CSC2106 Internet of Things: Protocols and Networks [2023/24 T2]

T29: Project Proposal

Name	Student ID
Abdul Halim Bin Abdul Rahim	2201627 / 2837343A
Mirza Anaqi Bin Muhammad Haizan	2200981 / 2837305M
Ang Guo Zheng Frederick	2203322 / 2837391A
Liew Jun Wei	2200829 / 2837290L
Wong Kar Long Keagan	2200517 / 2837260W

Date of Submission: 19 January 2024

Smart Warehouse Inventory Tracker

1. Executive Summary

The Smart Tracker project focuses on developing a comprehensive Smart Warehouse Inventory Tracker for indoor environments. The system aims to provide precise location information, catering to applications such as indoor asset tracking in warehouses. Utilising communication protocols, such as Bluetooth Low Energy (BLE) and/or WiFi, and microcontroller units (MCUs) such as M5StickC Plus, Super Mini ESP32-C3, and Raspberry Pi Pico, the project aims to demonstrate real-time tracking capabilities.

2. Problem Statement

The Smart Warehouse Inventory Tracker project aims to rectify the inherent limitations of contemporary IoT sensors, particularly concerning their efficacy in detecting objects in a vertical orientation within the intricate dimensions of a 3D indoor space. This deficiency becomes particularly pronounced and operationally challenging in vertical tracking scenarios, notably when applied to the precision monitoring of tools within a workshop environment. Recognising the urgency to overcome these limitations, the project introduces an innovative solution characterised by an augmentation of the sensor network, strategically increasing the number of nodes. This augmentation aims not only to bolster tracking accuracy but also to hone in on the nuanced demands of vertical tracking within the specific context of workshop tools.

Key Points:

- Vertical Tracking Limitation: The prevailing generation of IoT sensors grapples with challenges in effectively detecting objects oriented vertically within the intricate spatial landscape of a 3D indoor environment. This limitation poses a substantial impediment to achieving precise and reliable tracking outcomes.
- Enhanced Accuracy: The proposed solution unfolds as a meticulous strategy involving a discerning increase in the number of nodes within the sensor network. This deliberate augmentation is envisaged as a cornerstone for achieving a paradigm shift in tracking accuracy, transcending the constraints imposed by current sensor limitations.
- Workshop Tool Tracking: A focal point of this initiative is the tailored attention given to the tracking of tools within the dynamic setting of a workshop environment. Recognising the unique challenges presented by workshop scenarios, the solution aims to elevate the accuracy and reliability of tracking specifically for tools, thereby addressing a critical operational need.
- Scalability Consideration: The design philosophy underpinning the proposed solution is inherently forward-looking, with scalability positioned as a paramount consideration. Beyond immediate requirements, the solution is meticulously crafted to seamlessly expand and adapt to the evolving demands of future expansions, ensuring sustained relevance and operational efficiency.

3. Project Objectives

The primary objectives of the Smart Tracker project include:

- Develop an indoor asset tracking system using BLE and WiFi, potentially in a hybrid configuration.
- Explore and integrate indoor triangulation algorithms to enhance location precision.
- Deploy robust security measures to protect sensitive location data.

4. Justification of Communication Protocols

4.1 Indoor Asset Tracking

4.1.1 BLE and WiFi

- Precision: BLE and WiFi offer high precision for indoor asset tracking, ensuring accurate location data within confined spaces.
- Power Efficiency: BLE's energy efficiency is crucial for prolonged battery life in continuous indoor tracking scenarios.
- Widespread Compatibility: WiFi's availability and cost-effectiveness suit indoor environments with existing infrastructure.

4.1.2 Hybrid Approach

- Redundancy: Combining BLE and WiFi ensures continuous tracking even in the presence of signal interference or limitations of one protocol.
- Increased Coverage: Hybrid solutions offer broader coverage, mitigating potential blind spots.

5. Implementation Plan

The implementation plan involves:

- Integrating communication protocols into selected MCUs: M5StickC Plus, Super Mini ESP32-C3, and Raspberry Pi Pico.
- Developing indoor triangulation algorithms to enhance location precision.
- Implementing robust security measures to protect tracked data.
- Conducting prototype demonstrations to validate the system's functionality.

6. Programming Languages and Software Tools

6.1 Programming Languages

- M5StickC Plus, Super Mini ESP32-C3: C/C++ with the Arduino IDE.
  - Justification: Arduino IDE provides a user-friendly environment for rapid development and prototyping. C/C++ allows low-level programming, crucial for MCU applications.
- Raspberry Pi Pico: MicroPython or C/C++ with the official Raspberry Pi Pico SDK.
  - Justification: MicroPython offers simplicity for rapid development, while C/C++ with the SDK provides control over hardware-level interactions.

6.2 Software Tools

- Arduino IDE: Development and programming of MCUs, including M5StickC Plus and Super Mini ESP32-C3.
  - Justification: Simplifies the coding process, facilitates code upload, and supports libraries for BLE and WiFi functionalities.
- Thonny (for Raspberry Pi Pico): Programming and debugging for Raspberry Pi Pico.
  - Justification: Thonny is a beginner-friendly Python IDE supporting MicroPython, suitable for Raspberry Pi Pico development.
- PlatformIO (Optional): Cross-platform code building, debugging, and library management.
  - Justification: Enhances development, especially in complex projects involving multiple MCUs, supporting various platforms.
- Wireshark (for network analysis): Analysing network traffic and ensuring the security of communication protocols.
  - Justification: Used for debugging, analysing data packets, and ensuring the proper functioning and security of communication protocols.

6.3 Security Measures

- Secure Protocols: Implement secure versions of BLE, WiFi or other communication protocols (BLE Secure Connections, WPA3 for WiFi) to safeguard sensitive tracking data.
- Device Authentication: Implement device authentication mechanisms to ensure that only authorised devices can participate in the tracking system.
- Data Encryption: Encrypt tracked data to prevent unauthorised access and manipulation, ensuring the privacy and security of location information.

7. Core Functionality

- Location Tracking: Provide accurate location tracking in an indoor environment. Ensure performance in case of signal interference from physical obstructions
- Mesh Network: Use mesh networking libraries to enhance communication reliability.
- Define Predefined Areas: Allow users to set up predefined areas on a map using your application or web interface. These areas could be represented as geofences, which are virtual perimeters defined by geographical coordinates.
- GPS Data Collection: Continuously collect GPS data from the Smart Tracker device. This data should include the current location coordinates (latitude and longitude).

8. Possible Enhancements

- Power consumption: Focus on optimising power consumption to extend the battery life of your tracking devices, ensuring they can operate for extended periods without frequent recharging or battery replacement.
- Tamper Detection: Implement tamper detection features to alert users in case the tracker is tampered with, enhancing security and theft prevention.
- User Behaviour Analysis: Implement algorithms to analyse user behaviour based on tracking data, providing insights into asset movement patterns, crowded areas, and popular zones.
- Data Visualisation: Develop a user-friendly interface that visualises asset location data in real time, allowing users to monitor and analyse the movement of assets within the tracked area.

9. Work Division

- Project Manager (PM) - Halim:
  - Phase 1 (Week 1-5): Oversee the Literature Review phase, ensuring thorough research on current solutions and justifications for project decisions.
  - Phase 2 (Week 6): Supervise the Hardware Setup phase, ensuring the procurement and setup of selected MCUs and hardware compatibility verification.
  - Phase 3 (Week 7-10): Coordinate the Software Development phase, overseeing the development of MCU-specific code, indoor triangulation algorithms, and security measures.
  - Phase 4 (Week 10-12): Manage the Testing and Debugging phase, coordinating unit testing, system testing, and refining code as needed.
  - Phase 5 (Week 12-13): Oversee the Prototype Demonstration phase, ensuring the showcase of the working prototype with indoor tracking scenarios.

- Communication and Protocols Specialist (CPS): Mirza
  - Focus on integrating communication protocols into selected MCUs, ensuring seamless functionality and conducting tests to ensure throughput and bandwidth are optimised.

- Security and Data Encryption Expert (SDEE): Frederick
  - Lead the implementation of robust security measures, including device authentication mechanisms and data encryption protocols.

- Location Tracking and Algorithm Developer (LTAD): Keagan
  - Develop indoor triangulation algorithms to enhance location precision, ensuring accurate location tracking within the indoor environment.

- User Interface and Visualisation Developer (UIVD): Jun Wei
  - Core Functionality: Develop a user-friendly interface for data visualisation, focusing on real-time asset location tracking and user behaviour analysis.
  - Possible Enhancements: Contribute to the implementation of power consumption optimisation and tamper detection features in collaboration with other team members.

10. Project Timeline

The project will be divided into several phases:

- Phase 1 (Week 1-5):
  - Literature Review – Research current solutions and justify decisions made for the project.
- Phase 2 (Week 6):
  - Hardware Setup – Procure and set up selected MCUs, and verify hardware compatibility.
- Phase 3 (Week 7-10):
  - Software Development – Develop MCU-specific code, implement indoor triangulation algorithms, and integrate security measures.
- Phase 4 (Week 10-12):
  - Testing and Debugging – Conduct unit testing for individual MCUs, integrate MCUs and perform system testing, debug and refine code as needed.
- Phase 5 (Week 12-13):
  - Prototype Demonstration – Showcase the working prototype with indoor and outdoor tracking scenarios, collect feedback, and make final adjustments.
- Phase 6 (Week 12-13):
  - Project Report and Documentation – Compile a comprehensive project report, document codebase, and configurations for future reference.

11. Conclusion

The Smart Tracker project aims to create an advanced asset-tracking system using BLE and WiFi communication protocols. The chosen MCUs, programming languages, and software tools align with the project's specific requirements. This proposal acknowledges the early-stage nature of the project, allowing for adjustments and refinements during development. The proposed project timeline outlines sequential phases for successful development, testing, and documentation of the Smart Tracker system.