End User Documentations

Section 1: General System Description & Critical Data (Spec Sheet)

1. General Description

   This robot is designed to autonomously navigate around an unknown maze and locate two heat sources. The bot needs to launch three ping-pong balls near each heat source. Upon completion, the bot will climb up a ramp and find the third heat source to fire the last 3 ping pong balls. The system relies on TurtleBot 3 as the base, enhanced with LiDAR, AMG8833 thermal sensors and a projectile launcher.

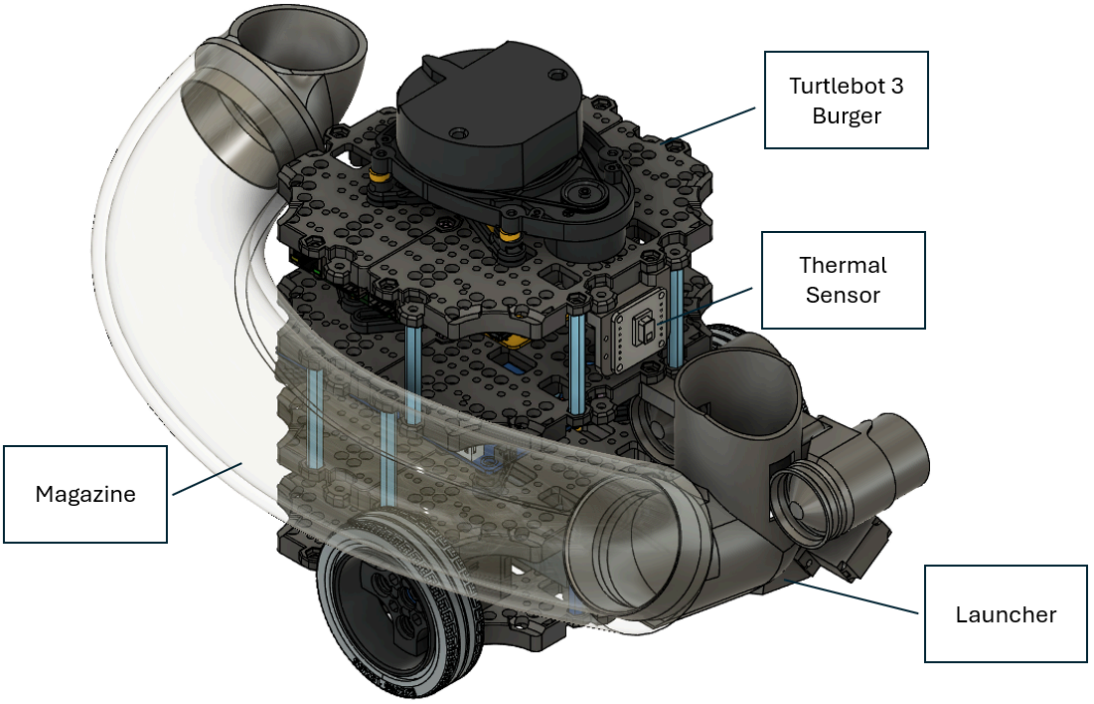
2. Critical Data / Specification Sheet

Model	Turtlebot 3 Burger
Processor & Control Unit	Raspberry Pi, OpenCR
Software version	ROS2 Humble, Ubuntu 22.04
Sensors Used	LiDAR: Used for SLAM-based navigation and mapping AMG8833: Thermal sensor for heat source detection
Actuators	Dynamixel Motors: For TurtleBot movement Servo Motors: For loading ping pong ball for launcher operation
Navigation & Communication	SLAM, ROS framework, Wi-Fi capabilities
Weight	Approx. 1.5 kg
Center of Gravity	Near the geometric center, slightly rear-biased due to battery placement
Battery Capacity	11.1V, 1800mAh Li-Po battery
Estimated Battery Life	1.42 hours (under worst-case power consumption)
Payload Capacity	Ability to carry and launch 9 ping-pong balls
Description of bot purpose	The autonomous robot has following features: 1. Navigate an unknown maze and record down the map on a computer. 2. Detect regions with heat higher than surrounding. 3. Fire ping pong balls at specific time intervals after heat signals are detected. 4. Climb a ramp at a known position.

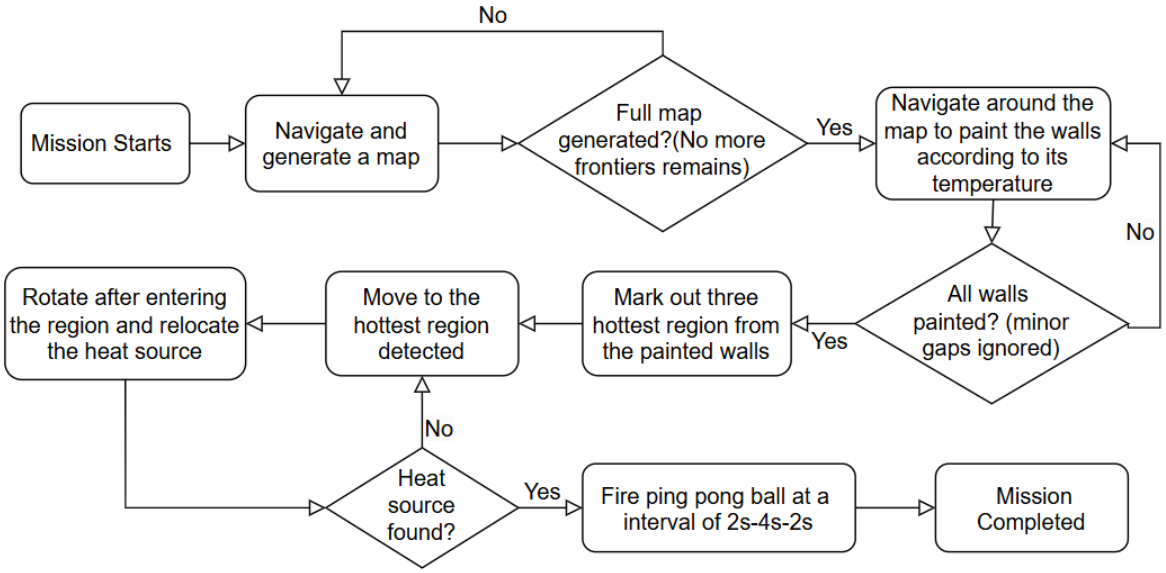
Refer to TurtleBot3 Burger Specifications found on ROBOTIS e-Manual website: https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#specifications

Section 2: Technical Guide

1. Section of the robot

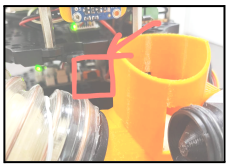


2. Robot running process



3. Steps for operating the robot

Steps	Rationale	Actions
Before the Run	Charging the Li-Po battery	Connect the battery to the blue charger's port with the black cable. The battery is done charging when the green LED indicator turns on.

0	Startup	1. FIRST, ensure that the laptop is connected to the network, then 2. Turn on Turtlebot Switch (see diagram below)  Wait for a startup tune to be played before proceeding to the next step.
1	Establishing SSH Connection to RPI	1. Ctrl + Alt + T 2. sshrp_disp_ip The turtlebotIP will be displayed to the terminal. Set the robot down in the maze
2	Start the program	1. cd ~/2310_workspace 2. tmuxinator start 2310_mission_lanuch ~/2310_workspace expected_ssh _time=16 turtlebotIP = 172.20.10.3 (replace with IP address get from step 1)

4. Functions
   (1) Navigation & Mapping (First time exploration)
   The robot utilizes Frontier-Based Exploration to discover new areas autonomously. It will move towards the direction of the furthest distance and will refind direction if no wall is detected on the left. The LiDAR scans the environment to generate a map in real time using SLAM (Simultaneous Localization and Mapping) and shows it on the laptop.
   (2) Heat Source Detection (Second time exploration)
   After the first exploration, the AMG8833 sensor is activated and continuously scans the environment after generating the map. The walls in the map formed will be painted in different colors according to the temperature it detects. Three regions with the highest temperature will be recorded. The robot will then navigate to each region recorded. It will spin until a heat source is redetected by the thermal sensor and approaches the heat sources.
   (3) Projectile Launcher
   The launcher consists of two flywheels powered by 12V motors, one loader (servo motor) and one tube. The tube can store 10 ping-pong balls.
   After the whole map exploration, three heat sources should be detected. The bot will navigate to each heat source. When the bot is within 10 cm of the heat source, the flywheels are activated. The loader (servo motor) will load the ping-pong balls in a time interval of 2-4-2s.
   (4) Error Handling & Safety Measures
   - When the robot gets stuck (i.e. it tries to reach a location it can not), the robot will stop the process and select to another destination to navigate towards.

- To avoid false detection of the heat source, the robot will mark out three general regions of the heat source during the mapping process. After all the mapping is done, the robot will move towards each region. When it enters the region, it will rotate until a heat source is redetected. It will then move toward the heat source and fire the ping pong ball.

Section 3: Acceptable Defect Log

1. Slight Sensor Inaccuracy: The heat sensor may have minor fluctuations (±0.5°C) but still detects sources effectively.
2. Launcher Speed Variation: Minor inconsistencies in projectile speed (±10%) as long as the ball clears obstacles.
3. Mapping Artifacts: Small inaccuracies in map generation due to SLAM drift (<10 cm deviation per meter).
4. Battery Efficiency Drop: If battery life is reduced to ~1.2 hours, it remains sufficient for mission execution.

Section 4: Factory Acceptance Test (FAT) Checklist

Test Item	Criteria	Status
Battery Test	Fully charged (11.1V)	
LiDAR Functionality	Detects obstacles correctly	
AMG8833 Heat Sensor	Detects heat sources (hand test at 1m)	
Motor Functionality	TurtleBot moves forward/backward correctly	
Launcher Test	Fires ping-pong ball above 1.5m	
SLAM Mapping	Generates map in an empty room	
Frontier Exploration Algorithm	Navigates a test environment	
Mechanical Assembly Checks	- Wheels and motors secured and operational - Ping-pong launcher mechanism functions correctly	

Section 4.5: Basic Debugging Steps

This section details the debugging steps for completing FAT and during the process of operating the robot. They are not exhaustive but are sufficient for basic debugging.

<u>Basic Command Line Interface (CLI) Commands</u>
ensure SSHed into the RPi before proceeding with these commands

sudo reboot now	reboots the RPi immediately
sudo shutdown now	shuts down the RPi immediately
rosbu	initialises LD08 LIDAR driver, OpenCR and Dynamixel drivers
rteleop	ensure rosbu is running on a separate terminal. enables teleoperation interface to test basic movements

Battery Level Check

Use battery level indicator device (included in kit, see below)

OR if SSHed into the RPi,

1. [RPi] run rosbu on the RPi
2. [REMOTE PC/RPi] on a separate terminal, run ros2 topic echo battery_state
3. read the 'battery percentage' parameter

LIDAR not scanning properly?

SSH into the RPi

1. [RPi] run rosbu on the RPi
2. [REMOTE PC] on a separate terminal, run rslam
3. Observe the SLAM map generated by the LIDAR for any defects

Shutdown the RPi, unplug and replug the LIDAR wires, replace the wires with new ones if needed.

AMG8833 output noisy?

SSH into the RPi

1. [RPi] run ros2 run casualty_detection pubCentroid
2. Observe the image in the command line for any defects

Shutdown the RPi, unplug and replug the AMG8833 wires, replace the wires with new ones if needed.

Launcher malfunctioning?

SSH into the RPi

1. [RPi] run ros2 run turtlebot_launcher launcher_service
2. [REMOTE PC/RPi] on a separate terminal run TRIGGER

Ensure servo and flywheel operation. Check wiring connected properly and to the correct GPIO ports.

Section 5: Maintenance and Part Replacement Log

1. **OpenCR**
   Issue: We suspected an issue with the OpenCR's IMU, however, it turned out that our Dynamixels were swapped.
2. **Raspberry Pi**
   Issue: Power was not off during changing of wires connections and one of the wires dropped and shorted the Raspberry Pi.