

Software Requirement Specification

PaddyRobot: Automation for Material Transfer Using Robotic Arm - Prototype

Version: 1.2.4 Date: 11/12/2022

Table of Contents

[2 Version Control 4](#_Toc121516342)

[3 List of Abbreviations 5](#_Toc121516343)

[4 List of Figures 6](#_Toc121516344)

[5 Introduction - Hafiidz 7](#_Toc121516345)

[5.1 Definitions 7](#_Toc121516346)

[5.2 Background 7](#_Toc121516347)

[5.3 System Overview – Chong CH 8](#_Toc121516348)

[6 Project Overview - Hafiidz 9](#_Toc121516349)

[6.1 Team and Members Details 9](#_Toc121516350)

[6.2 Github Link 9](#_Toc121516351)

[6.3 Aims and Objectives / Product’s Purpose 9](#_Toc121516352)

[6.3.1 Intended Audience and Intended Use 9](#_Toc121516353)

[6.3.2 Product Scope 9](#_Toc121516354)

[6.4 Constraints and Assumptions 10](#_Toc121516355)

[7 User Characteristics/Persona 11](#_Toc121516356)

[8 Hardware Requirements - Hafiidz 13](#_Toc121516357)

[9 Software Requirements - Sarah 15](#_Toc121516358)

[10 System Architecture – Chong CH 16](#_Toc121516359)

[10.1 State Diagram – Chong CH 16](#_Toc121516360)

[10.2 System Operation Flow Chart – Chong CH 17](#_Toc121516361)

[11 External Interface Requirements 22](#_Toc121516362)

[11.1 User Interfaces - Sarah 22](#_Toc121516363)

[11.2 Hardware Interfaces 22](#_Toc121516364)

[11.3 Software Interfaces 23](#_Toc121516365)

[11.4 Communications Interfaces 23](#_Toc121516366)

[12 Use Case - Sarah 24](#_Toc121516367)

[12.1 Use Case Chart 24](#_Toc121516368)

[13 Functional Requirements 25](#_Toc121516369)

[14 Robotic Arm 27](#_Toc121516370)

[14.1 Algorithm – Gary 28](#_Toc121516371)

[14.1.1 Algorithm Overview 28](#_Toc121516372)

[14.1.2 Justification of Chosen Algorithm 29](#_Toc121516373)

[15 Programmable Logical Devices (PLC) 31](#_Toc121516374)

[16 Sensors and Actuators 32](#_Toc121516375)

[16.1 Weight 32](#_Toc121516376)

[16.2 Pressure 32](#_Toc121516377)

[17 Implementation 33](#_Toc121516378)

[17.1 Data Structure 33](#_Toc121516379)

[18 References 34](#_Toc121516380)

# Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| No | Version | Date | Description of changes and person responsible for making changes |
| 1 | 0.1.0 | 06/12/2022 |  |
| 2 | 0.2.0 | 09/12/2022 | Update the functional requirement. (C.H. Chong)  Update the system architecture. (C.H. Chong) |
| 3 | 0.2.1 | 10/12/2022 | Update captions, description of use case and one UI screenshot (Sara) |
| 4 | 0.2.2 | 11/12/2022 | Update section 6 Project Overview |
| 5 | 1.2.2 | 11/12/2022 | Submit to SE Team for Static Test Development. |
| 6 | 1.2.3 | 17/12/2022 | Updated use case diagram (Sara) |
| 7 | 1.2.4 | 17/12/2022 | Modified and updated the aim and objectives of the project. (C.H. Chong) |
|  |  |  |  |
|  |  |  |  |

# List of Abbreviations

|  |  |  |
| --- | --- | --- |
| No | Abbreviation | Description |
| 1 | PLC | Programmable Logical Devices |
| 2 | GUI | Graphical User Interface |
| 3 | SRS | Software Requirement Specification |
| 4 | DOF | Degree of Freedom |
|  |  |  |
|  |  |  |
|  |  |  |

# List of Figures

[Figure 3‑1: System Overview Diagram 8](#_Toc121438954)

[Figure 5‑1: End User Persona 10](#_Toc121438955)

[Figure 8‑1: System Architecture Diagram 15](#_Toc121438956)

[Figure 8‑2: System State Diagram 16](#_Toc121438957)

[Figure 8‑3: Startup Logical Flow 17](#_Toc121438958)

[Figure 8‑4: Shutdown Logical Flow 18](#_Toc121438959)

# Introduction - Hafiidz

## Definitions

|  |  |
| --- | --- |
| Terms | Definition |
| PaddyRobot Project aka Project | Prototype hardware and user interface for robotic arm to solve paddy processing and packaging factory automation and optimization |
| End User | For prototype, the end user refers to project team members both from WOA7001 and WOC team  For target end user, this will be the workers/laborers, technician and admin in the paddy processing and packaging factory |

## Background

This Software Requirement Specification (SRS) document outlines the software and hardware requirement for PaddyRobot project which uses robotic arm to assist paddy processing and packaging.

This project is part of collaboratio with ICT-INOV initiative which promotes “modernizing ICT education for harvesting innovation” by ERASMUS and Programme of European Union. Robotic arm usage within paddy processing has been identified as one of the area to be studied.

## System Overview – Chong CH

The system is a design to model the material transfer process where goods or products are made from the manufacturing processes. The prototype or proof of concept for the system is proposed as shown below.

Graphical user interface

Description automatically generated with low confidence

Figure 4‑1: System Overview Diagram

The aim of the system is to move objects from a platform to another location to simulate the material transfer of the goods and products. The objects are chosen based on its weight to simulate the measurement process and transfer to a grid-like location where each grid can be specified from the software. The feature to pick specific location for transfer is to prototype the scalability and flexibility of the robotic arm in material transfer. Furthermore, the system should be fitted with a touch screen for end user operation on the system where basic trigger start or stop of operation can be done by operator.

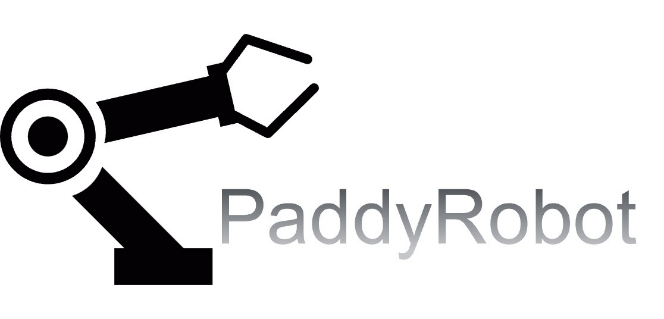
# Project Overview - Hafiidz

## Team and Members Details

The team consists of two sub team from Master of Computer Science (Applied Computing), MCS AC and Masters of Computer Science (Software Engineering), MSE.

| **No** | **Name** | **Matric** | **Program** |
| --- | --- | --- | --- |
| 1 | Chong Chia Hsing | S2159070/1 | MCS AC |
| 2 | Lee Wai Key | 17194567/2 | MCS AC |
| 3 | Mohd Hafiidz Hassan | S2147238 | MCS AC |
| 4 | Gary Yee Siew Wah | S2190954 | MCS AC |
| 5 | Normaisarah binti Ab Majid | 17150714 | MCS AC |
| 6 | Yulin Lei | s2152115 | MCS AC |
| 7 | Ahmad Allehyany | S2162429 | MSE |
| 8 | Nasib Ullah | S2019652 | MSE |
| 9 | Zhenfei Cao | S2142499 | MSE |
| 10 | WangShifeng | S2104917 | MSE |
| 11 | Hao Yang | S2028894 | MSE |

The team logo is made of robotic arm and the team name PaddyRobot highlighting the focus towards the success of the project.



## Github Link

The code and project information is available at the official project github link as provided below. Do note that to ensure privacy, the project link is private and access require admin approval. Please contact Hafiidz at S2147238@siswa.um.edu.my

<https://github.com/paddyrobot/woa7001>

## Aims and Objectives / Product’s Purpose

The aim of the project is to provide proof-of-concept for the automation of the material transfer process using the robotic arm which in turn can be applied in the context of rice milling industry.

The objectives of the projects are outline below:

* To build a prototype robotic arm which can perform the material transfer from 1 point to another location.
* To optimize the process of material transfer for rice milling factory in terms of the overall material cost, labour cost, power usage, and the average time to transfer.
* To optimize the robotic arm motion via suitable algorithm for more optimum usage in turn provide energy saving.

### Intended Audience and Intended Use

The intended audience also known as the End Users are outlined in the definition, where for the prototype, the audience is the team members itself. However, the final audience, would be the workers from paddy processing factory such as the laborers working directly in packaging and measurement section as well as the technician and the admin maintaining and supervising the units used.

### Product Scope

The project consists of both hardware prototype and software suites with testing suits design to test the prototype functionality and robustness.

In summary the scope are as follows:

1. Hardware prototype assembly
2. Software and algorithm MVP
3. Testing for both hardware prototype and software MVP

## Constraints and Assumptions

*To be defined*

# User Characteristics/Persona

Table 1: User's roles and access level.

|  |  |  |
| --- | --- | --- |
| Actor/Role | Description | Required Knowledge |
| Technician | Users can start and stop the robot arm  User can select features/routines from a list | Users need to know and understand the list of functions/routines of the robotic arm and are able to select and start/stop the robotic arm |
| Engineer | Users can perform all the Technician task.  Users can change the configuration or threshold of the robot. | Users need to possess the technical knowledge and understand the basic of programming for the robotic arm and the safety and risk associated with the configuration. |
| Admin | Users can perform all the Engineer tasks.  Users can create, delete, and change the Users. | Admin has the developer level understanding of the system and can provide customization and adjustment of the system based on the situation. |

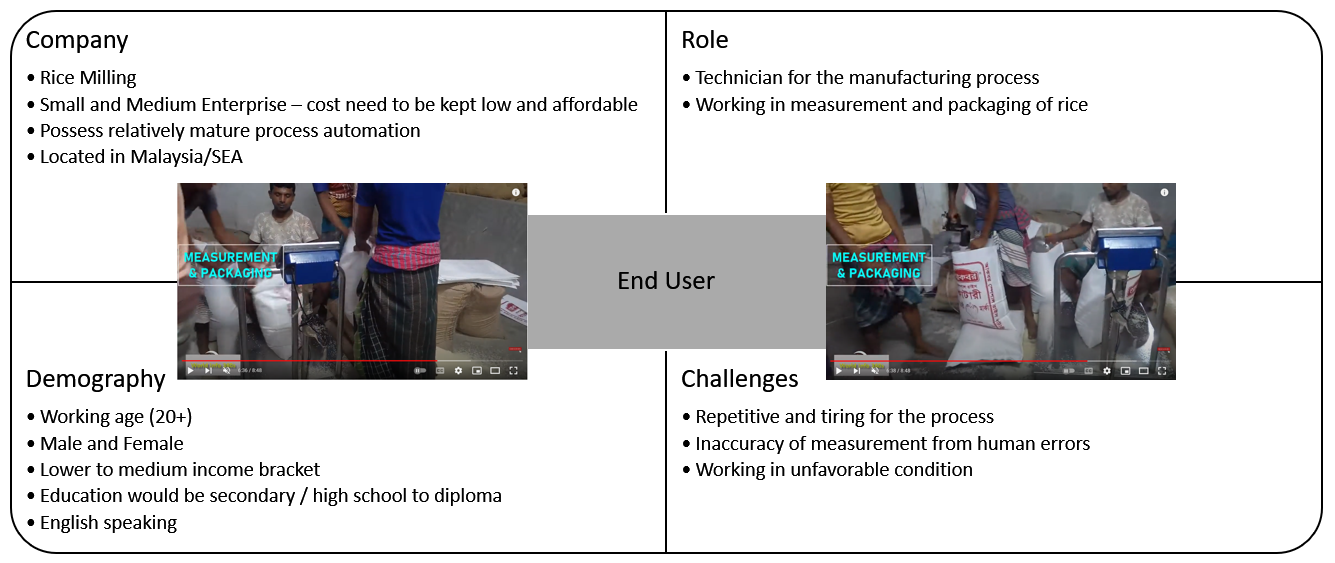


Figure 5‑1: End User Persona

# Hardware Requirements - Hafiidz

|  |  |  |
| --- | --- | --- |
| Feature ID | Hardware Feature Requirements | Description |
| HR001 | Electrical power supply/battery | To have a connection to power source  Input Voltage 12 to 24 Vdc |
| HR002 | 6 DOF Robotic Arm Frame | The frame that holds all components of robotic arm |
| HR003 | Programmable Logic Controller (PLC) or equivalent | To be able to program and link all sensors, controllers (UI/UX) and actuators |
| HR004 | Logging and Storage | To be able to store data from sensors, logs and other sources |
| HR005 | Switches/Touch User Interface | To be able to control the robotic arm |
| HR006 | Actuators  Servos | To be able to move/rotate robotic arm to specific angle and position as outlined by user or algorithm |
| HR007 | Sensors and Detection | Hardware has sensors to be able to detect/ measure/record the followings:   * Weight sensor * Touch/Pressure sensor * Visual/environment sensor |
| HR008 | Interfaces | To be able to interface with other relevant unit   * Modbus * Switches/Touch UI |

# Software Requirements - Sarah

|  |  |
| --- | --- |
| Software Requirements | Description |
| Operating System | Windows, MacOS, Linux |
| PC Memory | Minimum of 256MB RAM, 600MB Disk Space |
| Arduino IDE 1.8.0 | Software based on the C language used to program  the code to be integrated to the controller |
| Programming Language | Python, C or C++ |
| CPU | Intel Pentium 4 and above |
| Arduino Board USB Driver | Java JRE |
| Arduino Board Flash Memory | 32KB of which 4KB are used by bootloader |
| Web Application | Simple web application hosted on local computer as an interface for end users |

# System Architecture – Chong CH

The system architecture are shown in Figure 8‑1 where the relationship between each component of the system are linked together with an arrow indicating the flow of data or signal from and to the PLC which is the central controller for the system.

A picture containing diagram

Description automatically generated

Figure 8‑1: System Architecture Diagram

It should be noted that the bidirectional arrow between the touch screen and PLC means that the data are pass to and fro between these 2 components as the information of the current state of the system is stored in the PLC and would need to be display to touch screen while the input from user need to be pass from the touch screen to PLC.

## State Diagram – Chong CH

From the Figure 8‑2, there are 4 possible states shown in circles for the system: (1) off, (2) on, (3) standby, and (4) operation. The arrows between these states are the transitions between states where the logical sequences need to be performed by the program to achieve the the respective states.

The “off” state is the situation where the system is powered down with no possible logical operation performed by the program. The “on” state represent the powered up situation where there are sequence of initialization and checking performed but with no processes of operation by user or the robot. The external trigger and action is inhibit during this state.

Diagram

Description automatically generated

Figure 8‑2: System State Diagram

The “standby” state is when the system is ready to perform robotic action based on any external triggers and condition set by design. Lastly, the “operation” state indicate that the robotic arm is in motion and the intended function of the system is being realized or performed.

The logical sequence to transfer the system state from “off” to “on” is termed “startup” while “shutdown” is for the transition of “on” to “off”. On the other hand, “trigger start” and “trigger stop” is for transition of “standby” to “operation” and “operation” to “standby” respectively.

## System Operation Flow Chart – Chong CH

The logical sequence for the transition between system states are shown as flow charts in Figure 8‑3, Figure 8‑4, Figure 8‑5, and Figure 8‑6. These figures correspond to the “Startup”, “Shutdown”, “Trigger Start”, and “Trigger Stop” sequence respectively.

A picture containing diagram

Description automatically generated

Figure 8‑3: Startup Logical Flow

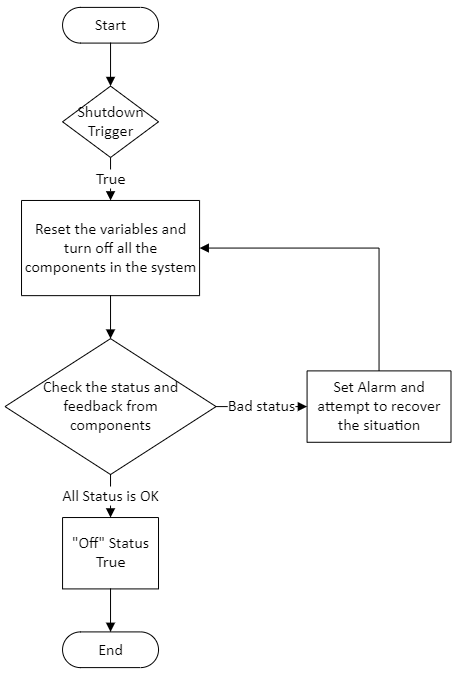


Figure 8‑4: Shutdown Logical Flow

Diagram

Description automatically generated

Figure 8‑5: Trigger Start Logical Flow

Diagram

Description automatically generated

Figure 8‑6: Trigger Stop Logical Flow

# External Interface Requirements

## User Interfaces – Sarah

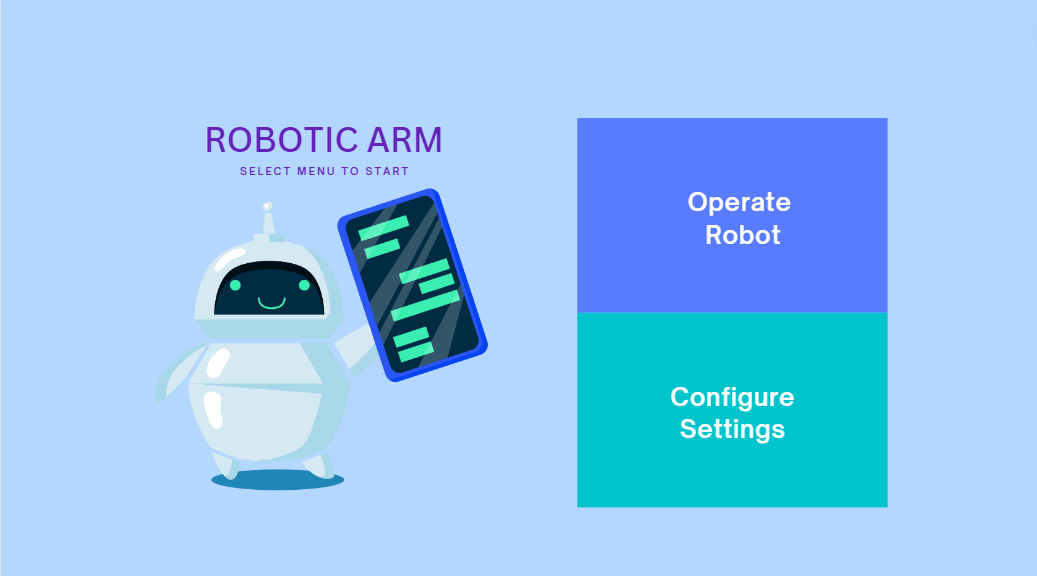


Figure X: User interface home page

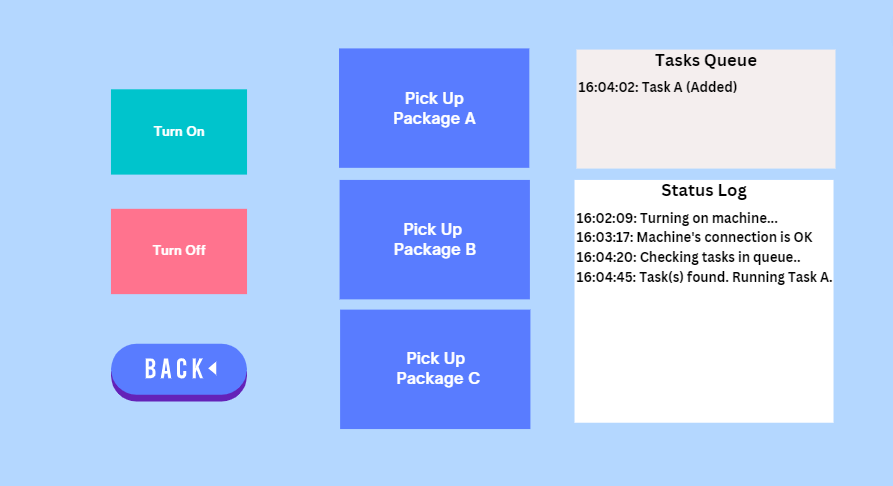


Figure X: Operate Robot page for end users

## Hardware Interfaces

*To be defined*

## Software Interfaces

*To be defined*

## Communications Interfaces

*To be defined*

# Use Case - Sarah

The use case here defined is to describe the operation of the system in reference to the operation of the end users.

## Use Case Chart

The use case has multiple personas interacting with each other, technician, engineer, admin and the robot arm. For basic tasks such as starting or shutting down the program, and selecting tasks to be executed can be done by all technician, engineer and admin which will also update the program logs for easy debugging.

The engineer will have an additional capabilities to update the configuration and settings such as updating the source code of the whole program, updating treshold and locations of the target starting point and destination.

An admin will have the same privileges as the technician and engineer but with an additional rights to manage the users of the system. The admin will be able to add or delete users and assign appropriate roles to each of them.

On the other hand, once any task is selected to be executed, it will send a trigger for the robotic arm to do the necessary movements to achieve its goal. The robotic arm is able to do arm extension and extraction, rotate, contract and release its grip from the object.

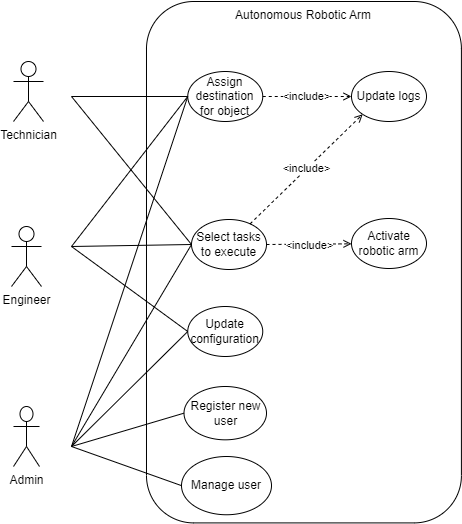


Figure XX: Use case diagram

# Functional Requirements

Functional requirements are a summary of all characteristics and features that had been stated in the previous sections. These requirements range from system, user, hardware to software. Technical details might be included to help describing the functionality herein contained.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Requirements | Priority | Features |
| 1 | The system can power up all the components and devices. | High | Startup |
| 2 | The user interface is available and showing the statuses of the system and components. | High |
| 3 | The system can perform some safety features and interlock to prevent mishandling of the system. | Medium | Safety |
| 4 | Set the maintenance mode for the system to disable the system ability to power on. | Low |
| 5 | There are 3 roles or access level: technician, engineer, and admin. | High | User Access |
| 6 | The admin can create the user account and assign the different roles and possess full access to the system. | High |
| 7 | The engineer can access and edit the configuration and threshold of the system and include the technician functionality. | High |
| 8 | The technician can operate the system, e.g., startup, shutdown, create task, trigger start, and trigger stop. | High |
| 9 | The user can create task where a task is one complete operation of moving object from point A to B. | High | Task Creation |
| 10 | The user can decide the final location (point B) of the object for the task. | High |
| 11 | The user can associate each different weight threshold for each task to decide the final location (point B) of the object. | High |
| 12 | The creation of more than 1 task. | High |
| 13 | The queueing of the tasks and first in first out in executing the tasks in queue. | Medium |
| 14 | The display of the tasks in user interface. | High | Task Display |
| 15 | The deletion of the tasks in user interface. | Medium |
| 16 | The timestamp for the tasks created. | Medium |
| 17 | The timestamp for the tasks completed. | Low |
| 18 | Trigger start the system and execute the task. | High | Execution of task |
| 19 | Unable to trigger start the system if any of the components is in alarm or unhealthy mode. | Low |
| 20 | Automatically execute the subsequent tasks in queue for multiple tasks. | Medium |
| 21 | Manually trigger the start of the next tasks 1 at a time. | High |
| 22 | Manually trigger the start of next tasks 1 at a time even though there are multiple tasks in queue. | Medium |
| 23 | Automatically stops when all the tasks in queue are completed or no task is in queue. | High | Stopping the execution of task |
| 24 | Trigger stop the system if any of the components is faulty during the operation. | Medium |
| 25 | Manually trigger stop the system and the operation cease when the tasks in execution reach the end. | High |
| 26 | Emergencies stop all the operation and power down the system. | Low |
| 27 | The robotic arm able to pick up the object based on the tasks created. | High | Operation of Robotic Arm |
| 28 | The robotic arm able to transfer the object based on the tasks created. | High |
| 29 | User able to manually operate the robotic arm. | Low |
| 30 | The robotic arm pressure sensors able to detect the pressure exerted on the object. | High |
| 31 | The weight sensor able to measure the weight of the objects. | High | Weight sensors for object detection |
| 32 | The system able to make decision on the weight of the objects and placed the object in the designated location. | High |
| 33 | The system can detect whether the destination location of the object is empty based on the weight sensors. | Low |
| 34 | The system able to make decision on the weight of the objects and placed the object in the most optimized location (nearest). | Low |

# Robotic Arm

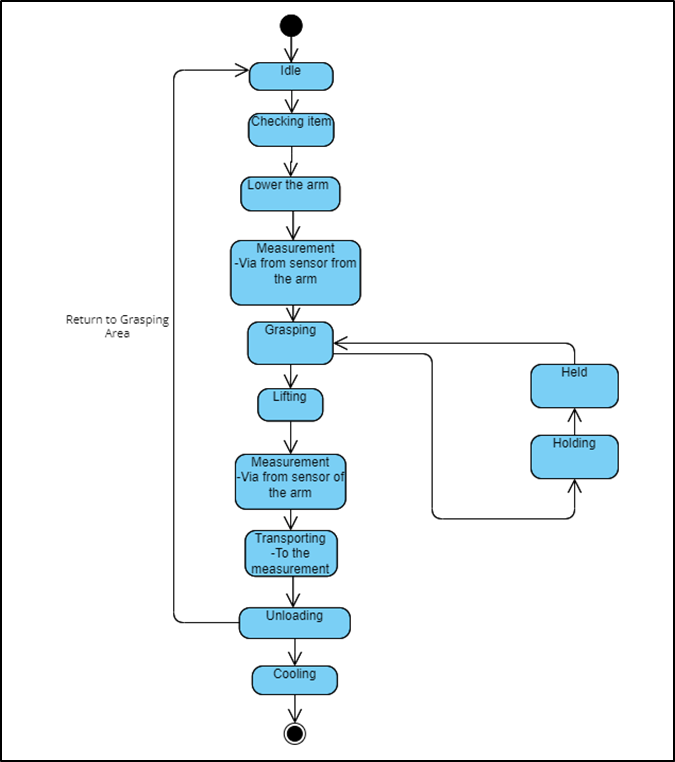


Figure 12:1 State Diagram of Robotic Arms

Flowchart describe – Yulin Lei

The machine is idle in its initial state. When we run the machine, the machine will run the robotic arm to the initial working position and point the camera at the assembly line. After the alignment operation is completed, the camera will measure the position of the cargo. At this time, the robotic arm can obtain the offset between the cargo position and the preset position. Substitute the data into the algorithm to obtain a precise trajectory and grab the goods and start moving towards the pallet. When the robotic arm moves to the preset position of the pallet, the camera will start to measure the cargo layer, and substitute the measured data to move the robotic arm to the top of the pallet. Then continue to make position corrections through the top cargo positioning point. Finally put the goods in the right place. Once the process is complete, the robotic arm can return to idle for the next job.

## Algorithm – Gary

### Algorithm Overview

Genetic Algorithm (GA) approach to optimisation is based upon the concept of the survival of the fittest. GA is the evolutionary algorithm which described as the strongest elements become stronger and weakest element to be eliminated in the natural world. The solution of the optimisation task using GA concept involves stochastic search of the solution space using strings of integers known as chromosomes (parameters to be optimised). Each integer from the chromosome called gene, and each gene has a decimal value between 0 and 9. The initial population of chromosome is generated at random and these are decoded to obtain the corresponding parameters. The parameters values are introduced into the system model. The fitness function will take a candidate solution to the problem as input and produces the output and finally determines how well it fulfils the criteria the algorithm is optimising for. The selection function takes the population and the results from the fitness function to determine who should reproduce. The reproduction function regulates how to expand the population based on the existing members. The reproduction involves retaining the best chromosomes for the next population. The other chromosomes will be replaced by new chromosomes using processes of crossover and mutation.

Crossover is the process involves two chromosomes from the parent chromosome engage in procedure in which some genes from one chromosome are interchanged with genes from the corresponding positions in the other. Mutation involves the selection from the certain number of the genes in the current population and random alterations are made in their values. Once the chromosomes are altered to form new population they have to be evaluated. The whole procedure is then repeated for number of iterations (generation) to produce final output.

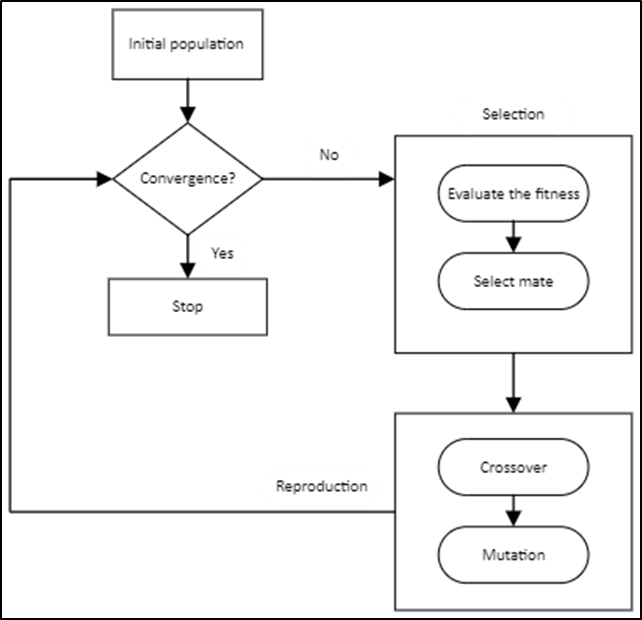


Figure 10.1 Genetic Algorithm Flowchart

### Justification of Chosen Algorithm

From Kumar et al. (2017), Genetic algorithm has been verified to be useful to treated optimisation problem. For many problems, GA may not be most sufficient way of determining the desired solution(s). GA techniques consists of selecting a large number of diverse solutions while builds on solutions that have found to have high fitness. GA can be excelled where other search algorithm will fail such as problems in which derivatives are extremely difficult or impossible to compute or problems which are ill-conditioned. Genetic Algorithm as opposed to traditional methods is the global optimisation strategy. The effectiveness of crossover allows the exchange of information between individuals separated by large distances in search spaces. It ensures the combination of individuals of high fitness, and the possibility that resulting solutions will inherit the strengths of both parents’ chromosomes.

According to Samet and Fakharian (2018), mutation of offspring solutions is great advantage as it permits to produce offspring trail solutions with “genetic segments” that vary from that of the parents to such a degree that these solutions are not confined to local optima. GA can search the parameter space in many directions simultaneously. Other heuristic methods perform iteration on single solution, GA make use of multiple offspring to explore a large number of regions at once with increasing the odds of a convergence upon the global optimum. From Friedrich et al. (2017) findings, the authors argue that GA works well in noisy environment condition. Therefore, Genetic Algorithm is generally successful at finding optimal solution to the problem.

Reference

Friedrich, T., Korzing, T., Krejca, MS., & Sutton, AM. (2017). The Compact Genetic Algorithm is Efficient Under Extreme Gaussian Noise. *IEEE Transactions on Evolutionary Computation*. Retrieved from https://ieeexplore.ieee.org/document/7577782

Kumar, S., Rani, K., & Banga, VK. (2017). Robotic Arm Movement Optimization Using Soft Computing. *IAES International Journal of Robotics and Automation (IJRA)*. Retrieved from https://www.researchgate.net/publication/332575161\_Robotic\_Arm\_Movement\_Optimization\_Using\_Soft\_Computing

Samet, V. & Fakharian, A. (2018). Energy efficiency in the robot arm using genetic algorithm. *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*. Retrieved from https://ieeexplore.ieee.org/document/8406625

# Programmable Logical Devices (PLC)

*To be defined*

# Sensors and Actuators

*To be defined*

## Weight

*To be defined*

## Pressure

*To be defined*

# Implementation

## Data Structure

# References

**There are no sources in the current document.**