Selective Compliance Assembly Robotic Arm (SCARA) Trainer for Engineering Students

Roviel Manfre C. Muldong [1], Salvador A. Loria Jr. [2]

[1]Student, Graduate School, Nueva Ecija University of Science and Technology, Cabanatuan City, Philippines.

[2]Faculty, Graduate School, Nueva Ecija University of Science and Technology, Cabanatuan City, Philippines.

Corresponding Author: rovielmuldong@gmail.com

Abstract: - Robotics is a sizeable evolving field when we talk about technology. It captures every attention when this knowledge comes into action. This work aims to construct and program a Selective Compliance Assembly Robotic Arm (SCARA) Trainer that can be used as an instructional package to teach robotic programming and its application. The system offers a such feature that can simulate programs in real-time. It can also perform instructions involving peripheral devices such as sensors, switches, and indicators. It will also acquaint users with the blending work of mechanical structure and electronic circuitry, or what is known as mechatronics. In general, this system is how the researcher shows the technological advancement in the field of robotics, mechatronics, and programming. The group wants to contribute to how engineering students can be exposed to programming robotics and its applications.

Key Words: — *Robotics, SCARA, Trainer, Module, Mechatronics, Electronics, Engineering.*

I. Introduction

Robots are machines that have interested the general population throughout history. People have wanted to use such equipment since simple devices were developed. The word robot comes from the Czech word robota, which means "servitude, forced labor" and was coined in 1923 (from dictionary.com). Since then, robots have been characterized as machines that look like humans. However, robots are more than machines that walk around yelling, "DANGER!". They are used in various tasks, from exciting space exploration to everyday work. They are complex and valuable machines that were employed in different industries.

Accompanying robots in the trend of technology is programming. Writing a program means writing instructions (apl.jhu.edu).

Manuscript revised November 10, 2022; accepted November 11, 2022. Date of publication November 12, 2022.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

As the programming becomes more creative and complex, the instructions to be executed by the programmed equipment will be as serviceable as programmed. A unit that can be programmed and can act as a brain for equipment behavior is a Microcontroller. A microcontroller is essentially a single-chip controller. The critical feature, however, is the microcontroller's capability to upload, store, and run a program, and it is easy to embed into electronic circuit designs (Carandang et al., 2009).

Technology is changing how we learn, how we interact, a how we think. Consequently, there is an increasing concern in teaching interdisciplinary, focusing on a particular subject or problem by simultaneously incorporating learning from multiple, traditionally separate, disciplinary perspectives.

The incorporation of robotics into the curriculum facilitates the curiosity of both the educator and the students alike. For example, this robot can be incorporated into traditional disciplinary learning activities, particularly programming, electronics, physics, etc.

Nowadays, robots are widely used in manufacturing, commercial establishments, home, and academic firms. However, Robotics remains uncommon for students due to a lack of knowledge of the subject.

The Selective Compliance Assembly Robotic Arm (SCARA) robot is an introductory module applicable to engineering students. It is a particular subclass of the robot family.

With the help of a SCARA trainer, the students will gain basic yet functional knowledge about robotics, which will be helpful in future applications. Also, this project can be integrated into the subject of Power Electronics, or if the curriculum is updated, Robotics can also be combined with Instrumentation and Control Engineering.

1.1 Statement of the Problem

General Problem: How to construct and program a SCARA Trainer that can be used as an instructional package to teach robotics programming and its application.

From the general problem stated, the following specific problems were also realized:

- Lack of exposure to robotics and its applications.
- Absence of equipment/materials in robotics.
- Lack of capability in SCARA programming.
- Insufficient knowledge of mechatronics.

1.2 Objective of the Study

General Objective: To program and construct our SCARA Trainer that will aid instructors in teaching robotics and help the students acquaint themselves with robotic programming and its applications. From the general objective stated, the following specific objectives were also conceptualized:

- To create a SCARA programming module that will be used in learning robotics programming.
- To be able to interface sensors on the SCARA Trainer programming.
- To learn basic principles of mechatronics.
- To expose the students to robotic programming and applications.

1.3 Scope and Limitations

The SCARA Trainer uses several motors that provide movement on a different axis. It can be programmed by the instructor or student the way they desire it and use sensors as an interface. It can also be used for the students to learn the mechanical assembly of the trainer. A module is provided for the usage instructions and basic programming lessons.

The SCARA Trainer has the following scope and limitations:

- It has two sensors motion sensor and a light sensor.
- Can only be used for low-power applications at a maximum weight load of 50 grams for the robotic arm.
- The rotation of each servo motor is limited to 180 degrees.
- There's a limited number of potentiometers and data switches; ten data switches and four potentiometers.
- The analog control of the SCARA tends to be unstable.

II. Methodology and Research Design

The research method used in the study is the following:

- *Descriptive method,* which uses a survey that identifies specific reactions and suggestions for the trainer.
- *Library and Online Research method* on related literature from Holy Angel University and De La Salle University libraries; and various web journals from the Institute of Electrical and Electronics Engineers (www.ieee.org) and journals from one of the best blogs for electronics engineering students and professionals (www.electronicshub.org).

2.1 Data Gathering Instruments

The researchers used different data-gathering instruments and techniques to gain further knowledge on the content of the proposed study. Data from other sources are essential to prove the feasibility of the proposed system.

- *Observation* - how students use the SCARA module and trainer, which improves their programming skills.
- *Interview* - which provides clarity of any information regarding the project. The researchers interviewed Engr. Herbert Parungao, Beverage Production and Maintenance Supervisor at Universal Robina Corporation; Engr. Anthony Tolentino, ECE Faculty at Don Honorio Ventura State University; and Engr. Mark Quintana, Mechatronics Laboratory Custodian at Holy Angel University, has ideas and is knowledgeable about the subject matter. A series of questions were asked to gain insights that would help construct the project.
- *Questionnaires* – a set of questions to evaluate reactions, suggestions, and deficiencies that need to be pointed out in the SCARA trainer. These questions were given to a few selected persons to obtain data for a survey.
- *Evaluation Form* – to measure how clearly, they had discussed the study's operation, principle, and application during the presentation and actual testing.

2.2 Method Used in Developing the System

In the developmental stage, the project study suggests many approaches to analyzing and characterizing different faces of methods, including the design, structure content, and project use. Additional phases are involved based on the concept, describing how the proposed study is developed and created.

In the planning phase, the team defines the solution in detail, what to build, how to make it, who will build it, and when it will be built. During this phase, the team works through the design process to create the solution in construction and design, writes the functional specification, and prepares work plans, cost estimates, and schedules for the various deliverables.

The first thing the proponents prepared was a timetable of things to be done and when to do them. The proponents looked for an idea from previous robotic arm models and programmed robots where they could gather data regarding the prototype they are building; gathered data serves as the basis of their design. After data was collected, the proponents proceeded to design the prototype, where the availability of the materials needed was first considered.

2.2.1 Design Phase

In this stage, the proponents were constructing the prototype. After analyzing what architecture is expected, the proponents arrived at components such as the poly glass for the structural body base and the track for the base movement of the robotic arm. Servo motors are motors that proponents decided to use for the body joints of the robot. The Tormax dc motor is used for the robot's movement on its base, accompanied by its circuit driver to control the behavior of the dc motor.

A microcontroller unit circuit is used for the project's programming side and the peripheral sensors' attachments to the MCU. A power supply that supplies the whole system is also constructed for the proper design of the system.

The PIC16F877A is used as the central processing unit of the SCARA Trainer. It is powered by a +5VDC power supply and has 32 I/O ports where servo motors and other interface modules, such as sensors and USART, are connected. It is responsible for sending instructions to the SCARA and receiving pulses or triggering from the sensors and switches. It also drives peripheral modules such as the LED, seven-segment displays, and buzzer.

There is a total of 7 servo motors used in this project. All serve for the different degrees of freedom of the SCARA: one servo motor for the gripper, one for the wrist rotation, one for the wrist bending, one for the elbow bending, two for the shoulder bend, and one for the shoulder rotation.

The Servomotor's movements are manipulated by a pulse width modulated (PWM) voltage produced by the MCU at the motor's control terminal. PWM is a time-variant signal; the duration of square waves of the PWM applied to the servo motors and the frequency used in programming determines the degree of movement of the engines. The researchers have used 3 Servo HS325HB, 1 Servo SG5010, and 3 Servo MG995.

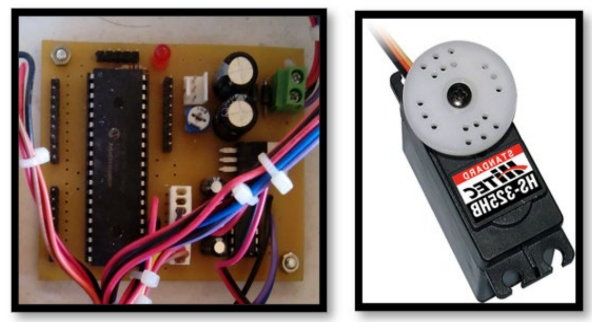

Fig.1. PIC16F877A Microcontroller (left), hs325hb servomotor (right)

The e PIC-Kit 2, which consists of the e PIC-Kit 2 software and hardware, is used in programming the MCU. By connecting the e PIC-Kit 2 to the on-circuit-programming connection of the MCU module and the USB port of the PC/Laptop, the PIC16F877A microcontroller can be programmed/reprogrammed using a HEX file imported and written utilizing the e PIC-kit 2 tool software.

The SCARA Trainer uses the USART interface to establish a connection between the PC/Laptop to the MCU module. Using a MAX232 circuit and USART to USB connector to connect the MCU to the PC/Laptop, the user can control the SCARA using a graphic user interface created using visual studio on its PC/Laptop.

Motion and light sensors are interfaced in this project so that the SCARA can perform selective compliance in some events. For example, limits or conditions are desired to be configured to the SCARA when the SCARA meets the certain condition sets by the sensor, shall comply with that condition, and perform appropriate actions.

The motion sensor will be triggered once a movement is detected near the motion sensor module. In comparison, the light sensor will provide a pulse triggering the MCU with a specific light intensity applied on the LDR.

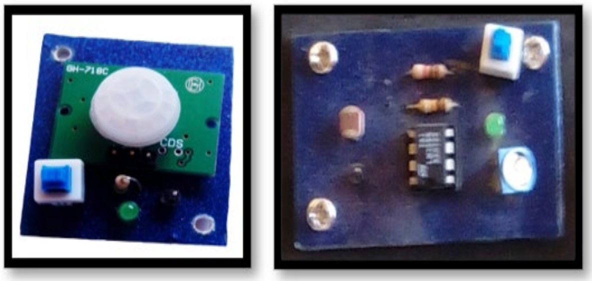

Fig.2. Motion Sensor (left), Light Sensor (right)

The following are other peripheral components of the SCARA trainer and their functions.

Potentiometers use analog-to-digital converter functions of the MCU, whereas they can be programmed to control the movement of SCARA by turning their shaft.

The data switches can trigger a single servo motor movement or start movements of the SCARA.

Fig.3. Potentiometer

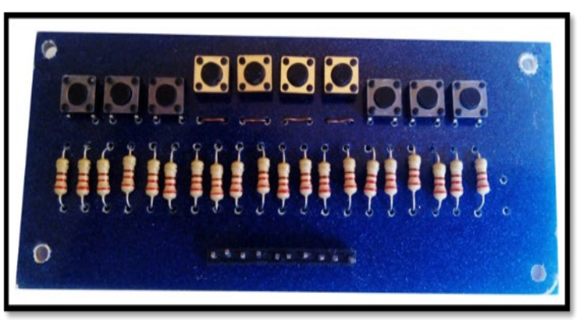

Fig.5. Data Switches

The buzzer is used on the trainer as a notification for starting or accomplishing SCARA Movement/Action.

These optoelectronics components of the SCARA Trainer provide visual indications, signaling, and countdowns for the SCARA. They can be programmed to produce light signals as the SCARA is functioning.

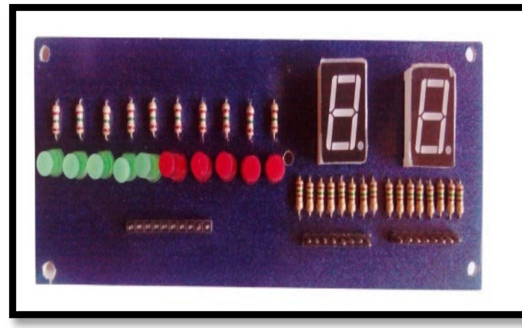

Fig.5. LED and Common Cathode Seven Segment Display

The breadboard is added to the SCARA modules so the users can integrate additional components and circuits on the trainer.

2.2.2 Development Phase

The development phase involves improving and enhancing the project until the desired output is achieved. For the design of the whole architecture of the project, Google sketch up PRO is used. On the other hand, Proteus software is the circuit simulator for the system. Proton IDE is the programming language used by the proponents, and in tandem with the programming language is the PICkit2 program compiler to compile programs to the microcontroller. Moreover, the cutting of poly glass was made by a Computer Numerical Control machine (CNC) to accurately cut the poly glass that will provide the skeleton of the robotic arm.

2.2.3 Testing Phase

The testing phase identifies the potential errors in the system. Using this phase, the researchers detected failures or malfunctions in the system's program and the hardware. Testing was done to guarantee that the system's performance had achieved its objective. The desired output can be achieved by adjusting and adapting the project.

At first, simulations for the motor of the robotic arm are accomplished to identify and correct the possible malfunctioning of each motor concerning the desired operation of the robotic arm trainer. Motors are primarily given consideration specifically on their torque specifications, whether the torque value of the motors is appropriate to achieve the desired output operation. Next, testing is on each part of the robotic arm to determine whether they work correctly in different scenarios.

2.2.4 Maintenance Phase

The researchers made a flexible system capable of adapting changes for future improvement and can be assured using a maintenance process. Upon the said process, the system will maintain its effectiveness in its operation.

For the maintenance of the SCARA Trainer, there should be a necessary regular check-up of the action of each motor to determine whether they are still working smoothly or not. In some instances, there is a need to put a tiny amount of grease in the servo gears to avoid misalignment that may lead to the breakage of the teeth of the gears.

Wirings must also be checked, especially those on each servo, because of the frequent action of the SCARA Trainer. There might be unexpected desoldering of wires for such instances.

Applying a power supply with an unknown current and voltage rating on the peripheral module must be avoided. This might bust the components, and large-scale replacement will be needed.

III. Presentation Of Data, Test, And Results, And Interpretation Of Data

The final components of the proposed system as per the design phase are the following: PIC16F877A Microcontroller, Servo Motors, 5V Current Servo Amplifier, Tormax DC Motor Driver, e PIC-kit 2, USART Interface, Motion Sensor, Light Sensor, Potentiometers, Tact Switches, Light Emitting Diodes (LEDs), Common Cathode Seven Segment Display, Breadboard, and a Buzzer.

The proposed study is a SCARA Trainer, which can be programmed using a PC/Laptop to perform a specific action. It uses an e-PIC-kit 2 to program the MCU with the desired function. It also has GUI software wherein the SCARA can be controlled in real-time. The trainer has peripheral modules which can be interfaced with the MCU and then to the SCARA to execute the desired action.

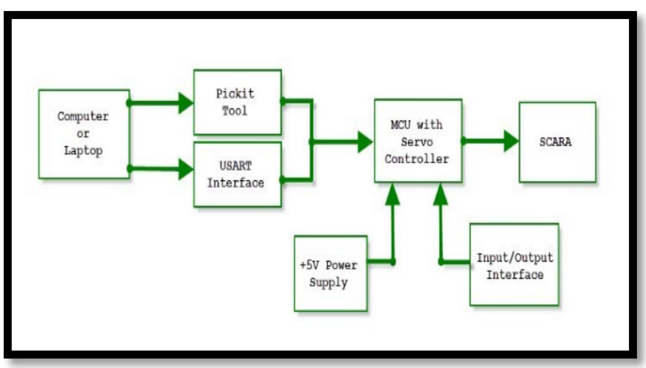

Fig.6. Block Diagram for SCARA trainer

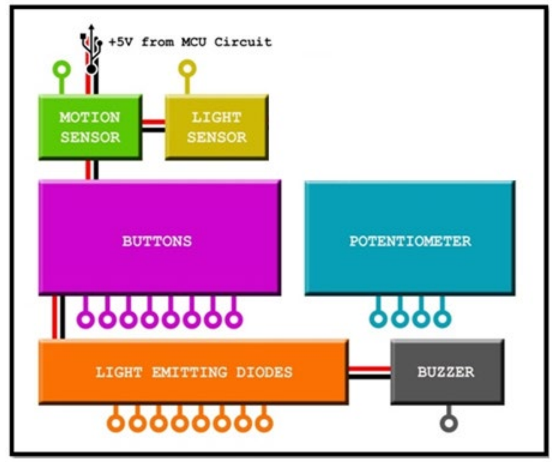

Fig.7. Block Diagram for Peripheral Module

3.1 System Development

In establishing the project, the succeeding components are considered part of the system development process and requirements for the proposed project's completion. The proponents gathered information and data about these components and discussed their application to the project and how they would work in the system.

3.1.1 Design

The proposed project's design is conceptualized to create our SCARA trainer and module used by Electronics Engineering and Electrical Engineering students and other related courses that tackle the subject of Robotics. The plenum search for different types of robotic arms, their construction, and their component led to the creation of a SCARA Trainer. The trainer has a robotic arm with seven degrees of freedom, reprogrammable PIC16F877A MCU, and peripheral modules. A +5VDC power supply powers it.

Establishing the Robotic Arm - the design of the robotic arm has seven servo motors providing seven degrees of freedom. And an additional engine for the robotic arm's forward and backward movement was added. Three HITEC HS-325HB Standard Ball Bearing Servo, one Servo SG5010, and three Servo tower Pro MG995 were used for the robotic arm, while one TORMAX DC motor was used for the forward-backward movement. The frame of the robotic arm was created using polymer glass cut using CNC cutting technology. The MCU module controls the robotic arm, where the control pins of all servo motors are connected to a servo motor amplifier that amplifies the control signals generated by the MCU.

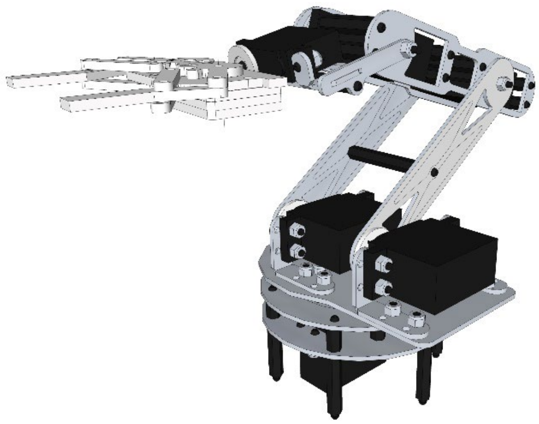

Fig.8. SCARA Robotic Arm

Establishing the Microcontroller Module - MCU is the brain of the system. The instructions are stored in the Electrically Erasable Programmable Read Only Memory (EEPROM). It reads using Central Processing Unit (CPU) and initiates the action programmed on the microcontroller. The MCU module comprises a PIC16F877A IC, MAX232 circuit, and a +5VDC and 12VDC power supply. The MCU module's design also has an on-circuit-programming pin allowing the PIC16F877A to be programmed easily by connecting the e PIC-Kit 2. The power supply on the MCU will provide power to the peripheral components and the robotic arm with a servo amplifier. The design used for the MAX232 circuit is the schematic provided in the MAX232 IC datasheet, which connects the MCU to the PC/Laptop.

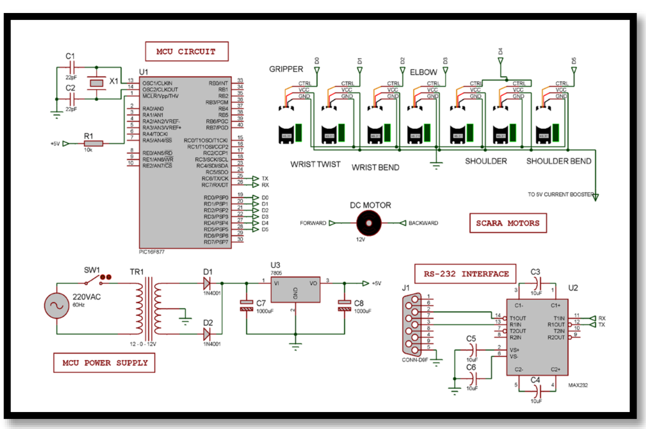

Fig.9. Microcontroller Module with Power Supply and RS232 Interface

Establishing the Peripheral Modules - The researchers had come up with SCARA peripheral modules that will aid in attaining the objectives of the project. They searched for different sensors that could be interfaced with the trainer, which led to the development of the following sensors: motion and light. They also added components that will provide auxiliary functions for the SCARA: a buzzer, four potentiometers, ten LEDs, two common cathode seven-segment displays, and ten data switches. A breadboard is also provided for integrating other components into the SCARA.

Establishing the Training Module - The project has a training/programming module that uses Proton IDE as the programming software and Proteus ISIS for the simulation. The module tackles both SCARA programming and mechatronics with each programming exercise. The programming exercises are the following:

Programming Exercise No.1 - Driving a Servo Motor
Programming Exercise No.2 - Familiarizing the SCARA and Multiple Servo Control
Programming Exercise No.3 - Creating a SCARA Trainer Movement Sequence
Programming Exercise No.4 - SCARA Control Using Data Switches
Programming Exercise No.5 - SCARA Trainer Control Using Potentiometers
Programming Exercise No.6 - Interfacing the SCARA with a Buzzer
Programming Exercise No.7 - SCARA Trainer LED and Seven Segment Interface
Programming Exercise No.8 - SCARA Trainer Sensor Interface
Programming Exercise No.9 - SCARA Control Using Serial or USART Interface

3.2 Testing

Testing is divided into three: robotic arm, peripheral modules, and graphic user interface (GUI). The hardware is also tested with their respective SCARA programs and interfacing the SCARA to the peripheral modules.

3.2.1 Robotic Arm Testing

The robotic arm was tested using motors, power supplies, and servo motor controllers. At first, the researchers used a two amperes power supply for the robotic arm, which proved insufficient to drive all the servo motors. Learning that each servo must have at least 500 milliamperes, the proponents shifted to six amperes to ensure that all eight motors would be provided with enough current.

Primarily, we used the HS-325HB servo motors for the robotic arm and the TORMAX DC motor for the base, but upon testing, the motors on the shoulder part could not lift the entire arm properly. As a result, we replaced it with larger servo motors. But the result was the same, so we focused on changing the servo motor controller. In the first test, we used E-Gizmo Servo Motor Controller, which sends pulses on the servo one at a time, causing one servo to lock to its position. Still, after controlling the different motors, the previous motor unlocks, and the current motor locks up. With this condition, the unlocked servo motor may not be able to carry the robotic arm and its load. We resolve this problem by directly connecting the supply of each servo to the power supply and the control signals to the microcontroller's output pins.

We could control all servo motors during the final testing, including the TORMAX DC motor. We pre-programmed a pick-and-place routine that carries a maximum 300g load using a six-ampere transformer, the same servo motors, and a servo motor controller using the PIC16F877A microcontroller.

3.2.1.1 SCARA Trainer Load Capability Testing - To determine the maximum load in terms of weight the SCARA Trainer can carry. This is done by Carrying the different weights of loads on each program script routine the robotic arm will execute, providing the time it will take, from picking up to dropping the load, for the robotic arm to accomplish a particular routine for the specific weight of the shipment.

Table.1. Data and Results in Testing the Pick and Place Routine 1

Test No.	Load	Time 1	Time 2	Time 3
1	20g	34.42	34.42	34.42
2	30g	34.47	34.47	34.47
3	40g	34.65	34.65	34.65
4	50g	34.87	34.87	34.87
5	60g	34.96	34.96	34.96
6	70g	FAILED	-	-

Table.2. Data and Results in Testing the Pick and Place Routine 2

Test	Load	Routine time	Distance	No. of position
1	20g	27.18s	18 cm	11
2	30g	27.27s	18 cm	11
3	40g	26.86s	18 cm	11
5	50g	27.04s	18 cm	11
6	60g	27.04s	18 cm	11
7	70g	27.22s	18 cm	11

Table.3. Data and Results in Testing the Pick and Place Routine 3

Test	Load	Routine time	Distance	No. of position
1	20g	18.58s	49cm	9
2	30g	18.57s	49cm	9
3	40g A	18.27s	49cm	9
4	40g B	18.76s	49cm	9
5	50g	18.67s	49cm	9
6	60g A	18.45s	49cm	9
7	60g B	26.5s	49cm	9

3.2.1.2 Individual Robotic Arm Joint Load Capability Testing - The Testing is done in each joint in carrying different loads that vary in weight, bending freely at its range of degree of freedom.

Table.4. Data and Results in Testing the Wrist Twist to Gripper (from 500us to 2500us), Wrist Bend to Gripper, and Base to Gripper.

Weight	Assessment
20g	OK at its full range
30g	OK at its full range
40g	OK at its full range
50g	OK at its full range
60g	OK at its full range
70g	OK at its full range

Table.5. Data and Results in Testing the Elbow to Gripper

Weight	Assessment
20g	OK at its full range
30g	OK at its full range
40g	OK at its full range
50g	OK at its full range
60g	OK at its full range
70g	OK from 500us up to 800us of PWM

Table.6. Data and Results in Testing the Shoulder to Gripper

Weight	Assessment
20g	OK at its full range
30g	OK at its full range
40g	OK at its full range
50g	OK from 500us to 1500uS of PWM
60g	OK from 500us to 1250uS of PWM
70g	OK from 500us to 1250uS of PWM

3.2.1.3 Linear Track Forward/Backward Speed Testing - The DC motor is a program to rotate forward and backward for a certain period. The distance it travels then is measured in millimeters. The period of rotation then is varied.

Table.7. Data and Results for Linear Track Forward/Backward Speed

Period of Rotation, T_R (mS)	Distance Traveled, D (mm)	Speed, D/T_R (mm/S)
2500	45	18
5000	98	19.6
7500	148	19.73
8800	178	20.4545

Average Speed = 19.44 mm/s

3.2.1.4 Servo Motor Degrees of Freedom Testing - The test is done by creating different programs that test the parts of the SCARA Trainer. Each program consists of an increasing value PWM determined by the servo command and generated by the microcontroller. The programs will be burned on the microcontroller one at a time. Once a specific part moves, the angle of movement is manually measured using a ruler and a protractor; an 8 sec time interval is provided between each movement.

Each robotic arm joint follows the following servo number assignment: S0 – Gripper, S1 - Wrist Twist, S2 - Wrist Bend, S3 – Elbow, S4 – Shoulder, and S5 – Base.

Table.8. Data and Results in Testing the Servo Motors Degrees of Freedom from 500us to 1500us

Servo	500us	750us	1000us	1250us	1500us
S0	0 °	25 °	45 °	70 °	90 °
S1	0 °	25 °	45°	70°	90°
S2	0°	10°	30°	55°	90°
S3	0°	40°	65°	85°	115°
S4	0°	45°	90°	120°	155°
S5	0°	30°	55°	85°	100°

Table.9. Data and Results in Testing the Servo Motors Degrees of Freedom from 1750us to 2500us

Servo	1750us	2000us	2250us	2500us
S0	90 °	90 °	90 °	90 °
S1	120°	145°	165°	185°
S2	115°	140°	170°	200°
S3	155°	185°	225°	255°
S4	170°	195°	195°	195°
S5	130°	145°	180°	185°

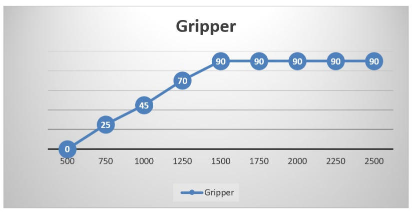

Fig.10. Graph of Gripper Testing. Degrees of Freedom Vs. PWM Signal

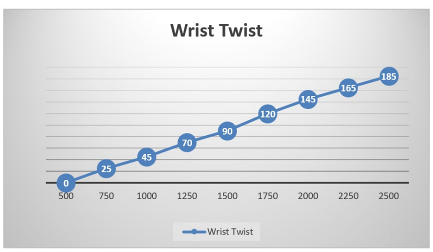

Fig.11. Graph of Wrist Twist Testing. Degrees of Freedom Vs. PWM Signal

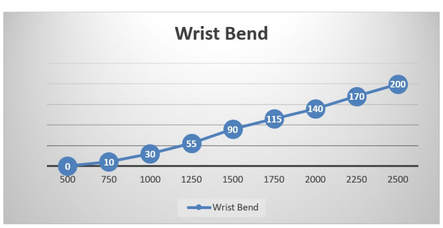

Fig.12. Graph of Wrist Twist Testing. Degrees of Freedom Vs. PWM Signal

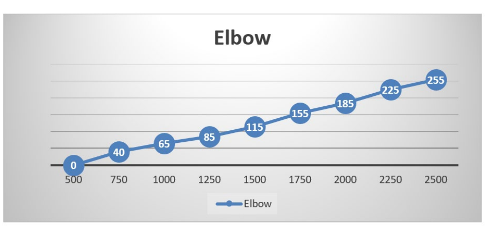

Fig.13. Graph of Elbow Testing. Degrees of Freedom Vs. PWM Signal

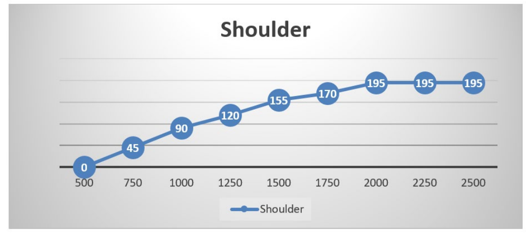

Fig.14. Graph of Shoulder Testing. Degrees of Freedom Vs. PWM Signal

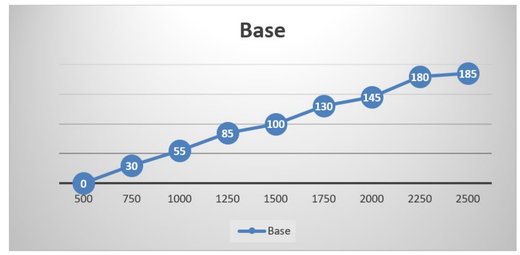

Fig.15. Graph of Base Testing. Degrees of Freedom Vs. PWM Signal

3.2.2 Peripheral Modules Testing

Peripheral modules are tested depending on their functions in coordination with the SCARA trainer. They are manually tested using test circuits with a program burned on the microcontroller.

3.2.2.1 Data Switches and Potentiometer Testing - The SCARA Trainer can be preprogrammed and manually controlled using data switches and a potentiometer. We manipulate the robotic arm using specific buttons, switches 1 and 2 for opening and closing the gripper, switches 3 and 4 for wrist movement, and so on, depending on the microcontroller program. The data switches are only limited to ten. Therefore, we only controlled five individual servo motors with left and proper movement. But depending on the program, it will be able to handle multiple servos at the same time using a pair of switches.

3.2.2.2 LEDs, Seven-Segment Displays and Buzzer Testing - The primary purpose of these peripheral modules is to provide visual and audio indications for the SCARA Trainer. Generally, we tested this module using a 5v supply connected to it and tried a program that triggers the buzzer, LEDs, and seven-segment displays while the SCARA is moving. We have programmed the red LEDs to light when the SCARA is moving and green to indicate it has stopped. The buzzer also rings when the SCARA routine is finished. A zero-to-nine counting program is also tested on the seven-segment displays.

3.2.2.3 Motion and Light Sensor Testing - Programming two ports of the microcontroller test the sensors wherein the sensors are the input triggering of one port that tells the other port to go high when the sensors are activated. This triggering is observed using an LED connected to the second port.

The 2nd test will use the same program but with the SCARA robotic arm as the load; the SCARA movement will depend on what sensor is triggered.

Motion Sensor Sensitivity Test - The PIR motion sensor is tested with movements located at variable distances from the sensor. The length will be measured using a meter stick or other measuring tool. The sensor is covered by a one-inch-high enclosure which is open on the top to make sure that the sensor will only detect movements parallel to that opening and avoid false triggering due to the side sensitivity of the sensor.

Table.10. Data And Results of Motion Sensor Sensitivity Test

DISTANCE	RESULT
20cm	High (On)
40cm	High (On)
60cm	High (On)
80cm	High (On)
100cm	High (On)
120cm	High (On)
140cm	High (On)
160cm	High (On)
180cm	Low (Off)
200cm	Low (Off)

Light Sensor Sensitivity Test - The testing will be based on the distance of a constant light source from the light sensor to determine how far the light source should be to trigger the sensor. Different lengths will also be tested with variable potentiometer resistance; thus, resistance is responsible for the sensor's sensitivity.

The distance is measured using a ruler, while an ohm meter tests the potentiometer resistance. The luminance of the constant light source is determined by its specification sheet.

Table.11. Data And Results of Light Sensor Sensitivity Test

Distance	500Ω	2.5kΩ	5kΩ	7.5kΩ	10kΩ
2cm	High(On)	High(On)	High(On)	High(On)	High(On)
5cm	High(On)	High(On)	High(On)	High(On)	Low(Off)
10cm	High(On)	High(On)	High(On)	Low(Off)	Low(Off)
20cm	High(On)	High(On)	Low(Off)	Low(Off)	Low(Off)
35cm	High(On)	High(On)	Low(Off)	Low(Off)	Low(Off)
40cm	High(On)	Low(Off)	Low(Off)	Low(Off)	Low(Off)

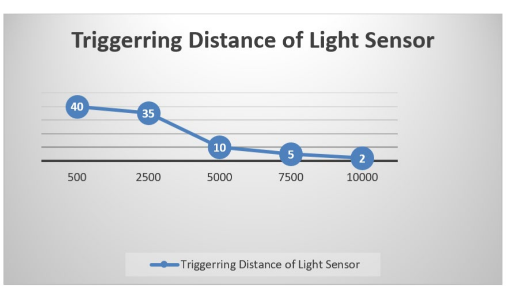

Fig.16. Graph of the Triggering Distance of Light Sensor. Distance (cm) Vs. Resistance (ohms)

3.2.3 Graphic User Interface Testing

In testing Visual Basic 6 GUI, we could control the SCARA Trainer using a PC/Laptop by connecting the USB to USART cable from the D9-connector of the RS-232 Circuit to the USB port of a PC/Laptop.

The GUI comprises sliders for each servo motor, providing real-time SCARA control. It also has saved the position and playback option, whereas it can save each position of the SCARA trainer and play them back with the desired execution speed.

In the testing, the researchers varied the refresh rate of the microcontroller, which determines the sensitivity of the SCARA trainer to the variation on the sliders.

3.3 Troubleshooting

The proponents have found some glitches in some PCB connections or continuity. Some paths intended for close contact are at some points opened, as shown. This causes some components in some circuits or modules to malfunction. Another problem that needs attention is the cold soldering or improper soldering of some components to the PCB that causes the component not to work correctly. Moreover, some busted components are identified and are to be replaced.

In addition, some malfunctions in the motor behavior are identified and considered, such as the contradicting rotation of the motors to each other on some parts of the robotic arm because of the wrong positioning of the motors. Furthermore, checking the polarity of polarized devices needs serious attention to avoid unnecessary damage to the device and the whole equipment itself. The proper designation for each port for the intended use must be accurately done to avoid confusion and malfunctioning of the entire project or the anticipated outcome operation of the robotic arm.

3.3.1 Other Troubleshooting Guides

1. Servo motor does not lock on its position.

a. Power requirement supplied to the servo motor is insufficient, or there is a lack of power supply. Check if the servo motor is connected correctly to the power supply.

b. Check the teeth of the gears of servo motors if they are complete. Change gear/s with a defect if there is/are stock of the same gear/s or change the servo motor.

c. Review your program output configuration to see if the port register of the malfunctioning servo motor is configured correctly.

2. No power on the servo motor.

a. Check the connecting wire to the power regulator. It might be cut or open. Change the connecting wire.
b. Check components at the power regulator circuit to see any burnt or loose components.
c. Check the primary AC power source. Brownout or a temporary voltage drop may be a cause.

3. Servo motors generate too much noise.

a. Check the gears to see if there are any missing tooth/teeth. Breakage of a tooth is caused by too much load carried while in operation. Change or replace the defected gear/s.
b. There may be a tilted gear. Arrange the equipment properly.
c. Check your program's clock frequency if configured to XTAL = 4.

4. Loose SCARA Trainer parts.

a. Check loosed nuts and screws. Tight them.

5. Servo motors do not respond to GUI.

a. Check if the female serial port is correctly interfaced with the serial driver and is properly grounded.
b. Check the cable used. Use other wires available.
c. Open the device manager and search for the used COMM PORT of the USB to serial cable; if not matched with the GUI, change the MSComm1.CommPort value.
d. Try to reset/restart the program and debug if necessary.

6. Forward/Backward movement of DC motor not working.

a. Check if the DC motor is correctly connected to the DC supply.
  i. Check the connecting wires if properly soldered and if there are cold solder pump solder and resolder.
  ii. Check the connecting wire. If open, change the connecting wire.
  iii. Check components at the PCBs of the power regulator and the DC motor driver. Suppose there are any burnt or open changes in such components.
b. If all the solutions above are done already, the problem is with the DC motor itself. Replace the DC motor immediately.
c. Check the program to see if the proper port for the DC motor is configured and if one command does not overlap with another.
d. If data switches are used in controlling the DC motor, make sure that the switch pins are configured as inputs.

7. No power on the peripheral.

a. Check how wires are soldered at the PCB and the power supply – there must be no cold solder. Desolder and solder wire again.
b. Check components at the power regulator circuit to see if there are any burnt or open changes in such components.
c. Check the primary AC power source. Brownout or a temporary voltage drop may be a cause.

8. SCARA Trainer does not respond to a specific peripheral.

a. Check the connecting wire used. It might be open. Change the connecting wire immediately.
b. Check the output of the peripheral. It may not satisfy the required input of the microcontroller, or such a peripheral is not used correctly.
c. Make sure that the MCU port is configured correctly for that peripheral.
d. Don't forget to turn ON the dedicated power switch of some peripheral modules.

9. *Unable to burn the program to ePicKit 2 or unable to complete the program using Proton IDE.*

a. Check the installation of ePicKit 2 to the microcontroller circuit and the PC. It must be installed appropriately.
b. Unplug and plug ePicKit 2 into the PC and the microcontroller circuit.
c. Rename the Proton IDE program file and the folders where the program is addressed. Make them shorter. Too long source addresses make the ePicKit 2 unable to burn the program.
d. Do not use numbers in naming your Proton IDE program.
e. Make sure that the import path is correctly set.

IV. CONCLUSION

Upon completing the proposed study, the proponents had proven the achievements of their objectives.

The proponents had built a programmable and instructional robotic arm entitled the Selective Compliance Assembly Robotic Arm (SCARA) Trainer. And with the use of the SCARA training modules composed of ten sample programs with their respective objectives, discussions, exercises, and applications, the students could learn the basic principles of mechatronics and the applications of robotics in the industry incorporated with training in the field of programming.

With the knowledge acquired in the training module, the students may be able to create their programs and use the different peripheral modules to instruct and explore the SCARA Trainer. The GUI provided using visual studio software has preprogrammed instructions to be embedded in the microcontroller to guide the user in familiarizing themselves with the robotic arm behavior, and a choice of reprogramming the instructions given at the GUI specified is for the exploration and training of the user.

The peripheral modules consisting of sensors, buzzers, data switches, potentiometer, LEDs, breadboard, and seven-segment displays can be interfaced on the SCARA trainer and program the microcontroller in compliance with each module.

The proponents successfully applied the principles and knowledge learned in the Electronics Engineering course in constructing the prototype. And assured that the components used for the whole system are in average condition and that devices interfaces function well.

Survey results compiled and evaluated show how the respondents highly agree with the project and its implementation.

5.1 Recommendation

a. For the further improvement of the Selective Compliance Assembly Robotic Arm Trainer, the proponents recommend the following:

- Include additional sensors like a color sensor, induction sensor, and audio/sound sensor.
- Create interchangeable replacements for the gripper, such as a video camera, screwdriver, soldering iron, etc.
- Improve the graphic-user interface by adding a feature like actual 3D model simulation.
- Adding more peripheral modules like 4x16 LCD, touch screen interface, Zigbee wireless interface, joystick controller, and keypads.
- Increase the number of data switches and potentiometers; and improve the size of the buzzer.
- Try other programming languages to explore and train more while finding the most appropriate programming language for the user where they will be more comfortable in programming.

References

[1]. Carandang, et al. (2009). Microcontroller-Based Humanoid Robotic Arms with Computer-Aided Programming. (De La Salle University, 2009).

[2]. Vairojanawong (1996). Development of a Selective Compliance Assembly Robotic Arm Using Microcontroller. (Dissertation, De La Salle University, 1996).

[3]. Dee, et. al (2000). Computer – Controlled Robotic Arm Using Fuzzy Logic for Object Gripping. (De La Salle University, 2000).

[4]. Bolos, et. al (2005). Robotic Arm Sensor – Controlled through Actuators and Logic (RASCAL). (De La Salle University, 2005).

[5]. Dizon, et. al (1991). A Programmable SCARA Type Robotic Arm. (De La Salle University, 1991).

[6]. Angeles, et. al (2010). EMG – based Robotic Arm Controller. (De La Salle University, 2010).

[7]. Menegatti& Moro (2010). Educational Robotics from High school to Master of Science. (Journal from University of Padova, Padova, Italy, 2010).

[8]. Fischer, et. al (2009). Development of a Laboratory Kit for Robotics Engineering Education. (Journal from Worcester Polytechnic Institute, Worcester, Massachusetts, 2009).

[9]. Catlin & Robertson (2012). Using Educational Robots to Enhance the Performance of Minority Students. (Journal from California State University San Bernardino, San Bernardino, California, United States, 2012).

[10]. Ruzzenente, et. al (2012). A Review of Robotics Kits for Tertiary Education. (Journal from University of Sussex, United Kingdom, 2012).

[11]. Gareth Branwyn (2003). Absolute Beginner's Guide to Building Robots. United States of America: Que Publishing.

[12]. Bishop, R.H. (2002). The Mechatronics Handbook. New York City: CRC PRESS.

[13]. PROTON + Compiler. Development Suite (2004). England: Crownhill Associates Limited.