BBM460
Software Design Document

Locating Earthquake Victims Using Bluetooth

Group 18
Buğra Kağan Acar (2210356097) – İlker Avcı (2210356061)

1. Introduction

This document provides a detailed design overview for our project. It outlines the system's architecture, components, interfaces, and data flow based on the initial project proposal.

1.1. Purpose

The purpose of this SDD is to detail the software design for the Bluetooth-based victim location system. It is intended for the project development team (that's us!) and other stakeholders (the TAs in this case) involved in the BBM460 project. This document contains some images that we'd normally omit in a formal setting. However we know that reading 20 documents as a TA with a workload is tedious work, so we wanted to sprinkle in some fun so that it is at least somewhat entertaining.

1.2. Scope

Software Product: The primary software product is a system designed to locate individuals in danger by detecting and analyzing the Bluetooth signals from their phones.

Functionality: The software will:

- Receive Bluetooth signal strength data from multiple deployed probes (ESP32 microcontrollers).
- Utilize the MQTT protocol for data transmission between probes and a central system.
- Aggregate data from probes in a central processing unit.
- Apply triangulation algorithms to calculate the estimated location of the signal source (victim) based on signal strength data from at least three probes.
- Provide timely and accurate location information to rescue teams.

Non-Goals: The system is not designed to establish direct communication with the victims' devices or handle other types of radio signals beyond Bluetooth. It relies on the victims' devices actively emitting Bluetooth signals.

Benefits, Objectives, and Goals: The main goal is to enhance the efficiency of search and rescue operations in natural disasters by quickly and accurately locating victims when traditional communication fails. The objective is to use Bluetooth signals to provide data to first responders.

1.3. Definitions, Acronyms, and Abbreviations

Figure 1 – Relatable image portraying a “meme” (may-may) that’s currently hip with the youngsters.

SDD: Software Design Document
ESP32: Low-cost, low-power system-on-a-chip microcontrollers with integrated Wi-Fi and Bluetooth.
Bluetooth: A short-range wireless technology standard for exchanging data between fixed and mobile devices over short distances.
MQTT: Message Queuing Telemetry Transport. A lightweight publish-subscribe network protocol that transports messages between devices.
Probe: An ESP32-based device deployed in the field to detect Bluetooth signals and measure their strength.
Central Processing Server: The central component responsible for receiving data from probes via MQTT, aggregating it, and performing triangulation calculations.
Multilateration: Multilateration is a geolocation technique used to determine the position of an object by calculating its distance from at least three known reference points, which can be based on time-of-flight (signal travel time), angle, or signal strength. In signal strength-based multilateration, the distance is estimated by measuring the strength of the signal received from the object at multiple reference points, with weaker signals indicating greater distance.
Signal Strength: An indicator of the power level of the received Bluetooth signal, often correlated with distance.

1.4. References

- https://tomorrowdesk.com/info/protocol
- https://www.clarify.io/learn/industrial-protocols
- https://library.dev.fiveable.me/wireless-sensor-networks/unit-9/localization-fundamentals-challenges-wsns/study-guide/RTpt7B9t2ewqyPgq

1.5. Overview

This SDD is organized as follows:

Section 1 (Introduction): Provides the purpose, scope, definitions, references, and overview of this document.
Section 2 (System Overview): Briefly describes the system's context and background.
Section 3 (System Design): Details the design methodology and the decomposition of the system into components.
Section 4 (Component Description): Provides a detailed description of each major software component.

2. System Overview

Figure 2 – Former US President Barack Obama searching haystacks, like us searching the rubble (we are better)

The system is designed for disaster scenarios where locating victims quickly is critical, and traditional communication infrastructure may be unavailable. It leverages the prevalence of Bluetooth-enabled mobile devices. The system employs a distributed network of probes (ESP32s) that detect Bluetooth signals. These probes transmit signal strength data via the lightweight MQTT protocol to a central system. The central system aggregates this data and uses triangulation algorithms to estimate the victims' locations. This approach aims to be scalable and robust in low-connectivity environments.

3. System Design

3.1. Design Method

The proposal implies a distributed system architecture utilizing a publish-subscribe communication pattern (MQTT). The core logic involves data acquisition (probes), data transmission (MQTT), and data processing/analysis (central system multilateration). A component-based design approach seems appropriate, separating the concerns of signal detection, communication, and location calculation.

3.2. Decomposition Description

The system can be decomposed into the following major components:

Probe Software (running on the ESP32s):

- Continuously scanning for Bluetooth signals.
- Measuring and recording signal strength.
- Publishing the collected data (signal strength, device identifier if available, timestamp) to specific MQTT topics.

MQTT Broker:

A standard self-hosted MQTT broker instance (probably Mosquitto) responsible for routing messages between publishers (probes) and subscribers (central server).

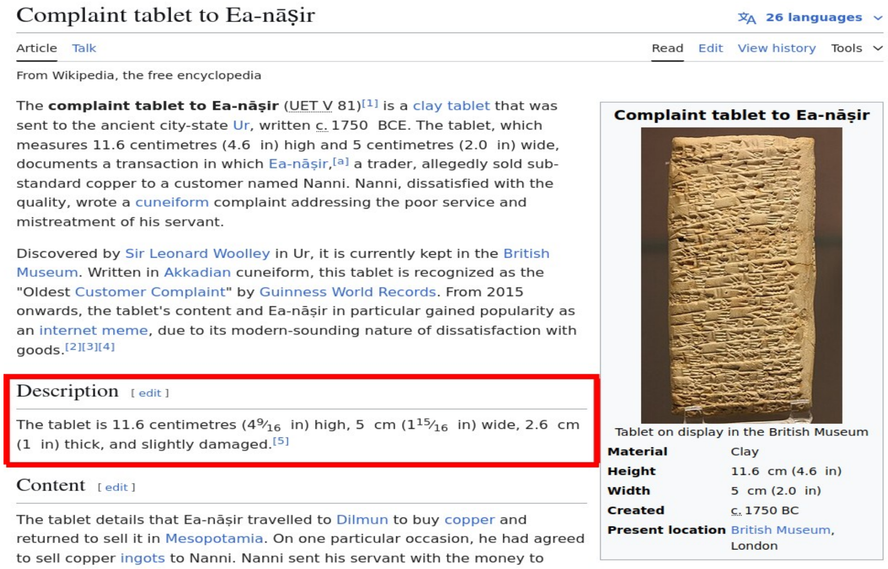
Central Processing Server:

- Subscribes to relevant MQTT topics to receive data from all active probes.
- Aggregates signal strength data from multiple probes for the same detected device.
- Implements multilateration algorithms to calculate the location based on data from four probes.
- Potentially includes a user interface or API to present the calculated location data to rescue teams.

Hierarchy/Control Flow: Probes operate independently, pushing data to the MQTT Broker. The Central System pulls data from the Broker and performs calculations. Control is decentralized at the probe level and centralized for processing.

Data Flow: Bluetooth Signal -> Probe (Scan & Measure) -> MQTT Message (Signal Strength Data) -> MQTT Broker -> MQTT Message -> Central System (Subscribe & Aggregate) -> Multilateration Algorithm -> Location Estimate.

4. Component Descriptions



Complaint tablet to Ea-nāṣir

From Wikipedia, the free encyclopedia

The **complaint tablet to Ea-nāṣir** (UET V 81)[1] is a clay tablet that was sent to the ancient city-state Ur, written c. 1750 BCE. The tablet, which measures 11.6 centimetres (4.6 in) high and 5 centimetres (2.0 in) wide, documents a transaction in which Ea-nāṣir,[a] a trader, allegedly sold sub-standard copper to a customer named Nanni. Nanni, dissatisfied with the quality, wrote a cuneiform complaint addressing the poor service and mistreatment of his servant.

Discovered by Sir Leonard Woolley in Ur, it is currently kept in the British Museum. Written in Akkadian cuneiform, this tablet is recognized as the "Oldest Customer Complaint" by Guinness World Records. From 2015 onwards, the tablet's content and Ea-nāṣir in particular gained popularity as an internet meme, due to its modern-sounding nature of dissatisfaction with goods.[2][3][4]

Description [edit]

The tablet is 11.6 centimetres ($4\frac{9}{16}$ in) high, 5 cm ($1\frac{15}{16}$ in) wide, 2.6 cm (1 in) thick, and slightly damaged.[5]

Content [edit]

The tablet details that Ea-nāṣir travelled to Dilmun to buy copper and returned to sell it in Mesopotamia. On one particular occasion, he had agreed to sell copper ingots to Nanni. Nanni sent his servant with the money to

Complaint tablet to Ea-nāṣir

Tablet on display in the British Museum

Material	Clay
Height	11.6 cm (4.6 in)
Width	5 cm (2.0 in)
Created	c. 1750 BC
Present location	British Museum, London

Figure 3 – World's First Written "Complaint Description", wait we're talking about "Components" not "Complaints"?

4.1. Component: Probe Firmware

Type: Embedded software (firmware) running on ESP32 microcontrollers. Physical component: ESP32 microcontroller.
Purpose: To detect nearby Bluetooth devices and report their signal strength to the central system. Corresponds to the data acquisition requirement.
Function: Initializes ESP32 hardware (Bluetooth radio). Continuously scans for Bluetooth advertisements. Extracts signal strength (RSSI) for detected devices. Formats data into an MQTT message. Connects to the Wi-Fi network and MQTT broker. Publishes data to a predefined MQTT topic. Handles potential connection errors.
Subordinates: ESP32 hardware drivers (Bluetooth, Wi-Fi), MQTT client library.
Dependencies: Requires Wi-Fi connectivity to reach the MQTT broker. Depends on the availability and configuration of the MQTT broker. Assumes target devices have Bluetooth enabled and are discoverable.
Interfaces:

Input: Bluetooth signals from the environment. Configuration parameters (MQTT broker address/topic).

Output: MQTT messages containing signal strength data published to the MQTT broker.

Data: Internal storage for Wi-Fi/MQTT configuration, temporary storage for scanned Bluetooth device data (MAC address, RSSI, timestamp).

4.2. Component: MQTT Broker

Type: Software Service (self-hosted Mosquitto instance). Physical component: Server.
Purpose: To facilitate communication between Probes and the Central System using the MQTT protocol.
Function: Receives messages published by Probes on specific topics. Routes these messages to any subscribed clients (the Central System). Manages client connections and subscriptions.
Subordinates: Underlying OS/Cloud infrastructure.
Dependencies: Network connectivity between Probes, Broker, and Central System.
Interfaces: Standard MQTT protocol interface for publishing and subscribing.
Data: Stores client connection states, subscriptions, and potentially queued messages (depending on configuration).

4.3. Component: Central Processing Server

Type: Software Application (e.g., Python script, server application). Physical component: Server.
Purpose: To collect data from probes and calculate victim locations. Corresponds to the data processing and analysis requirement.
Function: Connects to the MQTT broker and subscribes to data topics. Receives and parses incoming MQTT messages from probes. Stores and averages signal strength data per detected device from different probes. When sufficient data is available (e.g., readings from >= 3 probes for a device), applies multilateration algorithms to estimate location. Outputs/displays calculated locations. Manages probe status (optional).
Subordinates: MQTT client library, potentially a database or data storage mechanism, potentially a UI framework or API framework.
Dependencies: Network connectivity to the MQTT broker. Relies on the format and correctness of data sent by probes. Requires knowledge of the physical locations of the probes for multilateration.
Interfaces:

- **Input:** MQTT messages from the MQTT broker. Configuration data (probe locations).
- **Output:** Calculated location estimates (e.g., coordinates) presented via a UI, API, or log files.

Data: Stores incoming signal strength data (device ID, probe ID, RSSI, timestamp), probe location data, calculated victim location estimates.