German International University
Faculty of Engineering
Major of Automation and Control

Wireless Sensor Network for Robotic Fleet Coordination

Introduction to Computer Networks
Project Report

Team Members

Hassan Yousef	13006567
Pierre George Boshra	13007351
Mohamed Walid	13006513
Abdelhamid ElSharnouby	13006294
Khaled Khaled	14001048
Mahmoud Nasser	13006342

Supervised by
Dr. Yasmine Zaghloul
TA: Eng. Omar Hemeda

Winter Semester 2025
December 2025

Contents

1 Week 1: Research & Setup

extitNovember 17-23, 2025

1.1 Research Phase

To establish a solid foundation for our project, we conducted comprehensive research utilizing multiple academic and expert resources. Our research methodology included:

- **Academic Database Search:** We performed extensive literature review using the Egyptian Knowledge Bank (EKB) database, searching for relevant research papers and technical documentation on wireless sensor networks, ESP-NOW protocol, mesh networking architectures, and robotic coordination systems.

- **Expert Consultation:** We consulted with Professor Yasmine Zaghloul, our course instructor, to discuss the project scope, technical feasibility, and recommended implementation approaches. Her guidance helped us refine our project objectives and identify potential challenges.

- **Technical Advisory:** We met with TA Omar Hemeda to discuss the practical aspects of the project implementation, including hardware selection, programming methodologies, and project timeline management.

- **Laboratory Support:** We worked closely with Lab Engineer Amr Khaled to understand the available laboratory resources, safety protocols, and equipment handling procedures necessary for our project development.

1.2 ESP-NOW Protocol Learning

To gain practical understanding of the ESP-NOW protocol, we studied various resources with particular focus on the tutorial video "*ESPNOW for beginners! #ESP32 #ESP8266*" available at:

<div align="center">

https://www.youtube.com/watch?v=YdiOM3Xd_vs

</div>

This resource provided valuable insights into:

- The fundamentals of ESP-NOW as a fast, connectionless peer-to-peer communication protocol

- Setting up ESP-NOW connections between multiple ESP32 devices

- Broadcasting and receiving data packets

- Implementing callback functions for data handling

- Debugging common ESP-NOW implementation issues

1.3   Hardware Setup

We successfully configured the development environment for the ESP32 microcontrollers using the Arduino IDE. The setup process included:

- Installing the Arduino IDE (version 2.x)

- Adding ESP32 board support through the Boards Manager

- Installing necessary libraries (ESP-NOW, WiFi, sensor libraries)

- Testing basic functionality with simple sketch uploads

- Verifying serial communication for debugging purposes

1.4   Project Architecture Plan

After thorough analysis and team discussions, we finalized our system architecture as follows:

> **Information**
>
> **Hardware Configuration:**
>
> - **3 ESP32 Microcontrollers:** Serve as the core processing units
>
> - **2 Remote-Controlled Toy Vehicles:** Provide the mobile platform for our robotic agents
>
> - **Master-Slave Architecture:** Two ESP32 units mounted on vehicles act as slave nodes, while one ESP32 connected to PC via USB serves as the master coordinator

> **Note**
>
> **Sensor Configuration per Vehicle:**
>
> - **Ultrasonic Sensor (HC-SR04):** Detects obstacles and measures distances to objects in the robot's path
>
> - **Color Sensor (TCS3200/TCS34725):** Identifies the unique color of squares in the mapping area, enabling precise localization and obstacle mapping

1.5   System Design Specifications

1.5.1   Network Topology

The system implements a mesh network topology where all three ESP32 devices maintain peer-to-peer connections. This ensures:

- Redundant communication paths

- Distributed data processing

- Fault tolerance in case of node failure

- Real-time data sharing across all nodes

1.5.2   Mapping Area Design

The operational environment consists of a grid-based area with the following characteristics:

- Divided into uniquely colored squares

- Each square color serves as a unique identifier for localization

- Obstacles can be placed dynamically within the grid

- The grid provides a structured environment for testing coordination algorithms

1.5.3   Data Flow Architecture

1. **Slave Nodes (Mobile Robots):**

   - Continuously scan environment using ultrasonic sensors
   - Read ground color using color sensors
   - Broadcast obstacle and position data via ESP-NOW
   - Receive and process data from peer nodes
   - Update internal map representation

2. **Master Node (PC-Connected):**

   - Aggregates data from all slave nodes
   - Maintains global map of the environment
   - Provides visualization interface through serial communication
   - Coordinates robot movements (in later phases)
   - Logs data for analysis and documentation

1.6   Week 1 Deliverables

✓ Completed research on ESP-NOW protocol and mesh networking

✓ Set up Arduino IDE development environment

✓ Assembled 2-3 basic robot chassis with remote control capability

✓ Finalized project architecture and component selection

✓ Established communication with project advisors

✓ Created project timeline and milestone definitions

1.7   Challenges and Solutions

- **Challenge:** Understanding ESP-NOW protocol differences from traditional WiFi

- **Solution:** Studied multiple tutorials and documentation, conducted simple experiments

- **Challenge:** Selecting appropriate sensors within budget constraints

- **Solution:** Consulted with lab engineer and utilized available laboratory equipment

Additional Notes and Observations

2   Week 2: Core Module Development

extitNovember 24-30, 2025

2.1   Objective

Program one robot to read its ultrasonic sensor and broadcast obstacle data via ESP-NOW protocol.

2.2   Implementation Overview

[To be completed during Week 2]

2.3   Ultrasonic Sensor Integration

2.3.1   Hardware Connection

[Document sensor wiring and pin configuration]

2.3.2   Sensor Reading Code

Ultrasonic Sensor Data Acquisition

```
1  // Placeholder for ultrasonic sensor code
2  // Include: HC-SR04 initialization, distance measurement
       function,
3  // noise filtering, and calibration routines
```

2.4   ESP-NOW Broadcasting Implementation

2.4.1   Protocol Initialization

ESP-NOW Setup and Configuration

```
1  // Placeholder for ESP-NOW initialization code
2  // Include: WiFi mode setup, ESP-NOW init, peer registration,
3  // MAC address handling, and callback function definitions
```

2.4.2 Data Structure Design

Obstacle Data Structure

```
1  // Placeholder for data structure definitions
2  // Include: Struct for obstacle data, position information,
3  // timestamp, robot ID, and data packaging functions
```

2.5 Main Program Logic

Main Loop - Single Robot Broadcast

```
1  // Placeholder for main program code
2  // Include: Setup function, main loop with sensor reading,
3  // data broadcasting, timing control, and debug output
```

2.6 Testing and Validation

2.6.1 Test Cases

[Document test scenarios and results]

2.6.2   Observed Behavior

[Record actual system behavior during testing]

2.7   Week 2 Deliverables

☐ Functional ultrasonic sensor reading module

☐ Working ESP-NOW broadcast implementation

☐ Tested single-robot obstacle detection system

☐ Documented code with comments

☐ Performance metrics and range testing results

2.8   Challenges and Solutions

[Document any technical challenges encountered and their solutions]

Additional Notes and Observations

3 Week 3: Integration

extitDecember 1-7, 2025

3.1 Objective

Program a second robot to receive broadcasted data and implement the logic: "If I receive an obstacle alert, add it to my internal map and re-broadcast."

3.2 Implementation Overview

[To be completed during Week 3]

3.3 Data Reception Module

3.3.1 Receiver Callback Function

ESP-NOW Data Reception Handler

```
1  // Placeholder for ESP-NOW receive callback
2  // Include: Callback function to handle incoming data,
3  // data validation, parsing, and acknowledgment
```

3.4 Internal Map Implementation

3.4.1 Map Data Structure

Internal Map Structure

```
1  // Placeholder for internal map implementation
2  // Include: Map data structure (array/linked list),
3  // obstacle storage, coordinate system, map size definitions
```

3.4.2   Map Update Logic

Map Update and Management

```
1  // Placeholder for map update functions
2  // Include: Add obstacle function, update existing entries,
3  // remove outdated data, map query functions
```

3.5   Re-Broadcasting Logic

3.5.1   Selective Re-Broadcasting

Data Re-Broadcast Implementation

```
1  // Placeholder for re-broadcast logic
2  // Include: Function to re-broadcast received data,
3  // duplicate detection, broadcast management, TTL handling
```

3.6   Multi-Robot Coordination

3.6.1   Second Robot Main Program

Second Robot Complete Program

```
1  // Placeholder for second robot program
2  // Include: Setup with both TX and RX capabilities,
3  // main loop handling both sensing and receiving,
4  // integrated map management
```

3.7 Communication Protocol

3.7.1 Message Format

[Document the message structure and protocol specifications]

3.7.2 Collision Avoidance

[Document strategy to prevent network congestion]

3.8 Testing Multi-Robot System

3.8.1 Integration Test Scenarios

[Document test cases with two robots]

3.8.2 Map Consistency Verification

[Document how you verified both robots maintain consistent maps]

3.9 Week 3 Deliverables

☐ Working two-robot communication system

☐ Internal map implementation on both robots

☐ Functional re-broadcast mechanism

☐ Duplicate detection and handling

☐ Synchronized map between robots

☐ Communication protocol documentation

3.10 Challenges and Solutions

[Document any technical challenges encountered and their solutions]

Additional Notes and Observations

4 Week 4: Enhancement

extitDecember 8-14, 2025

4.1 Objective

Implement a simple "leader-follower" or "swarm" algorithm where robots try to maintain formation based on shared location data.

4.2 Implementation Overview

[To be completed during Week 4]

4.3 Algorithm Selection

4.3.1 Chosen Approach

[Document whether you chose leader-follower or swarm, and justify the choice]

4.4 Position Tracking System

4.4.1 Color-Based Localization

Color Sensor Integration

```
1  // Placeholder for color sensor code
2  // Include: Color sensor initialization, RGB reading,
3  // color identification, position mapping from color
```

4.4.2 Position Broadcasting

Location Data Sharing

```
1  // Placeholder for position broadcast
2  // Include: Extended data structure with position info,
3  // regular position updates, coordinate transformation
```

4.5 Coordination Algorithm Implementation

4.5.1 Leader-Follower Logic (if applicable)

Leader-Follower Algorithm

```
1  // Placeholder for leader-follower implementation
2  // Include: Leader selection, follower behavior,
3  // formation maintenance, distance control
```

4.5.2 Swarm Behavior Logic (if applicable)

Swarm Algorithm

```
1  // Placeholder for swarm implementation
2  // Include: Neighbor detection, flocking rules,
3  // separation/alignment/cohesion behaviors
```

4.6 Motor Control Integration

4.6.1 Movement Control System

Robot Movement Control

```
1  // Placeholder for motor control code
2  // Include: Motor driver initialization, movement functions,
3  // speed control, direction control, formation adjustment
```

4.7 Formation Maintenance

4.7.1 Distance Calculation

[Document how robots calculate relative positions]

4.7.2   Adjustment Strategy

[Explain how robots adjust their positions to maintain formation]

4.8   Master Node Coordination

Master Node Control Logic

```
1  // Placeholder for master node coordination
2  // Include: Global position tracking, command generation,
3  // formation monitoring, visualization output
```

4.9   Testing Coordination System

4.9.1   Formation Test Scenarios

[Document test cases for formation maintenance]

4.9.2   Performance Metrics

[Record formation accuracy, response time, stability]

4.10   Week 4 Deliverables

☐ Functional coordination algorithm implementation

☐ Color-based position tracking system

☐ Motor control integration

☐ Formation maintenance capability

☐ Master node coordination interface

☐ Algorithm documentation and parameters

4.11   Challenges and Solutions

[Document any technical challenges encountered and their solutions]

Additional Notes and Observations

5   Week 5: Final Testing & Documentation

extitDecember 15-22, 2025

5.1   Objective

Demonstrate the robots coordinating to map an area or avoid a new obstacle. Document the mesh protocol and finalize project deliverables.

5.2   Final System Integration

[To be completed during Week 5]

5.3   Comprehensive Testing

5.3.1   Test Scenario 1: Area Mapping

> **Note**
>
> **Test Description:**
> *[Describe the area mapping test scenario in detail]*
>
> **Expected Behavior:**
> *[Describe expected system behavior]*
>
> **Actual Results:**
> *[Record actual test results]*
>
> **Observations:**
> *[Note any interesting observations]*

5.3.2   Test Scenario 2: Dynamic Obstacle Avoidance

> **Note**
>
> **Test Description:**
> *[Describe the dynamic obstacle avoidance test]*
>
> **Expected Behavior:**
> *[Describe expected system behavior]*

Actual Results:
[Record actual test results]

Observations:
[Note any interesting observations]

5.3.3   Test Scenario 3: Coordination Under Communication Stress

> **Note**
>
> **Test Description:**
> *[Describe stress test with multiple obstacles and commands]*
>
> **Expected Behavior:**
> *[Describe expected system behavior]*
>
> **Actual Results:**
> *[Record actual test results]*
>
> **Observations:**
> *[Note any interesting observations]*

5.4   Mesh Protocol Documentation

5.4.1   Protocol Architecture

> **Information**
>
> **Network Topology:**
> *[Describe the final network topology implementation]*
>
> **Node Roles:**
>
> - Master Node: *[Describe responsibilities]*
>
> - Slave Nodes: *[Describe responsibilities]*
>
> **Communication Patterns:**

> *[Describe how nodes communicate with each other]*

5.4.2　Message Types and Formats

Protocol Message Definitions

```
1  // Final protocol message structures
2  // Include: All message types used in the system,
3  // data structures, packet formats, headers
```

5.4.3　Protocol State Machine

[Document the protocol state machine or flow diagram]

5.4.4　Error Handling and Recovery

[Document how the system handles communication errors, node failures, etc.]

5.5　Performance Analysis

5.5.1　Quantitative Metrics

Metric	Measured Value	Notes
Communication Range		
Message Latency		
Packet Loss Rate		
Sensor Accuracy		
Map Update Frequency		
Formation Error		
Battery Life		

5.5.2 Qualitative Assessment

[Provide qualitative assessment of system performance]

5.6 Demonstration Results

5.6.1 Video Documentation

[Reference video demonstrations, include timestamps of key features]

5.6.2 Photographic Documentation

[Include or reference photographs of the system in operation]

5.7 Complete System Code

Master Node - Final Code

```
1  // Final master node code
2  // Complete, documented, and tested implementation
```

Slave Node - Final Code

```
1  // Final slave node code
2  // Complete, documented, and tested implementation
```

5.8   Project Outcomes

5.8.1   Achievements

- *[List major achievements]*

5.8.2   Limitations

- *[List system limitations and constraints]*

5.8.3   Future Improvements

- *[Suggest potential enhancements]*

5.9   Lessons Learned

5.9.1   Technical Insights

[Document key technical lessons learned]

5.9.2   Project Management

[Document lessons about teamwork, timeline management, etc.]

5.10   Week 5 Deliverables

☐ Complete system demonstration

☐ Final tested code for all nodes

☐ Comprehensive protocol documentation

☐ Test results and performance metrics

☐ Video and photo documentation

☐ Final project report

☐ Presentation materials

Additional Notes and Observations

6   Conclusion

6.1   Project Summary

[To be completed after Week 5 - Provide comprehensive project summary]

6.2   Technical Contributions

[Highlight the technical contributions and innovations of your project]

6.3   Team Reflection

[Each team member's reflection on their contribution and learning experience]

6.4   Acknowledgments

We would like to express our sincere gratitude to:

- **Dr. Yasmine Zaghloul** for her invaluable guidance, expertise, and continuous support throughout this project

- **Eng. Omar Hemeda** for his technical assistance and practical advice during development

- **Eng. Amr Khaled** for providing laboratory resources and ensuring safe equipment usage

- **German International University** for providing the facilities and resources necessary for this project

7   References

1. ESP-NOW Protocol Documentation. Espressif Systems. `https://www.espressif.com/`

2. "ESPNOW for beginners! #ESP32 #ESP8266" Tutorial Video. `https://www.youtube.com/watch?v=Ydi0M3Xd_vs`

3. Egyptian Knowledge Bank (EKB). Accessed for wireless sensor network research. `https://www.ekb.eg/`

4. Arduino ESP32 Documentation. `https://docs.arduino.cc/`

5. *[Add additional references as needed during the project]*

A   Complete Source Code

[Include complete, final source code listings]

B   Circuit Diagrams

[Include detailed circuit diagrams and wiring schematics]

C   Component Specifications

[Include datasheets or specifications for major components]

D   Additional Test Results

[Include supplementary test data and analysis]