I-TOBOR
Project Manual

Team Members:

Mathilde Gaboriaud

Samuel Marajo

Wenzheng Wang

Lovinsky Jean Julien

Tamilarasan Raja

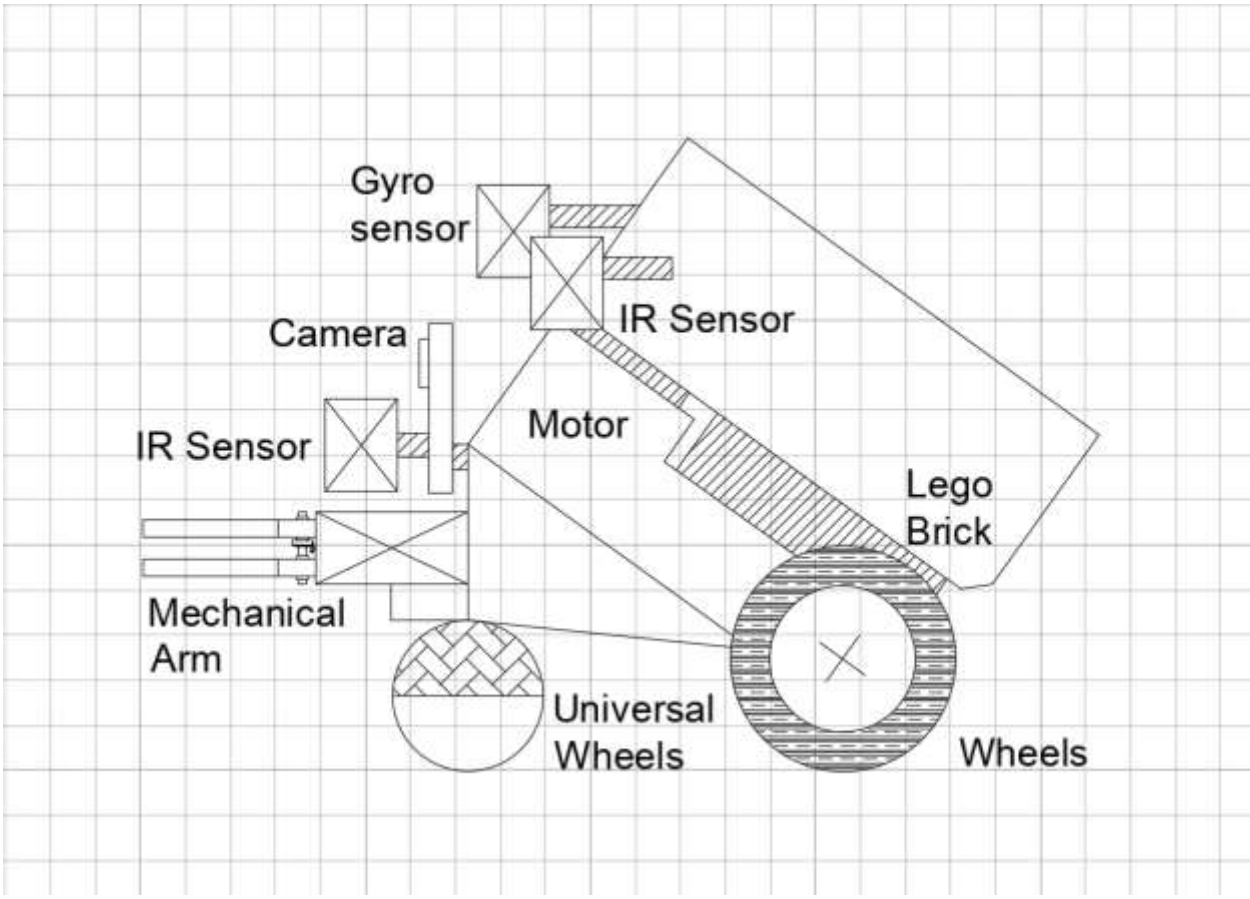
TABLE OF CONTENTS

Introduction

We are going to build a robot, which can search and find a defined object and to rescue it.

The Robot can explore the area within 1m and detect the target and is able to pick and comeback to the home.

Hardware Design:



- To maintain the symmetry of the robot, we use symmetrical parts for the two wheels in the hardware design.
- To ensure that the position of the wheels remains constant in the steering, I have added holding parts to the front and rear of the wheels to keep the wheel position constant.
- Due to the addition of the robotic arm in front of the car, 2 gimbals were added in front of the robot to ensure smooth running.
- To ensure a smooth return to home after object recognition, two IR sensors were added to keep the robot running properly.
- The gyroscope was added to ensure that the robot gets the correct angle when steering. The Lego brick was erected with different parts to keep more space in front of the robot.
- It is worth noting that the gyroscope should be kept in a horizontal position for the gyroscope to work properly.
- Before each robot run, the relevant parts of the wheels and robot arms should be checked for locking, this ensures symmetry in the robot architecture and also ensures reliable robot operation.

Components used:

- EV3 Brick
- Large and medium motors
- Infrared sensors
- Gyroscope sensor
- Pixy Camera
- Wheels

Large and medium motors

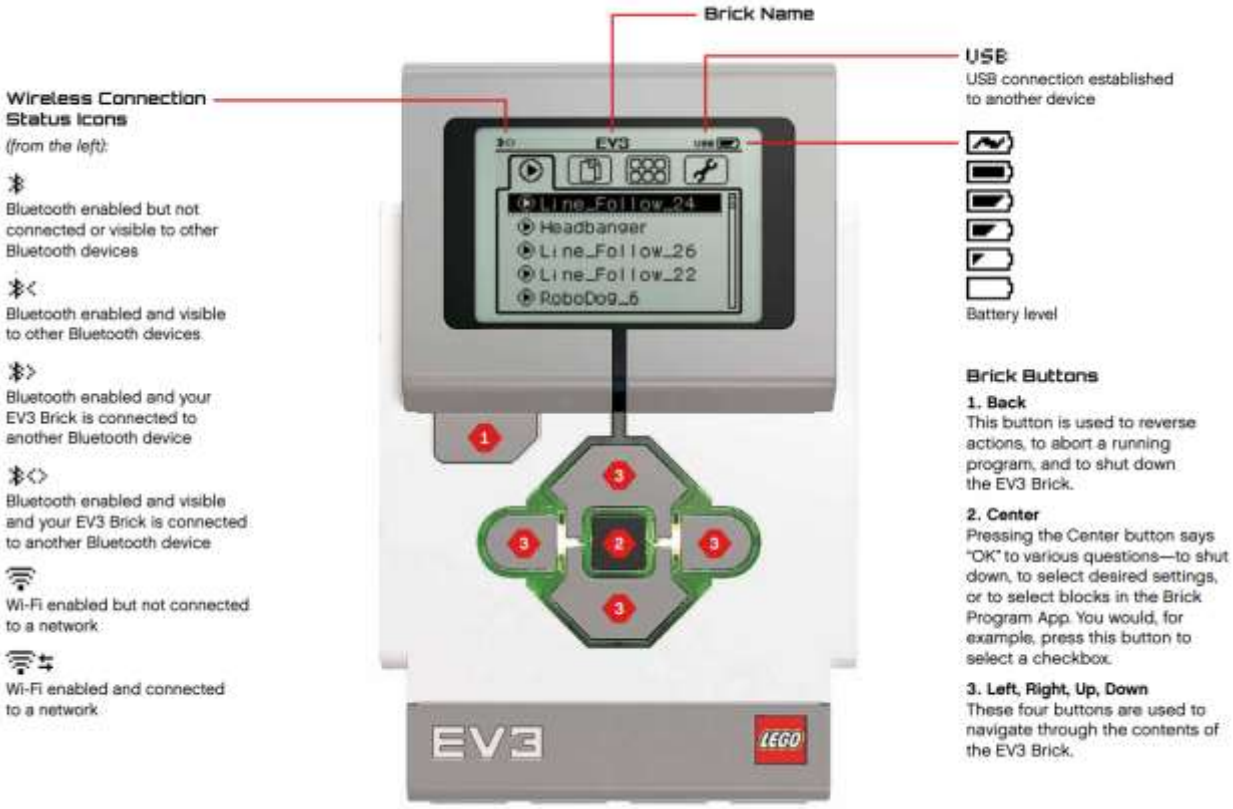
Pixey Camera

EV3 Brick

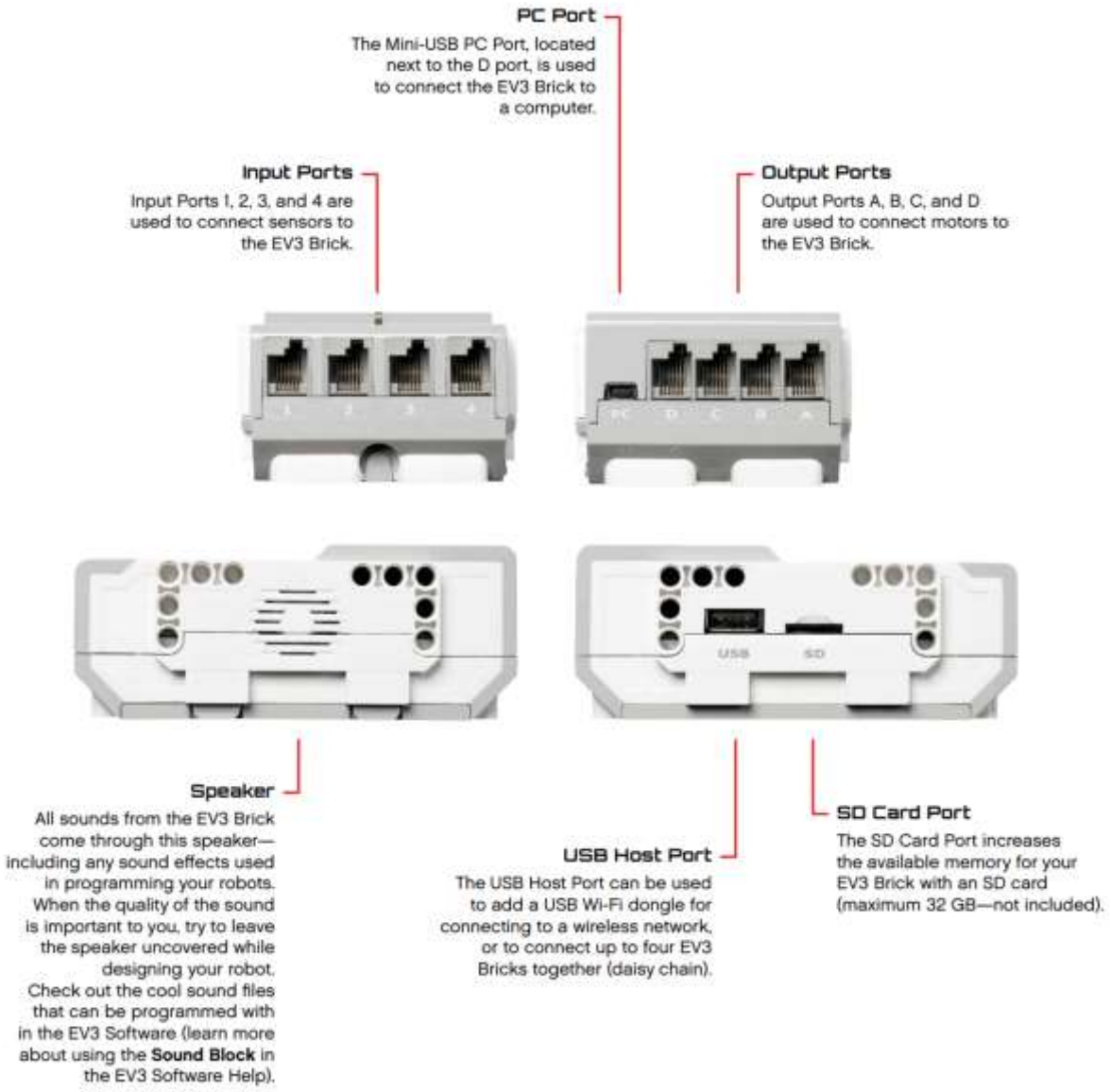
Gyroscopic sensor

Infrared sensor

EV3 Brick:



- The Display shows what is going on inside the EV3 Brick and enables you to use the Brick Interface
- It also allows you to add text, number or graphic responses into your program or experiments.
- The Brick Buttons allow you to navigate inside the EV3 Brick Interface
- Operating System - LINUX
- 300 MHz ARM9 Controller
- Flash Memory - 16 MB and RAM - 64 MB
- Brick Screen Resolution—178x128/Black & White
- USB 2.0 Communication to Host PC—Up to 480 Mbit/sec + USB 1.1 Host Communication—Up to 12 Mbit/sec
- Micro SD Card - Supports SDHC, Version 2.0, Max 32 GB
- Motor and Sensor Ports, Connectors - RJ12 + Support Auto ID + Power - 6 AA batteries



Large and Medium Motor:

Both the large and medium servo motors are equipped with a built-in Rotation Sensor with 1-degree resolution for precise control. The rotation sensor is used to measure how far a motor has turned (or has been turned). Rotation sensors can detect an amount of rotation in degrees or full rotations. You can also use the rotation sensor to find out what power level a motor is currently running at. Both motors are Auto ID supported.

The Large Motor runs at 160 - 170 rpm, with a running torque of 20 Ncm and a stall torque of 40 Ncm (slower, but stronger)

The Medium Motor runs at 240–250 rpm, with a running torque of 8 Ncm and a stall torque of 12 Ncm (faster, but less powerful).

Infrared Sensor:

The Infrared Sensor is a digital sensor that can detect infrared light reflected from solid objects. It can also detect infrared light signals sent from the Remote Infrared Beacon. The Infrared Sensor can be used in three different modes: Proximity Mode, Beacon Mode, and Remote Mode.

PROXIMITY MODE:

In Proximity Mode, the Infrared Sensor uses the light waves reflected from an object to estimate the distance between the sensor and that object. It reports the distance using values between 0 (very close) to 100 (far away), not as a specific number of centimeters or inches. The sensor can detect objects up to 70 cm away, depending on the size and shape of the object.

Gyroscope sensor:

The Gyro Sensor is a digital sensor that detects rotational motion on a single axis. If you rotate the Gyro Sensor in the direction of the arrows on the case of the sensor, the sensor can detect the rate of rotation in degrees per second. The sensor can measure a maximum rate of spin of 440 degrees per second. In addition, the Gyro Sensor keeps track of the total rotation angle in degrees. You can use this rotation angle to detect, for example, how far your robot has turned. This feature means you are able to program turns (on the axis the Gyro Sensor is measuring) with an accuracy of +/- 3 degrees for a 90-degree turn.

Pixy Camera:

PIXY is one of the camera modules specified for image processing, the recognition algorithm is color-based filtering. The main purpose of this camera is recognition colors and name them as a familiar object. This camera can “learn” what colors you “thought” it at first. In our project we choose RED color as target.

Wheels:

Two large wheels are attached in the back with two large motors for the movement of the Robot. Universal wheel is attached in the front to move freely and to help arm to pick accordingly.

Connections:

Sensors and Motors:

Motors and sensors must be connected to the EV3 Brick using the flat black Connector Cables, link sensors to the EV3 Brick using Input Ports 1, 2, 3, and 4. If you create programs while your EV3 Brick is disconnected from your device, the software will assign sensors to the following default ports: Port 1: Touch Sensor , Port 2: No sensor , Port 3: Color Sensor , Port 4: Infrared Sensor.

Port 1	Pixy Camera
Port 2	IR sensor
Port 3	Gyro sensor
Port 4	IR sensor
Port A	Medium motor
Port B	Large motor
Port C	Large motor

EV3 Brick to Your Computer:

We can connect the EV3 Brick to your computer by USB Cable or wirelessly using either Bluetooth or Wi-Fi. In our project, we connect EV3 brick using WIFI.

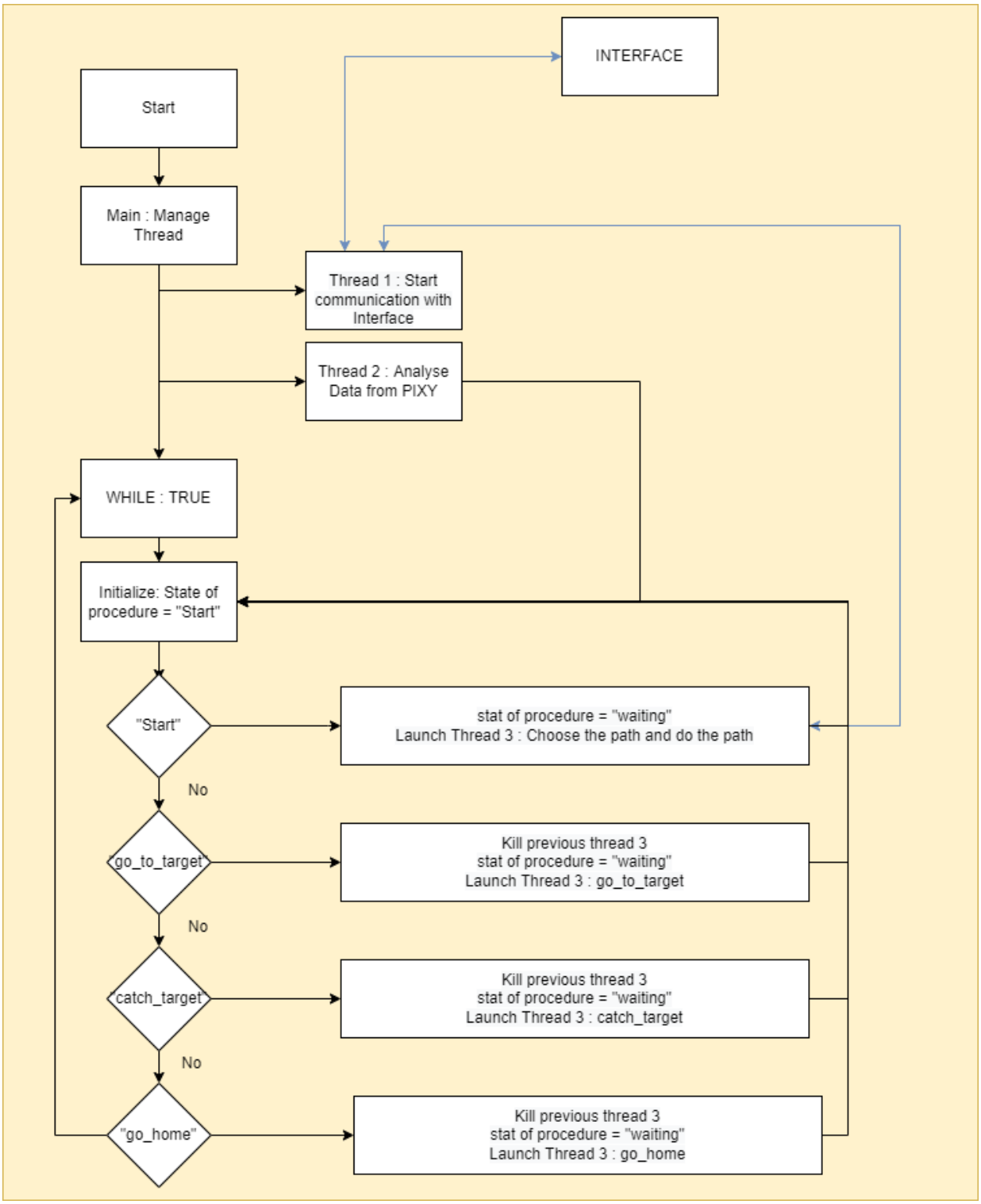
Wi-Fi:

First, we need Wi-Fi USB dongle to begin the setup. you must have access to a wireless network and to know the network's name and its password.

- Insert the Wi-Fi USB dongle into the EV3 brick's USB port.
- Turn on the EV3 brick and go to the "Wi-Fi" menu in the EV3 brick's settings. Select "Connect to Network" and enter the name and password of the wireless network you want to connect to.
- Once connected, you should be able to communicate with the EV3 brick from your computer using the EV3 programming software.
- Before you start programming, you should check the IP-address of the EV3 brick, this will be necessary to establish a connection between your computer and the EV3 brick.
- Once you have the IP-address, open the EV3 programming software on your computer, and enter the IP-address of the EV3 brick in the software to establish a connection.
- Once the connection is established, you can start programming your EV3 brick. Send your program to the brick, control the brick's motors and sensors, and more.

Software:

main.py	**class Robot:**
send_message():	left_rotation():
path_1():	right_rotation():
path_2():	get_angle_value():
path_3():	get_ir_value():
go_to_target():	get_us_value():
catch_target():	turn_right_with_precision():
find_target():	turn_left_with_precision():
	move_forward_with_us_sensor():
serveur.py	move_forward():
	move_target_forward():
class ThreadForClient:	move_target_forward_with_us_sensor():
class SendAndReceiveData:	stop(self):
class ExchangeWithUser:	turn_right():
	turn_left():
class odometry:	catch_target():
	drop_target():
update_odometry():	isPathOver():
update_compensation():	go_home():


INTERFACE
Start
Main : Manage Thread
Thread 1 : Start communication with Interface
Thread 2 : Analyse Data from PIXY
WHILE : TRUE
Initialize: State of procedure = "Start"
"Start"
stat of procedure = "waiting"
Launch Thread 3 : Choose the path and do the path
No
"go_to_target"
Kill previous thread 3
stat of procedure = "waiting"
Launch Thread 3 : go_to_target
No
"catch_target"
Kill previous thread 3
stat of procedure = "waiting"
Launch Thread 3 : catch_target
No
"go_home"
Kill previous thread 3
stat of procedure = "waiting"
Launch Thread 3 : go_home