus on developing hard skills in the form of practical knowledge that is more scientific and theoretical in nature, but also must provide a portion of soft skills that support one's thinking and behavior processes [3].

The public's view of robots and types of robots always assumes that robots are always humanoid in which this type of robot has human-like abilities in both function and action [4]. The following are several types of robots in terms of physical form such as: 1. Robot Manipulator (a robot whose shape is to manipulate or imitate the forms of living things); 2. Humanoid Robot (robots are human-like, meaning they have a head, two hands, a body, two legs, and so on); 3. Android Robot (robots are similar to Humanoid, but there are

differences, namely: they have skin, and are able to express themselves like humans); 4. Animaloid Robot (robot starts from the shape and behavior of mimicking animals); and 5. Wheeled Robot (robot that walks or moves using wheels) [5][6][7][8][9].

There are many benefits of learning programming robotics skills that can be applied in vocational students, including, 1. Stimulate systematic and structured thinking in solving a problem; 2. Improve motoric skills; 3. Improve imagination and innovation skills in designing a creative robot project; 4. Train cooperation in groups and increase self-confidence, both accepting and respecting the opinions of others as well as daring to display and express creative ideas; and 5. Train patience and persistence in making a project [10][11][12][13].

Programming robot movement to deal with some task for several students still found the difficulties [14][15]. There for, they should understand the programming language of C or C++ to handle the inverse or kinematic robotic motion. To deal with that problem, there are one software can be use for student easily on programming robotic motion its name is GrblGru [16].

GrblGru is a simulation 3D model robot movement and it can directly control the robot motion using a G-code of numerical control. Comparing the programming using Arduino it easier because in the Arduino need the library of each component such as stepper motor and in the GrblGru do not need adding such library.

II. STUDY LITERATURE

A. Arm Robot

Arm robot or better known as Robot Manipulator is one of the most widely used robot types by industrial society. A robotic manipulator that mimics the characteristics of a human arm is also called an articulated arm [17]. However, the articulated motion of a robotic arm is different from that of a human arm. While robot joints have less degrees of freedom or the so-called Degree of Freedom (DOF), the robot manipulator can move freely even from a certain angle with more freedom. For example, the elbow of an articulated robot can bend up or down whereas a person can only bend their elbow in one direction the rest only in a straight arm position [18].



The manipulators are grouped into classes according to the joint combination used in the construction of the robot manipulator. A Cartesian geometric arm uses only prismatic joints and can reach any position in a rectangular workspace by movement of the cartesian link. By replacing the cartesian arm joint with a revolute joint, a cylindrical geometric sleeve was formed[19][20].

With the combination of rotation and translation, this robot arm can reach any point in the workspace. If the shoulder joint is also replaced with a revolute joint, this means that the arm has geometric poles. The workspace of the articulated arm is a ball which is thick-walled, but the implementation is still quite complicated. The outer shell is a single ball, which can intersect each other. The robot manipulator is created from a sequence of links and joint combinations[21]. Links are rigid members that connect joints. The axes are the movable components of the robotic manipulator which cause relative motion between the links before or after them. The mechanical joints used to construct the manipulator arm robots are of five main types. Two of the joints are linear, in which the relative motion between adjacent links is non-rotational, and the three types of rotary, the relative motion of the robotic manipulator involves rotation between the links (Fig 1).

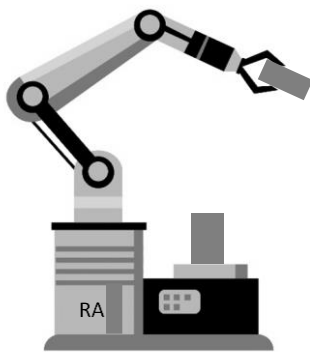


Fig. 1. The concept of robotic manipulator

B. GrblGru

GrblGru is the name of a free great open source software that can interpret G-code and use it to convert pulse and direction information to control the stepper motor power amps [16], [22]. The software and the operating instructions can be downloaded at <https://www.grblgru.com/>. The basic function of GrblGru is 1. a 3D simulator for milling and lathe; 2. a GCode Sender for Grbl, TinyG, Mega-5X, and g2core; 3. a CAM program for milling and lathe work; and 4. a Tool for learning CNC technology without needing a 'real' Machine [23].

The GrblGru have 2 types of operating modes. 1. Simulation Mode for Virtual 3D simulation of CNC machine control and 2. Control Mode for Real control of a CNC machine with Grbl, TinyG, Mega-5X or g2core operating system. The simulation mode is the original idea of the program. It was primarily intended to check existing NC programs on a virtual machine and to show any collisions or problems. This type of early 'virtual machine' has prevailed

in the industry more and more in recent years because it saves valuable machine time detecting possible errors in advance.

When GrblGru is started, the mode by default is set to the 'Simulation'. There is no connection to a real controller. The NC program to be examined is loaded and the NC commands are processed by the internal NC interpreter [24]. The loaded 3D model is animated accordingly NC = Numerical Control (Fig 2).

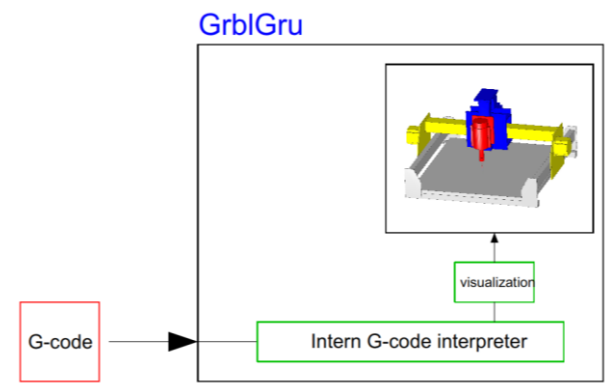


Fig. 2. The GrblGru in Simulation mode (GrblGru operating instructions, 2020)

In control operation mode, GrblGru can connect to an Arduino UNO, a TinyG, an Arduino Mega 2560, or an Arduino Due controller. The connection is made via the USB interface, which transfers both the NC commands to the controller and the information about the current axis positions back to GrblGru [24].

GrblGru uses this information to control the 3D model so that the 3D model is synchronized to the 'real' axes. When GrblGru is started, use the controller option to select Simulation, Grbl, TinyG, G2core, or GRBL-Mega-5x (Fig 3).

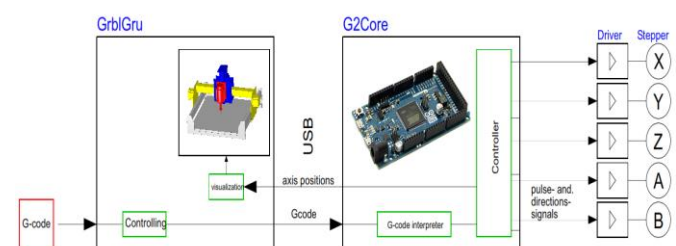


Fig. 3. The GrblGru in Control mode (GrblGru operating instructions, 2020)

III. METHOD

This study uses a research and development with the five phases, namely Analysis, Design, Development, Implementation, and Evaluation (ADDIE) model approach to developing and programming the prototype robotic arm (Fig 4) [25][27].

A. Analysis

The analysis phase, there are two important parts there are analysis the goals and required resources. The goals of this research are to deliver the competency of developing and

programming robotic arm, hence there are no gaps between students' knowledge and skills in studies robotic skills.

B. Design

In the design phases, the robotic was designed with Autodesk software. There are 5 part of robotic that designed to make complete of robotic arm, included: base, shoulder, elbow, wrist, and gripper.

C. Development

After all the part of robotic arm was finished designed, the next step was sending the model to 3D printer. Willing the 3D printer finish the model, the other electronic devices was prepared to be assembly with the other part.

D. Implementation

When the arm robot finished development, the next phase is making some programming of the robotic movement. The GrblGru was used in the implementation of robot programing. The robotic programming is from the basic movement to the advance.

E. Evaluation

The last phase of develop arm robot is evaluation. Evaluation was done to evaluate the performance of the arm robotic that finished assembly whether it have an error in the movement or already to uses in programming robotics courses.

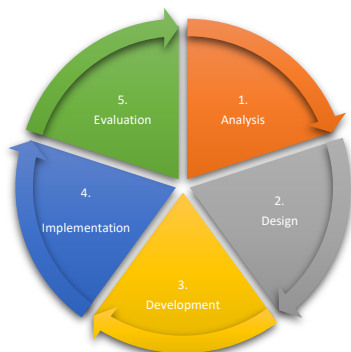


Fig. 4. The ADDIE model to develop 5 axis manipulator robots\

IV. RESULTS AND DISCUSSION

There are five steps on designing and programming of 5 axis manipulator robot with GrblGru software.

The main goals of this study are students easier to programming the arm robotic manipulator with limited knowledge about programming language such as C or C++. GrblGru software is one of alternative that can be use for the solution on programming with numerical control based. The requirement running GrblGru is using windows or Linux operating system.

There are 5 parts of robotic that designed to make robotic arm from FreeCAD software,

1. Robot base,
2. Shoulder,
3. Elbow,

4. Base line, and
5. Griper (Fig 5)

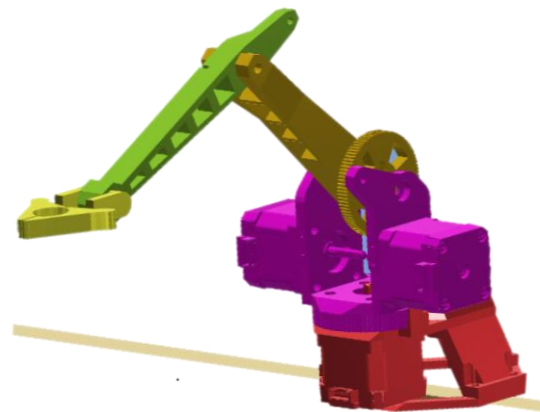


Fig 5. The design of 5 axis manipulator robot

After the model finished printed with 3D printer, the 5-axis manipulator robot was assembly with electronic component (Table 1) included stepper motor, servo motor, ramps v1.4, motor driver, Arduino mega 2560, sources power, and wire.

TABLE I. HARDWARE USES IN THE 5-AXIS MANIPULATOR ROBOT

NO	COMPONENT	SPECIFICATION
1	Controller	1-unit Arduino Mega 2560
2	Actuator	4-unit stepper motor nema 17; 1-unit servo motor
3	Power supply	1-unit 12 V 5 A
4	Ramps driver	1-unit Ramps V 1.4
5	Motor driver	5-unit driver A 4988
6	Wire	1-unit USB cable data

There for the final manipulator robot as seen fig 6. All the component 3D printer product included line, base, shoulder, elbow, and gripper attached with the component electronic (Arduino mega, ramps V 1.4, motor driver A 4988, stepper motor NEMA 17, servo motor).

After finished assembly, the robot was tested to make sure all the component gets the electricity sources. Arduino IDE was used to programming and running the motors. Programming the robot done with directly accessed the microcontroller Arduino mega 2560.



Fig. 6. The development of 5 axis manipulator robot

The step for programming robot using Arduino IDE as follow.

1. Open Arduino IDE software
2. Add the library stepper motor (StepperArmRobot.h) to the Arduino. Klik Klik Skecth → Add files, go to folder Documents → Arduino → libraries → StepperArmRobot → StepperArmRobot.h
3. After StepperArmRobot.h was opened on the Arduino, back to the main sketch to adding the library StepperArmRobot.h by insert #include name_lybrary.

```
# include "StepperArmRobot.h"

ArmRobot armRobot;

void setup ()
{

}
void loop ()
{

}
```

In the library StepperArmRobot there are basic functions that we can accessed, including:

- armRobot.begin()
- armRobot.set_angle(ID_motor, sudut)
- armRobot.set_steps(ID_motor, jumlah_steps)
- armRobot.set_position(x, y, z, sudut_rot_arm, sudut_rot_wrist, sudut_rot_grip)
- armRobot.sync_move(sudut_base, sudut_shoulder, sudut_elbow)
- armRobot.set_speed(ID, motor_speed)

1. Set Steps

Set steps is used to drive a stepper motor with a certain step. This function can only be used for the base, shoulder, elbow, and wrist joints. In this robot, the stepper motor used is NEMA 17. This motor has a movement revolution of 1.8° per one step, so that for the stepper motor to move 1 revolution (360°), the number of steps needed is 360 / 1.8 = 200 steps. The angle of the joints formed based on the number of steps is also influenced by the size or gear ratio of each joint. The gear ratios for each joint are as follows:

TABLE II. THE GEAR RATIO OF STEPPER MOTOR

Name	Revolution	Prescaler	Gear Ratio
Base	1.8	1/8	1 : 6
Shoulder	1.8	1/8	1 : 8
Elbow	1.8	1/8	1 : 8
Rotation Arm	1.8	1/8	1 : 1

To make a stepper motor move one revolution (360°), the number of steps required is:

$$n = \frac{\theta \times prescaler}{revolution} = \frac{360 \times 8}{1.8} = 1600$$

Because the robot arm joint uses a comparison gear, to calculate the number of steps in the robot joint is as follows:

$$n = \frac{\theta}{360} \times 1600 \times gear_ratio$$

Where n = number of steps, θ = joint angle

The function used to move the arm robot based on the step is:

set_steps().

Syntax:

```
armRobot.set_steps(ID_motor, jumlah_step);
```

example:

```
armRobot.set_steps(0,3200);
// Move the base (0) with 3200 steps
```

2. Programming motion of base angle:

```
#include <StepperArmRobot.h>

ArmRobot armRobot;

void setup()
{
  armRobot.begin();
  armRobot.set_speed(BASE, 15);
}

void loop()
{
  if(button)
  {
    delay(500);
    armRobot.stepper_attach();

    armRobot.set_angle(BASE, 0); delay(1000);
    armRobot.set_angle(BASE, 90); delay(1000);
    armRobot.set_angle(BASE, 180); delay(1000);
    armRobot.set_angle(BASE, 90); delay(1000);

  }
}
```

3. Set angle is the function that uses to move the arm robot with various angle with syntax. It should be noticed the maximum and minimum angle of stepper motor.

Syntax:

```
armRobot.set_angle(ID_motor, nilai_sudut);
```

example:

```
armRobot.set_angle(1, 90);
// Move the shoulder (1) with 90 degrees
```

Every motor has the differences angle range, including:

- a. Base angle : 0 – 180
- b. Shoulder angle : 10 – 127
- c. Elbow angle : 50 – 70
- d. Rotation Arm angle : 0 – 180
- e. Rotation Gripper angle : 0 – 180

4. Programming motion of shoulder angle

```
#include <StepperArmRobot.h>

ArmRobot armRobot;
void setup()
{
  armRobot.begin();
}
void loop()
{
  if(button)
  {
    delay(500);
    armRobot.stepper_attach();
    armRobot.set_angle(ELBOW, 90);

    armRobot.set_angle(SHOULDER, 90);
    armRobot.set_angle(SHOULDER, 110);
    armRobot.set_angle(SHOULDER, 70);
  }
}
```

5. Programming motion of elbow angle

```
#include <StepperArmRobot.h>

ArmRobot armRobot;

void setup()
{
  armRobot.begin();
}

void loop()
{
  if(button)
  {
    delay(500);
    armRobot.stepper_attach();
    armRobot.set_angle(SHOULDER, 90);

    armRobot.set_angle(ELBOW, 90);
    armRobot.set_angle(ELBOW, 120);
    armRobot.set_angle(ELBOW, 90);
    armRobot.set_angle(ELBOW, 70);
  }
}
```

6. Programming gripper

```
#include <StepperArmRobot.h>

ArmRobot armRobot;

void setup()
{
  armRobot.begin();
}

void loop()
{
  if(button)
  {
    delay(500);
    armRobot.stepper_attach();
    armRobot.gripper_cath(); delay(2000);
    armRobot.gripper_release(); delay(2000);
  }
}
```

The process of programming robot arm using Arduino IDE above for students still faces the difficulties. There for, second methods were applying programming arm robot with GrblGru. The first step is compiling the model design of manipulator robot to GrblGru program by selecting menu Extras → Machine Manager (Fig 7). When success compiling it seen in Fig 8.

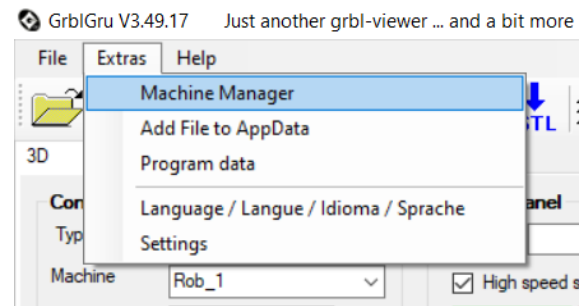


Fig. 7. Compiling the model to GrblGru program

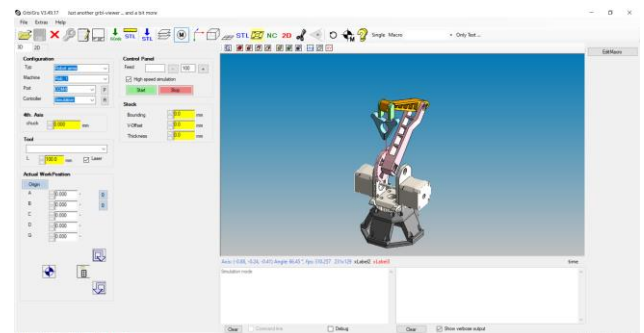


Fig. 8. The model of manipulator robot already compiling with GrblGru

The GrblGru software is programming with numerical control based, hence this software uses G-code language to be running the robot movement [28]. Table 3 shows the G-code and the meaning of the code [29], [30].

TABLE III. G-CODE AND THE MEANING

G-code	Meaning
G0/G00	Rapid positioning
G1/G01	Linear interpolation
G2/G02	Circular interpolation, clockwise
G3/G03	Circular interpolation, counter clockwise
G4/G04	Dwell (pause)
G10	Set Work Coordinate Origin (and resultant Offsets)
G17	Select the XY plane (for arcs)
G18	Select the XZ plane (for arcs)
G19	Select the YZ plane (for arcs)
G20	After this, units will be in inches
G21	After this, units will be in mm
G28	Go to Pre-Defined Position
G28.1	Set Pre-Defined Position
G28.2	Run homing cycle
G30	Go to Pre-Defined Position (Return to secondary home position)
G30.1	Set Pre-Defined Position
G38.1	Probe cycle
G38.2	Straight Probe
G38.3	Probe
G38.4	Probe
G38.5	Probe
G40	Tool radius comp off
G43.1	Dynamic tool length offset

In the GrblGru software the coordinate axes name is different, it can be seen in the interface (Fig 9).

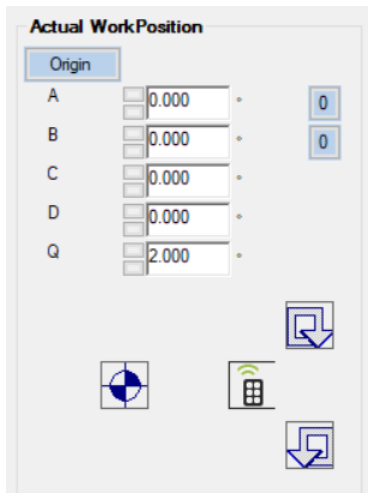


Fig 9. The actual work position in GrblGru

The coordinates name uses A, B, C, D, and Q. Hence the stepper motor axis name should adjust from X, Y, and Z to A, B, and C. However, this is only the name changing and has no effect on the function. A for programming line base motion, B for programming base motion, C for programming shoulder motion, D for programming elbow motion, and Q for programming the gripper motion.

Configuration are needed to make sure the machine has the same controller with the GrblGru. To configure the machine, select Type of machine in here used Robot arms; Machine select name of the machine, port autodetect, controller use the GRBL-Mega-5x (Fig 10) [24].

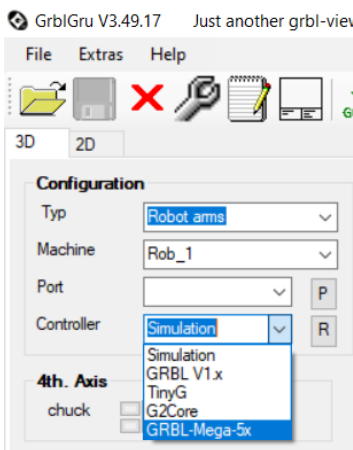


Fig. 10. Configuration in GrblGru

After finished the configuration, we can program the arm robot by open the macro toolbar in the configuration setting. To edit the robot references the motor stepper in the GrblGru we can go to Extras → Settings (Fig 10).

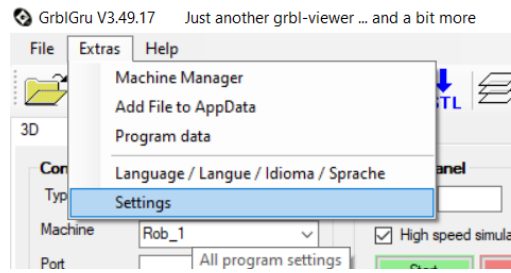


Fig. 10. Reference settings on GrblGru

Activated the macro toolbar by selecting yes option (Fig 11).

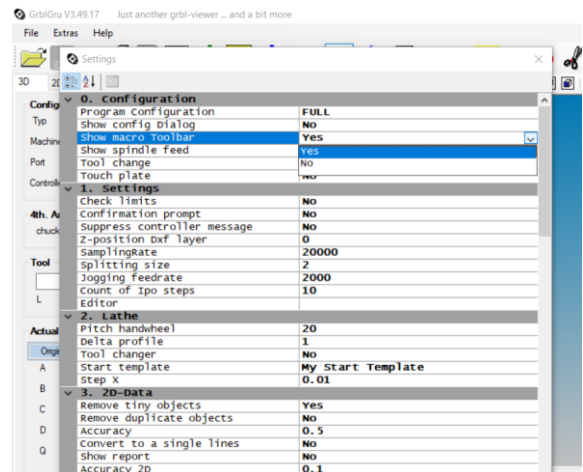


Fig. 11. Activate the macro toolbar on GrblGru

The example of the program is:

```
[Rotate the base]
G1 g90 f1000
A-40
A40
A-40
A40
A0

[Up and Down Motion of elbow]
G1 g90 f1000
C-40
C40
C-40
C40
C0

[Simple job]
G1 g90 f1000
A-40 B58 C13 D-70 E0 H0 Q10 ; Left
Q-25
A0 B0 C0 D0 E0 H0 ; Back to home
A40 B58 C13 D-70 E0 H0 ; Right
Q0
A0 B0 C0 D0 E0 H0 ; Back to home

[looping work]
G1 g90 f1000
#Counter = 3
```

```

%Inputs, Enter the following data, Number of strokes,
#Counter
while(#Counter > 0)
{
A-40 B58 C13 D-70 E0 H0 Q0
Q-25
A0 B0 C0 D0 E0 H0
A40 B58 C13 D-70 E0 H0
Q0
A0 B0 C0 D0 E0 H0
#Counter -1
}

```

V. CONCLUSION

The 5-axis manipulator robot was well design, programming and doing several tasks. There are 5 components was finished from 3D printer (Robot base, Shoulder, Elbow, Base line, and Griper) that assembly with electric component (Controller, Actuator, Power supply, Ramps driver, Motor driver, and Wire) to make complete manipulator robot.

Programming the robot using Arduino IDE was more difficult rather than using GrblGru software. In Arduino, the library off all the motor should be added on the Arduino library so the program can run, it is become problem when we did not have the motor library. Students look so frustrated when learn on programming the robot movement. With the limited basic language programing, students learn easier using GrblGru software to give the motion on arm robotic manipulator.

With numerical control based, GrblGru have simple programing compare with using Arduino IDE on the same robotic task. Also, with GrblGru software, student can have the simulation movement of the robotic arm in the GUI GrblGru before they are running the program directly to the arm robot. Hence, the students know simultaneously if the arm robot movement on the wrong way or on the other hand, they can easier predict the program will run smoothly on the arm robot.

REFERENCES

- [1] K. Phasale *et al.*, "Design , Manufacturing and Analysis of Robotic Arm with SCARA Configuration," *Int. Res. J. Eng. Technol.*, 2018.
- [2] Marsono, M. Sugandi, Tuwoso, and Purnomo, "Study the impact of internship on improving engineering students' competency," in *AIP Conference Proceedings*, 2017, vol. 1887, doi: 10.1063/1.5003547.
- [3] Marsono, "Development of a cooperative micro lesson study learning model to teaching creatively and teaching for the creativity of engineering students," *World Trans. Eng. Technol. Educ.*, vol. 14, no. 2, 2016.
- [4] T. B. Sheridan, "Human-Robot Interaction," *Human Factors*. 2016, doi: 10.1177/0018720816644364.
- [5] T. Brogårdh, "Present and future robot control development-An industrial perspective," *Annu. Rev. Control*, 2007, doi: 10.1016/j.arcontrol.2007.01.002.
- [6] S. Guo, T. Fukuda, and K. Asaka, "A new type of fish-like underwater microrobot," *IEEE/ASME Trans. Mechatronics*, 2003, doi: 10.1109/TMECH.2003.809134.
- [7] H. Hanafusa, *modern robotics*. 2017.
- [8] A. Cully, J. Clune, D. Tarapore, and J. B. Mouret, "Robots that can adapt like animals," *Nature*, 2015, doi: 10.1038/nature14422.
- [9] J. H. Oh, D. Hanson, W. S. Kim, I. Y. Han, J. Y. Kim, and I. W. Park, "Design of android type humanoid robot Albert HUBO," 2006, doi: 10.1109/IROS.2006.281935.
- [10] A. Ghanizadeh, A. H. Al-Hoorie, and S. Jahedizadeh, "Mindfulness," in *Second Language Learning and Teaching*, 2020.
- [11] J. Andrew Bagnell, "Reinforcement Learning in Robotics: A Survey," in *Springer Tracts in Advanced Robotics*, 2014.
- [12] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artif. Intell.*, 2008, doi: 10.1016/j.artint.2007.09.009.
- [13] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A Review of The Applicability of Robots in Education," *Technol. Educ. Learn.*, 2013, doi: 10.2316/journal.209.2013.1.209-0015.
- [14] V. Paramasivam, J. Huang, S. Elliott, and M. Cakmak, "Computer science outreach with end-user robot-programming tools," 2017, doi: 10.1145/3017680.3017796.
- [15] L. Mottola, K. Whitehouse, and C. Ghezzi, "Poster abstract: Towards spatial macroprogramming for sensing and actuating robot swarms," 2013, doi: 10.1145/2517351.2517407.
- [16] S. Jeon, "GRBL Wiki," *GRBL Wiki - github.com*, 2011. .
- [17] A. M. A. Ghiet and A. Baba, "Robot Arm Control with Arduino," *Univ. Turkish Aeronaut. Assoc.*, 2017.
- [18] T. Vignesh, R. Nivetha, G. Mowneka, and D. Selvakumar, "Design and Fabrication of 6-axis Gesture Controlled Robot," 2019, doi: 10.1109/ICACCS.2019.8728465.
- [19] R. M. Murray, Z. Li, S. S. Sastry, R. M. Murray, Z. Li, and S. S. Sastry, "Robot Dynamics and Control," in *A Mathematical Introduction to Robotic Manipulation*, 2018.
- [20] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control. Theory and Practice*. 2004.
- [21] L. Jin, S. Li, J. Yu, and J. He, "Robot manipulator control using neural networks: A survey," *Neurocomputing*, 2018, doi: 10.1016/j.neucom.2018.01.002.
- [22] S. S. Skogsrud, "Grbl," *Choosing a controller*, 2009. .
- [23] A. R. Sonawane, A. B. Rane, and D. S. S. Sudhakar, "Development of A3-Axis Cnc Milling Machine With an Open Source Controller," *Int. J. Res. Eng. Technol.*, 2017, doi: 10.15623/ijret.2017.0608002.
- [24] Gary. (GrblGru), "GrblGru - Operating Instructions," 2020, [Online]. Available: <https://cdn.website-editor.net/f7ae70c5e8d847db8c417786ea6ca337/files/uploaded/GrblGru%2520Manual.pdf>.
- [25] N. Aldoobie, "ADDIE Model Analysis phase," *Am. Int. J. Contemp. Res.*, 2015.
- [26] F. Hishamudin, "Model ADDIE," *Universiti Teknologi Malaysia*. 2016.
- [27] R. M. Shor, "ADDIE+: Adopting proven practices from the IT industry," *TD*, 2012.
- [28] X. Zheng *et al.*, "G-code," *Forbes*, 2016.
- [29] T. Loveland, "Understanding and Writing G & M Code for CNC Machines," *Technol. Eng. Teach.*, 2012.
- [30] B. Jayachandriah, O. V. Krishna, P. A. Khan, and R. A. Reddy, "Fabrication of Low Cost 3-Axis Cnc Router," *Int. J. Eng. Sci. Invent.*, 2014.