The German University in Cairo (GUC)

Faculty of Media Engineering and Technology
Computer Science and Engineering

Embedded System Architecture - CSEN 701

Parking Assistance System

Team Members:
Mahmoud Ahmed, #55-1653
Rola Ibrahim, #55-5230
Rubina Sameh, #55-2471
Sarah Fahmy, #55-4311
Michael Maged, #55-0615
Hagar Youssef, #55-2205
Tasneem Hesham, #55-1026

Team Number: Team #27

Under the supervision of:
Dr. Eng. Catherine M. Elias

Winter 2024

Contents Overview

This report includes the following sections:

— End of Contents —

1 Explanatory Notes for Each Subsystem

This section provides an explanation of the subsystems of the Parking Assistance System, along with their components and purposes.

1. IDLE Subsystem

Purpose: Initializes and powers up all sensors in the system.
Actions:

- Entry point into the system.
- Ensures all sensors (Infrared, Ultrasonic, and Sharp) are ready for operation.

2. Scanning Subsystem

Components:

- **Infrared Sensor:** Reads obstacle proximity for short-range detection.
- **Ultrasonic Sensor + Servo Motor:** Scans the surroundings by rotating the ultrasonic sensor to cover a wider area.
- **Sharp Sensor:** Performs medium-range obstacle detection.

Purpose: Collects distance data from all sensors for real-time analysis.

3. Detecting Subsystem

Components:

- **Processing Unit:** Processes the data collected from the sensors.

Purpose:

- Evaluates the collected data.
- Checks if the car's distance to obstacles falls within a critical or dangerous range.

4. Stop Subsystem

Components:

- **Buzzer:** Alerts the user when the distance is below a dangerous threshold.
- **DC Motor:** Stops the vehicle's motion.

Purpose:

- Stops the vehicle.
- Triggers a buzzer alert when an obstacle is too close.

5. Move Subsystem

Components:

- **DC Motor:** Controls vehicle motion.

Purpose:

- Activates or deactivates the DC motor to move the vehicle forward or backward, depending on the data from the detecting subsystem.

2 Component List and Diagrams

This section provides details of the hardware components used in the Parking Assistance System, along with their drivers, descriptions, and datasheet links.

1. DC Motor

Driver: dc-motor.c and dc-motor.h
Description:

- Simulates vehicle motion by moving forward or backward.

Datasheet Link: https://example.com/DCMotorDC Motor Datasheet

2. H-Bridge

Description:

- Enables bidirectional control of the DC motor.
- Allows forward and backward motion by managing motor polarity.

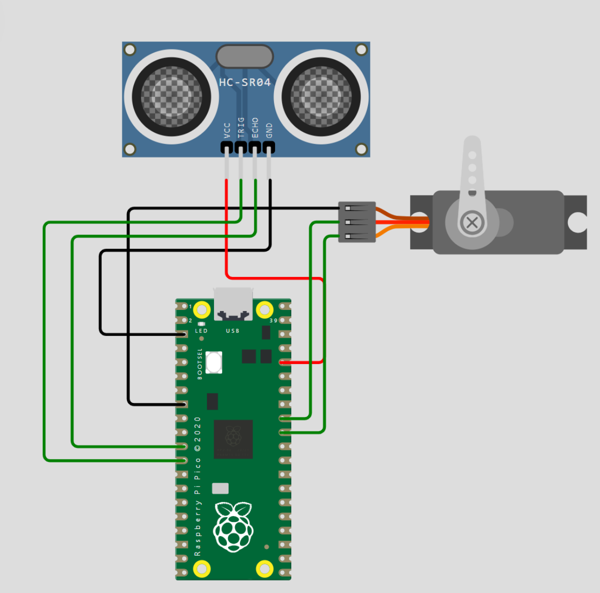
Datasheet Link: https://example.com/L298NL298N Datasheet

3. Servo Motor

Driver: `servo-control.c and servo-control.h`
Description:

- Rotates the ultrasonic sensor to scan a wider area.
- Operates within a range of 0-180 degrees for precise control.

Datasheet Link: http://www.ee.ic.ac.uk/pcheung/teaching/DE1-EE/stores/sg90-datasheet.pdf
Connection Diagram:

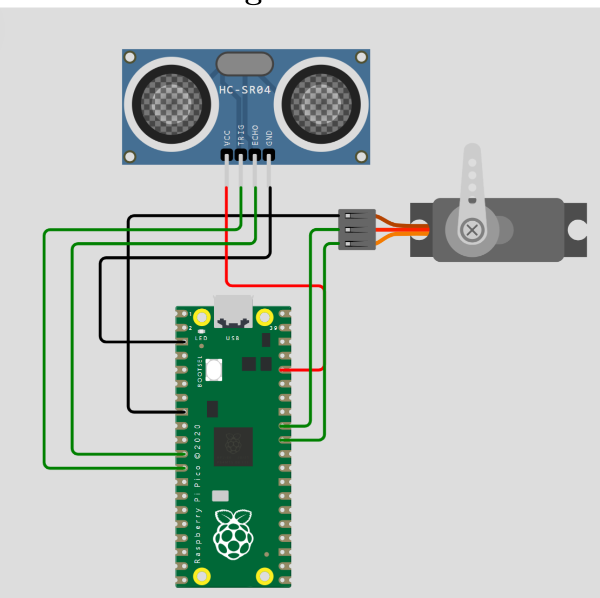


4. Ultrasonic Sensor

Driver: `ultraSonic-sensor.h` and `ultraSonic-sensor.c`
Description:

- Measures distance to nearby obstacles.
- Provides real-time obstacle detection data.

Datasheet Link: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
Connection Diagram:



5. Sharp Infrared Senso

Driver: sharp-infra-sensor.c and sharp-infra-sensor.h
Description:

- Detects short-range obstacles by measuring reflected infrared signals.
- Outputs analog data to indicate obstacle proximity.

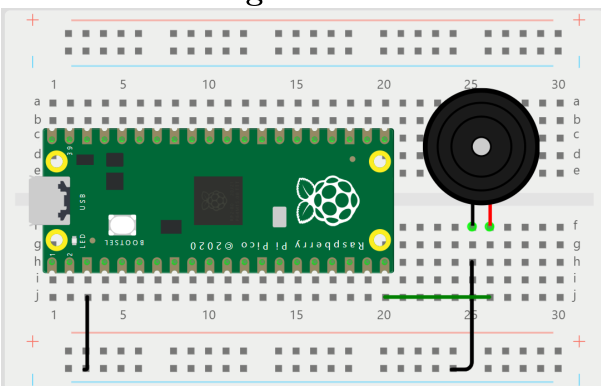
Datasheet Link: https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk$_e.pdf$

6. Buzzer

Driver:buzzer-sfm.c and buzzer-sfm.h
Description:

- Provides auditory feedback when an obstacle is detected.
- Activated when the vehicle is within a critical range of an object.

Datasheet Link: https://www.farnell.com/datasheets/2171929.pdfBuzzer Datasheet
Connection Diagram:

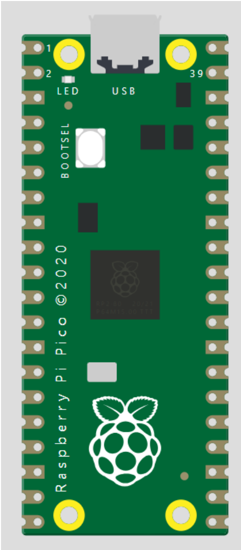


7. Raspberry Pi Pico Microcontroller

Description:

- Acts as the central control unit for the system, processing data and managing connected sensors and actuators.
- Provides GPIO pins for interfacing with components like the Sharp Infrared Sensor, ultrasonic sensor, and buzzer.
- Supports multiple programming environments (C, MicroPython, Arduino IDE).

Datasheet Link: https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf
Connection Diagram:


1
2
39
LED
USB
BOOTSEL
Raspberry Pi Pico ©2020

3 Updated State Flow Diagram

This section presents the updated state flow diagram of the Parking Assistance System. The diagram reflects the latest changes and additional adjustments made during the project development.

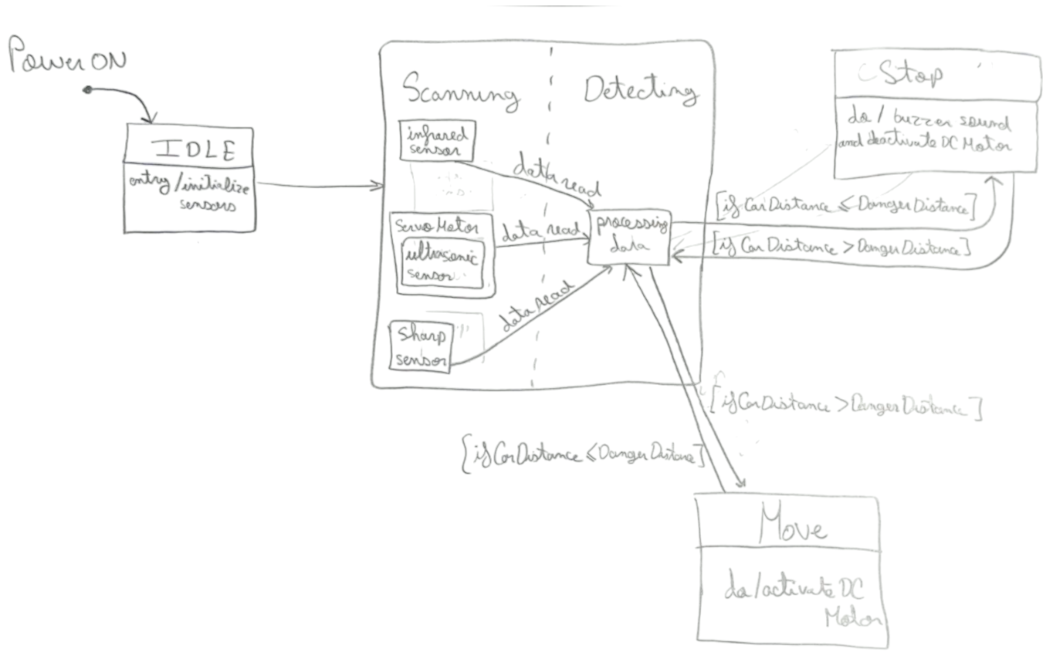


Figure 6: Updated State Flow Diagram.

The state flow diagram illustrates the transitions between different states of the system, such as:

- **Idle State:** The system awaits an input or sensor activation.
- **Obstacle Detection State:** The ultrasonic and infrared sensors detect objects in proximity.
- **Alert State:** The buzzer is triggered when obstacles are within a critical range.
- **Servo Rotation State:** The servo motor rotates to expand the scanning area.
- **Movement State:** The DC motor moves the vehicle forward or backward based on input feedback.

Transitions between these states are managed by sensor inputs, time intervals, and predefined threshold values.

4 Concurrency and Logic Explanation

The Parking Assistance System ensures real-time performance by implementing concurrency across its subsystems, as illustrated in the state flow diagram. Concurrency is managed through the following mechanisms:

Sensor Data Acquisition

Sensor readings from the ultrasonic sensor, Sharp infrared sensor, and the servo motor-mounted ultrasonic sensor are processed concurrently to maintain responsiveness:

- **Ultrasonic Sensor and Servo Motor**: The servo motor rotates the ultrasonic sensor periodically to scan the surroundings while the ultrasonic sensor continuously collects distance data.
- **Sharp Infrared Sensor**: Operates in parallel, providing medium-range obstacle detection.

Data from all sensors is continuously read and processed by the **Detecting Subsystem**, ensuring up-to-date obstacle information.

State Transitions and Decision Making

The system transitions between states based on real-time sensor data. Key decisions are handled by the **Detecting Subsystem**, which evaluates sensor inputs:

- If the car's distance to an obstacle is below the danger threshold, the system transitions to the **Stop Subsystem**, triggering the buzzer and halting the DC motor.
- If the obstacle is no longer within the danger range, the system transitions to the **Move Subsystem**, reactivating the DC motor.

Concurrent Component Control

The system ensures that components operate simultaneously to maintain performance:

- **Buzzer and DC Motor**: The buzzer alerts when obstacles are detected while the motor stops. Both are activated or deactivated based on the current state.
- **Servo Motor and Sensors**: The servo motor scans the surroundings, with the ultrasonic and Sharp infrared sensors operating independently but feeding data to the same processing logic.

Communication and Control

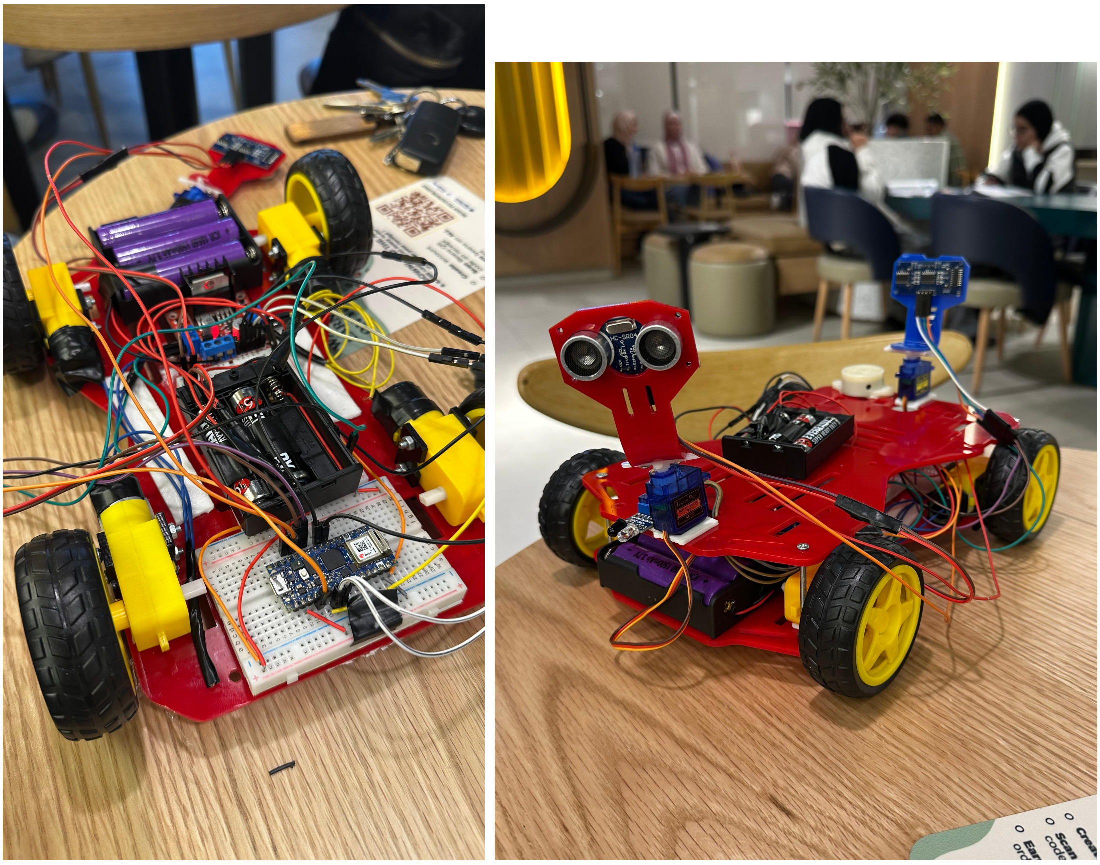
Shared variables and flags are used for communication between subsystems. For example:

- The **Detecting Subsystem** sets a flag when an obstacle is detected within the danger range, signaling the **Stop Subsystem**.
- When the obstacle is cleared, the flag resets, signaling the **Move Subsystem** to activate the DC motor.

This distributed logic ensures that each subsystem can function independently while maintaining real-time coordination.

5 Final Hardware Design

The final hardware design integrates all components of the Parking Assistance System. The figure below shows the complete hardware setup, including connections and placements.



Final Hardware Design.

The design includes the following components:

- Ultrasonic sensor mounted on a servo motor for scanning.
- Sharp IR sensor for medium-range obstacle detection.
- DC motor controlled via an H-Bridge for simulated movement.
- Buzzer for auditory feedback.
- Microcontroller as the central control unit.

All components are connected and powered through a regulated power supply to ensure smooth operation.

6 GitHub Project Link

The complete source code, diagrams, and additional documentation for the Parking Assistance System can be accessed at the following GitHub repository:

https://github.com/mahmoudd2/Embedded-Sys-Parking-Assistance-Car

The repository includes:

- Complete source code for the project.
- Connection diagrams and state flow diagrams.
- Instructions for replicating the system.