ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse186932020.pdf
https://doi.org/10.30534/ijatcse/2020/186932020

Alternative Control System for Robot Arm with Data Logger

Supriyono[1],Wahyu Widiyanto[2],Waluyo Adi Siswanto[3]
[1]Dept. of Mechanical Engineering, Universitas Muhammadiyah Surakarta, Indonesia,
supriyono@ums.ac.id
[2]Dept. of Mechanical Engineering, Universitas Muhammadiyah Surakarta, Indonesia
[3]Dept. of Mechanical Engineering, Universitas Muhammadiyah Surakarta, Indonesia

ABSTRACT

This study aims to make an alternative of robotic arm control system. The system developed will replace numeric control into a data logger system.The system records manually data movement and applied the recording data into robot arm. 6 potentiometers of 10 kΩacting as sensors, a robot arm with 6 motor servos of MG995 acting as actuators and a micro controller of Arduino Mega 2560 are used for the controller. The data recording is saved intoArduino Mega 2560EEPROM every 1000 mS. The results show that the Arduino mega 2560 EEPROM can record data for duration of 682000 mS. The 10 kΩ potentiometer has non-linearity of 0.37% and proper mapping is 0-1023 to 0-258 by shifting the initial angle of 20^{O}. The MG995 servo motor has non-linearity of 0.05%. The system established can record the movement and applied to the robot arm accurately.

Key words: Data Logger, Robot Arm, Potentiometer, Motor Servo, Arduino Mega

1. INTRODUCTION

The industrial technology is evolving rapidly today. Many developed countries are applying modern technology in their industries. Repeated tasks and high accuracy have become the needs of industrial process. Robots offer speed and accuracy that are not given by human. Robots can also reduce operating costs, reduce scrap, and be flexible for future use [1]. Due to customer needs, process in manufacturing industries has to be fast and accurate[2]. For example in car industry, arc welding robot has many advantages such as stability result, speed, reducing accident [3]. Another example is robot arm, where the end of the arm can be equipped with device as needed such as gripper, pump, vacuum, magnet or arc welding [4].
A number of situations are impossible for operators to carry out their activities, due to the level of danger or difficulty involved. In this situation, a robot arm can be used to replace operator. Users can control the robot arm without contact safely.[5]. However, the robot arm programming based on numerical control (NC) to determine its movement is not easy, and the investment are expensive. In this case, it is necessary to have an alternative beside NC in determining the robot arm's movement.

Arduino Mega2560 is one of microcontrollers based on ATmega2560. It has 54 digital input output, 16 analog inputs, 4 UARTs, a 16 MHz crystal oscillator, USB connection, power jack, ICSP header, and reset button. Arduino Mega 2560 is compatible with most shields designed for Uno [6].As a microcontroller Arduino can control many electronic devices [7]. There are numbers of research to apply Arduino in many fields [8].For examples, Alsmadi, et al.,[9] used Arduino for automatic control of a vehicle's safety device to prevent collisions. Supriyono, et al., [10] developed engine control unit (ECU)manipulator using Arduino to increase power and torque. Leni A. Bulan et al., [11] created an air conditioning units (ACU) controller based on Arduino for energy conservation in University of Batangas Philippines. Use the "Insert Citation" button to add citations to this document.

Other applications of the Arduino are to control robot arm. Many works have been reported in this fields. Kanchi and Gosala[12], did research on robot arm using Arduino Uno as a control and potentiometer as sensor to move motor servo. Similar researches did by Bhargava &Kumar [13]and Bora & Nandi [14]. In these works, they used 4 motor servos for a robot arm.

Servo motors are rotary actuators used for applications that require a large degree of accuracy for rotation angles [15]. The servo motor receives a control signal representing the output position and applies power to the DC motor until the shaft rotates to the correct position, determined by the position sensor[16]. Niranjan, Suhas, and Sreekanth[17], used proteus software to design schematic diagram for robot arm using Arduino. When they applied the design to build a robot arm, they had proper and accurate result. The gear transmission system on the motor servo is built to reduce the speed in order to have higher torque[18].

Purkayastha et al., [19], VanHuy et al. [20], and Oza& Mehta [21], examined robotic hands using flex sensors as inputs, Arduino as controllers and servo motors as actuators. The basic work of robotic hands is based on the conversion of the flex input (from the flex sensor) to the rotational output (to the Servo motor). The change of resistance value in bending of flex sensor is processed by Arduino and used as a variable to control the servo motor. So this change in resistance value can be used to control the servo motor. Soetedjo et al., [22] did a research to examine the relationship between rotation angle of potentiometer and the output voltage. The result shown a linear relationship between the angle and output voltage. The error of the linearity was about 0.39%. Whereas in other studies, a robot arm is controlled by Arduino Uno which interacts with an internet using the Arduino Ethernet Shield, the accuracy test shows that the actual output of a servo motor compared to the input sent to Arduino Uno via the internet is in between 97% to 99%[23].

Referring to previous research, there is an opportunity to make an alternative of the numeric programming of robot arm system. The alternative system is a data logger movement system.This study aims to make an alternative of robotic arm control system that uses NC into a data logger system. To have the data movement the robot arm will record the movement manually. In order to show that the alternative system works properly, a robot arm will be built using Arduino as a control, potentiometer as a sensor and motor servo as actuator.

2. METHODOLOGY

This research goes through several stages, namely designing block diagrams, designing system installations, making program flow chart diagrams and making servo motor speed settings.

2.1 Designing Block Diagrams

Block diagrams of this research is shown in Figure 1. Pushbutton and potentiometer are used as input for Arduino.Pushbutton is connected to the digital pin, while the potentiometer is connected to the analog pin. Potentiometer is used as input data angle (pulse) and then it used to determine servo motor movement.At the bottom of Figure 1, this shows the use of the Arduino Mega 2560 EEPROM memory. The EEPROM memory is used as a repository for data obtained from the potentiometer angular movements.The angular data from potentiometer is used to determine the angular motion of servo motor. There are three outputs from the Arduino Mega 2560 namely LCD, LED and servo motor. The LCD uses an additional module that is I2C. The function of this LCD is to be an interface to the system and LEDs are used as indicator lights of the system. Servo motors are connected to PWM (Pulse With Modulation) pins.

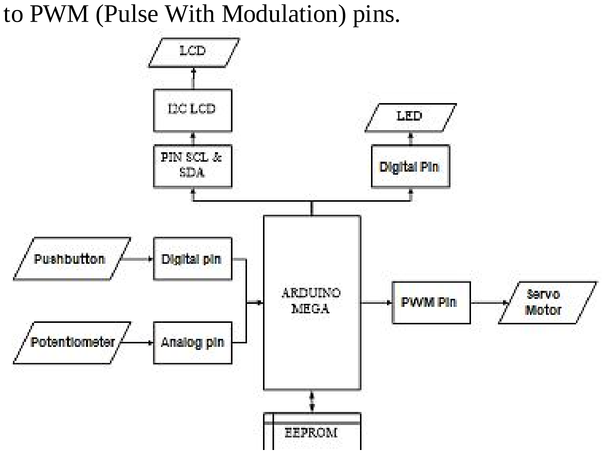

Figure 1: Block diagram of robot arm system.

2.2 Designing System Installation

The robot arm in this research is shown in Figure 2. The robot arm has 6 joints where each joint has a servo motor. While wiring diagram of this research is shown in Figure 3. The components of the system of this research are1 Arduino Mega 2560, 6 pushbuttons, 6 potentiometers of 10 kΩ, 6 LEDs of 5mm, 6 MG665 Servo Motors, and 1 LCD I2C.

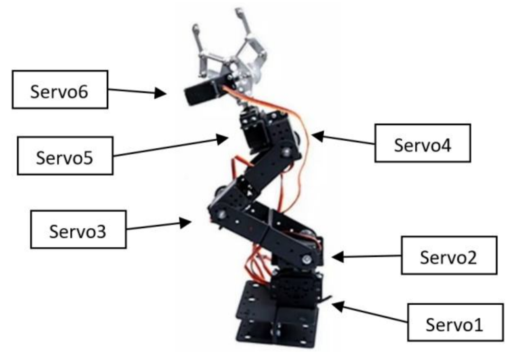

Figure 2: Installation of motor servo on arm frame

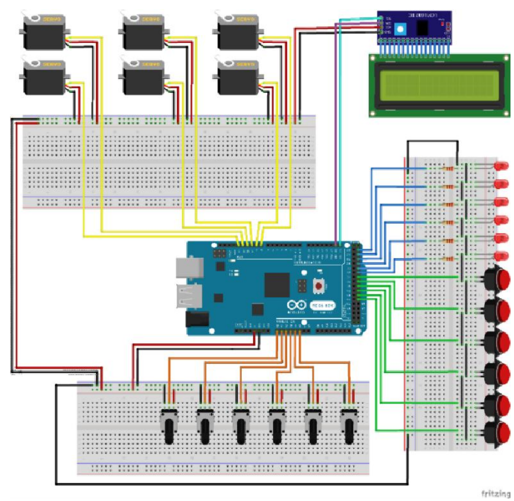

Figure 3: Wiring diagram

Figure 3 shows how to install components on the Arduino Mega 2560. The potentiometers as an input devices that transmits analog data are connected to the analog pins (A0 to A5). Pushbuttons as digital inputsare connected to the digital pin (pins 22, 24, 26, 28, 30, 32). LEDs as digital outputs are connected to the digital pins (pins 23, 25, 27, 29, 31, 33). The LCD used is added with the I2C module so that it is sufficient to be connected to the SCL and SDA pins. Servo motors require pulse data as motion reference, they are connected to the PWM pins (pins 8 to 13).

2.3 Flow Chart Diagrams

Figure 4 shows the flow chartto build source code of this research. There are two sub programs. The first one is called Record that is used to record robot arm movement manually. The movement is determined by potentiometer. The second sub program is called Replay. It is used torepeat the robot arm movement automatically by retrieving data from repository.

When the sub program Record is called, the system will read the analog potentiometer data (Potentiometer 1, 2, 3, 4, 5, 6) and map the analog data from 0 - 1023 to 0 - 258. The results of the analog data are sent to the motor servo

(servo 1, 2, 3, 4, 5, 6) and displays the data on the LCD screen and saves the data to the specified EEPROM address. EEPROM address will be added 6 automatically at each time to save data.

When the Replay sub program is called, the system will retrieve data at the specified EEPROM address (A1, A2, A3, A4, A5, A6). The data value that has been taken is sent to the servo motor according to the specified address (A1 to servo1, A2 to servo2, A3 to servo3, A4 to servo4, A5 to servo5, A6 to servo6) and displays the data on the LCD screen. If the EEPROM data retrieval is not read 90 ((A1, A2, A3, A4, A5, A6) = 90), then data retrieval based on the EEPROM address is added 6 ((A1, A2, A3, A4, A5, A6) plus 6) and repeat the data retrieval command. But if the EEPROM data reads 90, the system will call the direct sub program and stop the replay sub program.

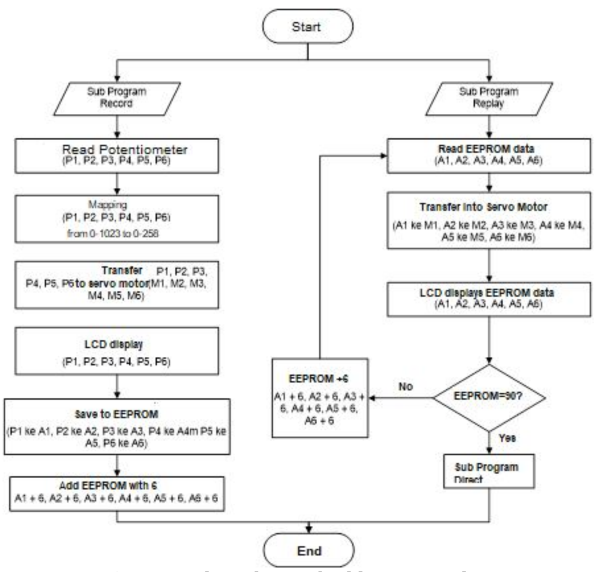

Figure 4: Flow chart to build source code

2.4 Servo Motor Speed Control

In controlling the speed of the servo motor, it must be known first how far the turning angle will be done. The following formula is used to find out the turning angle.

$$\Delta T = T_1 - T_0$$

where, ΔT : angle difference
T_1 : angle at the end of movement/final angle
T_0 : angle at thebeginning

Servo motor rotates step by step. Every step is 1 pulse or 1 degree. In order to adjust the rotating speed of the servo motor, the duration of each stephas to be adjusted. The duration of each step must also be adjusted according to the duration of data recording or dataretrieving. The duration of each data recording or retrieval is 1000 mS, then the formula is obtained to determine the duration of each step of the angular motion of the servo motor as follows.

If ΔT<0, then

$$w = \frac{z}{(\Delta T - (\Delta T \times 2))}$$

If ΔT>0, then

$$w = z/\Delta T$$

Where, ΔT : angle difference (Pulse)
w : duration each pulse (mS/Pulse)
z :Total time to reach the final angle (1000 mS)

Once the duration of each pulse obtained, the servo motor can be adjusted according to the speed at the time of data recording or retrieving which is every 1000 mS. The program to set the servo motor speed is as follows:

```
if (ΔT >= 0) {
   for (posisi1 = T0; posisi1 < T1; posisi1 +=1) {
     servo1.write(posisi1);
     delay(w);
   }
if (ΔT <= 0) {
   for (posisi1 = T0; posisi1 > T1; posisi1 -=1) {
     servo1.write(posisi1);
     delay(w);
   }
```

3. RESULT AND DISCUSSION

3.1 Allocation of Data Saving

The system developed has 6 potentiometers and 6 servo motors. Based on this fact, the system requires 6 storage addresses in once data retrieval. When the system is recording manual motion, the system retrieves data and stores that data every 1000 mS. In this research, EEPROM of Arduino Mega is used as repository. Therefore, it is necessary to do a test to read the maximum EEPROM address on Arduino Mega. This test is necessary in order to determine the maximum duration of data recording.

The test has been conducted and the result is shown in Figure 5. The result show that the Arduino Mega EEPROM has 4096 addresseswhich are allocated in number from 0 to 4095. In this case, Arduino Mega can record up to 682000 mS or it can take data for 682 times.

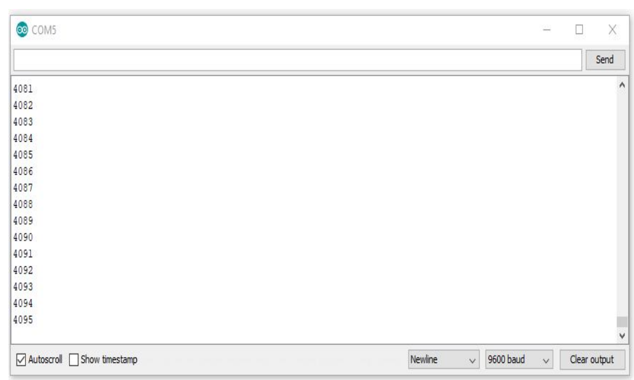

Figure 5: Result of testto read the maximum EEPROM address on Arduino Mega

3.2 Output Voltage and Angular Movement Relationship of Potentiometer

Analog pins of Arduino have voltage range in between 0 to 5 volts. On the other hand, potentiometer has angular movement in between 0 to 300 degree. In this study the potentiometer is acting as asensor. Potentiometer output voltage is used as a reference data to determine the movement of servo motor. In this case, it is necessary to have the relationship in between angular movement of potentiometer and output voltage. Measurement was conducted and the result is shown in Figure 6.

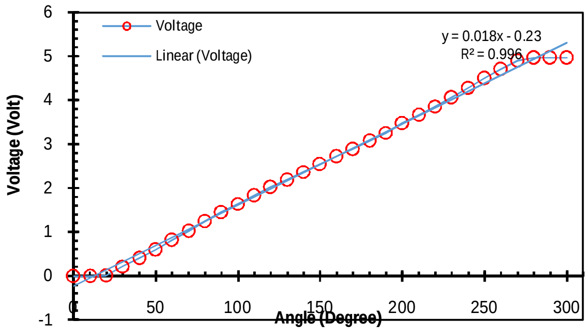

Figure 6. Relationship in between the output voltage and the potentiometer angle

Figure 6 shows that there are two variables X and Y. The variable X is the angular movement of the potentiometer and the variable Y is the output voltage of the potentiometer. From these data it produces a linear regression equation as follows:

$$Y = 0.0185x – 0.23$$

From the results, it can be said that each increases of potentiometer rotation angle by$1^{O}$the potentiometer output voltage increases by 0.0185 Volts.While the coefficient of determination (R Square) from the linear regression in Figure 6 is 99.63%. So the level of nonlinearity from the results is 0.37%.

3.3 Mapping Arduino Reading into Angular Movement

The Arduino reading on its analog pin will have the values from 0 to 1023. It means the angular movement of potentiometer as sensor will result in value from 0 to 1023. The conversion should map from 0-1023to 0-300 since the angular motion of potentiometer is from 0^{O} to 300^{O}.However, examination needs to be conducted to have proper mapping. In this work, some mapping strategies are examined. The mapping strategies are:

1. 0 – 1023 to 0 – 300
2. 0 – 1023 to 0 – 295
3. 0 – 1023 to 0 – 280
4. 0 – 1023 to 0 – 265
5. 0 – 1023 to 0 – 258

The examination result is shown in Figure 7. It shows the proper mapping strategy is the mapping from 0 – 1023 to 0 – 258. Although it shows the asynchronous between the angle of the potentiometer and the Arduino reading. This asynchronous caused a gap of 20^{O}. The gap of 20^{O} is consistent in the range of 0^{O} to 220^{O}. This angle range can be used to control the servo motor with an adjustment of -20^{O}.

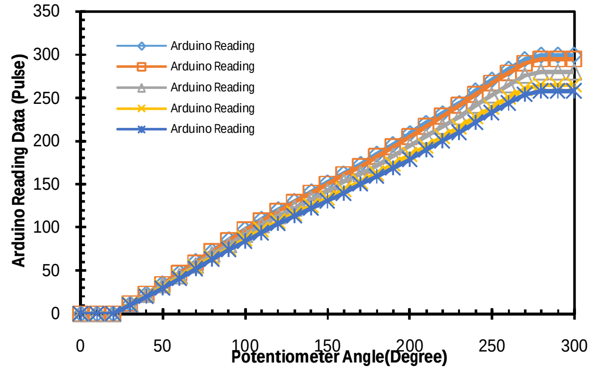

Figure 7: Result of mapping strategy examination

3.4 Mapping Strategy and Angular Motion of Servo Motor MG995

Once the proper mapping strategy is obtained, it can be used as a reference value for the angular motion of the MG995 servo motor. The relationship between the proper mapping and angular motion of MG995 servo motor is shown in Figure 8.

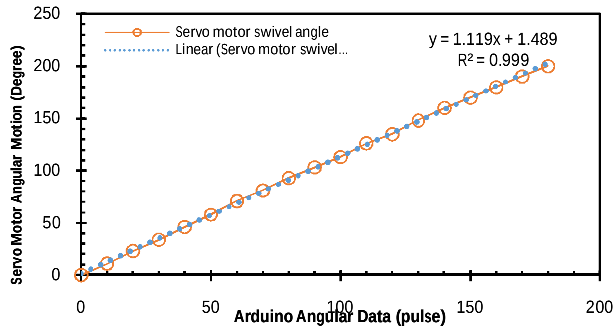

Figure 8: The relationship between the proper mapping strategy and angular motion of MG995 servo motor

From the graph in Figure 8 we get a linear regression equation as follows

$$y = 1.1191x + 1.4895$$

It can be said that each Arduino pulse data increases by 1, the angular motion of the servo motor increases by 1.1191 degree.While the coefficient of determination (R Square) is 99.95%. So the level of nonlinearity is 0.05%.

3.5 System Testing

When the system is recording a movement manually, its controller will read analog input from the pins where potentiometers are attached. In this case, potentiometer represents manual input to have manual angular movement. The results will be mapped from 0 to 1023 to 0 to 258. The mapping results will be the reference for servo motors to have angular motion. At the same time the mapping results will be saved in Arduino Mega EEPROM. Later the saving data can be retrieved to have automatic angular movement of servo motor. Figure 9 shows how the system record manual movement and replay the record automatically. It can be seen that the data record and the replay are exactly the same.

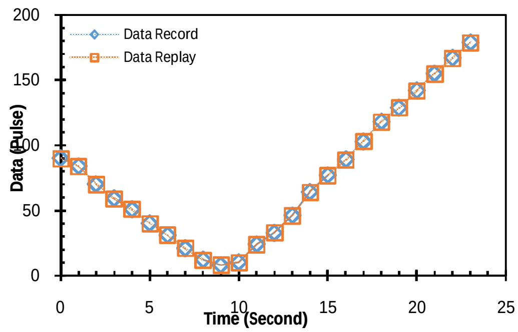

Figure 9: Testing result

4. CONCLUSION

The results show that the Arduino mega 2560 EEPROM can record data for duration of 682000 mS. The 10 kΩ potentiometer has non-linearity of 0.37% and proper mapping is 0-1023 to 0-258 by shifting the initial angle of 20^{O}. The MG995 servo motor has non-linearity of 0.05%. The system established can record the movement and applied to the robot arm accurately.

REFERENCES

1. B. Singh, N. Sellappan and P. Kumaradhas, **Evolution of Industrial Robots and their Applications**, *International Journal of Emerging Technology and Advanced Engineering*, pp. 763-768, 2013.
2. R. Shah and A. B. Pandey, **Concept for Automated Sorting Robotic Arm**, in *2nd International Conference on Materials Manufacturing and Design Engineering*, Gujarat, India, 2018. https://doi.org/10.1016/j.promfg.2018.02.058
3. P. Kah, M. Shrestha, E. Hiltunen and J. Martikainen, **Robotic arc welding sensors and programming in industrial applications**, *International Journal of Mechanical and Materials Engineering*, pp. 1-16, 2015.
4. K. S. Bhokare, **Overview on Industrial Robot, Scope and Application**, *IJSRD - International Journal for Scientific Research & Development*, pp. 508-515, 2017.
5. C. W. Chen, R. M. Hong and H. Y. Wang, **Design of a controlled robotic arm**, in *International Conference on Green Technology and Sustainable Development (GTSD)*, Kaohsiung, Taiwan, 2016.
6. D. D. Dessai, G. B. Gonsalves, M. R. Luis and M. S. Cardoso, **Dark Detector System for Paper Waste Detection**, *International Journal for Scientific Research & Development (IJSRD)*, pp. 873-875, 2017.
7. L. Louis, **Working Principle of Arduino and Using it as a Tool For Study and Research,***International Journal of Control, Automation, Communication and Systems (IJCACS)*, pp. 21-29, 2016. https://doi.org/10.5121/ijcacs.2016.1203
8. A. Nayyar and V. Puri, **A review of Arduino board's, Lilypad's & Arduino shields,** in *International Conference on Computing for Sustainable Global Development*, New Delhi, 2016.
9. O. M. Alsmadi, A. A. Al Jallad, Z. S. Abo-Hammoud and F. J. Al Majali, **Arduino-Based Automatic Safety Vehicle Control**, in *2nd International Conference on Research in Science, Engineering and Technology*, Dubai, 2014.
10. Supriyono, I. Mawardi and W. A. Siswanto, **The Development of Engine Control Module Manipulator Module Based on Arduino to Increase Power and Torque of Motorcycle Engine**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3135 - 3139, 2019. https://doi.org/10.30534/ijatcse/2019/76862019
11. L. A. Bulan, L. R. Maderazo and P. B. Asi, **Design and Development of Microcontroller-based Air Conditioning Units Controller as Input to Energy Conservation for University of Batangas**, *International Journal of Advanced Trends in Computer Science and Engineering*, pp. 1020 - 1025, 2019. https://doi.org/10.30534/ijatcse/2019/06842019
12. R. R. Kanchi and N. Gosala, **Design and Development of an Embedded System for Testing the Potentiometer Linearity**, *Sensors & Transducers Journal*, pp. 107-117, 2010.
13. A. Bhargava and A. Kumar, **Arduino controlled robotic arm**, in *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2017.
14. P. Bora and V. Nandi, **Low cost shadow function based articulated robotic arm**, in *International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, Shilong, India, 2015.
15. E. B. Mathew, D. Khanduja, B. Sapra and B. Brushan, **Robotic arm control through human arm movement detection using potentiometers,** in *International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, Noida, India, 2015.
16. K. Ghadge, S. More and P. Gaiwad, **Robotic Arm For Pick and Place Application**, *International Journal of Mechanical Engineering and Technology (IJMET)*, pp. 125-133, 2018.
17. L. Niranjan, A. Suhas and B. Sreekanth, **Design And Implementation Of Robotic Arm Using Proteus Design Tool And Arduino-UNO,***Indian Journal of Scientific Reasearch*, pp. 126-131, 2018.
18. S. Mitra, S. Biswas and M. Aus, **Arduino Based Bluetooth Operated Car Wiping Technique Using Android Mobile Phone,***International Research Journal of Engineering and Technology (IRJET)*, pp. 1794-1797, 2017.
19. A. Purkayastha, A. D. Prasad, A. Bora, A. Gupta and P. Singh, **Hand Gestures Controlled Robotic Arm,***Journal Of International Academic Research For Multidiciplinary*, pp. 234-240, 2014.

20. T. VanHuy, D. T. Minh, N. P. Kien and T. A. Vu, **SimpleRoboticHand in Motion Using Arduino ControlledServos**, *International Journal of Science and Research (IJSR)*, pp. 972-975, 2015.
21. V. Oza and P. Mehta, **Arduino Robotic Hand: Survey Paper**, in *International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, India, 2018. https://doi.org/10.1109/ICSCET.2018.8537312
22. A. Soetedjo, I. Somawirata and A. Irawan, **Human Arm Movement Detection Using Low-Cost Sensors for Controlling Robotic Arm,***Journal of Telecommunication, Electronic and Computer Engineering*, pp. 43-46, 2018.
23. W. M. H. W. Kadir, R. E. Samin and B. S. K. Ibrahim, **Internet Controlled Robotic Arm,** in *International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)*, Johor, Malaysia, 2012.
24. S. S. S. M. Soleh, M. M. Som, M. H. A. Wahab, A. Mustapha, N. A. Othman and M. Z. Saringat, **Arduino-Based Wireless Motion Detecting System**, in *IEEE Conference on Open Systems (ICOS)*, Langkawi Island, Malaysia, 2019.