

HackNation 2025

--Cyfrowy Nieśmiertelnik—

Zespół „Alive”

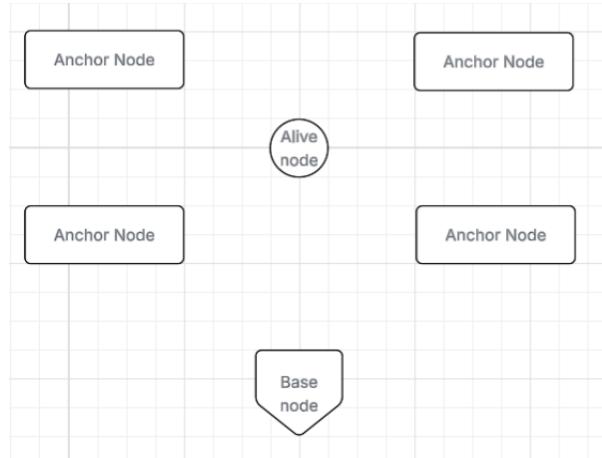
HARDWARE SOLUTION

Opracowanie jest dokumentacją techniczną warstwy hardware. Nasza praca, oprócz elementu koncepcyjnego składała się z eksperymentów na rzeczywistym sprzęcie, oraz tworzenie kodu obsługującego potrzeby eksperymentu w języku Python.

Kod: https://github.com/ciembor/hack_2025_psp/hardware

Warstwa hardware składa się z 3 elementów:

- anchor nodes
- moduł Alive
- base node



Relacja poszczególnych elementów:

Relacja Anchor Node => Alive node

Anchor nodes - ciągły nasłuch i odbiór wiadomości LoRa	Alive node - ciągły odczyt sensorów, rozgłaszenie-publikowanie wszystkich wiadomości LoRa
--	---

Relacja anchor node - anchor node

Nody na etapie inicjalizacji misji zostają uszeregowane na podstawie odległości wobec siebie i każdy z nich ma określone do którego Node'a wysyła zebrane informacje. Zbierane są pakiety co kilka sekund, lub ciągła komunikacja.

Relacja ostatni anchor node - Base node

ostatni Anchor node w szeregu przesyłania komunikuje się bezpośrednio z base node.	Base node = Anchor node którego za zadanie jest nasłuchiwać tylko pakietów od Anchor Nodes
--	--

Anchor nodes

Statyczne punkty bazowe o znanych koordynatach GPS. Montowane na statywach lub wózach strażackich.

Funkcje:

- znana lokalizacja GPS
- na podstawie lokalizacji GPS i odległości pomiędzy node'ami jednym z pierwszych kroków misji jest zdefiniowanie lokalnego układu współrzędnych Referencyjnego w dalszych etapach misji.
- ciągły nasłuch, odbierają informacje przesyłane od modułu Alive przez LoRa
- łączą informacje odczytane z modułu Alive z odczytem rssi LoRa, oraz rssi Wifi (moduł Alive to access point z ssid= ID np. FF-001)

Komponenty użyte na etapie eksperymentów i prototypowania

- Mikrokontroler ESP32-D0WD-V3
- Antenna RA-02 AI-Thinker LoRa SX1278 433MHz IPEX 3.3V
- GPS

Moduł Alive

Nasobne urządzenie zbierające sygnał z sensorów.

Funkcje:

- zbieranie informacji z sensorów
- działa w trybie Wifi Access Point
- wysyła ciągle pakiety danych z odczytami z sensorów poprzez LoRa
- gromadzi dane na karcie SD

Hardware

- Mikrokontroler ESP32-D0WD-V3
- Antenna RA-02 AI-Thinker LoRa SX1278 433MHz IPEX 3.3V
- IMU*
- Barometr + Termometr*

Docelowo do zintegrowania:

- Duży przycisk
- Laser punktowy
- Buzzer
- Czytnik kart microSD

*-używane naprzemiennie – patrz wnioski z doświadczeń

Nieśmiertelnik

Brak – moduł Alive kompleksowo spełnia trzy funkcje:

- Monitorowanie odczytów
- Czarna skrzynka
- Pomoc w identyfikacji i lokalizacji w przypadku, gdy ratownik nie jest w stanie wezwać pomocy- buzzer, laser pionowy (zadymienie), alert do pozostałych nodów

Warstwa danych

Moduły Alive przesyłają dane w formacie json. Wszystkie dane z wszystkich modułów zapisywane są w plikach json. Do etapu Base Node dane mają charakter surowy, bez interpretacji i obliczeń. W Base Node odbywać się będzie obliczanie lokalizacji, estymacje, korekty, przetwarzanie na potrzeby warstwy Frontend np. określanie aktywności na podstawie odczytów z IMU, korekty lokalizacji na podstawie LoRa, Wifi, BLE oraz IMU. Tu będą znajdować się algorytmy dead reckoningu oraz potrzebne modele Machine Learning wspomagające proces szacowania lokalizacji.

W Base Node znajdująć się będzie skrypt core.py (github) który będzie głównym silnikiem przetwarzania danych.

Dane będą przechowywane na kartach microSD w każdym z Anchor Node oraz w module Alive.

Base Node będzie zawierać historyzację danych w formie bazy danych pozwalającą na analizę zdarzeń w przeszłości.

Takie rozwiązanie stanowi wielopoziomowe zabezpieczenie przed utratą danych.

Wnioski z doświadczeń przeprowadzanych podczas HackNation:

- Należy rozważyć użycie podwójnych płyt ESP z uwagi na ograniczenia liczby pinów GPIO. Sugerowane rozwiązanie to zastosowanie w modułach Alive dwóch płyt: jedna skonfigurowana jako komunikacyjna, druga dedykowana do obsługi sensorów. Komunikacja między nimi powinna odbywać się poprzez BLE.
- Oprócz ograniczeń w liczbie GPIO pojawia się problem z równoczesnym korzystaniem z sensorów podłączanych tym samym protokołem (np. I2C). W przypadku jednoczesnego użycia IMU oraz Barometru konieczne jest zastosowanie dwóch mikrokontrolerów.
- Opcjonalnie moduł Alive może być wyposażony w rozszerzoną kartę pamięci, co umożliwia przechowywanie większej ilości danych i tym samym wydłuży czas obserwacji.

- Należy zoptymalizować pakiety składające się z informacji zebranych w json – wiadomość LoRa ma ograniczenie 255 byte'ow. Od tego zależy częstotliwość pakietowania, w trakcie eksperymentów rozpoczęto interwał pakietów co 10s, finalnie ograniczony do 1s. Można ograniczyć również nazwami kluczy w samym piku json
- Należy uzupełnić programy Python o obsługę komunikacji BLE. Mikrokontrolery ESP32 mają wbudowany bluetooth.

Poniższe zdjęcia mają na celu przedstawić poglądowo poligon działań – trzy Anchor Nodes, Base Node oraz FF01 – moduł Alive na strażaku. Układy na zdjęciu składają się jedynie z mikrokontrolera i modułu LoRa, bez dodatkowych sensorów. Zasilanie było zapewnione poprzez kable USB-USBc bezpośrednio do komputera aby umożliwić analizę zapisywanych danych oraz debuggowanie skryptów.

