Hochschule Reutlingen
Reutlingen University

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 - 22 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]

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-university.de/fileadmin/user_upload/2022_05_23_StuPro_AllgemTeil_WiSe_22-23_2022-05-11.pdf

Executive Summary

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.

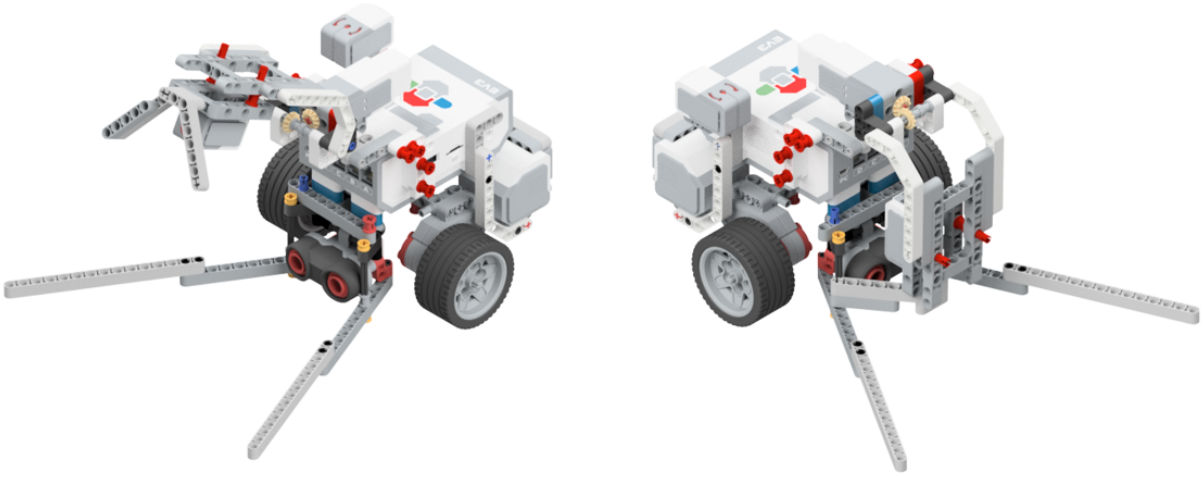


Figure 1: Fruta Oes

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 on page 2, followed in section 3 on page 6 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 on page 12 highlights the Agile Software engineering (SCRUM) procedure followed in this project. The SCRUM section will briefly discuss the project setup in section 4.1 on page 12. Then section 4.2, section 4.4,section 4.5 present the role and progress achieved in sprints 1, 2, and 3 respectively. Finally, discussions and reflections will be detailed in section 5 on page 18

Table of Contents

List of Figures

List of Tables

1 Introduction

Agricultural intensification, mechanization, and automation have all led to major increases in agricultural productivity throughout time [2]. 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]. 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 on page ii 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 on the following page 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.

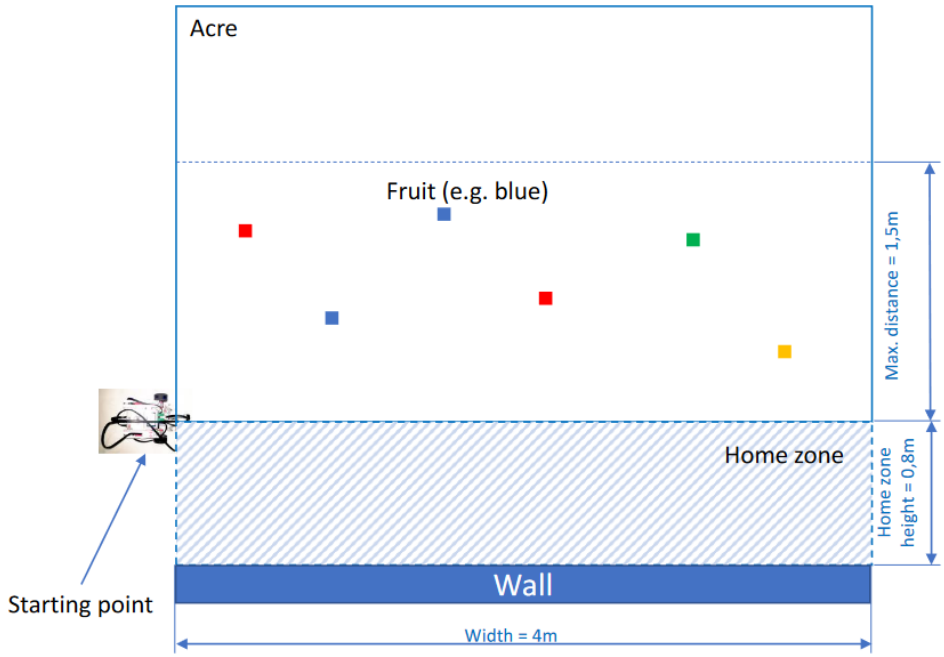


Figure 2: Field [1]

2 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 on the next page depicts the five main components of **Fruta Oes**.
The five main components of this robot are listed in table 1. **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 on page 5.

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 on the following page. **Item 4-Gripper and Colour Sensing Module** grips the object and makes sure it is in a good position for the colour sensor.

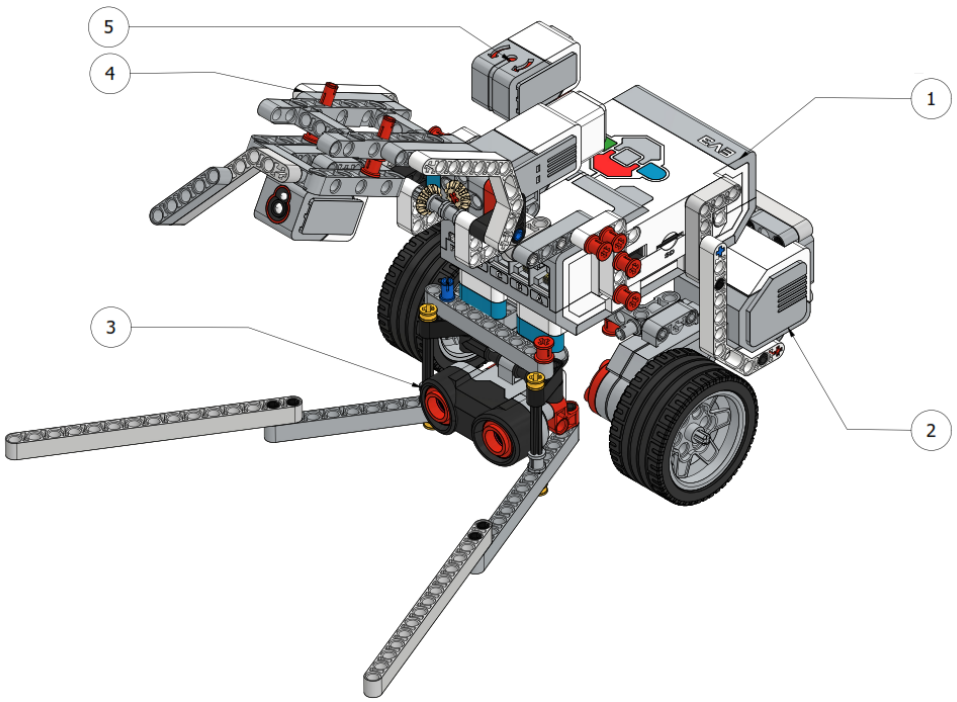


Figure 3: Main Components of Fruta Oes

2.1 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.

Table 2: Object Detection and Manipulator Module Components

Item	Component
1	Ultra Sonic Sensor
2	Manipulator Fangs

Figure 4 on the next page shows the object detecting manipulator, it is composed of two main parts listed in table 2: 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 $\pm$ 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.

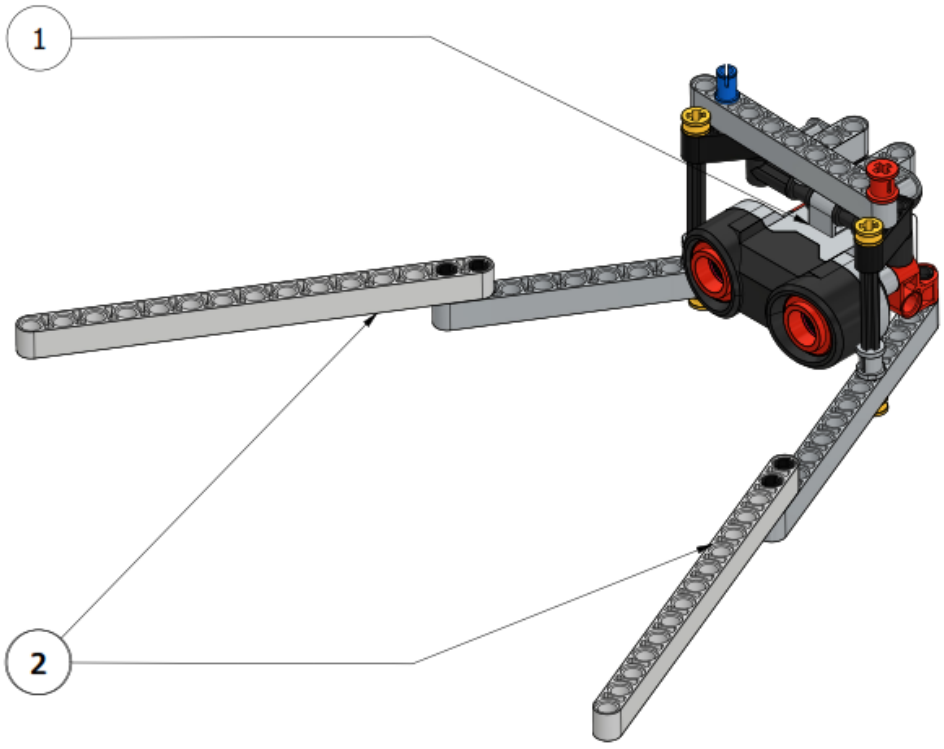


Figure 4: Object Detection and Manipulator Module

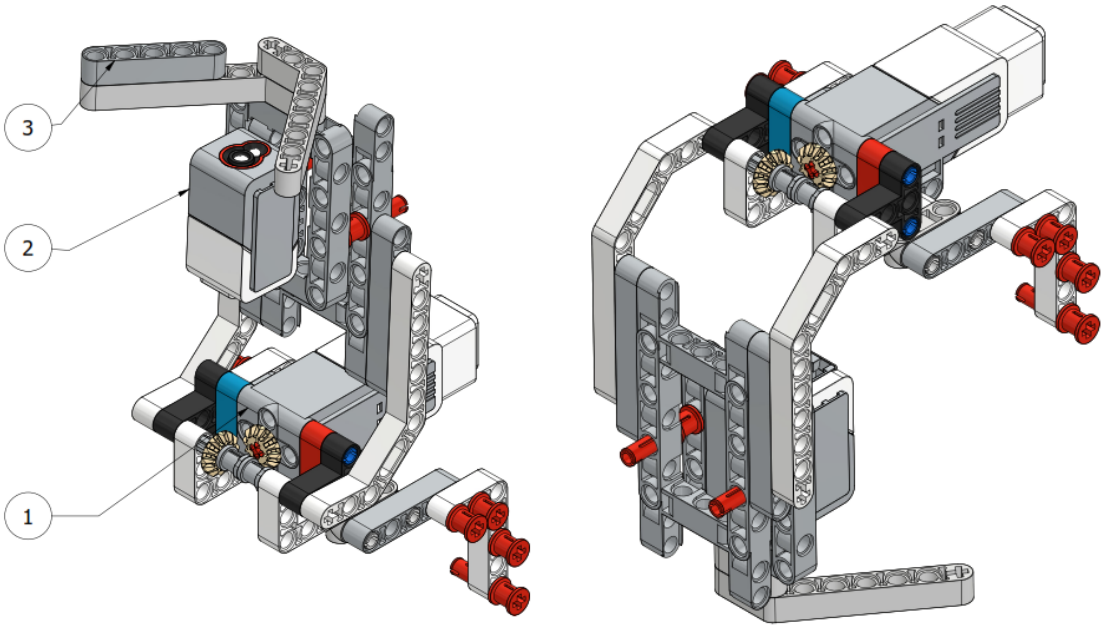
2.2 Gripper and Colour Sensing Module

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 on the previous page.

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 on the following page depicts te gripper and the colour sensing module. figure 5a on the next page shows the gripper in opened mode and figure 5b on the following page 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.



(a) Opened Gripper (b) Closed Gripper

Figure 5: Gripper and Colour Sensing Module

2.3 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. there is also a balancing wheel positioned underneath the base close to the rear but this wheel is not shown in figure 6 on the following page.

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. The intelligent motor may be coordinated with other robot motors to move in a straight line at the same speed by using the built-in rotation sensor. This motor has an angular velocity (ω) of 160 - 170 rpm, and can produce a running torque of 20 $N.cm$ and a stall torque of 40 $N.cm$ [2]. In the design of this drive base, each EV3 large motor is coupled with a 56 $milimeter$ diameter wheel. The dimensions of these wheels will become handy later when developing an algorithm to move the robot from one point to another for a specified distance. Appendix E shows a python function where a circumference of the wheels is used to move the robot forward and backwards, this function will be explain in depth in section 3 on the next page.

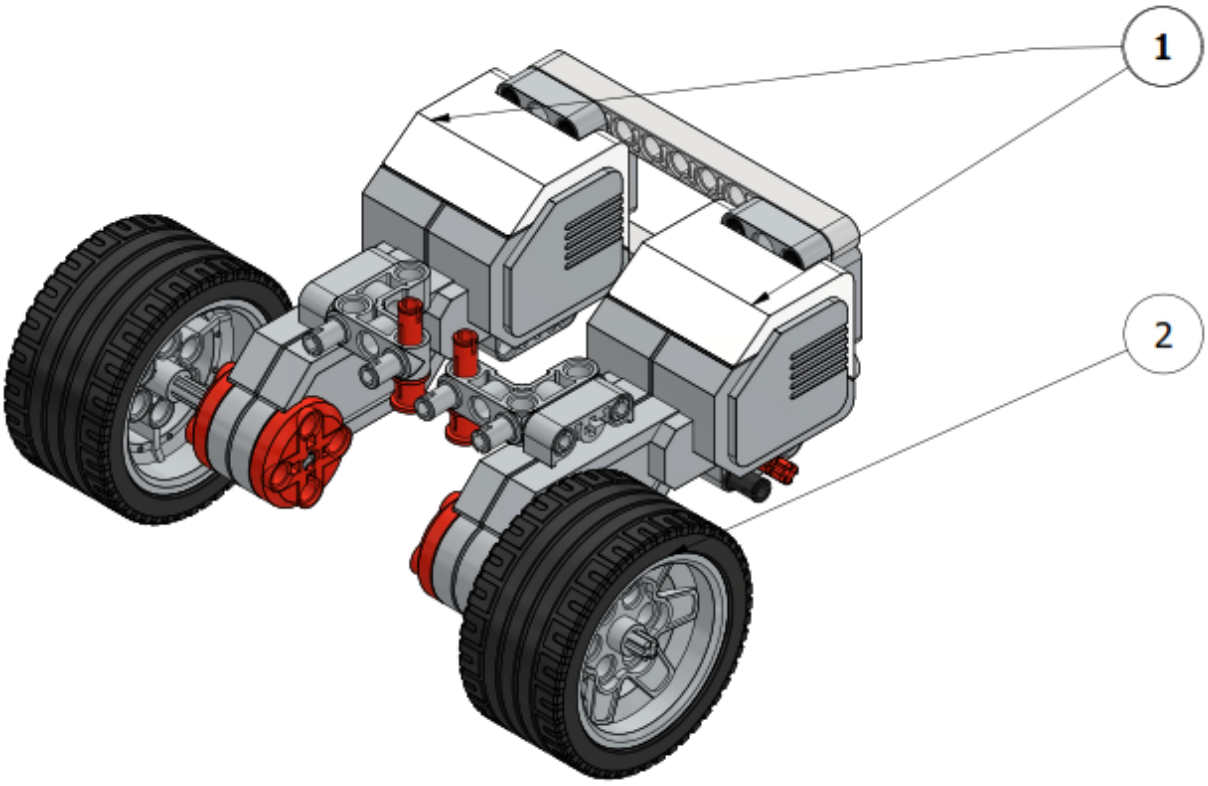


Figure 6: Drive Base

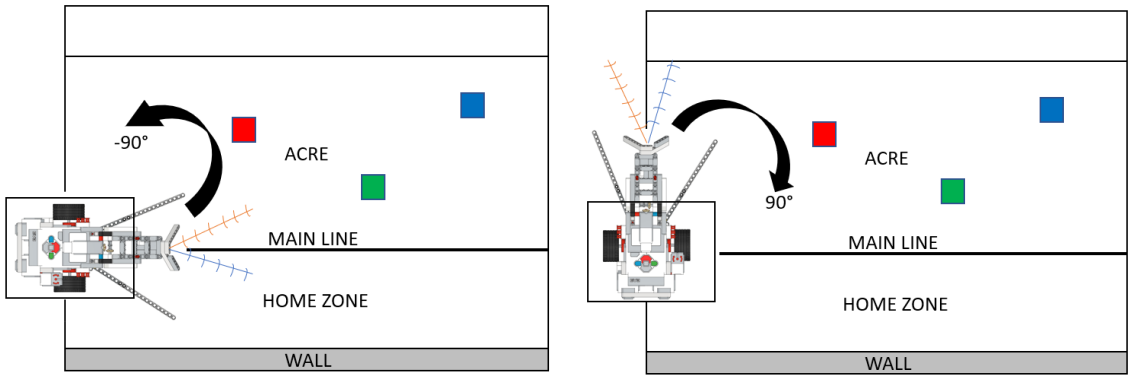
3 Software Design

3.1 Task Solution

A solution to harvesting specific fruit uses mainly the ultrasonic sensor for searching for objects in an area. The solution that will be presented in this software design requires one to understand terms used for sectioning the field. Figure 7 on the following page shows the field, the field consists of a part called **Acre**, this part will have a mixture of fruits, required fruits and the non-required fruits. There is a **Main Line**, this is the line that will be used to track the movement of the robot, the robot will have to return to this line after every task it completes. Then there is **Home-Zone** where required fruits will be delivered to from the **Acre**.

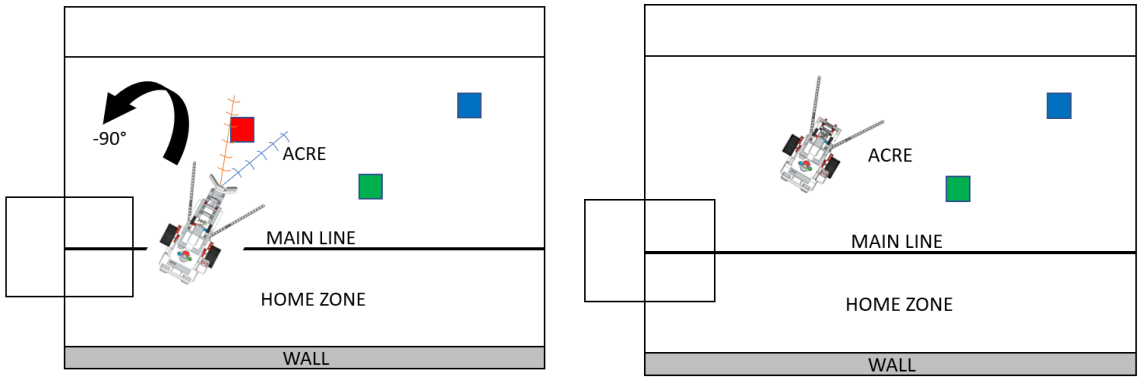
Figure 7a on the next page shows the robot in a starting position, the blue and orange lines from the robot illustrates the ultrasonic sound, the distance that is searched is also determined. The robot must turn left 90 degrees at a slow speed and determine is there are objects in the specified range (the robot pose after this turn is shown in Figure 7b on the following page), then for assurance, the robot searched again by turning right 90 degrees and pose parallel to the main line.
After a complete scan, the robot advances by a short distance, and do the similar search until it detects an object after which the object is approached (figure 8b on the next page). The robot places the approached fruit under the colour sensor to determine if it is the desired fruit or not. If the fruit

is not desired, the robot then pushes the fruit out of the way and then return to the main line to search again as shown in figure 9a. When a desired fruit is found (in this example a green object in figure 9b), the robot pushes the fruit to the home zone and return to the main line until two fruits are harvested.



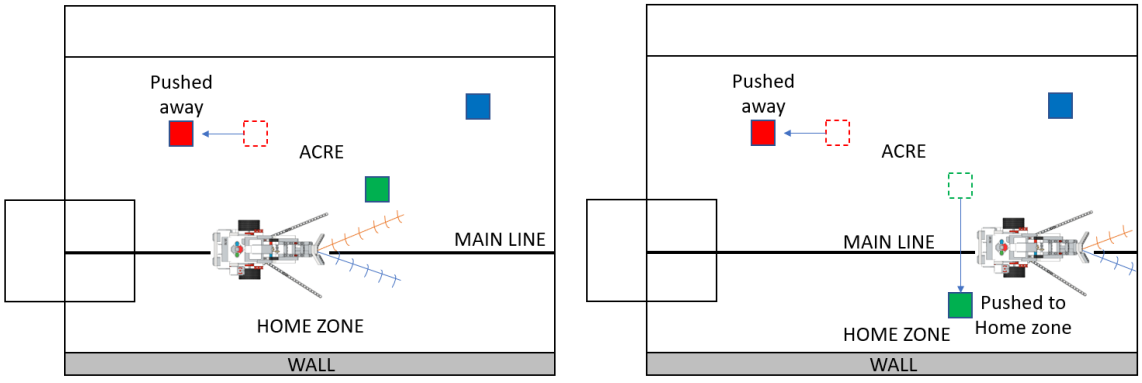
(a) Robot Starting Position (b) Position after 1 scan

Figure 7: Fruit Searching



(a) Object Detected (b) Object Approached

Figure 8: Fruit Searching in new position



(a) Object Detected (b) Object Approached

Figure 9: Undesired and desired fruit

3.2 Main Code

This section discusses the execution of the main function. Figure 10 presents the activity diagram for this main function. Firstly, Python need libraries are imported (e.g EV3 library, Math library, EV3 brick buttons library, Time library). This function deploys the robot to harvest fruits and only returns to the starting position if one of the two conditions is met: 1. if the robot harvested two fruits it must then return. 2. if a five minute times out while searching, the robot must return home. A safety condition to make sure the robot does not exceed the limits of the field was implemented. If the robot moved a distance of four minutes along the main line without finding fruits, it is then sent back to the starting position to do the search once again.

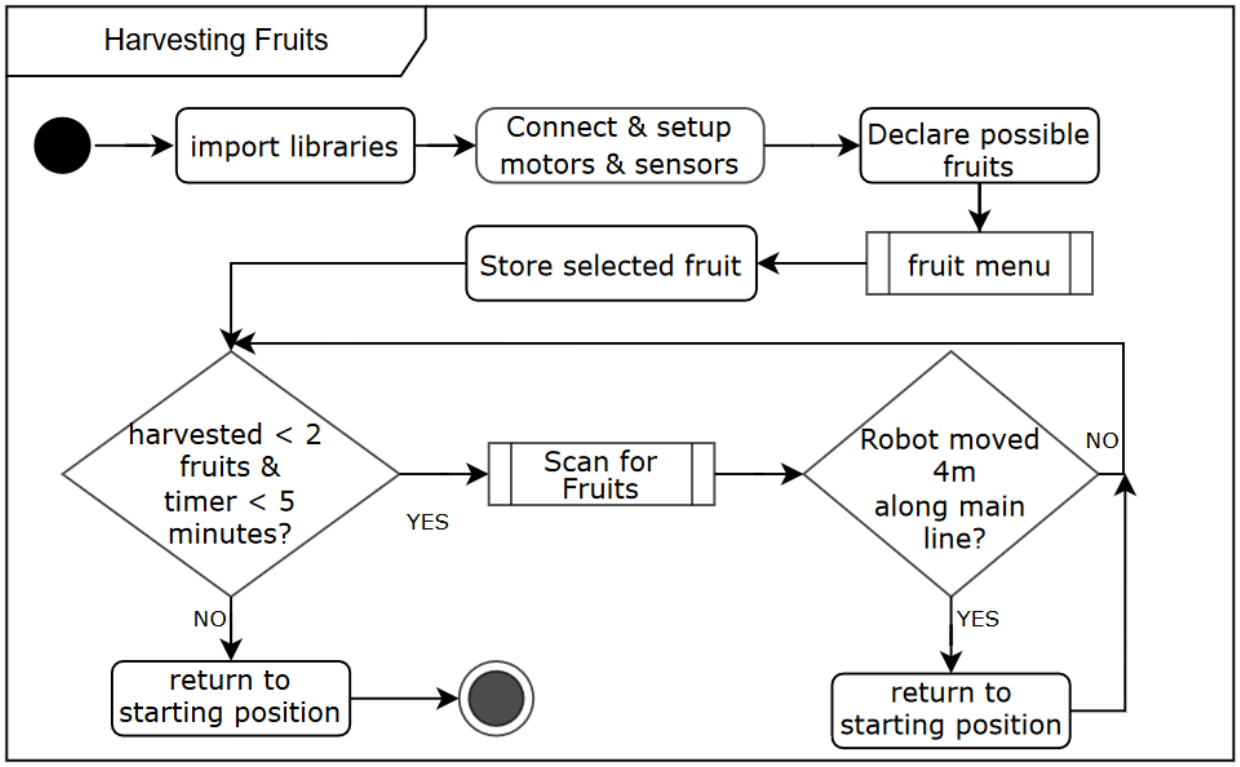


Figure 10: Main Code Activity Diagram

Algorithm 1 : Main Function Pseudo Code

```
Import libraries and connect devices;
Create a fruit dictionary with 4 fruits and 4 buttons;
Call fruit menu() function and store the selected fruit;
while harvested < 2 fruits and timer < 5 minutes do
    Call scanning() function;
    if Robot reached the end of the main line (4m)? then
        Return to the starting position;
    end if
end while
Return to the starting position;
```

3.3 User Menu for Selecting the Colour (fruit)

As it can be seen in figure 1 on page ii, 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 11 presents the activity diagram for User Menu function, the corresponding source code is attached on appendix B.

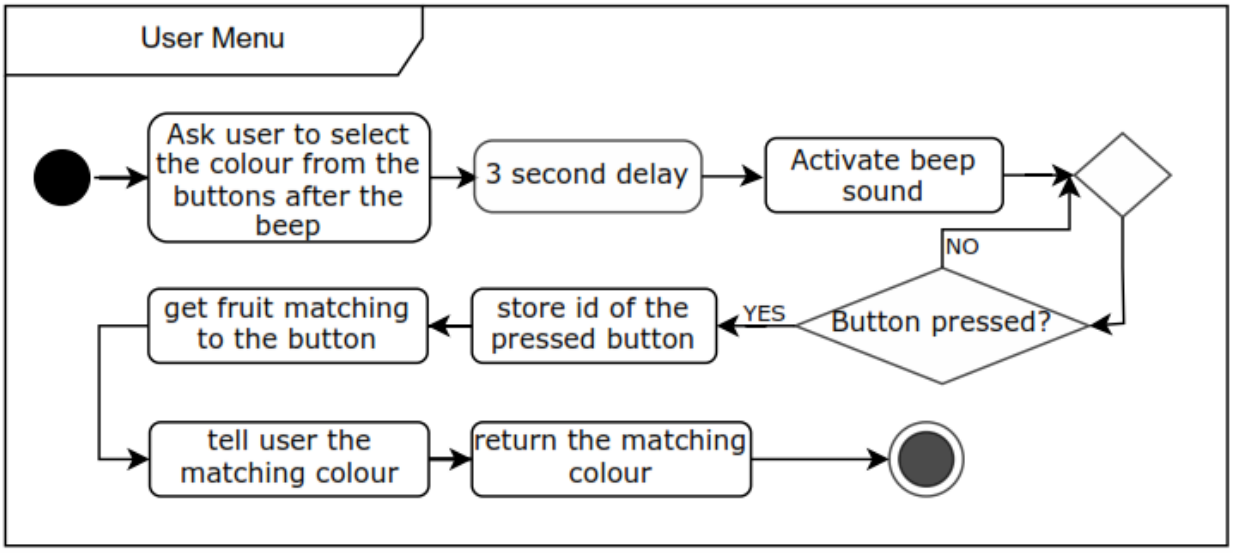


Figure 11: Menu Code Activity Diagram

Algorithm 2 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.

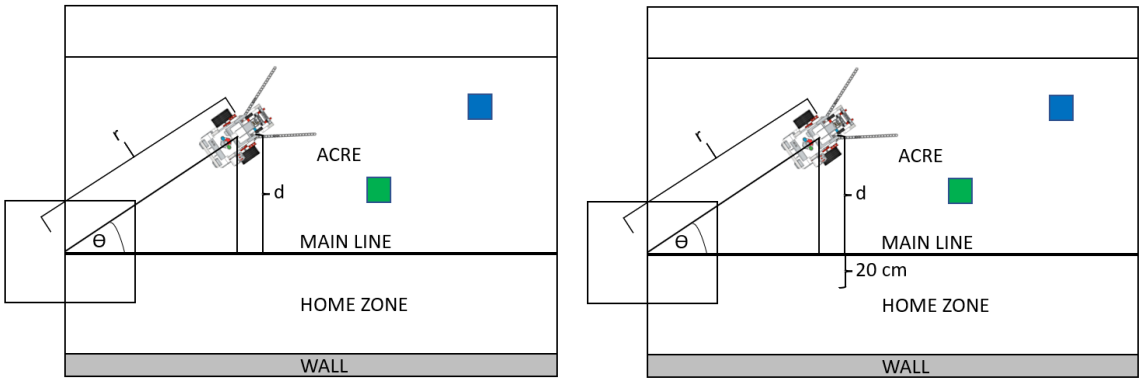
Algorithm 2 : User Menu Method Pseudo Code

```
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;
```

3.4 Driving from Acre to Main line and Home Zone

The position and orientation of the robot needed to be tracked to be sure that it drives to correct areas of the field. This was obtained by using trigonometry functions (equation (1)). Figure 12 shows the graphical analysis. The angle at which the object is detected is saved, also the distance the robot travelled to the object(hypotenuse). What is left is to find the opposite distance(d) of the triangle. For driving to the main line, the robot must drive the distance (d) (figure 12a) and to drive to the home zone, 20 cm are added to this distance (figure 12b).

$$opposite(d) = hypotenuse(r) \times \sin\theta \tag{1}$$

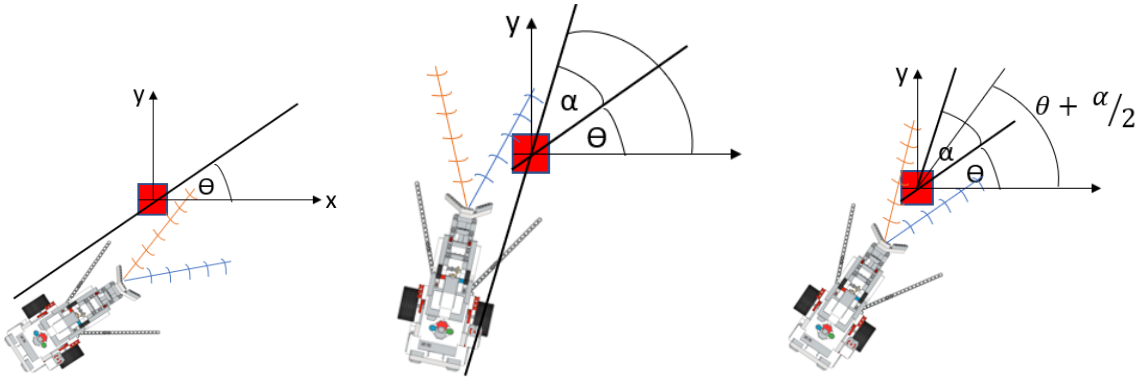


(a) Driving from Acre to Main line (b) Driving from Acre to home zone

Figure 12: Driving from Acre to Main line and Home zone

3.5 Scanning Function and Ultra-Sonic Sensor Optimization

The activity diagram for the scanning function is shown in figure 14 on the following page. This function makes use of the following algorithm to pin point the exact point of the object. The robot finds the edges of the object (figure 13a and figure 13b) and calculates the centre angle as shown in figure 13c.



(a) first Edge of object (b) Second Edge of object (c) Optimized angle

Figure 13: Ultra-Sonic Sensor Optimization

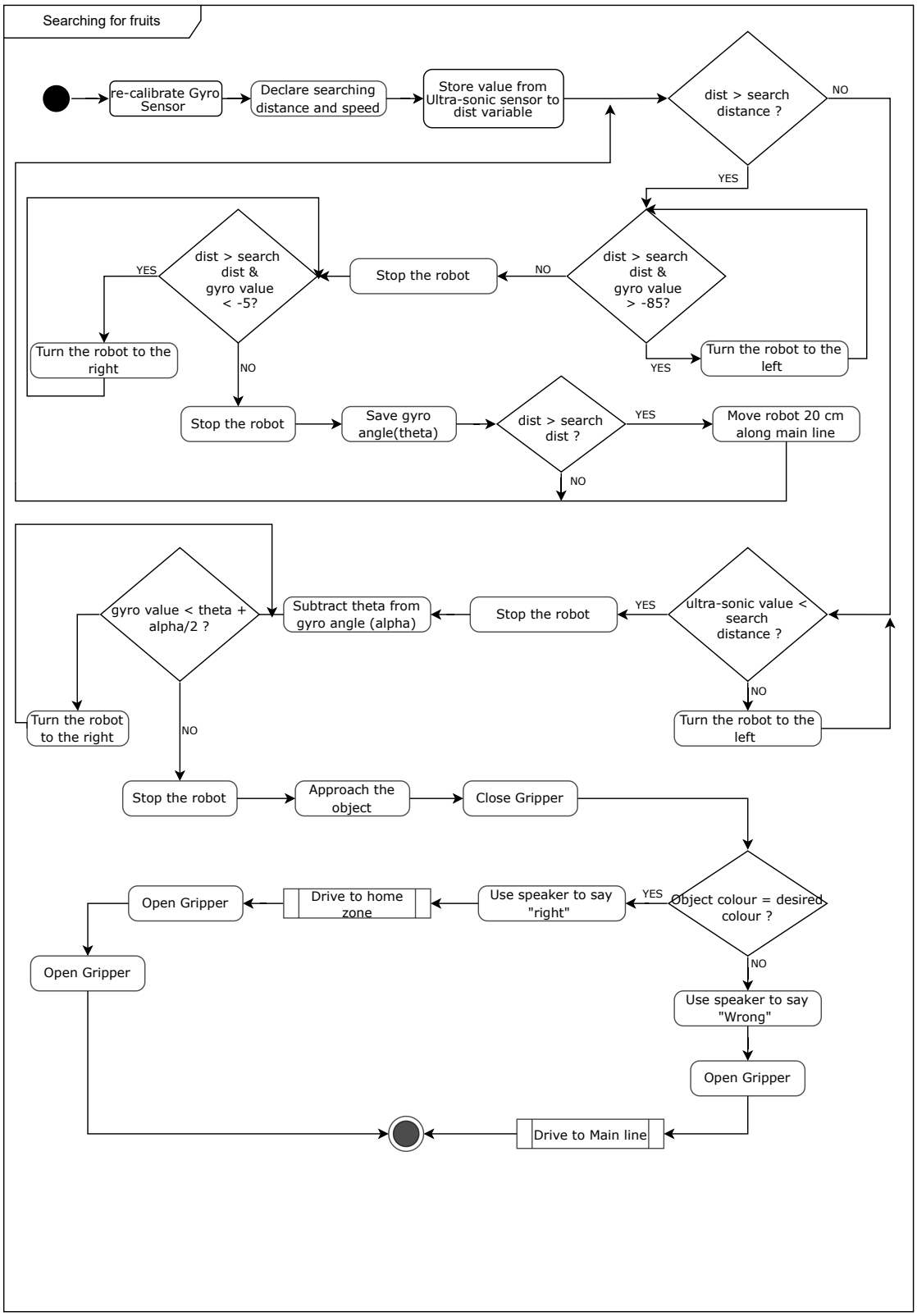


Figure 14: Scanning Code Activity Diagram.

4 Analysis: Plan how to tackle the problem

4.1 Project Setup

This section presents the project setup. The project vision is pitched in this section, the Scrum team is introduced, and the product backlog is presented.

4.1.1 Project Vision

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

4.1.2 SCRUM Team

Figure 15 presents the scrum team: Encarnacion Nunez-Ortega is the **Product Owner** and responsible 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 15: SCRUM Team

4.1.3 Product Backlog

Table 5 Present the Product backlog for the project. This table comprises ordered list of user stories that have been derived from the requirements. In other words, all the features of the product that is designed in this project are listed in this table. The first column shows the priority of the user stories from 1-10, priority 1 means it is the most important task.

Table 5: Product Backlog

Priority	User Story	Effort
1	As a user, I want the robot to be completely autonomous.	
2	As a user, I want the robot to ask me for the correct fruit and to confirm that it understood me.	
3	As a developer, I want the robot to be able to detect a 2.5cm cube.	
4	As a user, I want the robot to look for the specified fruit and check if the fruit is correctly grabbed.	
5	As a user, I don't want the robot to return to the home zone without the fruit grabbed.	
6	As a user, I want the robot to place the right fruit in the home zone.	
7	As a developer, I want the robot to keep count of the harvested fruits	
8	As a user, I want the robot to immediately return to the starting point after collecting two fruits within 5 minutes.	
9	As a user, I don't want the robot to search for fruits outside the acre.	
10	As a developer, I want the code to be efficient.	

4.2 Sprint 1

4.2.1 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.

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

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

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

4.2.2 Burn-down Chart

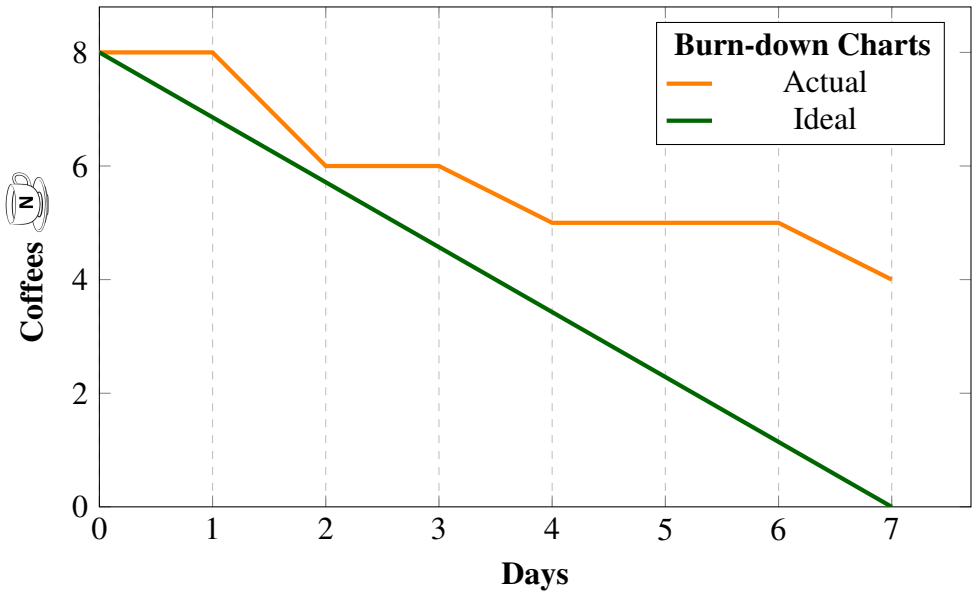


Figure 16: Sprint 1 Burn-down Chart

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.

4.3 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 understood me.*

We initially attempted to identify the colour of the fruit using a colour sensor and showing directly the cube, but it proved to be unreliable as it also picked up the colour 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 colour.

4.4 Sprint 2

4.4.1 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.

As a **user**, I want the robot to place the **right fruit** in the home zone. (3)	As a **user**, I don't want the robot to **return** to the home zone **without the fruit** grabbed. (4)	As a **user**, I want the robot to look for the **specified fruit** and check if the fruit is correctly grabbed. (5)	As a **user**, I want the robot to be completely **autonomous**. (1)

4.4.2 Burn-down Chart

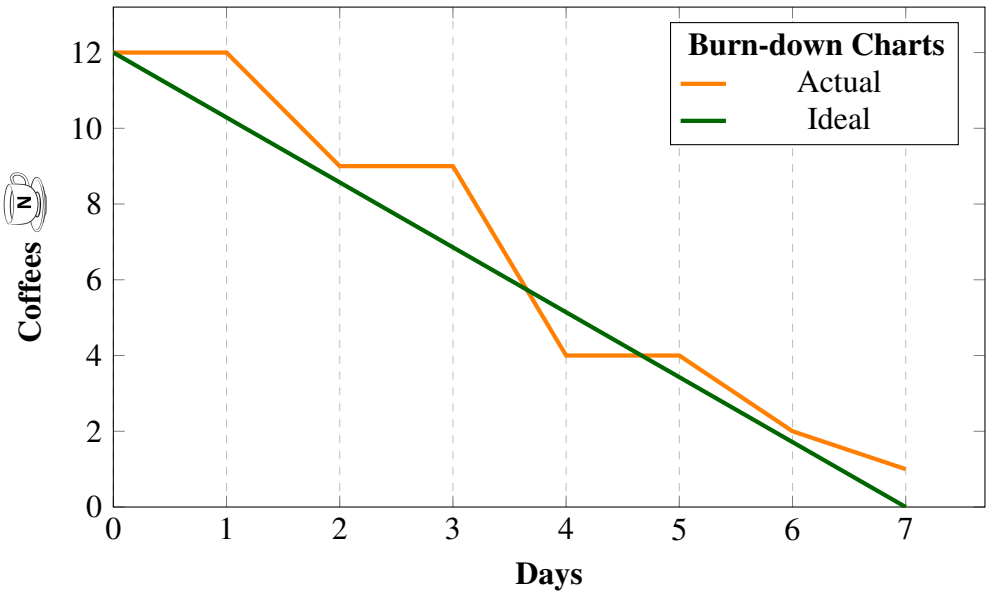


Figure 17: Sprint 2 Burn-down Chart

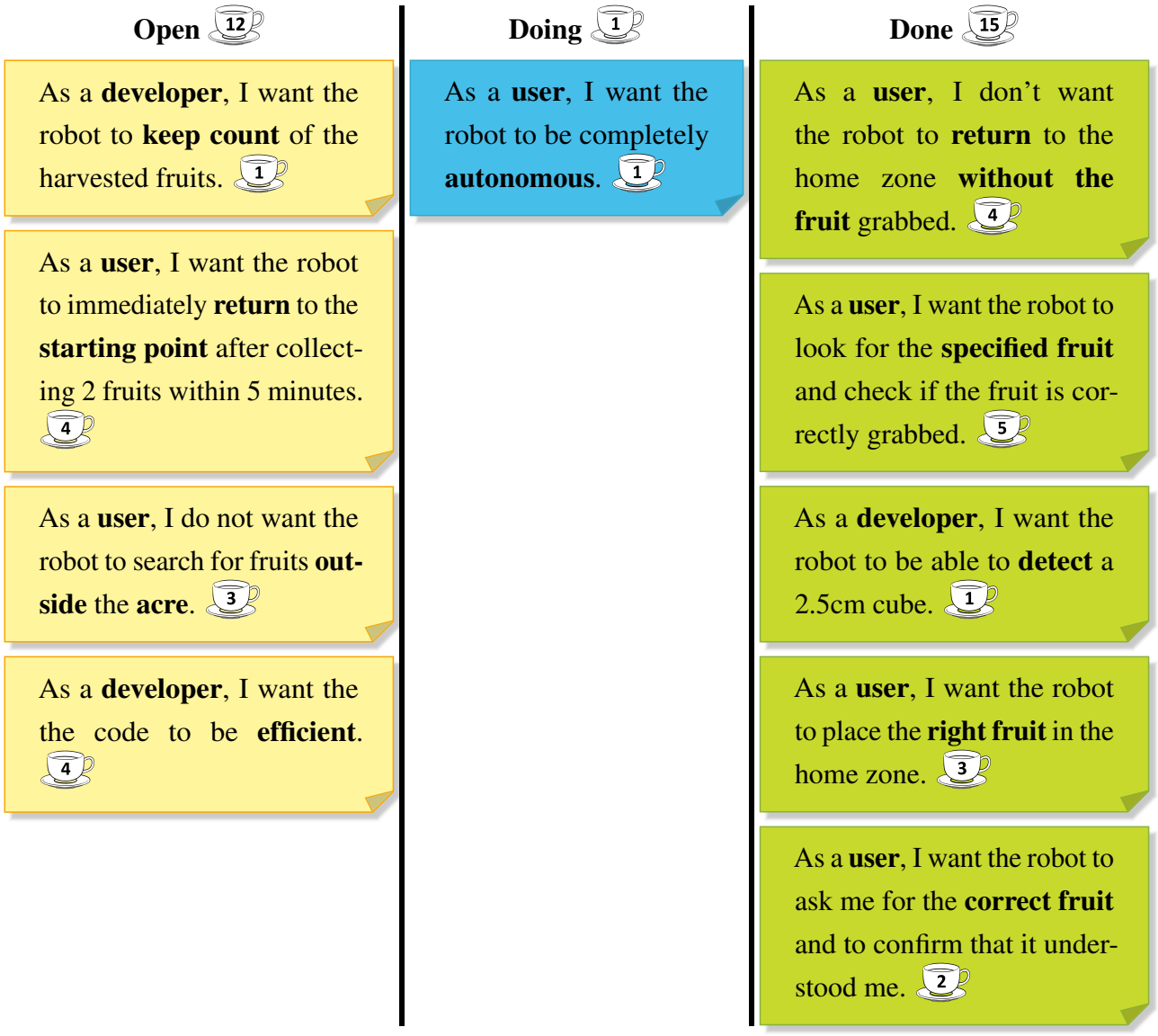
The increment of this Sprint was 11 instead of 12 coffees in view of the fact that we underestimated the tasks. We have an incomplete user story which states: As a user, I want the robot to be completely autonomous. This user story was deemed incomplete as the robot's brick did not detect the program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

4.4.3 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 section 3 on page 6). Through testing with various positions of the fruit, we determined that this approach was the most effective.

4.4.4 Product Backlog



4.5 Sprint 3

4.5.1 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.

As a **developer**, I want the robot to **keep count** of the harvested fruits. 1

As a **user**, I want the robot to immediately **return** to the **starting point** after collecting 2 fruits within 5 minutes. 4

As a **user**, I do not want the robot to search for fruits **outside** the **acre**. 5

As a **developer**, I want the the code to be **efficient**. 4

4.5.2 Burn-down Chart

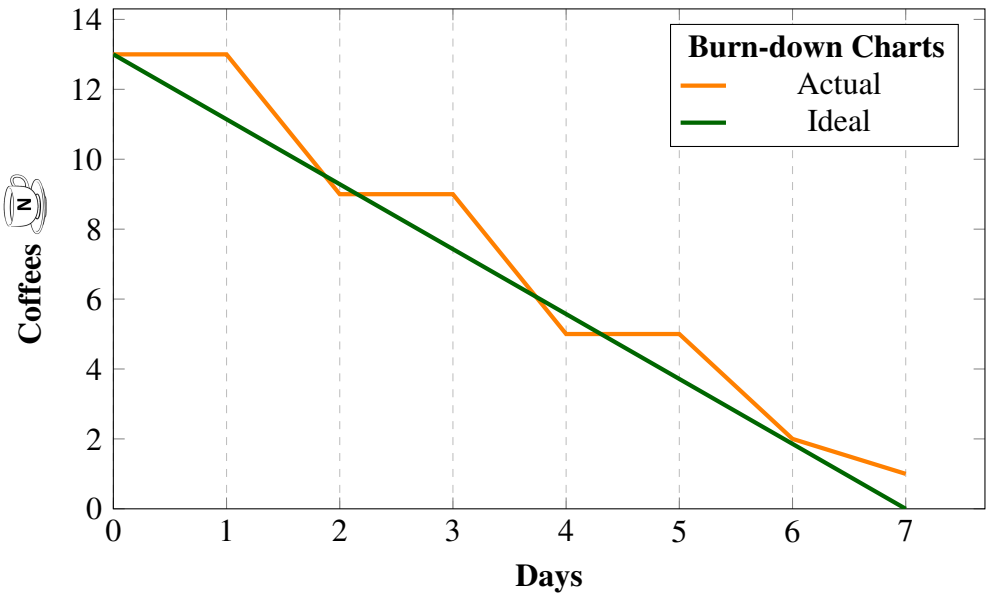


Figure 18: Sprint 3 Burn-down Chart

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.

4.5.3 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 Discussion and reflection

5.1 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.

5.2 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] 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 Visio 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 Python). No work done by other persons is presented as that of the report author.

Appendices

A Library importing and Device Setup

Code 1: Library importing and Device Setup

```
#!/usr/bin/env python3
import ev3dev.ev3 as ev3
from ev3dev.ev3 import * #to allow the use of brick buttons
import math #importing Math Library to allow usage of Mathematical
    functions
import ColorSensor2 #importing Color sensor library
from time import sleep #importing time library
import time
#-----------------------------------------------------------------
#Device Connections start
# Connect the outputs to the motor
motor_left = ev3.LargeMotor('outA')
motor_right = ev3.LargeMotor('outD')
gripper_motor = ev3.MediumMotor('outC')

#Gyro Sensor initialisation
gy = ev3.GyroSensor()

#Colour Sensor initialisation
cl= ColorSensor2.ColorSensor2()
cl.mode='COL-COLOR'

#Ultra-Sonic Sensor initialisation
us = ev3.UltrasonicSensor()
us.mode='US-DIST-CM'

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

#Global Variables (we know use of global variables is not a good
    practice, but in this case it is necessary)
harvested = 0  #keeping count of the harvested fruits
total_distance = 0  #keeping track of the distance travelled by the
    robot
gripper_opened = True   #keeping track of the state of the gripper
thetha = None   #To avoid the scanning angle from exceeding the
    limits
start = 0   #Variable to start the 5 minute timer
end = 0 #Variable to end the 5 minute timer
current_time = 0    #Variable to track the current time
#Device Connections ends
```

B User Menu for Selecting the Colour

Code 2: User Menu for Selecting the Colour

```
def menu():
#Function to select the desired fruit using the buttons from the Brick
global start
ev3.Sound.speak("Select the colour from the buttons after the beep").
    wait()
sleep(3)
Sound.beep().wait()
start = time.time() #Starting the timer
while True:
if btn.any():# Checks if any button is pressed.
button_id = btn.buttons_pressed
print(button_id[0])
break
else:
sleep(0.01)

option_text = " I will search for {0}" #Robot confirming the colour
ev3.Sound.speak(option_text.format(fruit_dict.get(button_id[0]))).wait
    ()
return fruit_dict.get(button_id[0]) #Returning the desired fruit
```

C Function for turning left

Code 3: Function for turning left

```
def turn_left(speed,angle):
#Function that turns left, it requires the turning speed and the angle
    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 = 'hold')
motor_right.run_forever(speed_sp=speed, stop_action = 'hold')
#print(gy.value())
motor_left.stop()
motor_right.stop()
```

D Function for turning right

Code 4: Function for turning right

```
def turn_right(speed,angle):
#Function that turns right, it requires the turning speed and the angle 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 = 'hold')
motor_right.run_forever(speed_sp=-speed, stop_action = 'hold')
#print(gy.value())
motor_left.stop()
motor_right.stop()
```

E Function for moving the robot forward

Code 5: Function for moving the robot forward

```
def move_distance_in_mm(distance_mm):
#Function to move a specified distance, it transforms rotation of the wheels to distance.
dist = (distance_mm/1000) * (360/0.176)
motor_left.run_to_rel_pos(position_sp = dist, speed_sp=400, stop_action = "brake")
motor_right.run_to_rel_pos(position_sp = dist, speed_sp=400, stop_action = "brake")
motor_left.wait_while('running')
motor_right.wait_while('running')
motor_left.stop()
motor_right.stop()
```

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

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

```
def drive_to_home_zone(hypotenuse, current_angle, add):
#Function to drive to the home zone from the Acre with a correct fruit.
global total_distance, harvested, start, end, current_time
distance = abs(hypotenuse*(math.sin(math.radians(current_angle)))) #
    Using Pythagorean theorem to calculate the distance from the fruit
    to the home-zone
total_distance += abs(hypotenuse*(math.cos(math.radians(current_angle))
    )) #Adding the distance moved forward to the total for returning to
    the starting point
turn_right(100,current_angle + 85) #turning right 90 degrees (we used
    85 because the Gyro-sensor was not accurate)
gy.mode = 'GYRO-RATE'
gy.mode = 'GYRO-ANG'
sleep(1)
move_distance_in_mm(abs(distance + add))
gripper("open") #Open the gripper to release the fruit in the home-zone
harvested += 1  #keeping count of the harvested fruits
move_distance_in_mm(-add) #returning to the main-line
turn_left(200,85) #Turning left to continue searching
end = time.time()
current_time += (end - start) #Checking the current time
```

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

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

```
def drive_from_acre_to_line(hypotenuse, current_angle):
#Function to drive to the main-line from the Acre without any fruit.
global total_distance, start, end, current_time
distance = abs(hypotenuse*(math.sin(math.radians(current_angle))))
print(distance)
turn_left(1000, 80) #moving a wrong fruit away so that it cannot be
    detected again.
sleep(1)
turn_right(200,current_angle) #turning the robot perpendicular to the
    main line after moving a wrong fruit.
total_distance += abs(hypotenuse*(math.cos(math.radians(current_angle))
    )) #Adding the distance moved forward to the total for returning to
    the starting point
gy.mode = 'GYRO-RATE'
```

```
gy.mode = 'GYRO-ANG'
sleep(1)
move_distance_in_mm(-abs(distance)) #Reversing the robot to the main-
    line
```

H Function for operating the Gripper

Code 8: Function for operating the Gripper

```
def gripper(command):
#Function to open and close the gripper using the Medium Motor
global gripper_opened
if (command == "close" and gripper_opened == True):
gripper_motor.run_to_rel_pos(speed_sp=150, position_sp=230, stop_action
    = 'brake') # closing
sleep(2)
gripper_opened = False
elif (command == "open" and gripper_opened == False):
gripper_motor.run_to_rel_pos(speed_sp=250, position_sp=-210,
    stop_action = 'brake') #opening
gripper_motor.wait_while('running')
gripper_opened = True
```

I Function for Searching the fruits

Code 9: Function for Searching the fruits

```
def scanning():
#Function for scanning the acre for nearby fruits
global total_distance, start, end, current_time, thetha
gy.mode = 'GYRO-RATE'
gy.mode = 'GYRO-ANG'
sleep(1)
search_space = 1500 #scanning for fruits withing 1.5m (Width of the
    acre)
scanning_speed = 50
dist = us.value() #Storing the distance given by the Ultra-Sonic sensor
while(dist > search_space or thetha == None): #Only detect fruits which
    are withing the search space
while(dist > search_space and gy.value()>-85): #It scans turning left
    90 degrees
motor_left.run_forever(speed_sp=-scanning_speed, stop_action = 'hold')
```

```
motor_right.run_forever(speed_sp=scanning_speed, stop_action = 'hold')
dist = us.value() #Distance at which the fruit is detected
motor_left.stop()
motor_right.stop()

while(dist > search_space and gy.value()<-5): #it scans turning right 90 degrees
motor_left.run_forever(speed_sp=scanning_speed, stop_action = 'hold')
motor_right.run_forever(speed_sp=-scanning_speed, stop_action = 'hold')
dist = us.value() #Distance at which the fruit is detected
motor_left.stop()
motor_right.stop()

thetha = gy.value() #Angle at which the fruit is detected
dist =us.value() #Distance at which the fruit is detected

if(dist > search_space): # If no fruits detected, advance the robot 20 cm
move_distance_in_mm(200)
total_distance += 200 #Adding the distance moved forward to the total for returning to the starting point

end = time.time()
current_time += (end - start) #Checking the current time

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
motor_left.run_forever(speed_sp=-50, stop_action = 'hold')
motor_right.run_forever(speed_sp=50, stop_action = 'hold')
motor_left.stop()
motor_right.stop()
alpha = gy.value() - thetha #Angle at which the fruit is lost when turning

while (gy.value()< (thetha + (alpha/2))): #Aligning the robot with the center of the fruit
motor_left.run_forever(speed_sp=50, stop_action = 'hold')
motor_right.run_forever(speed_sp=-50, stop_action = 'hold')
motor_left.stop()
motor_right.stop()

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

```
move_distance_in_mm(dist/2)
while(us.value() < search_space):
motor_left.run_forever(speed_sp=-50, stop_action = 'hold')
motor_right.run_forever(speed_sp=50, stop_action = 'hold')
motor_left.stop()
motor_right.stop()
alpha = gy.value() - thetha

while (gy.value()< (thetha + (alpha/2))):
motor_left.run_forever(speed_sp=50, stop_action = 'hold')
motor_right.run_forever(speed_sp=-50, stop_action = 'hold')
motor_left.stop()
motor_right.stop()
move_distance_in_mm(dist/2+10)
else:
move_distance_in_mm(dist+10) #Moving forward 1cm more to force the
    fruit to be below the colour sensor

sleep(1)
gripper('close') #Closing the gripper to allow colour sensing function

if (cl.getCalibratedColorString() == fruit): #if the sensored fruit is
    correct
ev3.Sound.speak("Right")
drive_to_home_zone(dist, abs(gy.value()), 300) #Drive it to the Home-
    zone
else:
ev3.Sound.speak("wrong") #if the sensored fruit is wrong
gripper('open') #open the gripper
drive_from_acre_to_line(dist, abs(gy.value())) #move the fruit away
turn_right(200,85)
end = time.time()
current_time += (end - start) #Checking the current time
```

J Main Code

Code 10: Main Code

```
fruit_dict ={   #Declaring the available fruits
  #Dictionary of all possible fruits (Coloured cubes) in the
     competition and corresponding selection button on the EV3 brick
  'up':"Red",
  "right":"Green",
  "left":"Blue",
```

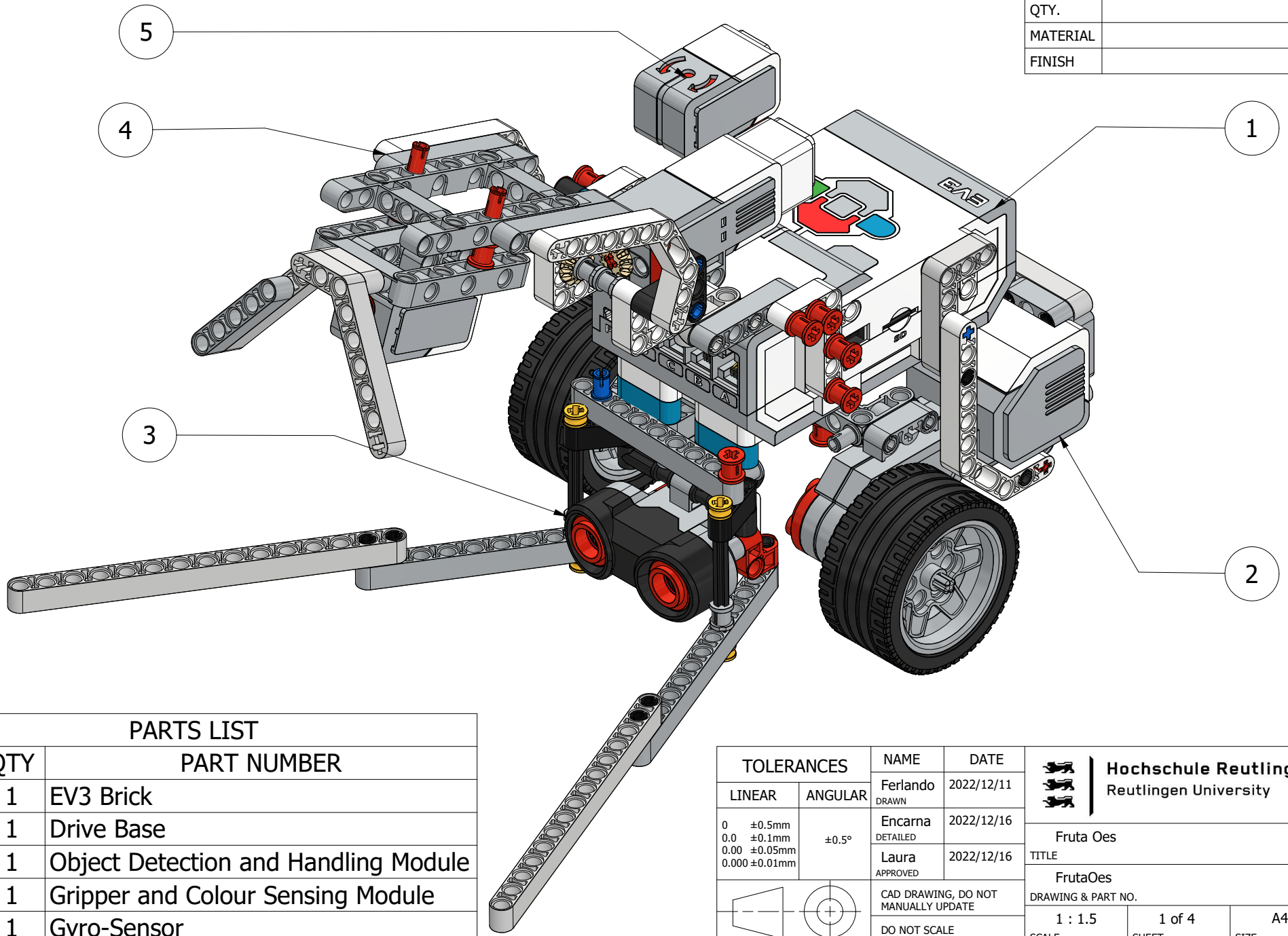
```
  "down":"Yellow",
}

fruit = menu() #Storing the desired fruit

while (harvested<2  and current_time<300): #when it reaches 300s (5
    minutes) or when it harvested 2 fruits, it must come back to the
    starting point
scanning()
if(total_distance >= 4000): #If the robot drove the whole length of the
     acre, it must return to the starting point and scan again
move_distance_in_mm(-total_distance)
total_distance = 0

move_distance_in_mm(-total_distance) #Reversing the robot back to the
    starting Position
```

ITEM NO.	
QTY.	
MATERIAL	
FINISH	



PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	EV3 Brick
2	1	Drive Base
3	1	Object Detection and Handling Module
4	1	Gripper and Colour Sensing Module
5	1	Gyro-Sensor

TOLERANCES		NAME	DATE
LINEAR	ANGULAR	Ferlando DRAWN	2022/12/11
0 ±0.5mm 0.0 ±0.1mm 0.00 ±0.05mm 0.000 ±0.01mm	±0.5°	Encarna DETAILED	2022/12/16
		Laura APPROVED	2022/12/16
		CAD DRAWING, DO NOT MANUALLY UPDATE	
		DO NOT SCALE	

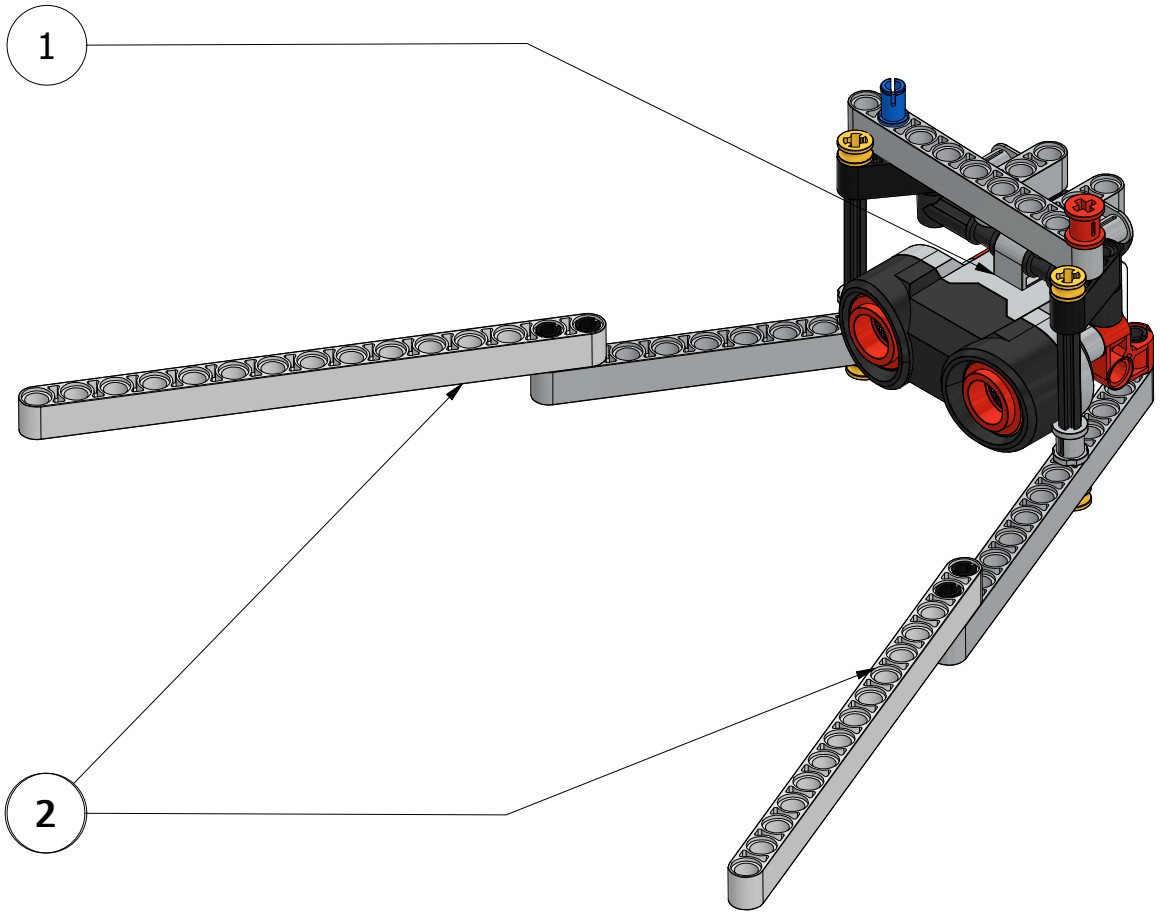
Hochschule Reutlingen
Reutlingen University

Fruta Oes
TITLE

FrutaOes
DRAWING & PART NO.

1 : 1.5	1 of 4	A4
SCALE	SHEET	SIZE

ITEM NO.	
QTY.	
MATERIAL	
FINISH	



PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	Ultra-Sonic Sensor
2	2	Manipulator Fangs

TOLERANCES		NAME	DATE
LINEAR	ANGULAR	Encarna DRAWN	2022/12/12
0 ±0.5mm 0.0 ±0.1mm 0.00 ±0.05mm 0.000 ±0.01mm	±0.5°	Laura DETAILED	2022/11/24
		Ferlando APPROVED	2022/12/26
		CAD DRAWING, DO NOT MANUALLY UPDATE	
		DO NOT SCALE	

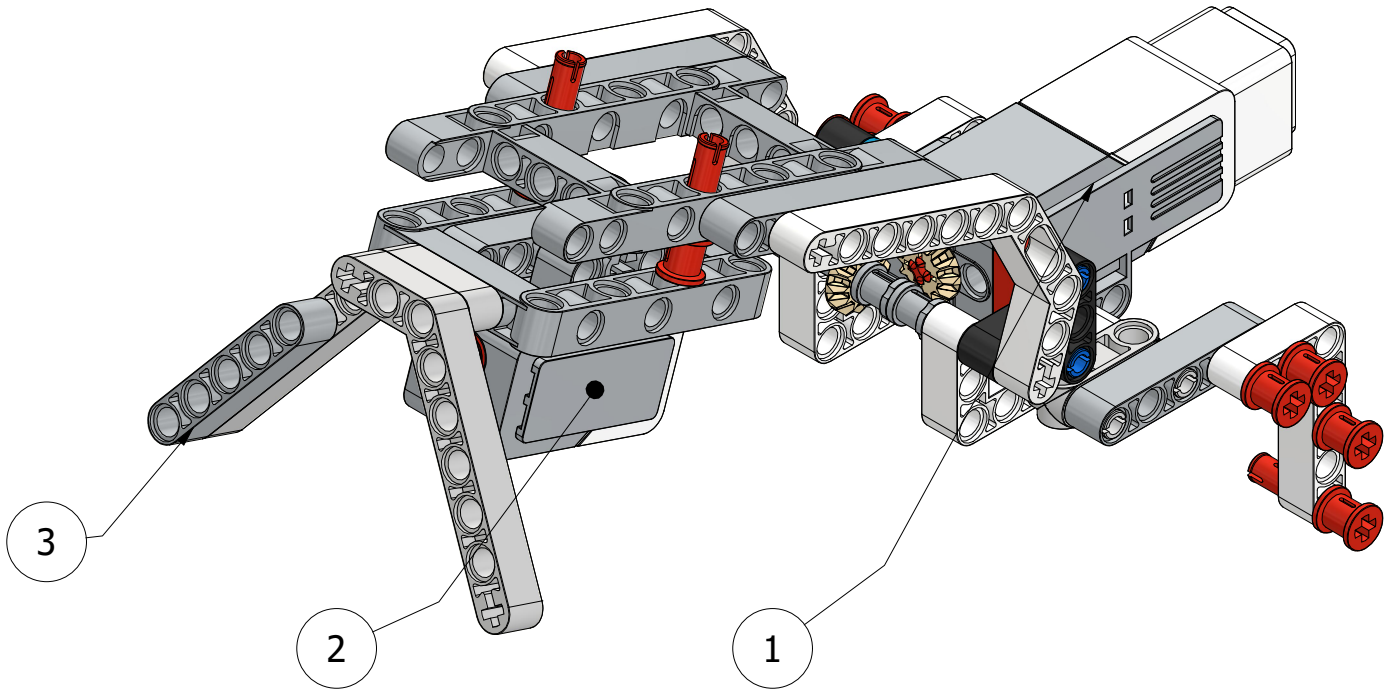
Hochschule Reutlingen
Reutlingen University

Object Detection and Handling Module
TITLE

Object Detection and Handling Module
DRAWING & PART NO.

1 : 1.5	2 of 4	A4
SCALE	SHEET	SIZE

ITEM NO.	
QTY.	
MATERIAL	
FINISH	



PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	EV3 Medium Motor
2	1	EV3Colour Sensor
3	2	Manipulator Fangs

TOLERANCES		NAME	DATE
LINEAR	ANGULAR	Laura DRAWN	2022/12/16
0 ±0.5mm 0.0 ±0.1mm 0.00 ±0.05mm 0.000 ±0.01mm	±0.5°	Encarna DETAILED	2022/12/01
		Ferlando APPROVED	2022/12/01
		CAD DRAWING, DO NOT MANUALLY UPDATE	
		DO NOT SCALE	

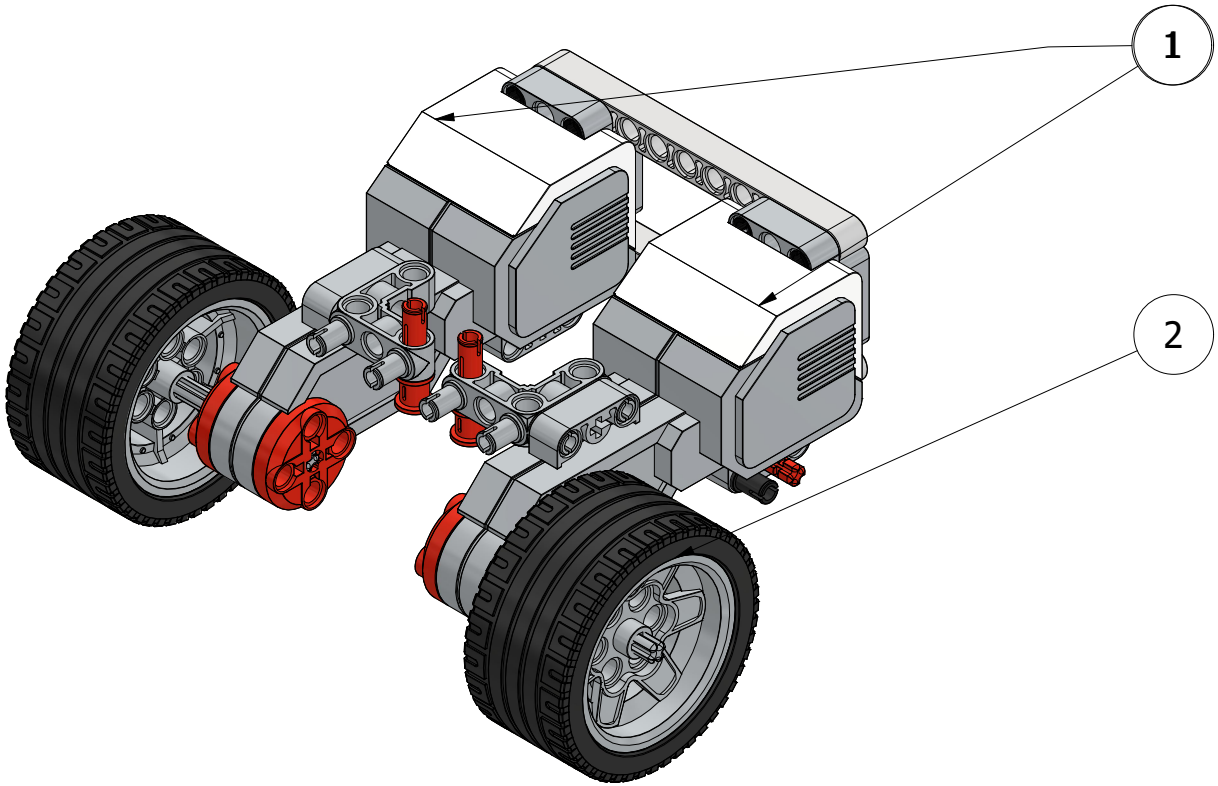
Hochschule Reutlingen
Reutlingen University

Gripper and Colour Sensing Module
TITLE

Gripper and Colour Sensing Module
DRAWING & PART NO.

1 : 1.2	3 of 4	A4
SCALE	SHEET	SIZE

ITEM NO.	
QTY.	
MATERIAL	
FINISH	



PARTS LIST		
ITEM	QTY	PART NUMBER
1	2	EV3 Large Motor
2	2	EV3 Wheels

TOLERANCES		NAME	DATE
LINEAR	ANGULAR	Ferlando DRAWN	2022/12/11
0 ±0.5mm 0.0 ±0.1mm 0.00 ±0.05mm 0.000 ±0.01mm	±0.5°	Encarna DETAILED	2022/12/06
		Laura APPROVED	2022/12/06
		CAD DRAWING, DO NOT MANUALLY UPDATE	
		DO NOT SCALE	

Hochschule Reutlingen
Reutlingen University

Drive Base 2022
TITLE

Drive Base
DRAWING & PART NO.

1 : 1.5	4 of 4	A4
SCALE	SHEET	SIZE