ANTI DRONE SYSTEM

Innova team

MARTIN Léa (Project Manager)

TIE Emmanuel (Hardware Engineer)

JIMENEZ MADRONA Antonio (Software Engineer)

PELLIZA Ezequiel (Test Engineer)

LI Zhenzhi (Documentation Manager)

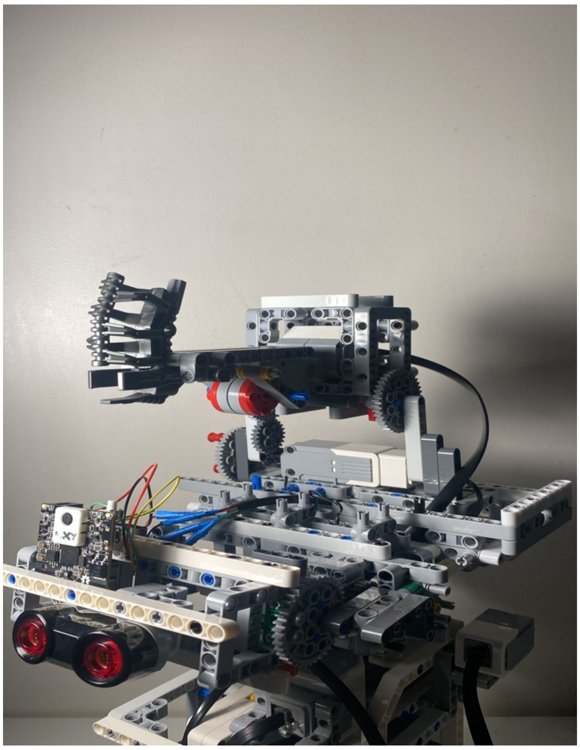
Table of contents

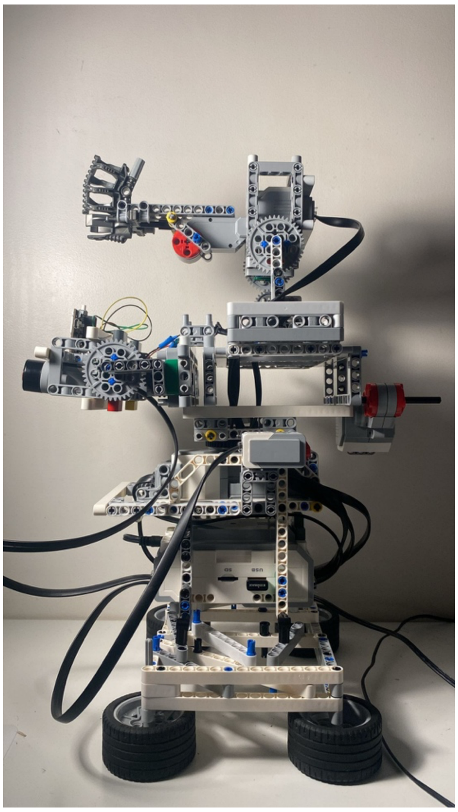
1.Introduction

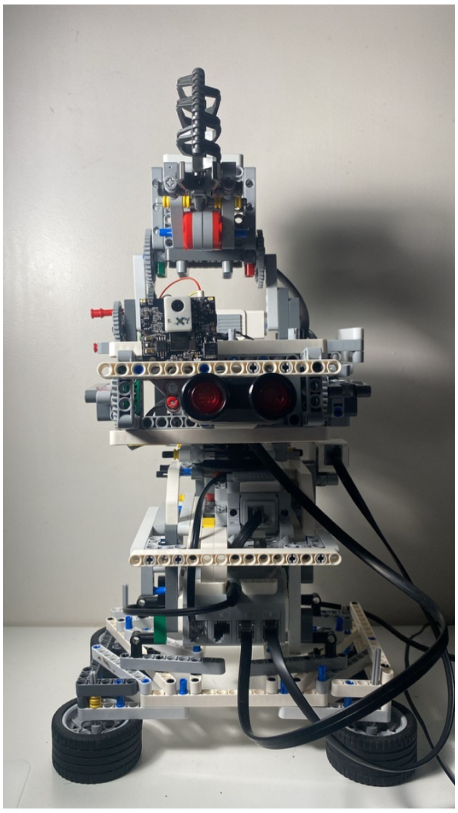
1.1 Purpose of the manual

- System Overview: Provides an overall overview of the Anti Drone system, including project goals, key specifications, and testing phases. This will help the user to understand the background and objectives of the system.
- User Interfaces: An introduction to the graphical user interfaces that may be involved in Gold testing, including the design, functionality, and operational description of the interface. This will provide the user with an understanding of how to interact with the system.
- Commands and Controls: Explain how users interact with the system, including how to send commands, obtain system status information, and perform specific actions on the system. Ensure that the user understands and is able to use command and control functions correctly.

1.2 Overview of the system







1.3 Project goals and objectives

Goals:

Achieve Target Recognition: Develop a robotic system capable of recognizing targets in the surrounding environment using a camera.

Implement Target Tracking: Once a target is identified, enable the robot to track the target's movement to ensure it remains within the field of view.

Implement Automatic Shooting: Develop a robotic system that can automatically adjust the shooting angle based on the target's position and distance, and execute the shooting action.

Integrate Touch Sensor: Utilize a touch sensor as a safety mechanism, allowing the robot to halt its movement when necessary to ensure operational safety.

Integrate Ultrasonic Sensor: Use an ultrasonic sensor to measure the distance between the robot and the target, aiding in target tracking and shooting precision.

Objectives:

Enhance Target Recognition Accuracy: Improve the robot's accuracy and stability in recognizing targets through algorithm optimization and parameter adjustments.

Enhance Target Tracking Precision: Improve the robot's target tracking algorithm using sensor data and real-time feedback to ensure accuracy and responsiveness.

Implement Multi-Motor Control: Enable the robot to perform full-scale scanning and shooting in three-dimensional space by appropriately controlling horizontal and vertical motors.

Ensure User-Friendliness: Design a user-friendly interface or command system, allowing users to interact with and control the robot seamlessly.

Ensure Safety: Throughout the project, ensure that the robot's movements and shooting operations comply with safety standards, minimizing potential risks of accidents.

Provide Comprehensive Documentation: Produce detailed documentation, including user manuals and technical documents, to facilitate user and developer understanding of the system's principles and operation.

1.4 Design and Implementation Limitations

- The project is programmed in python, EV3 core controller and GitHub platform.
- The whole machine is made of Lego plastic parts, so it can not be bumped and beaten.
- The code implemented by the program can be found on GitHub, which is convenient for users to better understand the meaning of the program.

2. Components

2.1 EV3 Brick

The **Brick Status Light** that surrounds the Brick Buttons tells you the current status of the EV3 Brick. It can be green, orange, or red, and can pulse. Brick Status Light codes are the following:

- Red = Startup, Updating, Shutdown
- Red pulsing = Busy
- Orange = Alert, Ready
- Orange pulsing = Alert, Running
- Green = Ready
- Green pulsing = Running program

Input ports:

Input Ports 1, 2, 3, and 4 are used to connect sensors to the EV3 Brick.

Output ports:

Output Ports A, B, C, and D are used to connect motors to the EV3 Brick.

PC port:

The Mini-USB PC Port, located next to the D port, is used to connect the EV3 Brick to a computer.

2.2 Touch sensor

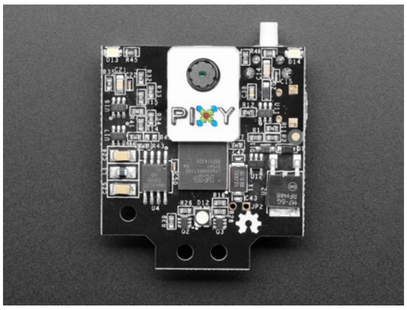
The Touch Sensor is an analog sensor that can detect when the sensor’s red button has been pressed and when it is released. That means the Touch Sensor can be programmed to action using three conditions—pressed, released, or bumped (both pressed and released).

The role of the touch sensor here is to provide an external condition that immediately stops the motion or operation of the robot when the user presses the touch sensor. This mechanism can be used for safety control, where the user can stop the robot at any time by pressing the touch sensor in case of an accident or the need for an emergency stop.

2.3 Ultrasonic sensor

The role of the ultrasonic sensor here is to be used to detect the presence of a target in front of the robot and to provide information about the distance to the target. This information can be used to adjust the robot's movements, such as adjusting the firing angle to ensure that the target is hit.

2.4 Pixy2 camera



It can make image recognition easier, supports multi-object recognition, and has powerful multi-color color recognition and color block tracking capabilities.

Processor: NXP NXP LPC4330, 204 MHz dual-core.

Image sensor: Aptina MT9M114, 1296×976 resolution,

integrated image stream processor.

Lens field of view: horizontal 60 degrees, vertical 40 degrees.

Power consumption: 140 mA typical.

Power input: USB input (5V) or Vin input (6V~10V).

RAM: 264K bytes.

Flash memory: 2M bytes.

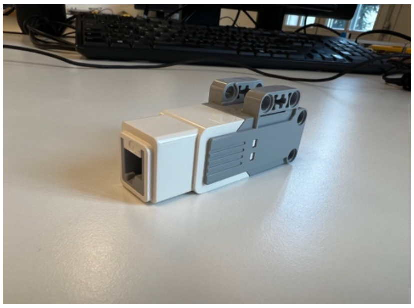
Available data outputs: UART serial, SPI, I2C, USB, digital, analog.

Dimensions: 1.5" x 1.65" x 0.6".

Weight: 10g.

Integrated light source: about 20 lumens.

2.5 Motor



Medium Motor can be programmed to turn on or off, control its power level, or to run for a specified amount of time or rotations.The motors run through the program, combining various gears and Lego parts, and then the parts of the machine move independently. The motor is the most important source of power.

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).

The Large Motor is a powerful “smart” motor. It has a built-in Rotation Sensor with 1-degree resolution for precise control. The Large Motor is optimized to be the driving base on your robots.

By using the Move Steering or Move Tank programming block, the Large Motors will coordinate the action simultaneously.

3. Getting Started

3.1 Remote connexion

1. Creation of the Wifi network

In order to communicate with the robot, you will need to establish an SSH connec(on from your computer or any device that can do this connec(on, to the block robot. You will need to be connected to the same network.

2. Connection of the robot to WiFi

You are able to connect to you robot by the Wifi to send document or launch a script. First of all, you need to plug the Wifi dungle to one of the USB port of the lego block. On the interface block, you need to go on the Wifi menu and click on the “scanning network” line.

Then, you select the network, write the network password and the robot is connected to Wifi.

3. Set up the SSH connection

A. By password

There are several methods to connect by ssh to the robot, but we choose to connect by password.

From your computer, you open a terminal to send the ssh command: *ssh robot@IP_adress*. Then, you will need to write the password robot block which is define by default as *maker*.

You are now connected to the device on a terminal, as if you were in a local terminal. You can now send document or launch any script on the block robot !

3.2 Cloning your Git hub repository

In order to have your git hub repository on the local space of your robot, you need to be connected by an SSH connec(on. You will do the following steps in the terminal of your robot.

If you have a github repository, you can clone it on the local repository of your robot. Thanks that, you have the last modifica(on from the last commit you pushed. For that, you just need to copy the url link and send the command: *git clone repo_url*

If your project has several submodules, you have to write the command: *git clone repo_url –recursives-submodules*

You have now a directory with the name of your project. We will consider for the next step that the repository only has one submodule. You send the command: *git pull;* then you have the last commit from your repository.

If you work with different branches, you will send the command: *git branch -a.*

To select the branch you are working on, you send the command: *git checkout branch_name*.

4. System function description

Welcome to our robot system, equipped with the capabilities of recognizing, tracking, and shooting targets. Here is a series of steps to guide you on how to operate and utilize these features.

Step1: Start the robot

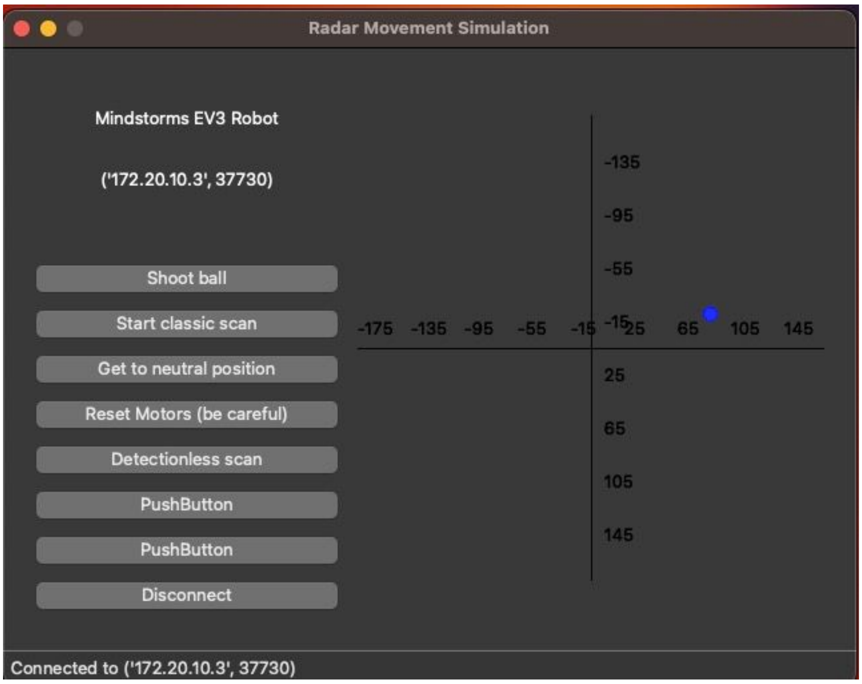
Ensure that the robot is in the powered-on state. Connect the EV3 main controller and power it up.

Step2: Connect to External Device

Connect the robot to an external device through WiFi or other communication methods, ensuring that both the external device and the robot are on the same network.

Step3: Send Commands

Use the external device to send commands to the robot. You can trigger different actions with the following commands:



Step4: Scanning Operation

Horizontal Scanning:

The horizontal motor will gradually rotate, allowing the camera to horizontally scan the surrounding environment. This helps the robot detect the approximate direction of the target.

Vertical Scannning:

At each position of the horizontal scan, the vertical motor will gradually rotate, enabling the camera to vertically scan. The ultrasonic sensor will detect the distance to the target.

Target Detection:

If a target is detected, the robot will emit an audible alert and adjust the angle of the vertical motor to ensure shooting accuracy.

Step5: Shooting

If the target is correctly detected and tracked, the robot will automatically adjust the shooting angle and execute the shooting action. This can be triggered by sending the corresponding command or by algorithmic means.

Notes:

Ensure a safe surrounding environment while using the robot to avoid collisions or other unexpected events.

Parameters in the code can be adjusted to adapt to different environments and task requirements.

Prior to using the robot system, read relevant documentation to understand the operational principles and safety protocols.

By following these steps, you will be able operate the robot effortlessly and achieve the functionality of target recognition, tracking, and shooting. Enjoy your use!