PROJECT BATTLEBOT DOCUMENTATION

By Abu Hasib Shanewaz

Contents

Chapter 1: BattleBot Hardware

This section provides detailed information about the hardware components used for the BattleBot. Each component plays a crucial role in ensuring the BattleBot operates efficiently and meets its designed objectives.

1.1 Arduino NANO Microcontroller

The Arduino NANO serves as the brain of the BattleBot. It is a compact, yet powerful microcontroller based on the ATmega328P. It manages the robot's operations, including reading sensor inputs, processing data, and controlling the motors and gripper. The small size of the Arduino NANO makes it ideal for compact robot designs.

1.2 PCB with Arduino UNO Footprint for Additional Shields

A custom PCB with an Arduino UNO footprint is used to accommodate additional shields and expand the functionality of the BattleBot. This setup allows for easy integration of various modules and sensors, facilitating a modular and versatile design.

1.3 Metal Chassis

The metal chassis provides a sturdy and durable framework for the BattleBot. It houses all the electronic components and ensures the robot can withstand the mechanical stresses encountered during operation. The metal construction also adds weight, improving traction and stability.

1.4 Powerbank (10,000 mAh)

A 10,000 mAh powerbank supplies the necessary power to the BattleBot. This high-capacity battery ensures the robot can operate for extended periods without frequent recharging. The powerbank powers both the logic circuits and the motors, providing consistent and reliable energy.

1.5 On/Off Switches

Two On/Off switches are used: one for the logic circuit and one for the motor circuit. These switches allow for easy and independent control of the robot's power systems, enabling safe and convenient operation.

1.6 Gripper with Servo Motor

The BattleBot is equipped with a gripper controlled by a servo motor. This gripper allows the robot to grasp and manipulate objects. The servo motor provides precise control over the gripper's movements, enabling secure gripping and accurate placement of objects.

1.7 Electromotors

The BattleBot is driven by two electromotors, providing the necessary propulsion. These motors are powerful enough to move the robot and carry objects while maintaining control and stability.

1.8 Ultrasonic Distance Sensor

An ultrasonic distance sensor is used for object detection. This sensor measures the distance to objects in front of the robot, allowing it to detect obstacles and interact with them using the gripper.

Chapter:2 Requirement Analysis

2.1. Essential Functions

Autonomy

- After it is begun, the robot must function independently without assistance from a human.

Line Tracking

- On a white surface, the robot needs to be able to recognize and follow a black line.
- A responsive and accurate line-following system is necessary to guarantee that the robot remains on the line at all times.

Identifying Objects

- The robot needs to be able to use an ultrasonic sensor to identify objects that are in its path.
- To ensure rapid reaction, objects should be detected within an 8 cm range.

Gripper Function

- A servo-operated gripper that can identify and grasp objects is a must for the robot.
- When needed, the gripper should be able to open to release things after closing securely on them.

Mechanism of Stopping

- When the robot gets to a specified spot (such as a black box), it has to stop.
- When stopping, the robot has to release whatever object it is holding.

2.2. Non-Operational Conditions

Achievement

- The robot should react quickly to obstacles and follow the line.
- To grab and release things, the gripper needs to work fast and effectively.

Dependability

- Over several runs, the robot must reliably follow the line and recognize items.

- Robust hardware and software are necessary to provide uninterrupted operation without errors.

Utilization

- With little assistance from the operator, the robot should be simple to start and operate.
- A serial monitor should be used by the system to offer feedback for debugging and status updates.

Security

- If any of the sensors identify a black surface, the robot will not be able to continue along its intended path and must stop.
- Safe handling without overheating or overloading should be ensured by the power supply and motor operations.

2.3. Hardware Specifications

Microcontroller

- Robotics is controlled via an Arduino Nano.

Sensors

- Eight line follower sensors are used to find the black line.
- Ultrasonic object detection sensor (HC-SR04).

Motors

- The robot is propelled by two DC motors.
- A motor driver is used to manage the motors.

Gripper

- Servo motor for driving the gripper mechanism (SG90).

Power Source

- A power bank that can power the robot and its parts with a minimum capacity of 10,000 mAh.

Framework

- A sturdy metal chassis to safely contain every component.

Flips

- The motor circuit control and logic circuit are separated by two On/Off switches.

2.4. Necessary Software

Functions of Sensor Reading

- Features that allow line sensor analog values to be read and, using preset thresholds, converted to digital values.

Functions of Motor Control

- Functions to regulate the motors' direction and speed in response to sensor input.

Functions of Ultrasonic Sensors

- Calculates the distance to detected objects and initiates the ultrasonic sensor.

Gripper Control Features

- Operates the servo motor gripper in order to grasp and release items.

Primary Control Loop

- A loop function that runs the gripper, checks for things, reads sensor values continuously, and controls the motors.

2.5. Environmental Conditions

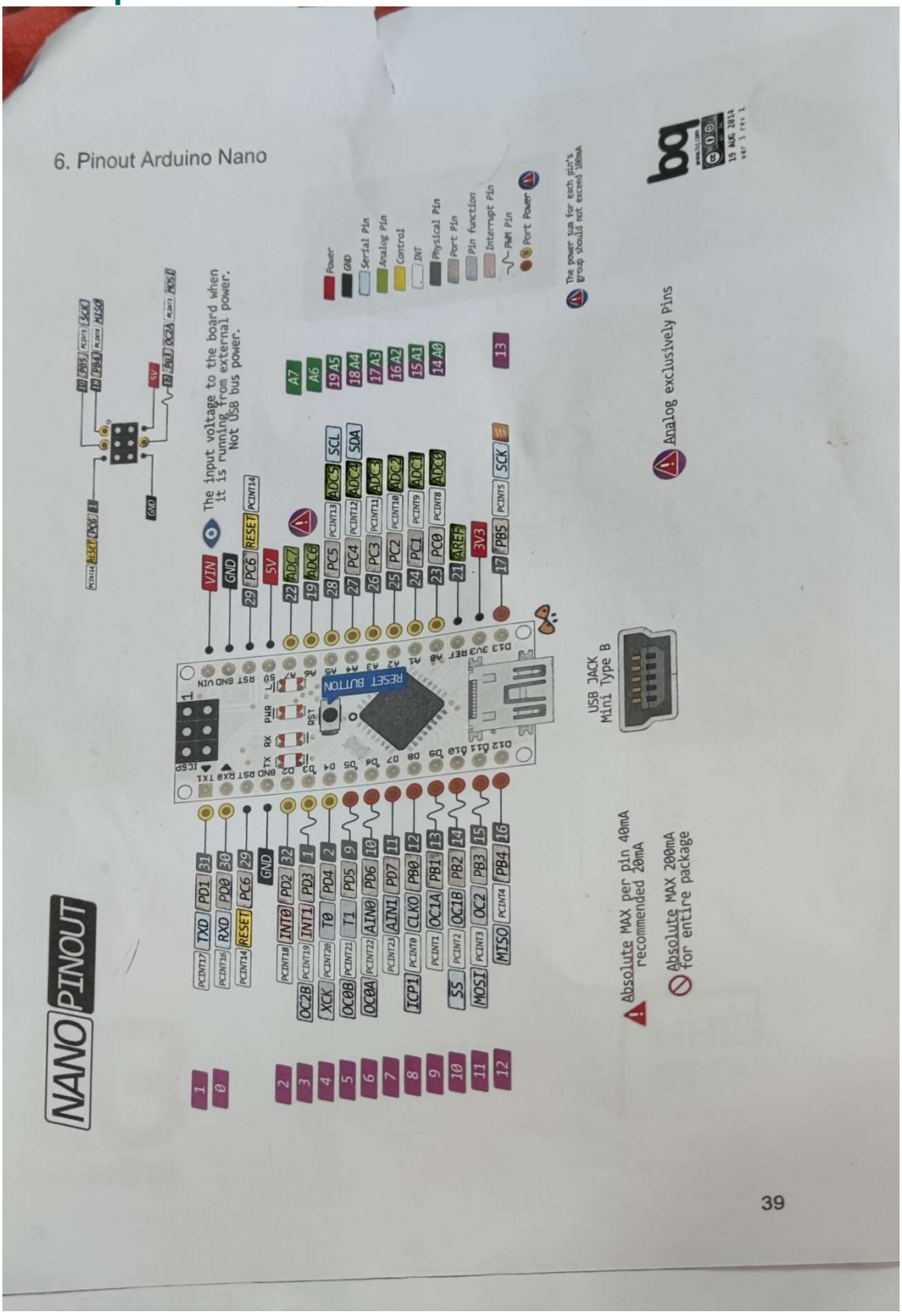
Surface of Operations

- A clean, white surface with a distinct black line should be used for the robot's operations.

Lighting Specifications

- Sufficient and uniform illumination to guarantee precise sensor measurements.

Chapter 3: Arduino Pinout



Chapter 4: IO List

A detailed IO list is provided below.

Pin Number	Function	Connected Component
D3	Motor Right Forward	Motor Driver
D4	Motor Right Backward	Motor Driver
D5	Motor Left Forward	Motor Driver
D6	Motor Left Backward	Motor Driver
D8	Echo Pin	Ultrasonic Sensor (HC-SR04)
D9	Trigger Pin	Ultrasonic Sensor (HC-SR04)
D11	Servo Control	Gripper Servo Motor (SG90)
D13	LED	Onboard LED
A0	Line Sensor 1	Line Tracking Sensor
A1	Line Sensor 2	Line Tracking Sensor
A2	Line Sensor 3	Line Tracking Sensor
A3	Line Sensor 4	Line Tracking Sensor
A4	Line Sensor 5	Line Tracking Sensor
A5	Line Sensor 6	Line Tracking Sensor
A6	Line Sensor 7	Line Tracking Sensor
A7	Line Sensor 8	Line Tracking Sensor

Chapter 5: Reflection

Reflection on the Process

Planning and Design

The process of developing the BattleBot involved several key steps, each contributing to the final functionality of the robot. The approach was methodical, starting with basic wiring and progressing to more complex integrations.

1. **Initial Wiring and Pin Testing**:
    - **Wiring the Robot**: The first step was to wire the robot, connecting all the essential components, including the Arduino Nano, motors, IR sensors, ultrasonic sensor, servo motor, and NeoPixel LED.
    - **Pin Testing**: Ensuring that each component was connected to the correct pin on the Arduino Nano was crucial. This involved testing each pin to verify its functionality. For example, each motor pin was tested to ensure the motors responded correctly, and each IR sensor pin was checked for accurate readings.
2. **Functionality Development**:
    - **Line Tracking**:
        - **Pin Testing for Motors and IR Sensors**: After wiring, the next step was to ensure the motors and IR sensors were correctly connected and operational. This involved writing simple test code to make the motors move and the IR sensors read values.
        - **Threshold and Speed Calibration**: The IR sensors required calibration to determine the correct threshold values for detecting the black line. Similarly, the motors needed to be set to the correct speeds to ensure smooth and responsive movement.
        - **Adjusting Turns**: Once the basic line-following functionality was working, the code was refined to improve the robot's ability to navigate turns. This involved adjusting the motor speeds dynamically based on the sensor readings to ensure the robot stayed on the line.
    - **Object Detection and Gripper Control**:
        - **Ultrasonic Sensor Testing**: The ultrasonic sensor was tested to determine the distance to objects. This required finding the correct pin configuration and calibrating the sensor to provide accurate distance measurements.
        - **Gripper Speed and Control**: The servo motor controlling the gripper was calibrated to determine the correct speed and angles for opening and closing.

This involved experimenting with different speeds to find the optimal settings for reliable object manipulation.

3. **Integration and Testing**:
    - **Combining Functionalities**: After developing the individual functionalities, the next step was to integrate them into a cohesive system. This involved combining the line tracking, object detection, and gripper control code into a single program.
    - **Debugging and Refinement**: The integrated system required extensive testing and debugging. This included ensuring that the robot could follow a line, detect objects, and manipulate the gripper seamlessly without conflicts between the different functionalities.
4. **NeoPixel Integration**:
    - **Pinout and Combinations**: The NeoPixel LED required finding the correct pinout and experimenting with different combinations to achieve the desired blinking patterns. This added a visual element to the BattleBot, enhancing its overall functionality and aesthetics.

Challenges Faced

- **Wiring Complexity**: Managing the wiring for multiple components was challenging. Ensuring each connection was correct and secure required careful attention to detail.
- **Sensor Calibration**: Calibrating the IR sensors and ultrasonic sensor to provide accurate readings took significant effort. This involved testing under various conditions and adjusting the thresholds and timing accordingly.
- **Code Integration**: Combining the different functionalities into a single codebase and ensuring they worked together smoothly required careful debugging and refinement.

Learning Outcomes

- **Technical Skills**: I gained a deeper understanding of Arduino programming, sensor integration, and motor control. This project enhanced my ability to troubleshoot and debug complex systems.
- **Problem-Solving**: The challenges faced during the project helped me develop problem-solving skills. I learned to approach problems methodically, breaking them down into smaller parts and testing solutions iteratively.