

# EV3 Hardware and Programming

## International Project Engineering

Computer Science for Engineers IPEB3 WS22/23

## by

Encarnacion Ortega - 808412 Ferlando Mkiva - 808238 Laura Orozco - 808415

**03 November 2022 - 19 January 2023**

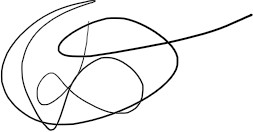


# Declaration

### We have read and understand the Allgemeine Studien- und Prüfungsordnung für das Bachelor- und Masterstudium der Hochschule Reutlingen - § 13 Täuschung, Ordnungsverstoß, Plagiat [1](#_bookmark0)

Therefore, we hereby declare that the work contained within this document is of our own and that all sources and their authors have been listed and appropriately recognized for their contribution in the completion of this document.

**STUDENT SIGNATURES**



### Encarnacion Ortega (808412) Ferlando Mkiva (808238) Laura Orozco (808415)

1[https://www.reutlingen-uni](http://www.reutlingen-university.de/fileadmin/user_upload/2022_05_23_StuPro_AllgemTeil_WiSe_22-)versity[.de/fileadmin/user\_upload/2022\_05\_23\_StuPro\_AllgemTeil\_WiSe\_22-](http://www.reutlingen-university.de/fileadmin/user_upload/2022_05_23_StuPro_AllgemTeil_WiSe_22-)

23\_2022-05-11.pdf

# Executive Summary

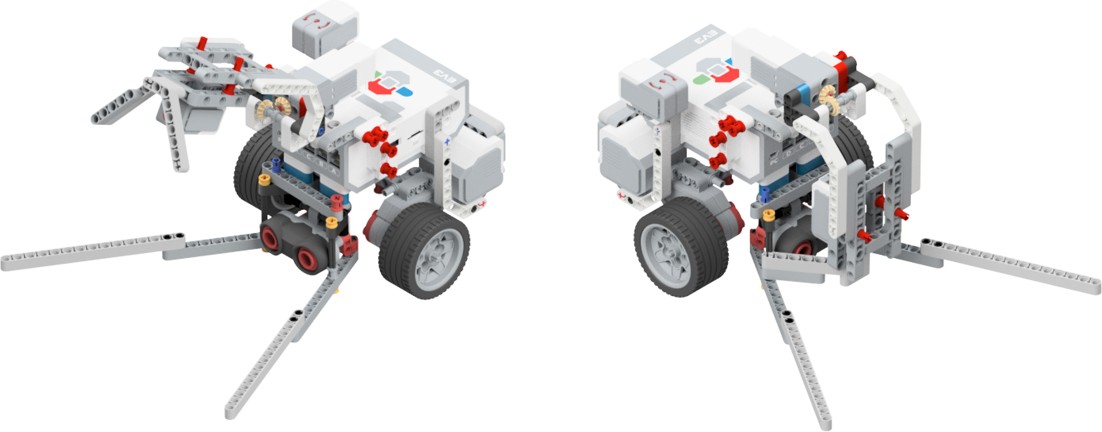


Figure 1: Fruta Oes

This document illustrates the process to develop a robot using the Scrum methodology. It was part of the Computer Science lecture and was designed to introduce students to Scrum and software development.

The Scrum methodology will be used to manage the project, which involves dividing the project into small, manageable parts called sprints. Each sprint will have specific goals and will last for a week.

The team will begin by defining the requirements of the robot and creating a product backlog. The team will then hold a sprint planning meeting to determine which tasks will be completed during the next sprint. During each sprint, the development team will work on building the robot, and the Scrum Master will hold daily stand-up meetings to ensure that everyone is on track and that any obstacles are addressed. The team will also hold a sprint review meeting at the end of each sprint to review what has been accomplished and plan for the next sprint.

Once the robot is completed, it will be tested to ensure that it meets the requirements and that it is able to perform the task for which it was designed. The team will hold a retrospective meeting to review the project and identify any areas for improvement for future projects.

By using the Scrum methodology, the project will be well-organized and efficient, and the final product will be a high-quality, functional robot that is capable of performing its main task: harvesting fruits.

This task makes the robot’s mechanical structure the most important part. Thus, this document will discuss in detail all the sub-components of the mechanical design in section [2](#_bookmark6) on page 2[,](#_bookmark6) followed in section [3](#_bookmark18) on page [6](#_bookmark18) by the software design where a full discussion on all the algorithms used, activity diagrams and testing will be provided. The next step will be the management of the software development: section [4](#_bookmark24) on page [8](#_bookmark24) highlights the Agile Software engineering (SCRUM) procedure followed in this project. The SCRUM section will briefly discuss the project setup in section [4.1](#_bookmark25) on page [8.](#_bookmark25) Then sections [4.2,](#_bookmark31)  [4.3,](#_bookmark35) and [4.4](#_bookmark37) present the role and progress achieved in sprints 1, 2, and 3 respectively. Finally, discussions and reflections will be detailed in section [5](#_bookmark38) on page [11](#_bookmark38) .

|  |  |
| --- | --- |
| **Table of Contents** |  |
| [**List of Figures**](#_bookmark2) | **iv** |
| [**List of Tables**](#_bookmark3) | **iv** |
| [**1 Introduction**](#_bookmark4) | **1** |
| [**2 Mechanical Design**](#_bookmark6) | **2** |
| [2.1 Object Detection and Manipulator Module](#_bookmark9) . . . . . . . . . . . . . . . . . . . . . | 3 |
| [2.2 Gripper and Colour Sensing Module](#_bookmark12) . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| [2.3 Drive Base](#_bookmark15) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| [**3 Software Design**](#_bookmark18) | **6** |
| [3.1 Main Code](#_bookmark19) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| [3.2 User Menu for Selecting the Colour (fruit)](#_bookmark21) . . . . . . . . . . . . . . . . . . . . . | 7 |
| [**4 Agile Software Engineering**](#_bookmark24) | **8** |
| [4.1 Project Setup](#_bookmark25) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| [4.1.1 Project Vision](#_bookmark26) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| [4.1.2 SCRUM Team](#_bookmark27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| [4.1.3 Product Backlog](#_bookmark29) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
| [4.2 Sprint 1](#_bookmark31) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |
| [4.2.1 Goals of the sprint](#_bookmark32) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |
| [4.2.2 Burn-down Chart](#_bookmark33) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |
| [4.3 Sprint 2](#_bookmark35) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
| [4.3.1 Goals of the sprint](#_bookmark36) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
| [4.4 Sprint 3](#_bookmark37) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
| [**5 Recommendations and Conclusion**](#_bookmark38) | **11** |
| [**References**](#_bookmark38) | **12** |
| [**Appendices**](#_bookmark43) | **13** |
| [**A Library importing and Device Setup**](#_bookmark44) | **14** |
| [**B User Menu for Selecting the Colour**](#_bookmark45) | **15** |
| [**C Function for turning left**](#_bookmark46) | **15** |
| [**D Function for turning right**](#_bookmark47) | **16** |
| [**E Function for moving the robot forward**](#_bookmark48) | **16** |
| [**F Function for moving the robot from Acre to Home-Zone**](#_bookmark49) | **17** |
| [**G Function for moving the robot from Acre to Main-Line**](#_bookmark50) | **17** |

[H Function for operating the Gripper](#_bookmark51) 18

1. [Function for Searching the fruits](#_bookmark52) 18
2. [Main Code](#_bookmark53) 20

List of Figures

[1 Fruta Oes](#_bookmark1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ii

[2 Object Detection and Manipulator Module](#_bookmark5) [[1]](#_bookmark39) . . . . . . . . . . . . . . . . . . . 1

[3 Main Components of Fruta Oes](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1. [Object Detection and Manipulator Module](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . 3
2. [Gripper and Colour Sensing Module](#_bookmark13) . . . . . . . . . . . . . . . . . . . . . . . . 4

[6 Drive Base](#_bookmark16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

[7 Main Code Activity Diagram](#_bookmark20) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

[8 Menu Code Activity Diagram](#_bookmark22) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

[9 SCRUM Team](#_bookmark28) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

[10 Sprint 1 Burnd-own Chart](#_bookmark34) 10

List of Tables

[1 Main Components](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1. [Object Detection and Manipulator Module Components](#_bookmark11) . . . . . . . . . . . . . . 3
2. [Gripper and Colour Sensing Module Components](#_bookmark14) . . . . . . . . . . . . . . . . . 4

[4 Drive Base Components](#_bookmark17) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

[5 Product Backlog](#_bookmark30) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

# Introduction

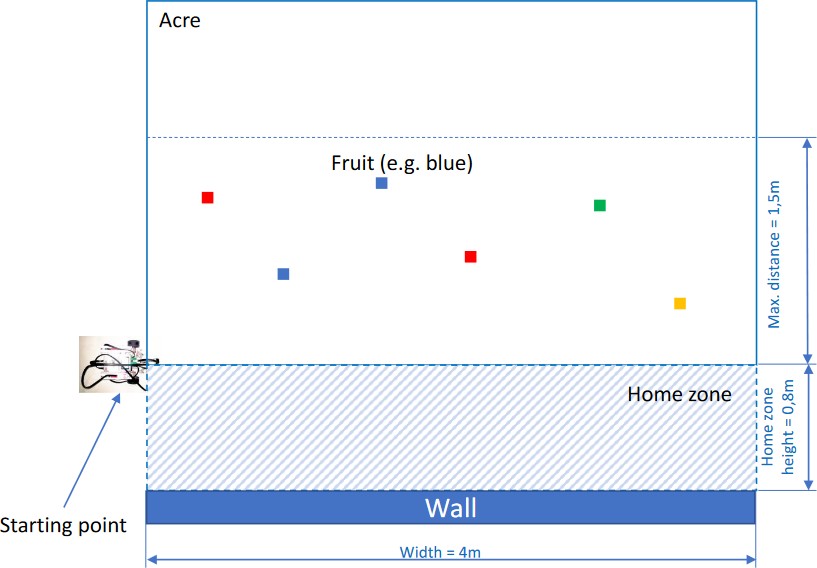


Figure 2: Object Detection and Manipulator Module [[1]](#_bookmark39)

Agricultural intensification, mechanization, and automation have all led to major increases in agricultural productivity throughout time [[2](#_bookmark40)]. Robots are intelligent machines that may be trained to carry out particular jobs, make choices, and take immediate action. They are needed in a variety of industries, and they function best in stable environments where consistent accuracy and high productivity are required [[3].](#_bookmark41) Robots have become increasingly important in today's companies as they offer a wide range of benefits, including increased efficiency, improved safety and reduced labor costs.

With the fast pace of technological advancement, it is expected that robots will play an even greater role in companies in the future. Companies that invest in robots will be better able to compete in today's global economy, and will also be well positioned to take advantage of new opportunities as they arise.

Another important aspect of robots in today's companies is their ability to be eco-friendly. Robotics technology has advanced to such a level that it is now possible to design robots that are energy-efficient and use sustainable materials. Additionally, many robots are designed to be lightweight and compact, which reduces their overall energy consumption and carbon footprint.

Moreover, robots can also be used in the field of agriculture and farming, where they can help to improve crop yields and reduce the need for pesticides and other chemicals. This can help to reduce the environmental impact of farming, while also increasing food production. As robots continue to evolve and become more advanced, it is important for companies to consider the environmental impact of their usage and take steps to make them as eco-friendly as possible. By doing so, they can not only improve their bottom line, but also contribute to a greener world.

Considering this, the aim of this project is to create a product that can be useful for the companies and also eco-friendly, due to the increasing demand of less environment-harmful features.

LEF BOTS GmbH is a robot specialized company. The name of the firm is created from the initials of the stakeholders: **L**aura; **E**ncarnacion; **F**erlando **(LEF)**. In this text, the firm presents the development of its major product; the fruit harvesting robot. The robot is given an intuitive name **Fruta Oes**, which is a combination of two languages Spanish and Afrikaans. Fruta means fruits in Spanish and Oes means to harvest in Afrikaans, thus, the name of the robot **Fruta Oes** means fruits harvester. One might be asking themselves why Spanish and Afrikaans, well the answer is simple, the stakeholders of the firm are from Spain (Laura and Encarnacion) and South Africa (Ferlando). Figure [1](#_bookmark1) Presents **Fruta Oes**.

This robot main task is to ask for a fruit color, collect the 2 desired fruits and place them in the home zone (Figure 2) within 5 minutes. Figure 2 shows the home zone position, the dimensions of the acre and some additional details. The mentioned fruits are wooden cubes colored blue, red and green, with an edge dimension of 2.5cm. Our design is to give input to the robot by pressing the button with the fruit’s color they want to harvest, but apart from that the robot is completely autonomous. Fruta Oes has to return to the starting point after completing its main task, with an error margin of 20cm.

A Lego EV3 Mindstorms set is used as a basis for the product. The Mindstorms set includes different Sensors and motors that are operated by a microcontroller brick, which serves as the power supply and the main control unit to the other technical parts. It is capable of storing small programs, programed via Python 3.

# Mechanical Design

This section presents the mechanical design of **Fruta Oes** robot. The section will start presenting how the sub-components fit and work together, then proceed by discussing each component in detail. Figure [3](#_bookmark7) depicts the five main components of **Fruta Oes**.

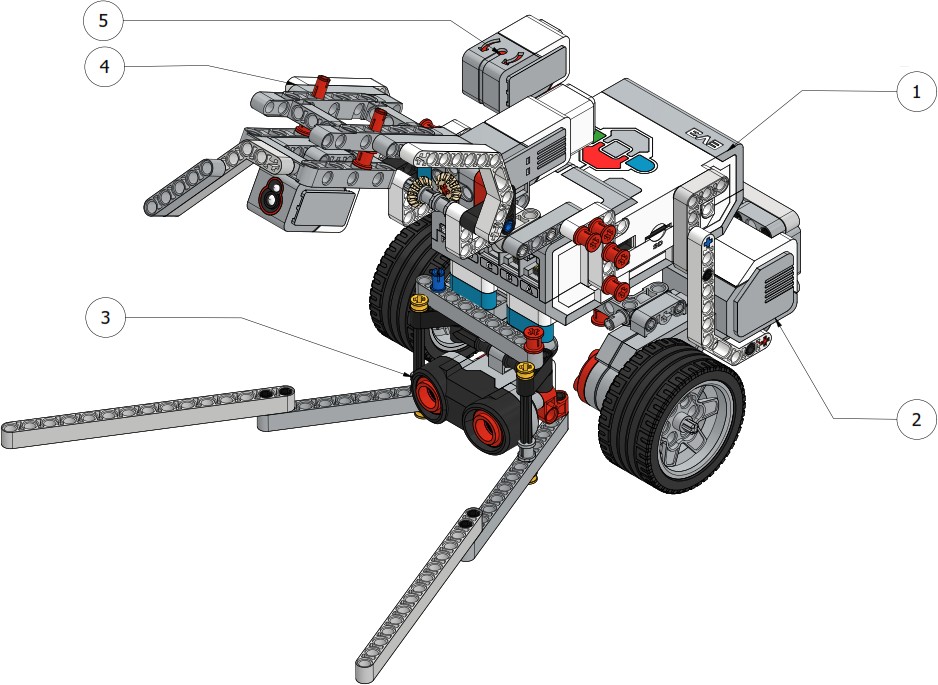


Figure 3: Main Components of Fruta Oes

The five main components of this robot are listed in table [1.](#_bookmark8) **Item 1-EV3 Brick** is the heart and brain of the robot, the processing of the software takes place in this module. **Item 2-Drive Base** serves two purposes; providing stability and mobility, the in-depth design of this module will be discussed in section [2.3](#_bookmark15) on page [5.](#_bookmark15)

Table 1: Main Components

### Item Component

* 1. EV3 Brick
  2. Drive Base
  3. Object Detection and Manipulator Module
  4. Gripper and Colour Sensing Module
  5. Gyro-Sensor

**Item 3-Object Detection and Manipulator Module** not only enable object detection capabilities but also makes sure that the object is manipulated to be in a good position for the gripping action. Object Detection and Manipulator Module will be fully discussed in section [2.1](#_bookmark9) on the next page. **Item 4-Gripper and Colour Sensing Module** grips the object and makes sure it is in a good position for the colour sensor.

* 1. **Object Detection and Manipulator Module**

This section discusses the object detection mechanism. This module not only enable object detection capabilities but also makes sure that the object is manipulated to be in a good position for the gripping action.

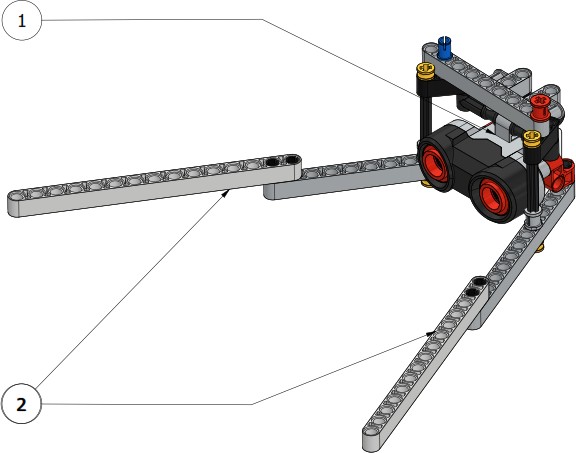


Figure 4: Object Detection and Manipulator Module

Table 2: Object Detection and Manipulator Module Components

**Item Component**

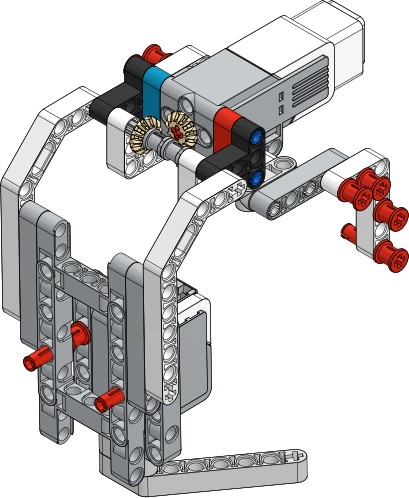
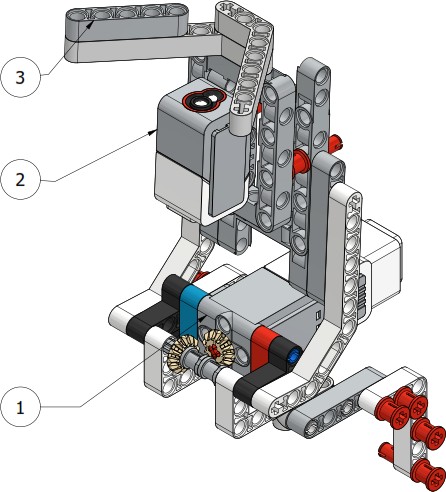
* + 1. Ultra Sonic Sensor
    2. Manipulator Fangs

Figure [4](#_bookmark10) shows the object detecting manipulator, it is composed of two main parts listed in table [2:](#_bookmark11) the ultrasonic Sensor which enables the fruit detection capability for the robot and the manipulator fangs. The ultra-sonic sensor is positioned as low as possible, because the objects to be detected are of 2.5cm height. This sensor is also used to provide distances to the robot as it is also capable of computing the distance to the detected object. The EV3 sensor has a measuring range of 250 cm and an accuracy range of 1cm. During the testing phase of the sensors, it was noted that the sensor is sometimes losing the object when the robot is approaching the detected object, this lead to the introduction of the manipulator fangs which are just LEGO beams positioned in such a way that they manipulate the object to slide and ends up in-front of the sensor. This mechanism improved the object detection and made the gripping operation and colour sensing easier. The latter will be discussed later in the document.

*±*

## Gripper and Colour Sensing Module

This section presents the design of the colour sensing and gripping mechanism. This module is also equipped with manipulating fangs. This module is designed to with two states, opened and closed, the module must be fully opened during the searching of fruits in the acre to avoid distracting the signal of the ultrasonic sensor module that was discussed in section [2.1](#_bookmark9) on the preceding page.



1. Opened Gripper (b) Closed Gripper

Figure 5: Gripper and Colour Sensing Module

Table 3: Gripper and Colour Sensing Module Components

**Item Component**

* 1. EV3 medium servo motor
  2. EV3 Colour Sensor
  3. Gripper Manipulator fangs

Figure [5](#_bookmark13) depicts te gripper and the colour sensing module. figure [5a](#_bookmark13) shows the gripper in opened mode and figure [5b](#_bookmark13) shows the gripper in closed mode. This module comprises of three major components: EV3 medium servo motor, this motor is a great solution when the design of the robot requires shorter response times like the harvesting robot designed in this project. The motor is used for opening and closing mechanism of the gripper, it is equipped with 240 to 250 rpm and 0.08 Nm of running torque. The next component is an EV3 Colour sensor, the main function of this component is to enable the robot to distinguish between the required fruits and non-required fruits. The EV3 Colour Sensor is capable of detecting seven colours plus the absence of colour. It can tell the difference between colour or black-and-white or among blue, green, yellow, red, white, and brown. This sensor samples at a rate of 1 kHz, it must be positioned 1cm away from the object for better results.

## Drive Base

In this section, the assembling of the drive base for the robot is discussed. it is made up of two parts: EV3 large motors and the wheels, also listed in table [4.](#_bookmark17) there is also a balancing wheel positioned underneath the base close to the rear but this wheel is not shown in figure [6.](#_bookmark16)

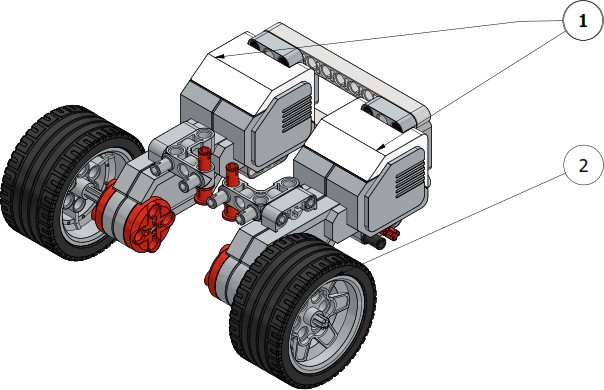


Figure 6: Drive Base

Table 4: Drive Base Components

**Item Component**

1. EV3 Large Motors
2. Wheels

The EV3 Large Servo Motor uses tacho feedback for accurate control to within one degree [[4].](#_bookmark42)

# Software Design

## Main Code

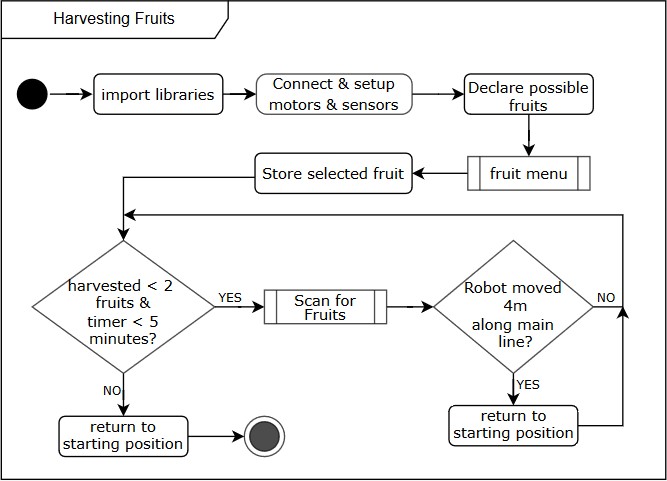


Figure 7: Main Code Activity Diagram

## User Menu for Selecting the Colour (fruit)

As it can be seen in figure [1](#_bookmark1) on page [ii,](#_bookmark1) the EV3 brick buttons are coloured with all possible colours of the project: Left button for Blue, Up button for Red and right button for green. The user must wait for the beep after the robot asked them to select the colour from the buttons. Based on the pressed button, the robot will confirm the chosen colour and proceed to search for that specific fruit in the acre. Figure [8](#_bookmark22) presents the activity diagram for User Menu function, the corresponding source code is attached on appendix B.

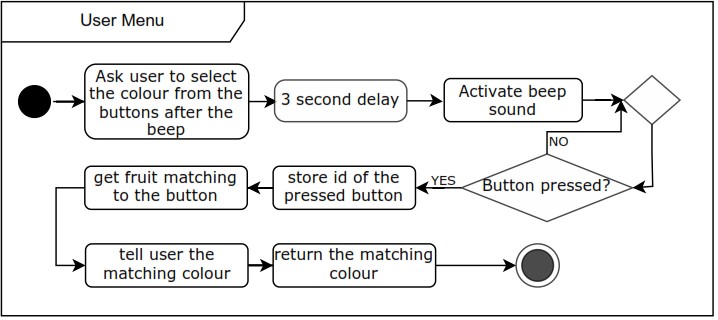


Figure 8: Menu Code Activity Diagram

Algorithm [1](#_bookmark23) presents a pseudo code for this user menu method. A five minute timer is started at the beginning of the function, because one of the requirements is that the robot must return to the starting position after five minutes and this time must be measured as soon as the robot starts interacting with the user. This algorithm is constructed from the activity diagram above.

**Algorithm 1** : User Menu Method Pseudo Code

1: Start the 5 minutes timer;

2: Use EV3 Speaker function to ask the user to select the colour from the buttons after the beep;

3: sleep for 3 seconds to allow the sound command to complete;

4: Activate the EV3 Beep Sound;

5: **while** *True* **do**

6: **if** Any button is pressed **then**

7: Store the id of the pressed button;

8: break out of the while loop;

9: **end if**

10: **end while**

11: Use the id of the pressed button to get matching fruit from the fruit dictionary;

12: Use EV3 speaker to tell the user the matching colour;

13: Return the matching colour to the rest of the code;

# Agile Software Engineering (Analysis: Plan how to tackle the problem?)

## Project Setup

~~LEF BOTS GmbH is a robot specialising company, the name of the firm is created from the initials of the stakeholders:~~ **~~L~~**~~aura;~~ **~~E~~**~~ncarnacion;~~ **~~F~~**~~erlando~~ **~~(LEF)~~**~~. In this text, the firm presents the development of its major product; the fruit harvesting robot. The robot is given an intuitive name~~ **~~Fruta Oes~~**~~, which is a combination of two languages Spanish and Afrikaans. Fruta means fruits in Spanish and Oes means to harvest in Afrikaans, thus, the name of the robot~~ **~~Fruta Oes~~** ~~means fruits harvester. One might be asking themselves why Spanish and Afrikaans, well the answer is simple, the stakeholders of the firm are from Spain (Laura and Encarnacion) and South Africa (Ferlando). Figure~~ [~~1~~](#_bookmark1) ~~on page~~ [~~ii~~](#_bookmark1) ~~Presents~~ **~~Fruta Oes~~**~~.~~

### Project Vision

"For harvesters who care for the environment. We believe in a world where harvesters can harvest being eco-friendly"

### SCRUM Team

Figure [9](#_bookmark28) presents the scrum team: Encarnacion Nunez-Ortega is the **Product Owner** and respon- sible for determining what needs to be done. Laura Ponce-Orozco is the team’s **Scrum master** and responsible for removing all impediments. The **development team** consist of Ferlando Mkiva, Encarnacion Nunez-Ortega and Ponce-Orozco together will determine how to deliver chunks of work in frequent increments.



Figure 9: SCRUM Team

### Product Backlog

Table 5: Product Backlog

### Priority User Story Effort

* + - 1. As a user, I want the robot to be completely autonomous.



* + - 1. As a user, I want the robot to ask me for the correct fruit and to confirm that it understood me.



* + - 1. As a developer, I want the robot to be able to detect a 2.5cm cube.
      2. As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed.



* + - 1. As a user, I don’t want the robot to return to the home zone without the fruit grabbed.



* + - 1. As a user, I want the robot to place the right fruit in the home zone.
      2. As a developer, I want the robot to keep count of the harvested fruits
      3. As a user, I want the robot to immediately return to the starting point after collecting two fruits within 5 minutes.



9

10

As a user, I don’t want the robot to search for fruits outside the acre.

As a developer, I want the code to be efficient.

## Sprint 1

### Goals of the sprint

The goal of this sprint was to tackle the following user stories. In total eight cups of coffees must be consumed as the effort of this sprint.



**Priority 2**



As a **user**, I want the robot to ask me for the **correct fruit** and to confirm that it under- stood me.

**Priority 3**

As a **developer**, I want the robot to be able to **detect** a 2.5cm cube.

**Priority 4**

As a **user**, I want the robot to look for the **specified fruit** and check if the fruit is cor- rectly grabbed.

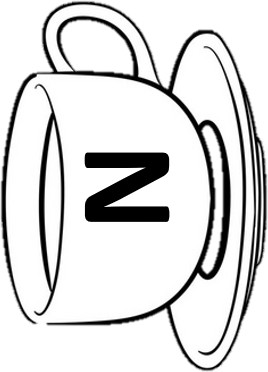


### Burn-down Chart

8

**Burn-down Charts**

Actual Ideal

6

4

**Coffees**

2

00 1 2 3 4 5 6 7

### Days

Figure 10: Sprint 1 Burn-down Chart

The increment of this Sprint was 4 instead of 8 coffees in view of the fact that we underestimated the tasks. We have an incomplete user story which states: As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed. This user story was deemed incomplete as the robot's capabilities were limited to searching for any fruit and not a specific one, as well as the inability to properly grasp the fruit. As a result, we had to add 4 extra story points to the next Sprint and address the incomplete user story during the next Sprint Planning.

**Test: expected and actual results, bug fixing.**

User story: As a **user**, I want the robot to ask me for the **correct fruit** and to confirm that it under- stood me.

We initially attempted to identify the color of the fruit using a color sensor and showing directly the cube, but it proved to be unreliable as it also picked up the color of the floor. To address this, we limited the sensor to only detect blue, green, and red. However, this caused confusion when distinguishing between red and green. To improve accuracy, we implemented a manual method, using buttons on the brick to identify the color.

## Sprint 2

### Goals of the sprint

The goal of this sprint was to tackle the following user stories. In total twelve cups of coffees must be consumed as the effort of this sprint.

**Priority 2**



As a **user**, I want the robot to place the right fruit in the home zone.

**Priority 3**

As a user, I don’t want the robot to return to the home zone without the fruit grabbed.

**Priority 4**

As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed.



**Priority 4**

As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed.



Burn-down chart

The increment of this Sprint was 11 instead of 12 coffees in view of the fact that we underestimated the tasks. We have an incomplete user story which states: As a user, I want the robot to be completely autonomous. This user story was deemed incomplete as the robot's brick did not detect the program correctly. As a result, we had to add 1 extra story point to the next Sprint and address the incomplete user story during the next Sprint Planning.

**Test: expected and actual results, bug fixing.**

User story: As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed

Initially, we aimed to calculate the angle at which we needed to rotate the robot in order to detect the specified fruit. To do this, we measured the detection range of the ultrasonic sensor by positioning a cube at varying distances and marking the points at which the sensor first detected the cube. We then calculated the center of this detection range as a point of reference for approaching the fruit. However, we found that this method was not consistently accurate as the detection range varied. As an alternative, we decided to rotate the robot to the left by 90 degrees and then save the angle at which the fruit was first detected (see Software Design). Through testing with various positions of the fruit, we determined that this approach was the most effective.

* 1. **Sprint 3**

### Goals of the sprint

The goal of this sprint was to tackle the following user stories. In total thirteen cups of coffees must be consumed as the effort of this sprint.

Burn-down chart

The increment of this Sprint was 11 instead of 13 coffees in view of the fact that we underestimated the tasks. We have an incomplete user story which states: As a developer, I want the code to be efficient. As a result, we spent the last week improving our code.

**Test: expected and actual results, bug fixing.**

User story: As a developer, I want the robot to keep count of the harvested fruits.

To ensure the functionality of our code, we conducted multiple tests using varying quantities of fruits to be harvested. The results were satisfactory and the code performed as intended.

# 5 ~~Recommendations and Conclusion~~(discussion and reflection?)

**Results of the sprint retrospectives:** In regard to the first sprint retrospective, we identified utilizing a shared screen as a significant positive aspect as it allowed for greater efficiency and improved error detection through simultaneous code review. A negative aspect identified was the time wasted on experimenting with various approaches to the same task. We always continued implementing the highlights and discarding the lowlights in all following sprints.

For the second sprint retrospective, we recognized a positive aspect in that we learned about the working styles of each group member and were able to capitalize on that knowledge. On the negative side, we acknowledged that we spent a significant amount of time attempting to execute a specific approach, instead of exploring new ideas.

For the third sprint retrospective, we considered that thinking creatively and embracing new ideas was a highlight. However, the team's motivation was negatively impacted by a series of errors related to the sensors.

**Final discussion**: Using Scrum as a didactical method for this project has been a valuable and beneficial experience. Scrum is a widely used and well-established framework for Agile software development, and its principles and practices can be applied to a wide range of projects.

One of the advantages of using Scrum in this context is that it promotes active participation and collaboration among team members. The Scrum meetings, such as sprint planning, daily stand-ups, and retrospectives, provide opportunities for the team to communicate and work together effectively, which is crucial for a successful project.

Furthermore, Scrum enables a focus on delivering value to the end-user through the use of user stories, which is a key aspect of the project, in this case, the end-user will be the farmer or the person who will use the harvesting robot.

In conclusion, working on a real-world project with Scrum has been a great opportunity to learn about Agile development and gain practical experience in project management, teamwork, and problem-solving, acquiring skills very useful for a student.

**References**

1. Danner M, Wachter M. Computer Science for Engineers: Phase 3 Challenge - Harvester. Reutlingen University - International Project Engineering. 2022;IPEB3 - WS22/23.
2. Nof SY, Nof SY. Springer handbook of automation. Springer; 2009.
3. Nof SY. Emerging trends and industry needs handbook of industrial robotics. John Wiley & Sons; 1999.
4. Robots LR. Raising Robots MINDSTORMS EV3 Large Servo Motor; 2023. Available from:

<https://raisingrobots.com/product/large-servo-motor/>.

Note: diagrams and graphs are works of the report authors. Specifically, drawings are created using Microsoft Viso Professional 2019 and Autodesk Inventor Student version 2023, algorithms are constructed using the LATEX "algpseudocode" package, and Code appendices are constructed using the LATEX "listings" package (with data obtained from MATLAB® R2016a Simulink). No work done by other persons is presented as that of the report author.

**Appendices**

# Library importing and Device Setup

**Code 1: Library importing and Device Setup**

1 #!/ usr / bin / env python3

2 import ev3dev . ev3 as ev3

3 from ev3dev . ev3 import \* # to allow the use of brick buttons

4 import math # importing Math Library to allow usage of Mathematical functions

5 import Color Sensor 2 # importing Color sensor library

6 from time import sleep # importing time library

7 import time

8 # -----------------------------------------------------------------

9 # Device Connections start

10 # Connect the outputs to the motor

11 motor\_left = ev3 . Large Motor (' outA ')

12 motor\_right = ev3 . Large Motor (' outD ')

13 gripper\_motor = ev3 . Medium Motor (' outC ')

14

15 # Gyro Sensor initialisation

16 gy = ev3 . Gyro Sensor ()

17

18 # Colour Sensor initialisation

19 cl= Color Sensor 2 . Color Sensor 2 ()

20 cl. mode =' COL - COLOR '

21

22 # Ultra - Sonic Sensor initialisation

23 us = ev3 . Ultrasonic Sensor ()

24 us. mode ='US - DIST - CM '

25

26 btn = Button () # activate the use of brick buttons

27

28 # Global Variables ( we know use of global variables is not a good practice , but in this case it is necessary )

29 harvested = 0 # keeping count of the harvested fruits

30 total\_distance = 0 # keeping track of the distance travelled by the robot

31 gripper\_opened = True # keeping track of the state of the gripper

32 thetha = None # To avoid the scanning angle from exceeding the limits

33 start = 0 # Variable to start the 5 minute timer

34 end = 0 # Variable to end the 5 minute timer

35 current\_time = 0 # Variable to track the current time

36 # Device Connections ends

# User Menu for Selecting the Colour

**Code 2: User Menu for Selecting the Colour**

1 def menu ():

2 # Function to select the desired fruit using the buttons from the Brick

3 global start

4 ev3 . Sound . speak (" Select the colour from the buttons after the beep "). wait ()

5 sleep (3)

6 Sound . beep (). wait ()

7 start = time . time () # Starting the timer

8 while True :

9 if btn . any ():# Checks if any button is pressed .

10 button\_id = btn . buttons\_pressed

11 print ( button\_id [0])

12 break

13 else :

14 sleep ( 0. 01)

15

16 option\_text = " I will search for {0} " # Robot confirming the colour

17 ev3 . Sound . speak ( option\_text . format ( fruit\_dict . get ( button\_id [0]) )). wait ()

18 return fruit\_dict . get ( button\_id [0]) # Returning the desired fruit

1. **Function for turning left**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code 3: Function for turning left** | | | | |
| 1  2  3  4  5  6  7  8  9  10  11 | def turn\_left ( speed , angle ):  # Function that turns left , it requires the turning it must turn  gy. mode = ' GYRO - RATE ' gy. mode = ' GYRO - ANG ' sleep (1)  while gy. value () >- angle :  motor\_left . run\_forever ( speed\_sp =- speed , stop\_action motor\_right . run\_forever ( speed\_sp =speed , stop\_action # print ( gy. value ())  motor\_left . stop () motor\_right . stop () | speed and  = ' hold ')  = ' hold ') | the | angle |

# Function for turning right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code 4: Function for turning right** | | | | |
| 1  2  3  4  5  6  7  8  9  10  11 | def turn\_right ( speed , angle ):  # Function that turns right , it requires the turning it must turn  gy. mode = ' GYRO - RATE ' gy. mode = ' GYRO - ANG ' sleep (1)  while gy. value () <angle :  motor\_left . run\_forever ( speed\_sp =speed , stop\_action = motor\_right . run\_forever ( speed\_sp =- speed , stop\_action # print ( gy. value ())  motor\_left . stop () motor\_right . stop () | speed and  ' hold ')  = ' hold ') | the | angle |

1. **Function for moving the robot forward**

**Code 5: Function for moving the robot forward**

1 def move\_distance\_in\_mm ( distance\_mm ):

2 # Function to move a specified distance , it transforms rotation of the wheels to distance .

3 dist = ( distance\_mm / 1000) \* ( 360 / 0 . 176 )

4 motor\_left . run\_to\_rel\_pos ( position\_sp = dist , speed\_sp =400 , stop\_action

= " brake ")

5 motor\_right . run\_to\_rel\_pos ( position\_sp = dist , speed\_sp =400 , stop\_action = " brake ")

6 motor\_left . wait\_while (' running ')

7 motor\_right . wait\_while (' running ')

8 motor\_left . stop ()

9 motor\_right . stop ()

# Function for moving the robot from Acre to Home-Zone

**Code 6: Function for moving the robot from Acre to Home-Zone**

1 def drive\_to\_home\_zone ( hypotenuse , current\_angle , add ):

2 # Function to drive to the home zone from the Acre with a correct fruit .

3 global total\_distance , harvested , start , end , current\_time

4 distance = abs ( hypotenuse \*( math . sin ( math . radians ( current\_angle )))) # Using Pythagorean theorem to calculate the distance from the fruit to the home - zone

5 total\_distance += abs ( hypotenuse \*( math . cos ( math . radians ( current\_angle ))

)) # Adding the distance moved forward to the total for returning to the starting point

6 turn\_right (100 , current\_angle + 85) # turning right 90 degrees ( we used 85 because the Gyro - sensor was not accurate )

7 gy. mode = ' GYRO - RATE '

8 gy. mode = ' GYRO - ANG '

9 sleep (1)

10 move\_distance\_in\_mm ( abs ( distance + add ))

11 gripper (" open ") # Open the gripper to release the fruit in the home - zone

12 harvested += 1 # keeping count of the harvested fruits

13 move\_distance\_in\_mm (- add ) # returning to the main - line

14 turn\_left (200 ,85) # Turning left to continue searching

15 end = time . time ()

16 current\_time += ( end - start ) # Checking the current time

# Function for moving the robot from Acre to Main-Line

**Code 7: Function for moving the robot from Acre to Main-Line**

1 def drive\_from\_acre\_to\_line ( hypotenuse , current\_angle ):

2 # Function to drive to the main - line from the Acre without any fruit .

3 global total\_distance , start , end , current\_time

4 distance = abs ( hypotenuse \*( math . sin ( math . radians ( current\_angle ))))

5 print ( distance )

6 turn\_left (1000 , 80) # moving a wrong fruit away so that it cannot be detected again .

7 sleep (1)

8 turn\_right (200 , current\_angle ) # turning the robot perpendicular to the main line after moving a wrong fruit .

9 total\_distance += abs ( hypotenuse \*( math . cos ( math . radians ( current\_angle ))

)) # Adding the distance moved forward to the total for returning to the starting point

10 gy. mode = ' GYRO - RATE '

11 gy. mode = ' GYRO - ANG '

12 sleep (1)

13 move\_distance\_in\_mm (- abs ( distance )) # Reversing the robot to the main - line

# Function for operating the Gripper

**Code 8: Function for operating the Gripper**

== False ):

position\_sp = -210 ,

False

" open " and gripper\_opened

9 gripper\_motor . run\_to\_rel\_pos ( speed\_sp =250 , stop\_action = ' brake ') # opening

10 gripper\_motor . wait\_while (' running ')

11 gripper\_opened = True

closing

= ' brake ') #

6 sleep (2)

7 gripper\_opened =

8 elif ( command ==

1 def gripper ( command ):

2 # Function to open and close the gripper using the Medium Motor

3 global gripper\_opened

4 if ( command == " close " and gripper\_opened == True ):

5 gripper\_motor . run\_to\_rel\_pos ( speed\_sp =150 , position\_sp =230 , stop\_action

# Function for Searching the fruits

**Code 9: Function for Searching the fruits**

10 while ( dist > search\_space or thetha == None ): # Only detect fruits which

are withing the search space

11 while ( dist > search\_space and gy. value () > -85): # It scans turning left 90 degrees

12 motor\_left . run\_forever ( speed\_sp =- scanning\_speed , stop\_action = ' hold ')

1.5 m ( Width of the

the Ultra - Sonic sensor

thetha

1 def scanning ():

2 # Function for scanning the acre for nearby fruits

3 global total\_distance , start , end , current\_time ,

4 gy. mode = ' GYRO - RATE '

5 gy. mode = ' GYRO - ANG '

6 sleep (1)

7 search\_space = 1500 # scanning for fruits withing acre )

8 scanning\_speed = 50

9 dist = us. value () # Storing the distance given by

13 motor\_right . run\_forever ( speed\_sp = scanning\_speed , stop\_action = ' hold ')

14 dist = us. value () # Distance at which the fruit is detected

15 motor\_left . stop ()

16 motor\_right . stop ()

17

18 while ( dist > search\_space and gy. value () <-5): # it scans turning right 90 degrees

19 motor\_left . run\_forever ( speed\_sp = scanning\_speed , stop\_action = ' hold ')

20 motor\_right . run\_forever ( speed\_sp =- scanning\_speed , stop\_action = ' hold ')

21 dist = us. value () # Distance at which the fruit is detected

22 motor\_left . stop ()

23 motor\_right . stop ()

24

25 thetha = gy. value () # Angle at which the fruit is detected

26 dist =us. value () # Distance at which the fruit is detected

27

28 if( dist > search\_space ): # If no fruits detected , advance the robot 20 cm

29 move\_distance\_in\_mm (200)

30 total\_distance += 200 # Adding the distance moved forward to the total for returning to the starting point

31

32 end = time . time ()

33 current\_time += ( end - start ) # Checking the current time

34

35 while ( us. value () < search\_space ): # optimizing the exact angle at which the fruit is detected , turn the robot until the fruit is lost / no\_longer\_detected

36 motor\_left . run\_forever ( speed\_sp = -50 , stop\_action = ' hold ')

37 motor\_right . run\_forever ( speed\_sp =50 , stop\_action = ' hold ')

38 motor\_left . stop ()

39 motor\_right . stop ()

40 alpha = gy. value () - thetha # Angle at which the fruit is lost when turning

41

42 while ( gy. value () < ( thetha + ( alpha /2) )): # Aligning the robot with the center of the fruit

43 motor\_left . run\_forever ( speed\_sp =50 , stop\_action = ' hold ')

44 motor\_right . run\_forever ( speed\_sp = -50 , stop\_action = ' hold ')

45 motor\_left . stop ()

46 motor\_right . stop ()

47

48 if( dist > 500) : # If the fruit is more than 0.5 m away , the robot must stop halfway and scan again .

49 move\_distance\_in\_mm ( dist /2)

50 while ( us. value () < search\_space ):

51 motor\_left . run\_forever ( speed\_sp = -50 , stop\_action = ' hold ')

52 motor\_right . run\_forever ( speed\_sp =50 , stop\_action = ' hold ')

53 motor\_left . stop ()

54 motor\_right . stop ()

55 alpha = gy. value () - thetha

56

57 while ( gy. value () < ( thetha + ( alpha /2) )):

58 motor\_left . run\_forever ( speed\_sp =50 , stop\_action = ' hold ')

59 motor\_right . run\_forever ( speed\_sp = -50 , stop\_action = ' hold ')

60 motor\_left . stop ()

61 motor\_right . stop ()

62 move\_distance\_in\_mm ( dist / 2+10)

63 else :

64 move\_distance\_in\_mm ( dist +10) # Moving forward 1 cm more to force the fruit to be below the colour sensor

65

66 sleep (1)

67 gripper (' close ') # Closing the gripper to allow colour sensing function

68

69 if ( cl. get Calibrated Color String () == fruit ): # if the sensored fruit is correct

70 ev3 . Sound . speak (" Right ")

71 drive\_to\_home\_zone ( dist , abs ( gy. value ()), 300) # Drive it to the Home - zone

72 else :

73 ev3 . Sound . speak (" wrong ") # if the sensored fruit is wrong

74 gripper (' open ') # open the gripper

75 drive\_from\_acre\_to\_line ( dist , abs ( gy. value ())) # move the fruit away

76 turn\_right (200 ,85)

77 end = time . time ()

78 current\_time += ( end - start ) # Checking the current time

# Main Code

**Code 10: Main Code**

1 fruit\_dict ={ # Declaring the available fruits

2 # Dictionary of all possible fruits ( Coloured cubes ) in the competition and corresponding selection button on the EV3 brick

3 ' up ':" Red ",

4 " right ":" Green ",

5 " left ":" Blue ",

6 " down ":" Yellow ",

7 }

8

9 fruit = menu () # Storing the desired fruit

10

11 while ( harvested <2 and current\_time <300) : # when it reaches 300 s (5 minutes ) or when it harvested 2 fruits , it must come back to the starting point

12 scanning ()

13 if( total\_distance >= 4000) : # If the robot drove the whole length of the acre , it must return to the starting point and scan again

14 move\_distance\_in\_mm (- total\_distance )

15 total\_distance = 0

16

17 move\_distance\_in\_mm (- total\_distance ) # Reversing the robot back to the starting Position