BOLT: Bluetooth Operated Line Tracker

Embedded Systems, Cyber-Physical Systems and Robotics Project Report

Sherniyaz Nurlankul, Ansat Euler, Zahari Nedev, Shivam Bhatt,
Hejunjie Cao, Shibin Dong, YunXiang Miao
Technical University of Munich, B.Sc. Information Engineering

July 2025

Abstract

"BOLT" (Bluetooth Operated Line Tracker) is a smart robotic platform developed to automate small-scale delivery tasks in office environments. Built around the STM32F103C8T6 microcontroller, BOLT integrates line-following capability, ultrasonic obstacle avoidance, and Bluetooth manual control. The system uses embedded protocols like UART, I2C, GPIO, PWM, and timers. This report presents the design methodology, implementation challenges, performance evaluation, and future potential of the system.

1 Introduction

Manual transport of lightweight items within office spaces leads to time inefficiencies. This project aims to reduce such overhead by introducing an autonomous delivery robot – BOLT.

Project Objectives:

- Automate indoor delivery tasks using line tracking
- Enable real-time remote control via Bluetooth
- Integrate affordable embedded hardware
- Demonstrate CPS integration with feedback control

2 Methodology

The design process followed these stages:

1. **Requirement Analysis:** Identifying key functions (line tracking, obstacle avoidance, manual override, real-time data transfer, remote control).
2. **Hardware Selection:** Choosing STM32F103C8T6 for its GPIO flexibility and cost-efficiency.
3. **Modular Design:** Separating sensor, control, and actuation logic.
4. **Prototyping:** Breadboarding and assembling components for testing.
5. **Testing and Tuning:** Sensor calibration, PWM tuning, control loop iteration.

3 System Overview

BOLT is structured in five main subsystems:

1. **Perception:** Detects line and obstacles.
2. **Control:** STM32 processes sensor data and coordinates movement.
3. **Actuation:** DC motors (with driver) and SG90 servo for directional scanning.
4. **Communication:** Bluetooth UART for mobile control, I2C for OLED.
5. **Power Management:** Regulated 9V source using buck converters.

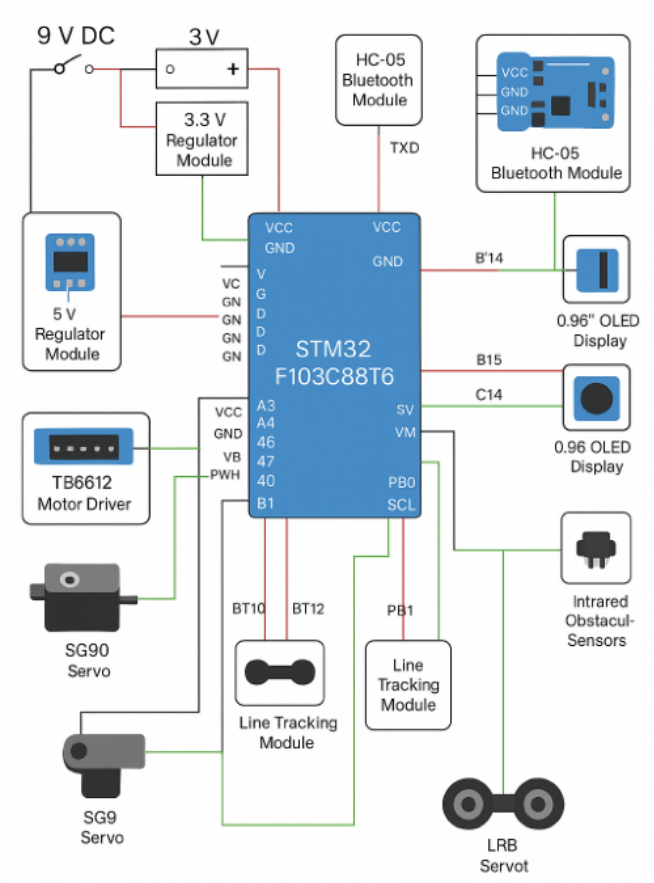

Figure 1: Functional block diagram of BOLT

4 Hardware Components

Microcontroller: STM32F103C8T6

ARM Cortex-M3 MCU, 64 KB Flash, 20 KB SRAM, UART, I2C, PWM, Timers.

Sensors and Actuators

- IR Line Sensor (YB-MVX01)
- Ultrasonic Sensor (HC-SR04)
- IR Obstacle Sensors (MH series)
- Servo Motor (SG90)
- DC Motors & TB6612 Motor Driver

Communication and Power

- HC-05 Bluetooth via UART
- ASRPro Voice Module (optional)
- OLED Display via I2C
- 9V to 5V/3.3V regulated power system

5 Bill of Materials

Component	Quantity	Purpose
STM32F103C8T6 Dev Board	1	Main controller
ST-Link V2 Programmer	1	Programming/debugging STM32
HC-05 Bluetooth Module	1	Wireless communication
HC-SR04 Ultrasonic Sensor	1	Obstacle detection
YB-MVX01 Line Tracker	1	Line following
MH IR Obstacle Sensors	2	Side obstacle detection
TB6612 Motor Driver	1	Driving 4 DC motors
0.96" OLED Display	1	Visual feedback
SG90 Servo Motor	1	Scanning ultrasonic sensor
DC Motors + Wheels	4	Locomotion
5V + 3.3V Buck Converters	1 each	Voltage regulation
Plastic Chassis	2	Structural frame
Jumper Wires, Screws, Nuts	Various	Connections and assembly

Table 1: Bill of Materials for BOLT

6 Circuit Wiring

The system integrates power management, control logic, communication, and actuation into a compact, efficient embedded circuit. Key wiring highlights are as follows:

Power Regulation

A 9V input is stepped down via two buck converters:

- **5V Regulator (U3):** Powers the TB6612 motor driver and SG90 servo.
- **3.3V Regulator (U2):** Feeds the STM32F103C8T6 MCU, Bluetooth module, OLED display, and sensors.

Microcontroller Connections (STM32F103C8T6)

- Powered by regulated 3.3V.
- GPIOs handle sensor inputs and motor control signals.

Bluetooth Communication Module – HC-05

- VCC → 3.3V
- GND → GND
- TXD → STM32 pin A10 (UART RX)
- RXD → STM32 pin A9 (UART TX)

Sensors

Ultrasonic Distance Sensor – HC-SR04

- VCC → 3.3V
- GND → GND
- TRIG → STM32 pin B14
- ECHO → STM32 pin B15

Infrared Obstacle Sensors – MH Series

- VCC → 3.3V
- GND → GND
- OUT → STM32 pins C14, C15

Line Tracking Module – YB-MVX01

- VCC → 3.3V
- GND → GND
- OUT → GPIOs BT10–BT13 on STM32

Display – OLED (0.96" I2C)

- VCC → 3.3V
- GND → GND
- SDA → STM32 pin B9
- SCL → STM32 pin B8

Motor Driver – TB6612

- VCC → 3.3V (logic)
- VM → 5V (motor power)
- GND → GND
- AIN1, AIN2 → STM32 A5, A4
- BIN1, BIN2 → STM32 A6, A7
- PWMA, PWMB → STM32 A1, A0
- AO1, AO2, BO1, BO2 → Motor outputs

Servo Motor – SG90

- VCC → 5V
- GND → GND
- PWM Signal → STM32 pin B1

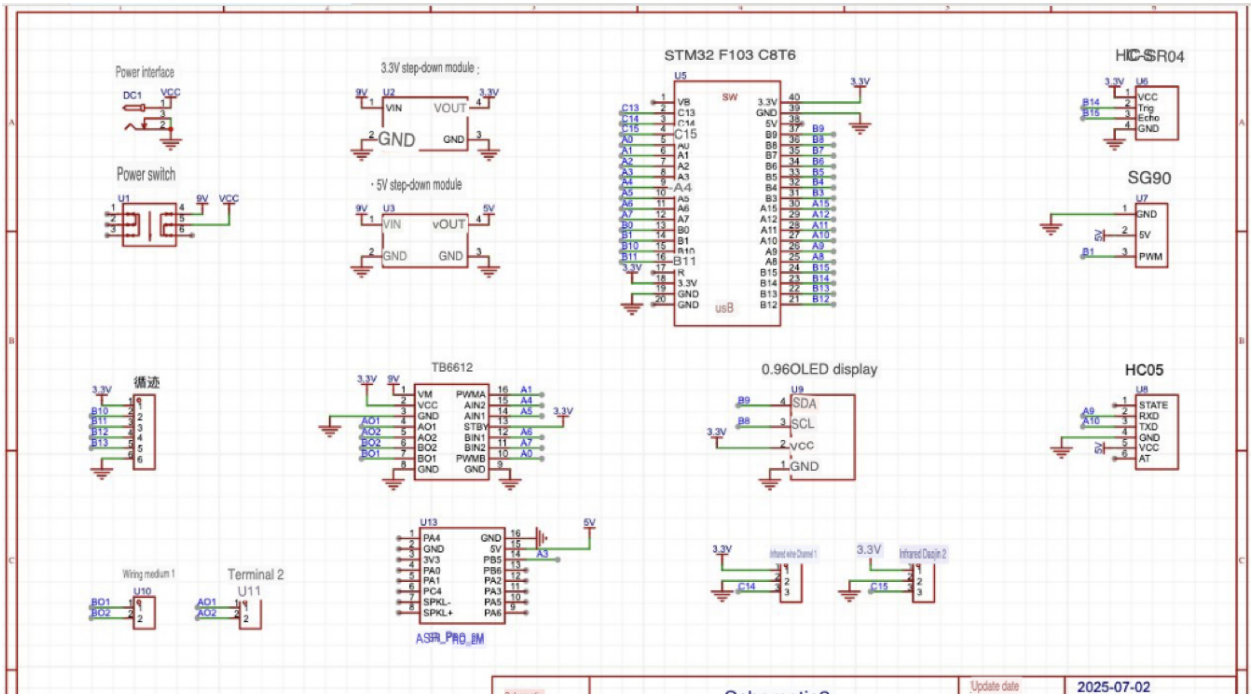

Figure 2: Wiring diagram of BOLT

7 Software Architecture

Developed in C using STM32CubeIDE.

- **Main Loop:** Prioritized line following and obstacle avoidance.
- **ISR:** UART RX and ultrasonic timer capture.
- **Modular Files:** 'motor.c', 'sensor.c', 'bluetooth.c', etc.
- **Bluetooth Parser:** Reads `MoveForward, MoveBackward, TurnLeft, TurnRight, Stop, ClockwiseRotation, AntiClockwiseRotation` from app.
- **OLED Update:** Mode and distance feedback in real-time.

Operating Modes

1. Autonomous
2. Manual via Bluetooth

8 Problems Faced

During development and testing, the following challenges were encountered:

- **Precision Turning:** Achieving accurate 90-degree turns was difficult due to motor calibration inconsistencies and variable traction.
- **Bluetooth Debugging:** Intermittent disconnections and signal delays required extensive troubleshooting.
- **Response Lag:** A slight delay between Bluetooth commands and movement affected real-time control.
- **Line Tracking Confusion:** When both the surface and line were black, the sensor often failed to differentiate.
- **Wheel Disconnections:** Mechanical instability in wheel mounting caused frequent disruptions.
- **IR Sensitivity:** Shiny or overly bright surfaces interfered with accurate IR sensor readings.

9 Limitations and Constraints

- **Power Limitations:** The 9V battery limits runtime and voltage stability under load.
- **Surface Dependency:** IR line sensors perform poorly on reflective or dark glossy floors.
- **No Feedback Loop:** Lacks encoders or closed-loop control for precise motor feedback.

10 Key Observations

- Simpler hardware is more stable when properly calibrated.
- Interrupt-driven designs enhance responsiveness compared to polling.
- Minor mechanical flaws (like loose wheels) have outsized effects on performance.
- Sensor mounting position dramatically impacts field of view and reaction time.

11 Future Improvements

To improve robustness and functionality:

- Introduce PID control with encoders for precise movement.
- Use a lithium-ion power bank for longer runtime and consistent voltage.
- Add computer vision or a camera module for dynamic navigation.
- Implement OTA Bluetooth firmware updates.
- Enable multi-agent communication using WiFi.
- Implement voice control module

12 Team Contributions

- YunXiang Miao: Hardware, testing
- Hejunjie Cao, Shibin Dong: Firmware, control logic
- Sherniyaz Nurlankul, Ansat Euler: Bluetooth interface, documentation
- Zahari Nedev, Shivam Bhatt: Project Management, documentation

13 Conclusion

BOLT showcases a practical embedded robotics application, integrating sensors, communication, and real-time control in a modular framework. Despite minor hardware and sensor challenges, it achieved its primary objectives of autonomous delivery and Bluetooth-based manual override. Future iterations can transform BOLT into a fully autonomous and intelligent indoor logistics platform.

References

- J. Xie, Jiangxie Keji, 2025. `https://jiangxiekeji.com/download.html`
- STM32CubeIDE: STMicroelectronics
- HC-05, TB6612, YB-MVX01 Datasheets
- ASRPro V2.0 Voice Module Docs