Malaysian Journal of Industrial Technology (MJIT), Volume 6, No.1, 2022
eISSN: 2637-1081

MJIT 2022 Malaysian Journal of Industrial Technology

PROTOTYPE DEVELOPMENT: ROBOTIC PALM OIL PICKING ARM

Mohd Nasrulddin Abd Latif.
Unit Academic and Continuing Education, Kolej Komuniti Kemaman, Lot K-148E, Bangunan Majlis Perbandaran Kemaman, Kijal 24100 Kemaman, Terengganu.
Email: nasrulddin@kkck.edu.my

Nurul Ashykin Abd Aziz.
Faculty of Industrial Management, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300, Gambang Kuantan, Pahang.
Email: ashykin.aziz@gmail.com

Mohamad Rohieszan Ramdan.
Faculty of Management and Economics, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak.
Email: rohieszan@fpe.upsi.edu.my

Nor Hafiza Othman.
Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan, Locked Bag 36, Pengkalan Chepa, 16100, Kota Bahru, Kelantan.
Email: hafiza.o@umk.edu.my

*Corresponding author's email: ashykin.aziz@gmail.com

ABSTRACT

Despite being the world's second-largest palm oil producer, Malaysia still has trouble harvesting palm fruit due to a labor shortage. This industry is still highly dependent on labor from other nations, particularly in the plantation sector and activities requiring palm fruit harvesting. This led to the development of an innovative project to improve the harvesting process of palm oil by deploying a robotic arm in the real world. This was done to boost the picking activity's output. Thus, the objective of this project is to develop a robotic arm that is not only user-friendly but also capable of being put to work in the activity of picking palm oil. This forward-thinking idea uses an Arduino microcontroller to run the autonomous suspension system for palm oil harvesting activities. The process of constructing a robotic arm encompasses six distinct phases. These stages include requirement analysis, system design, implementation, system testing, system deployment, and system maintenance. The following pieces of hardware are used in this project: an Arduino Uno, a range finder ultrasound ultrasonic sensor HC-SR04 HC SR 04, an L298N dual h bridge motor drive controller board module, 18650 battery charger, a Robotic Arm, and an Active Buzzer Module that links through Bluetooth. For picking palm oil, a successful design, testing, and implementation of a microcontroller, ultrasonic sensor, and wireless Bluetooth were accomplished, as well as the system's setup of the robotic arm. Even though the development of robotic arms is still in its infancy, this innovative project indicates that this could potentially be the future approach for automating the picking activity in the palm oil industry.

Keywords: *Robotic Arm; Arduino Microcontroller; Palm Oil Industry*

1.0 Introduction

The development of robotics must be accelerated to assist the basic agricultural industry to grow more rapidly. The robotic system does provide an excellent opportunity for the agro-based industry to assist in the launch of automated production processes involving complex production (Talaviya et al., 2020). Traditional methods, in general, can handle more fixed and static objects (Wan Ismail, 2010). Due to rising demand, the conventional approach is no longer one of the best ways to kickstart production (Iskandar, 2018). In addition to the unpredictability of Malaysia's climate, traditional methods can no longer resolve these issues, particularly those involving the plantation industry (Martin et al., 2015).

Since it was first planted commercially in 1917, the area of oil palm plantations in Malaysia has reached 5.85 million hectares (Kushairi et al., 2019). Essentially, the industry continues to rely on foreign labour to assist in the production process. Dependence on foreign labour considers the ability to manage the farm within the time frame set in addition to taking care of the workers' health (Mohd Nawi et al., 2016). However, workers in the palm oil plantation management system continue to use traditional or semi-mechanical methods. According to Azman et al. (2017), the level of mechanisation in this industry is only 1.9% for harvesting and 16.2% for collection operations. This demonstrates that mechanisation management in oil palm plantations remains low. To address this issue, improvements to the existing product design must be made using a more sophisticated robot system, and employees must be trained to use the new system (Mohd Nawi et al., 2015).

Although the oil palm plantation industry now has a robotic system, neat harvesting and more efficient collection of oil palm fruits into trucks remain the central issue in launching the production process (Verheye, 2010; Lim, 2021). This complex and challenging process strongly emphasizes developing a better robotic system than the existing one. Previous research literature discovered that the existing robotic system for harvesting methods is inefficient due to the motorised cutter (Cantas) being limited by height (less than 5 m palm height) (Abdul Razak et al., 2008). Furthermore, this robotics lacks a control system for the collection method of oil palm fruits.

Many ongoing studies are being conducted by the Malaysian Palm Oil Board (MPOB) to ensure a more sophisticated robotic system to help harvest and collect palm oil (Mohd, Abd Rahim, & Norman, 2021). Currently, the industry relies on traditional robotics methods, despite the introduction of some robotics by research institutes and universities and tests conducted by the industry itself. This is because existing robotics are still expensive to own and inefficient to use in assisting with the start-up of the manufacturing process. As a result, a more detailed study of robotics with harvesting that reaches the maximum height and a collection system that does not affect damage to palm oil fruit is required.

2.0 Literature Review

Malaysia Sustainable Palm Oil (MSPO) is a Malaysian standard for oil palm management certification and supply chain that includes requirements established by the Malaysian Palm Oil Certification Council (MPOCC) (Lim et al., 2021). The MSPO project shares similar goals with the Roundtable on Sustainable Palm Oil (RSPO), which seeks to reduce environmental impact while ensuring the socioeconomic well-being of Malaysia's palm oil industry. Oil palm harvesting and evacuation are often labor-intensive processes that require growers to tour vast plantation sites, engage in hand harvesting, and then evacuate fresh fruit to collection points. With technological advancements, most businesses are heading toward automation, digitalization, innovative production, or "Industry 4.0." An encouraging development to address the labor scarcity issue is the automation and mechanization of the process toward contemporary plantation operation (Kushairi et al., 2017).

The productivity of tree crops such as oil palm is immensely complicated. To understand and mitigate these risk factors, a collection of multi-layered large data sets is required (Jelani et al, 2008). Furthermore, advanced analytics are required to integrate those highly heterogeneous datasets to generate insights about the key constraints on yields at the tree and field scales. Due to the height of oil palm trees and the distance of fronds within fruit bunches from the focus point level of the harvesters' eyes, the harvesting of tall oil palms is typically challenging. In addition, for palms taller than 2.5 metres, a sickle attached to a long pole is used. By pulling the sickle downwards, the sharpness, shape, and profile of the sickle contribute to the effectiveness of the cutting operations. Cutting energy comes from the harvester's endurance and tool sharpness and self-skill, such as lifting and handling the long pole and cutting fronds and fruit bunches (Jelani et al., 2008).

According to Wan Ishak et al. (2011), the current research in agricultural robotics may serve as the foundation for future work on an intelligent robot eye for harvester robots. The creation of the robot eye was investigated by utilizing non-contact measurements, such as video-grammetry, to identify the item and measure it in 3D coordinates (Razali et al., 2020). Robots are now typically represented and their motions are modelled using the Denavit and Hartenberg (D-H) representation (Denavit and Hartenberg, 1995). The first step of the method is to systematically assign and label an orthonormal (x, y, and z) coordinate system to each robot joint. Then, it will be possible to connect one joint to the next and eventually put together a complete model of a robot's geometry. The D-H convention was mostly used in robot manipulators, which have an open kinematic chain with one DOF in each joint and either a revolute or prismatic joint. The following four parameters, referred to as D-H parameters, which are for length, α for twist, d for twist and θ for angle, were used to describe the transformation.

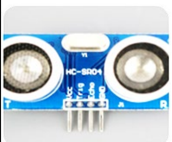
Through the concept of forced learning, camera vision can detect fruit maturity (Wan Ishak et al., 2011). This concept implies that the matured fruit will be connected when it is placed on the harvester's arm if harvesting is done manually. Camera vision, however, applied the current hue value to a digital image of an oil palm in a field and outside. The programmed system compares the colour value between this trained or dummy output, which can be viewed as the standard for the input of defined mature fruit in reality. Outdoors, the environment's temperature changes, humidity on the target colour's surface, and variations in illumination brought on by sunlight will all impact the vision system (Razali et al., 2020).
Besides, the monitoring system on the oil palm plantation site has been improved significantly due to advances in autonomous vehicle technology with object detecting capacity provided by

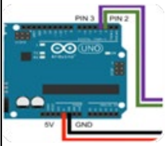
multi-camera systems and radar (Chong et al., 2017). Small tractors and self-manipulating robots can be used to accomplish this and provide farmers with information about the location of ripe oil palm fruits and the ability to start an auto-harvesting procedure in the future. According to Lim et al. (2021), adopting smart sensors for palm fruit ripeness detection and a cloud computing system to optimize the harvesting route could improve the harvesting process at the palm oil plantation site. Additionally, palm trees can be monitored using remote sensing technologies to detect trees that produce poorly and increase the efficiency of the entire harvesting process (Kassim et al., 2014). It can be used to assess the status of the environment and agricultural conditions.

3.0 Materials and Components

To produce the development of a prototype of robotic palm oil picking arm, these hardware components are required as follows:

Table 1. Materials and Components

ardware	Description	Function
 Laptop	A portable computer with the same capabilities as a desktop computer - The ability to run any software (Kay, 2016).	Run the Arduino IDE software and setup the Arduino UNO using the programming language (Sudarsana et al., 2019).
 Arduino UNO R3	The Atmega328-based Arduino Uno is a microcontroller board. It features a total of 20 digital input/output pins (Kaswan, Singh, & Sagar, 2020).	Allows a programmer to develop modular code that performs a specific job and then returns to the region of code from where the function was "called." The most common reason for writing a function is to do the same operation several times in a programme. It contains all of the components required to support the microcontroller (Louis, 2016).
 Range Finder Ultrasound Ultrasonic Sensor HC-SR04	The HC-SR04 is a compact, low-cost ultrasonic sensor with a 15° detection angle and an easy-to-use packaging that can measure distances ranging from 2cm to 4 metres with consistent performance and good range accuracy (Abd Latif, Abd Aziz, Ramdan, & Othman, 2021).	To identify items at a distance without requiring the robot to make physical touch with them. Sound pulses may be used to calculate distance in the same manner as bats and submarines do (Abdulkhaleq, Hasan, Abdul, & Salih, 2020).

 L298n Dual H Bridge Motor Drive Controller Board Module	Low heat, excellent anti-interference performance. High working power up to 46v, huge current up to 3A MAX, and continuous current up to 2A, power up to 25w (Kaswan et al., 2020).	To convert a low-current control signal into a higher-current signal capable of driving a motor. This will aid in controlling the robotic arm to turn in the desired direction (Mateenuddin & Jahagirdar, 2017).
 HC-06 Bluetooth Modules For Arduino	The HC-06 is a Bluetooth class 2 slave module intended for transparent wireless serial communication (Parves, Mahin, Ahmed, Mia, & Sarkar, 2022).	Its functionality becomes apparent to the user after it is associated with a master Bluetooth device. Connect it to the smart robotic arm's remote control, in this example the smartphone (Parves et al., 2022).
 18650 BATTERY CHARGER	A device that forces an electric current through a secondary cell or rechargeable battery to provide energy. The holder must also establish electrical contact with the battery terminals when using dry cells. a plastic box with the shape of the housing moulded as a compartment or compartments accepting a battery or batteries, or a separate plastic holder affixed using screws, eyelets, adhesive, double-sided tape, or other way (Yu, 2016).	To power the 18650 battery. A battery is held in one or more compartments or chambers. It serves as the robotic arm's power supply (Abd Latif et al., 2021).
 Arduino Uno R3 Extension Shield V5.0	It is useful to connect the sensors directly to the Arduino Uno without using a breadboard. It supports plug-and-play connections to numerous modules like as sensors, servos, relays, potentiometers, and so on (Kaswan et al., 2020).	Increase the number of ports or pins on the Arduino UNO (Louis, 2016).
	Motor Operating Voltage: 3.3v - 6v	To automate the process of placing goods or products onto pallets (Swamardika,

Robotic ARM	Model: HC02-48 2 Way Motor Shaft Speed (No Load) : 3v(125RPM), 5V(208RPM)	Budiastra, Setiawan, & Ngurah, 2017).
 Active Buzzer Module	A mechanical, electromechanical, or piezoelectric audio signalling device (Abd Latif et al., 2021).	An audio signalling device to alert user.

4.0 Methodology

In this project, the robotic arm employs the waterfall concept. The waterfall model is a linear-sequential life cycle model, which means that each phase must be finished before the next phase can begin, and no phases overlap. Requirement analysis, system design, implementation, testing, deployment, and maintenance are part of the waterfall methodology.

Phase 1: Requirement Analysis
The first stage of the waterfall model is requirement analysis. It is the stage where innovators define the project requirements, analyze them, and thoroughly comprehend the difficulties. Giving someone a question and interviewing them are two methods for gathering information (Casteren, 2015).

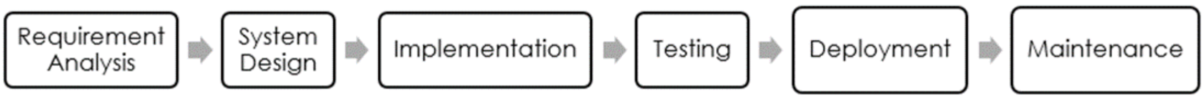


Figure 1. Waterfall Methodology

Phase 2: System Design
It serves in the specification of hardware and system requirements throughout the system design process, as well as the definition of overall system architecture. The innovators will provide a technological solution to the difficulties outlined in the product requirements, which will include scenarios, layouts, and data models. In a nutshell, this is the phase in which the innovators build the project proposal (Casteren, 2017).

Phase 3: Implementation
Once the design has been accepted, the technical execution step begins. The innovators will include the design papers into our project. This is frequently the longest period (Aroral, 2021).

Phase 4: Testing
Following the completion of full implementation, testing must take place before the product can be provided to the client. The innovators will develop their test cases using the design papers, personas, and user case scenarios provided by the supervisor. Testing is carried out to ensure that the client has no problems when utilising the robotic arm (Sharma & Singh, 2021).

Phase 5: Deployment
After functional and non-functional testing, the robotic arm is installed in the client environment or released to the market during the deployment phase (Pedamkar, 2022).

Phase 6: Maintenance
If something goes wrong with the product at the final step, which is maintenance. It will either improve or develop again (Mariana, 2019).

Functional Requirement
The requirement specification is a software engineering technique consisting of several processes establishing the demands or conditions that must be met for a new or changed product while considering competing client expectations. It specifies the tasks the components should perform, the technique used, the amount of coding the project should have, and the user interfaces of the apps (Dabbagh & Peck Lee, 2015). Functional requirements illustrate the project's internal working, the project description, and the explanation of each component. The following are the functional requirements:

i. Microcontroller: Using an Arduino UNO to build an electric robotic arm.
ii. Ultrasonic sensor: Detects any impediments in its path.
iii. Buzzer: It used to alert the user by sound.
iv. Wireless Bluetooth: to allow the robotic arm to be controlled over Bluetooth.
v. Remote XY: A Bluetooth-enabled application for controlling the robotic arm.

Non-Functional Requirement
A qualitative demand for a product, service, system, process, document, location, infrastructure component or facility is a non-functional requirement. Whereas functional requirements define what something accomplishes, non-functional requirements define its characteristics. In certain circumstances, non-functional requirements are intangible items that require human judgment to develop and assess, such as sensory analysis (Chung, Sampaio, & Leite, 2009). It describes system characteristics concerned with how the system meets functional requirements. They are as follows:

i. Security: To use the remote XY applications to operate the robotic arm, the user must first provide the right password.
ii. Reliability: If the ultrasonic sensor detects the object/obstacles, the buzzer will inform the user and the robotic arm will stop when they are too close.
iii. Performance: It is dependent on its power and motor. If the electricity is sufficient to feed the motor, it will function normally.
iv. Maintainability: When a component needs to be replaced, it is easily accessible.

5.0 Results and Discussion

Most of industrial robots are built to tackle repetitive and heavy production operations. Typically, robots are assigned to certain jobs in the sector. The most prevalent manufacturing robot is the robot arm. A robot arm is essentially a mechanical arm that may be programmed to perform the same duties as a human arm.

System Configuration
The specification of a certain computer system, from its hardware components to the software and numerous operations that execute within that system, is referred to as system configuration. It refers to the types and models of devices installed, as well as the software used to run the various sections of the computer system. System configuration also refers to certain operating system settings that have been established by default, either automatically or manually, by a specified application or the user.

```
1  #include <SoftwareSerial.h>
2  #include <RemoteXY.h>
3
4  // RemoteXY connection settings
5  #define REMOTEXY_SERIAL_RX 2
6  #define REMOTEXY_SERIAL_TX 3
7  #define REMOTEXY_MODE__SOFTSERIAL
8  #define REMOTEXY_SERIAL_SPEED 9600
9  #define REMOTEXY_ACCESS_PASSWORD "smarttrolley"
10
11 // RemoteXY configurate
12 #pragma pack(push, 1)
13 uint8_t RemoteXY_CONF[] =
14   { 255,3,0,0,0,55,0,8,13,2,
15   2,1,8,14,22,11,20,23,22,11,
16   2,26,31,31,79,78,0,79,70,70,
17   0,5,16,54,16,30,30,17,50,30,
18   30,2,26,31,129,0,9,7,19,6,
19   21,16,20,6,17,66,117,122,122,101,
20   114,0 };
21
22 struct {
23   // input variable
24   uint8_t switch_1; // =1 if switch ON and =0 if OFF
25   int8_t joystick_1_x; // =-100..100 x-coordinate joystick position
26   int8_t joystick_1_y; // =-100..100 y-coordinate joystick position
27
28     // other variable
29   uint8_t connect_flag;  // =1 if wire connected, else =0
30 } RemoteXY;
31 #pragma pack(pop)
```

Figure 2. Coding for configure Bluetooth Connection

Lines 1 and 2 are for including the Software Serial and Remote XY libraries. Lines 5 to 8 create the Bluetooth connection, while line 9 sets the password for the remote XY app to robotic arm. The setting of the remote XY interface is shown on lines 12 to 20. Lines 22-30 define the structure for the remote XY function.

```
33 #define trigPin A0
34 #define echoPin A1
35 #define buzzPin 13
36
37 //define right motor control pins
38 #define right_motor_A 8
39 #define right_motor_B 9
40 #define right_motor_speed 11 //enable pin
41
42 //define left motor control pins
43 #define left_motor_A 6
44 #define left_motor_B 7
45 #define left_motor_speed 10 //enable pin
46
47 uint8_t RightMotor[3] = {right_motor_A, right_motor_B, right_motor_speed};
48 uint8_t LeftMotor[3] = {left_motor_A, left_motor_B, left_motor_speed};
49
50 //speed control of motors
51 void Wheel (uint8_t * motor, int v)
52 {
53   if (v > 100) v=100;
54   if (v < -100) v=-100;
55   if (v > 0){
56     digitalWrite (motor [0], HIGH);
57     digitalWrite (motor [1], LOW);
58     analogWrite (motor [2], v * 2.55);
59   }else if ( v<0 ){
60      digitalWrite (motor [0], LOW);
61     digitalWrite (motor [1], HIGH);
62     analogWrite (motor [2], (-v) * 2.55);
63   }else{
64     digitalWrite (motor [0], LOW);
65     digitalWrite (motor [1], LOW);
66     analogWrite (motor [2], 0);
67   }
68
```

Figure 3. Coding for Ultrasonic Sensor and Buzzer

The component pins are defined by lines 33 to 45, beginning with line 33-34 for the ultrasonic sensor, line 35 for the buzzer, and lines 38-40 and 43-45 for the L298N driver. Lines 47 and 48 are used to set the array value for both the left and right motors. Lines 51 through 68 include the code for regulating the robotic arm's movement.

```
70 //buzzer
71 void buzzer()
72 {
73   if(RemoteXY.switch_1 == 1){
74     int duration, distance;
75     // Output pulse with 1ms width on trigPin
76     digitalWrite(trigPin, HIGH);
77     delay(1);
78     digitalWrite(trigPin, LOW);
79     // Measure the pulse input in echo pin
80     duration = pulseIn(echoPin, HIGH);
81     // Distance is half the duration devided by 29.1 (from datasheet)
82     distance = (duration/2) / 29.1;
83     // if distance less than 1 meter and more than 0 (0 or less means over range)
84     if (distance <= 100 && distance >= 0) {
85       //Stop when distance <= 30cm
86       if (distance <= 30 && distance >= 0) {
87         digitalWrite (6, LOW);
88         digitalWrite (8, LOW);
89         digitalWrite(buzzPin, HIGH);
90       }else{
91         digitalWrite(buzzPin, HIGH);
92       }
93     } else {
94       // Don't buzz
95       digitalWrite(buzzPin, LOW);
96     }
97     // Waiting 60 ms won't hurt any one
98     delay(60);
99   }else{
100    digitalWrite(buzzPin, LOW);
101  }
102 }
```

Figure 4. Coding for Collusion Avoiding System

Lines 71-102 are for the collusion avoidance system. Line 73 is for the button that determines whether the collusion avoidance mechanism is engaged or not. Line 74 declares the required variable. Lines 76 to 78 cause the ultrasonic sensor to generate a sound pulse for one second and then cease. Meanwhile, lines 80 and 82 are used to calculate the range of any obstacles in front of it. Lines 84 to 101 are used to assess whether the distance between the obstruction and the robotic arm is less than or equal to 100cm. If the distance is less than 30cm, the buzzer will still beep, and the robotic arm will stop.

```
104 void setup()
105 {
106   RemoteXY_Init ();
107   pinMode(trigPin, OUTPUT);
108   pinMode(echoPin, INPUT);
109   pinMode(buzzPin, OUTPUT);
110
111   //initialization pins
112   pinMode (right_motor_A, OUTPUT);
113   pinMode (right_motor_B, OUTPUT);
114   pinMode (left_motor_A, OUTPUT);
115   pinMode (left_motor_B, OUTPUT);
116 }
117
118 void loop()
119 {
120   RemoteXY_Handler ();
121   //If lost connection it stop all motor and buzzer
122   if(RemoteXY.connect_flag == 1)
123   {
124     buzzer();
125     //manage the motor
126     Wheel (RightMotor, RemoteXY.joystick_1_y - RemoteXY.joystick_1_x);
127     Wheel (LeftMotor, RemoteXY.joystick_1_y + RemoteXY.joystick_1_x);
128   }else{
129     digitalWrite (10, LOW);
130     digitalWrite (11, LOW);
131     digitalWrite (7, LOW);
132     digitalWrite (6, LOW);
133     digitalWrite (8, LOW);
134     digitalWrite (9, LOW);
135     digitalWrite (buzzPin, LOW);
136   }
```

Figure 5. Coding for Setup Function of the Robotic Arm

Lines 104 to 116 are for the setup function, in which the innovators specify the variable as input or output, and lines 118 to 136 are for the loop function, in which the innovators want the code to loop again. Lines 122 to 136 are for an extra safety feature that will instantly stop the robotic arm if the Bluetooth connection is broken.

Security Requirement

A functional security requirement outlines the functional behaviour that ensures security. It is directly testable and observable. Functional requirements include requirements for access control, data integrity, authentication, and erroneous password lockouts. Before utilising the Remote XY applications through Bluetooth to operate the robotic arm, the user must provide a password.

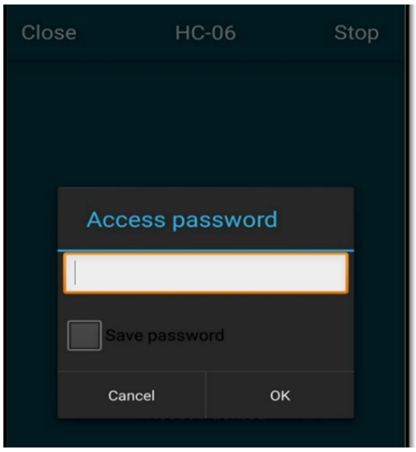


Figure 6. Security Requirement

Logical Design

A conceptual, abstract design is logical design. The innovators are not concerned with actual implementation specifics yet, but rather with the sorts of information that are required. The logical design method entails organising data into a set of logical relationships known as entities and characteristics.

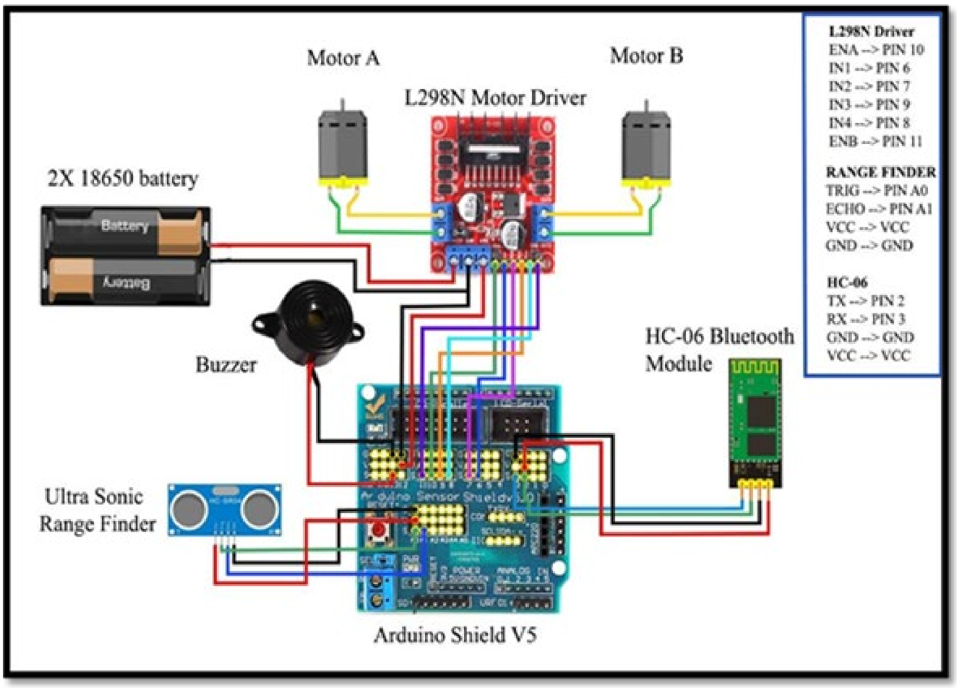


Figure 7. Logical Diagram

The figure above shows the logical design of our project's robotic arm. The innovators utilise the Arduino shield V5 in this project to fit all the other components to the Arduino UNO R3 as shown in the figure. The ENA and ENB pins on the driver are connected to digital pins 10 and 11 on the Arduino, while the IN1 pin is connected to digital pin 6, IN2 to digital pin 7, IN3 to digital pin 9, and IN4 pin is connected to digital pin 8. Also, the innovators employ two 18650 batteries for electricity (7.4v). The positive wire is connected to the CVV on the driver, while the negative wire is connected to the ground (GND). The innovators connect the 5V and GND pins on the driver to the voltage (V) and ground (G) pins on the shield to power the Arduino and other components attached to it. The motor is used by the inventors to link it to both outputs on the driver. For the buzzer, we connect the positive pin to the signal (S) pin so we can regulate when it buzzes, and the negative pin to the ground (G) pin. Next, because the inventors are using the hc-06, the Bluetooth will only have four pins: TX, RX, GND, and VCC. So, we connect the TX pin to digital pin 2, the RX pin to digital pin 3, the GND pin to ground (G), and the VCC pin to voltage (V). The ultrasonic range finder is the last but not least. It has four pins for TRIG, ECHO, VCC, and GND. We connect the range finder TRIG pin to the A0 pin, the ECHO pin to the A1 pin, the VCC pin to the voltage (V) pin, and the GND pin to the ground (G) pin.

Prototype Development

The figure above shows the robotic arm prototype. The prototype was made by the inventors since we are having difficulty finding a suitable engine for the full-size of robotic arm but concentrating on the technology within the prototype. The robotic arm is created by the creators simply using a basket and picking function.

Figure 8. Prototype of Robotic Arm.

6.0 Test Description and Result

Unit Testing

Unit testing is a type of software testing in which individual programme units/components are tested. The goal is to ensure that each component of the software works as intended. A unit is the smallest testable component of any piece of software. It typically has one or more inputs and a single output. The unit testing plan for this project is shown in the table below. If the testing fails, it is utilised to identify and resolve the problem.

Table 2. Unit Testing Plan

Test step	Test Procedure	Expected Result	Actual Result	Status (Pass/ Fail)
Arduino uno r3	Connect the Arduino to laptop and upload the default code	The code uploads successfully	The code is uploaded successfully	Pass
Connection of Bluetooth	Pairing the hc-06 Bluetooth with the smartphone	The LED on hc-06 Bluetooth will stop blinking which mean it successfully pair with another device	The LED on the HC-06 Bluetooth stop blinking	Pass
Sound from buzzer	Upload simple tone code on Arduino that connected with buzzer	Produce sound when the Arduino is power up	The buzzer produce sound	Pass

L298N driver output	Upload a code that make driver produce an output voltage	The driver will produce an output of 7.4v	The driver produces an output 7.4v	Pass
Detection of ultrasonic sensor of obstacles	Upload a code that will display the range of the obstacle on Arduino ide serial monitor	Display the range of object if it detect any	Display the range of object on Arduino ide serial monitor	Pass

Integration Testing Plan

Integration testing is the phase of software testing in which distinct software modules are joined and tested as a group. It takes place after unit testing but before validation testing. Integration testing takes unit-tested modules as input, collects them into bigger aggregates, applies tests described in an integration test plan to those aggregates, and gives the integrated system suitable for system testing as output. The table below depicts our project's integration testing strategy. This testing will demonstrate the relationship between each component's functions. If the testing fails, it is utilised to identify and resolve the problem.

Table 3. Integration Testing Plan

Test step	Test Procedure	Expected Result	Actual Result	Status (Pass/ Fail)
App function to control robotic arm and connection via Bluetooth.	Upload a code that control the movement of the robotic arm using smartphone	The robotic arm will move according to the control on the smartphone	The robotic arm moves according to the movement of the control on the smartphone	Pass
Detection of ultrasonic sensor of obstacles and alert of buzzer	Upload a code that make buzzer produce sound when the ultrasonic sensor detect obstacle within 1 meter	The buzzer produces sound when the obstacle is within 1 meter or less	The buzzer produce sound	Pass
Connection of motor driver and motor	Upload a code that make the motor to move forward	The motor will move forward	The motor moves forward	Pass

		when the power is connected		

7.0 Contributions/Advantages

A range of technological advancements has been made as a result of innovation for the oil palm industry. Mechanization of oil palm operations will solve the industry's labour scarcity issue. Adopting robot harvesters will help simplify the harvesting process without needing expert labour and ensure worker safety. The microcontroller, ultrasonic sensor, and wireless Bluetooth were successfully designed, tested, and implemented for picking palm oil, and the robotic arm setup for the system was also successful. The robotic arm delivers significant cost savings, improved quality assurance, increased output, and the capacity to operate under challenging environments. Even though the development of robotic arms is still in its infancy, this innovative project indicates that this could potentially be the future approach for automating the picking activity in the palm oil industry.

8.0 Conclusion

Currently, the traditional method of harvesting and collecting oil palm fruits into trucks is simply inefficient. This is because conventional methods require significant labour and time to begin production. Furthermore, the loose fruit collection system makes accelerating work in oil palm plantations difficult. Various types of collections are available today, such as disc plate collectors, roller pickers, etc. However, the produced robotics cause significant damage to the palm fruit and provide less comfort to workers when operating the robotics. In practice, the harvest and collection method must be suitable for operation to maintain the product quality and the workers' health. As a result, new robotics criteria must be developed and tested to assist in resolving problems encountered by oil palm plantation workers and, as a result, to assist the Malaysian oil palm industry in achieving higher performance.

9.0 References

[1] Abd Latif, M. N., Abd Aziz, N. A., Ramdan, M. R., & Othman, N. H. (2021). Design and Development of Smart Automated Clothesline. Malaysian Journal of Industrial Technology, 6(1).

[2] Abdulkhaleq, N. I., Hasan, I., Abdul, N., & Salih, J. (2020). Investigating the resolution ability of the HC-SRO4 ultrasonic sensor Investigating the resolution ability of the HC-SRO4 ultrasonic sensor. IOP Conference Series Materials Science and Engineering, 745(1).

[3] Abdul Razak, J., Ahmad, H., Johar, J., Malik, N., Yosri, G., & Omar, A. (2008). A tool for efficient harvesting of oil palm fresh fruit bunches. Journal of Oil Palm Research, 20(12), 548-558.

[4] Aroral, H. K. (2021). Waterfall Process Operations in the Fast-Paced World: Project Management Exploratory Analysis. International Journal of Applied Business and Management Studies, 6(1).

[5] Azman, I., Siti Mashani, A., Norhidayu, A., Abd Razak, J., & Abd Rahim, S. (2017). An economic study on mechanisation in oil palm plantations in Malaysia. Palm Oil Economic Review and Outlook Seminar 2017. Pullman Kuala Lumpur City Centre, 17 January 2017.

[6] Casteren, W. Van. (2015). The Waterfall Model and the Agile Methodologies: A Comparison by Project Characteristics. In Paper on Software Development Methods (pp. 1–6).

[7] Casteren, W. Van. (2017). The Waterfall Model and the Agile Methodologies.

[8] Chung, L., Sampaio, J. C., & Leite, P. (2009). On Non-Functional Requirements in Software Engineering. In Modeling: Foundations and Applications (pp. 363–379). Springer Berlin / Heidelberg.

[9] Chong,K.L., Kanniah, K.D., Pohl, C. & Tan, K.P. (2017) A review of remote sensing applications for oil palm studies, Geo-spatial Information Science, 20:2, 184-200.

[10] Dabbagh, M., & Peck Lee, S. (2015). Functional and Non-Functional Requirements Prioritization: Empirical Evaluation of IPA, AHP-based, and HAM-based approaches. Soft Computing, 20(11).

[11] Denavit, J., & Hartenberg, R. S. (1995). A kinematic notation for lower-pair mechanisms based on matrices. Journal of Applied Mechanics, (23): 215－221.

[12] Iskandar, M. J., Baharum, A., Anuar, F. H., & Othaman, R. (2018). Palm oil industry in Southeast Asia and the effluent treatment technology—A review. Environmental technology & innovation, 9(2018), 169-185.

[13] Jelani, A.R., Hitam, H., Jamak, J., Noor, M., Gono, Y. & Ariffin, O. (2008). A Tool For The Efficient Harvesting Of Oil Palm Fresh Fruit Bunches. Journal Of Oil Palm Research, 20 (12): 548-558.

[14] Kassim, M.S.M., Ismail, W.I.W., Ramli, A.R. & Bejo, S.K. (2014). Image Clustering Technique in Oil Palm Fresh Fruit Bunch (FFB) Growth Modeling. Agriculture and Agricultural Science Procedia, 2(1): 337-344.

[15] Kaswan, K. S., Singh, S. P., & Sagar, S. (2020). Role of Arduino in Real World Applications. International Journal of Scientific and Technology Research, 9(1), 1113–1116.

[16] Kay, R. H. (2016). Assessing Laptop Use in Higher Education: The Laptop Use Scale. Journal of Computing in Higher Education, 28(1), 18–44.

[17] Kushairi, A., Ong-Abdullah, M., Nambiappan, B., Hishamuddin, E., Bidin, M. N. I. Z., Ghazali, R., ... & Parveez, G. K. A. (2019). Oil palm economic performance in Malaysia and R&D progress in 2018. Journal of Oil Palm Research, 31(2), 165-194.

[18] Lim, C. H., Cheah, Z. H., Lee, X. H., How, B. S., Ng, W. P. Q., Ngan, S. L., ... & Lam, H. L. (2021). Harvesting and evacuation route optimisation model for fresh fruit bunch in the oil palm plantation site. Journal of Cleaner Production, 307(2021), 1-11

[19] Louis, L. (2016). Working Prinnciple of Arduino and Using IT As a Tool for Study and Research. International Journal of Control, Automation, Communication and Systems, 1(2), 21–29.

[20] Mariana, A. . (2019). Implementation of the Waterfall Model for the Development of the DKI Jakarta PKK Current Mail Information System, 6(5), 453–459.

[21] Martin, S., Rieple, A., Chang, J., Boniface, B., & Ahmed, A. (2015). Small farmers and sustainability: Institutional barriers to investment and innovation in the Malaysian palm oil industry in Sabah. Journal of Rural Studies, 40(2015), 46-58.

[22] Mateenuddin, F. S., & Jahagirdar, V. . (2017). An Android Controlled Mini Rover for Real Time Surveillance Using Raspberry Pi 3. International Journal of Advanced Research in Engineering and Management, 3(5).

[23] Mohd Nawi, N. S., Md Deros, B., Ab Rahman, M. N., Nordin, N., & Hani, S. Ezrin. (2015). Mechanized tools and technovation machinery in palm oil plantations: Utilization or rejection? Paper presented at the International Symposium on Advancement in Ergonomics and Safety 2015. Universiti Malaysia Perlis, Perlis. 30 November 2015.

[24] Mohd Nawi, N. S., Md Deros, B., Sukadarin, E. H., & Nordin, N. (2016). Malaysian oil palm workers are in pain: Hazards identification and ergonomics related problems. Malaysian journal of public health medicine, 16(s1), 50-57.

[25] Mohd, R. K., Abd Rahim, S., & Norman, K. (2021). Mechanising Oil Palm Loose Fruits Collection-A Review. Journal of Oil Palm Research, 33(1), 1-11.

[26] Parves, K. . T., Mahin, M. K., Ahmed, K. T., Mia, P., & Sarkar, M. M. R. (2022). Bluetooth Control Car with Arduino. Retrieved from file:///C:/Users/user/Downloads/GroupEReport.pdf

[27] Pedamkar, P. (2022). Introduction to Waterfall Model. Retrieved June 21, 2022, from https://www.educba.com/waterfall-model/

[28] Razali MH, Bakar, MA, & Mohd Sabri, MS. (2020). Conceptual development of automated harvester for tall oil palm tree. Adv Agri Food Res J 2020; 1(2): a0000151.

[29] Sharma, L., & Singh, V. (2021). WaterFall Model. Retrieved June 21, 2022, from https://www.toolsqa.com/software-testing/waterfall-model/

[30] Sudarsana, I. K., Putra, I. P. A. W., Anam, F., Istianti, T., Pandin, M. G. R., Bhawika, G. W., Laili, N. R. (2019). The Function of Technology

[31] y and Device Laptop for Education Purpose. Journal of Physics Conference Series, 1363(1), 012062.

[32] Swamardika, A., Budiastra, I. ., Setiawan, N., & Ngurah, I. E. (2017). Design of Mobile Robot with Robotic Arm Utilising Microcontroller and Wireless Communication. International Journal of Enginerring and Technology, 9(2),838–846.

[33] Talaviya, T., Shah, D., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. Artificial Intelligence in Agriculture, 4(2020), 58-73.

[34] Verheye, W. (2010). Growth and production of oil palm. In Land use, land cover and soil sciences. UNESCO-EOLSS Publishers.

[35] Wan Ismail, W.I. (2010). Research and development of oil palm harvester robot at Universiti Putra Malaysia. International Journal of Engineering and Technology, 7(2), 87-94.

[36] Yu, B. (2016). Design and Experimental Results of Battery Charging System for Microgrid System. International Journal of Photoenergy, 1–6.