

Advanced software for connected Automated Guided Vehicle Systems Final Project

Overview:

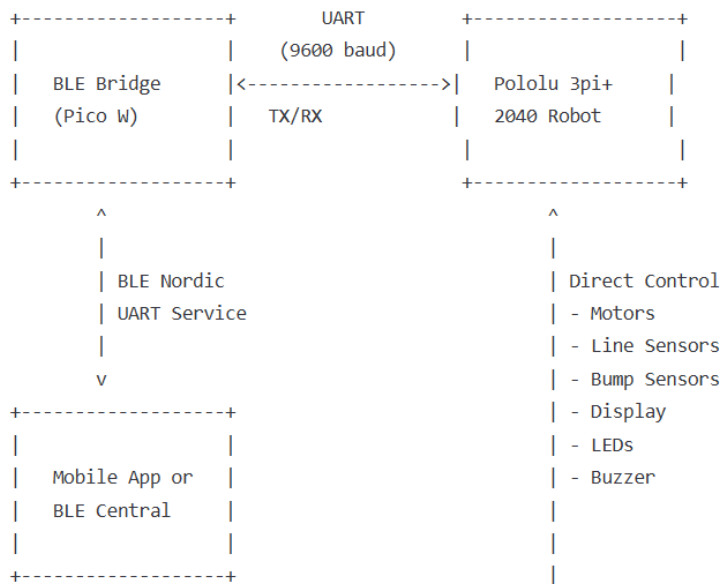
This report describes the implementation of BLE communication bridge for the Pololu 3pi+ robot, which utilizes a Raspberry Pi Pico W board as a wireless communication bridge between the Bluetooth-enabled devices and the robot.

System Architecture

The implementation follows a two-component architecture that separates wireless communication from robot control logic:

1. BLE Bridge Component (Pico W): Acts as a wireless gateway that handles Bluetooth communication with external devices.
2. Pololu 3pi+ 2040: Manages the robot's sensors, motors, and line-following algorithms, communication with the BLE bridge via UART interface.

Block diagram:



Communication Flow:

1. BLE central (in this case LightBlue mobile app) connects to the Pico W using the Nordic UART service
2. Commands sent from the central device are received by the BLE Bridge

3. The BLE bridge forwards commands to the robot via UART
4. Robot processes commands and returns responses
5. Responses are forwarded to the BLE Central through the bridge
6. Robot can send autonomous status updates to app

Implementation

The BLE Bridge is implemented on a Raspberry Pi Pico W using MicroPython and the built-in Bluetooth stack, key features are as follows:

- Nordic UART service: implements the standard BLE UART service with TX and RX characteristics
- Advertising: Advertises as “PICO_ROBOT_BRIDGE_2” for easy discovery
- UART communication: Communicates with the robot using UART at 9600 baud
- Message forwarding: Bidirectional message forwarding between BLE and UART
- Status monitoring: Regular heartbeat messages and status updates
- Visual feedback: LED indicators for connection status and activity

The robot control code runs on the robot and handles:

- Movement controls: Forward, backward, left, right, and stop commands
- Line following: Calibration and PID-based line following algorithm
- Collision detection: Using robot's bump sensors
- Status reporting: Battery level, mode, speed, sensor states
- Visual feedback: display updates showing current state