Imitator Dalmierza w symulatorze Naprowadzania Artylerii

Tomasz Kozubski

1.Cel projektu

Celem projektu było opracowanie urządzenia imitującego dalmierz laserowy, wykorzystywanego do pomiaru dystansu w środowisku symulacyjnym. Projekt powstał na potrzeby zwiększenia realizmu szkoleń z naprowadzania artylerii.

2. kontekst projektu

Urządzenie zaprojektowano jako element wspomagający immersję podczas ćwiczeń w symulatorze wojskowym (np. w grze Arma 3). Zamiast korzystać z klasycznego interfejsu ekranowego, uczestnicy szkolenia mogą w fizyczny sposób „zmierzyć” odległość do wirtualnego celu – zupełnie jak w rzeczywistości.

Po wciśnięciu przycisku, włączany jest wskaźnik laserowy, który ułatwia precyzyjne celowanie. Symulator rejestruje dystans w grze i zapisuje go do pliku `.txt`, który udostępniany jest przez prosty serwer oparty na Pythonie (Flask). Mikrokomputer ESP32 pobiera ten dystans i wyświetla go na ekranie OLED.

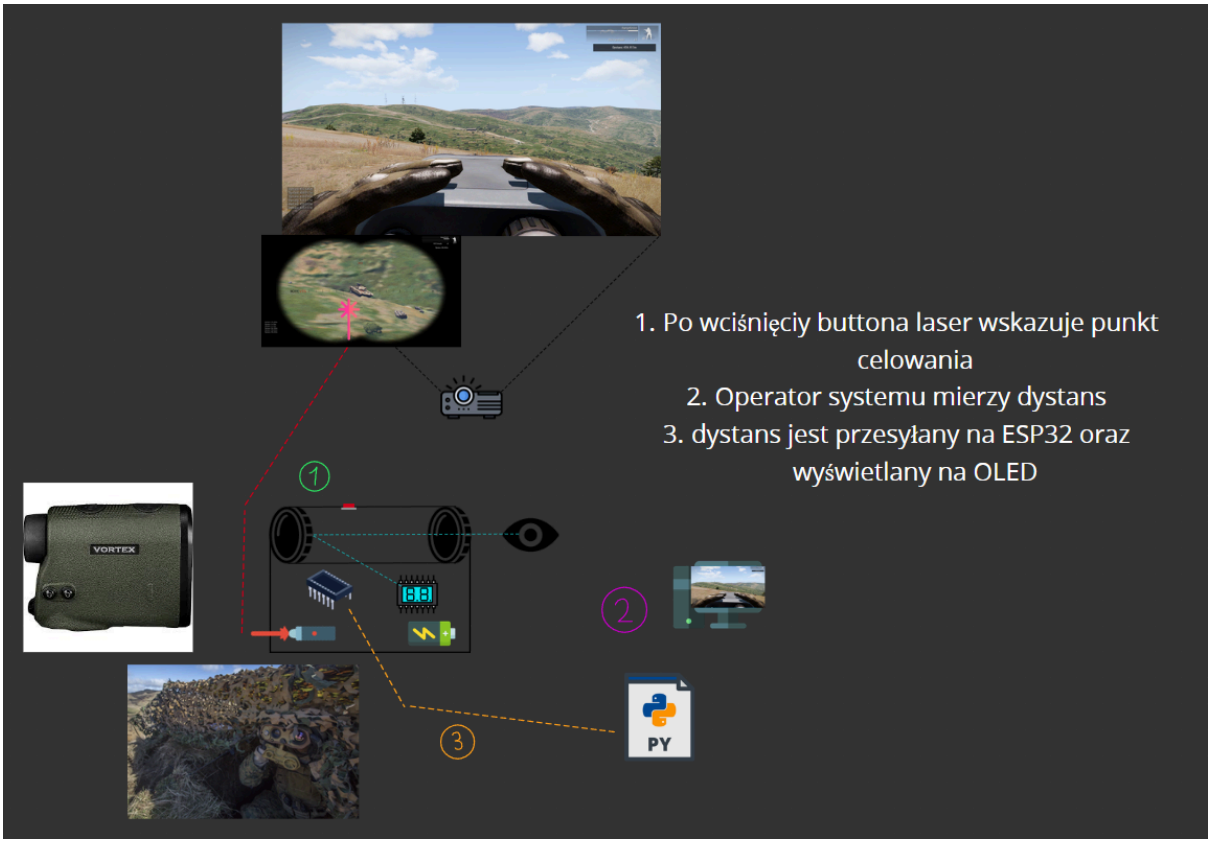
3. Opis działania

Urządzenie wyposażone jest w przycisk imitujący wyzwalacz dalmierza.

Po jego naciśnięciu:

- Aktywowany jest wskaźnik laserowy, wskazujący punkt pomiaru.
- ESP32 łączy się z lokalnym serwerem HTTP i pobiera aktualny dystans zapisany przez grę.
- Wynik wyświetlany jest na ekranie OLED.

Gdy przycisk nie jest wciśnięty, wyświetlacz pokazuje kropkę, imitującą celownik red-dot.



4. Użyte komponenty

Dioda laserowa 5mW czerwona 650nm 5V - kropka

ESP32 WiFi + BT 4.2- platforma z modułem ESP-WROOM-32 zgodny z ESP32-DevKit

Wyświetlacz OLED niebieski graficzny 1,3" 128x64px I2C v2 - niebieskie znaki

Tact Switch 12x12mm z nasadką - grzybek czarny

Tranzystor bipolarny NPN BC337-40 45V/0,8A

Komponent	Opis
ESP32 DevKit (ESP-WROOM-32)	Płytka główna z Wi-Fi i Bluetooth
Dioda laserowa 5mW, 650nm (5V)	Wskaźnik celu
Wyświetlacz OLED 1.3" 128x64 I2C	Ekran do prezentacji dystansu
Przycisk Tact Switch 12x12mm	Fizyczny trigger pomiaru
Tranzystor NPN BC337-40	Sterowanie laserem

5. Wykorzystane Biblioteki

#include <WiFi.h>
Biblioteka do obsługi Wi-Fi w ESP32.

#include <HTTPClient.h>
Umożliwia wykonywanie zapytań HTTP (GET, POST itp.) z ESP32 przez Wi-Fi.

#include <Wire.h>
Obsługa komunikacji I2C (Inter-Integrated Circuit) – magistrali do podłączania urządzeń takich jak czujniki i wyświetlacze.

#include <U8g2lib.h>
Obsługa wyświetlaczy graficznych OLED, LCD, eInk itp.

6. Link do filmiku
https://youtu.be/y-EmwAmccjE?si=hS3CarU2m4iT1atb

7. Kod

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Wire.h>
#include <U8g2lib.h>

const char* ssid = "SSID";
const char* password = "PASSWORD";
const char* serverUrl = "SERVER_ADDRESS";
```

```
U8G2_SH1106_128X64_NONAME_F_HW_I2C
u8g2(U8G2_R0, U8X8_PIN_NONE, 22, 21);
const int buttonPin = 18;
const int laserPin = 23;

void reconnectWiFi() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Próba ponownego połączenia z Wi-Fi...");
    WiFi.disconnect();
    WiFi.begin(ssid, password);
    unsigned long start = millis();
    while (WiFi.status() != WL_CONNECTED && millis() - start <
10000) {
      delay(500);
      Serial.print(".");
    }
    Serial.println();

    if (WiFi.status() == WL_CONNECTED) {
      Serial.println("Połączono ponownie z Wi-Fi!");
      Serial.print("IP: ");
      Serial.println(WiFi.localIP());
    } else {
      Serial.println("Nie udało się ponownie połączyć z Wi-Fi");
    }
  }
}

void setup() {
  Serial.begin(115200);
  delay(1000);
  u8g2.begin();
  u8g2.clearBuffer();
```

```
  u8g2.sendBuffer();

  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(laserPin, OUTPUT);
  digitalWrite(laserPin, LOW);

  WiFi.begin(ssid, password);
  Serial.print("Łączenie z Wi-Fi");
  unsigned long start = millis();
  while (WiFi.status() != WL_CONNECTED && millis() - start <
10000) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("Połączono z Wi-Fi!");
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("Nie udało się połączyć z Wi-Fi");
  }
}

void loop() {
  reconnectWiFi();

  bool buttonPressed = (digitalRead(buttonPin) == LOW);

  if (buttonPressed) {
    digitalWrite(laserPin, HIGH);
```

```
    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;
      http.begin(serverUrl);
      int httpCode = http.GET();
      if (httpCode == 200) {
        String response = http.getString();
        Serial.println("Dystans: " + response);
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_ncenB14_tr);
        u8g2.setCursor(0, 40);
        u8g2.print(response);
        u8g2.print(" m");
        u8g2.sendBuffer();
      } else {
        Serial.println("Błąd HTTP: " + String(httpCode));
      }
      http.end();
    } else {
      Serial.println("Brak połączenia Wi-Fi");
    }
  } else {
    digitalWrite(laserPin, LOW);


    u8g2.clearBuffer();
    int centerX = 64;
    int centerY = 32;
    u8g2.drawDisc(centerX, centerY, 2, U8G2_DRAW_ALL);
    u8g2.sendBuffer();
  }

  delay(1000);
}
```

Mod do Army
Zapisuje dystans do schowka

```
while {true} do {
    private _distance = player distance cursorObject;

    hint format ["Dystans: %1m", _distance];
    systemChat format ["Dystans: %1m", _distance];


    copyToClipboard str _distance;

    sleep 1;
};
```

Python Serwer Flask

```
from flask import Flask, Response

app = Flask(__name__)

@app.route('/')
def home():
    return "Serwer działa. Wejdź na /data, by zobaczyć
zawartość pliku."

@app.route('/data')
def data():
    try:
```

```
        with
open(r"C:\Users\Tom\Desktop\armaRangefinderTest\arma_dist
ance.txt", 'r', encoding='utf-8') as f:
            content = f.read()
        return Response(content, mimetype='text/plain')
    except FileNotFoundError:
        return "Plik nie został znaleziony.", 404
    except Exception as e:
        return f"Wystąpił błąd: {e}", 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Python czyta dystans

```
import pyperclip
import time

file_path =
"C:\\Users\\Tom\\Desktop\\armaRangefinderTest\\arma_distanc
e.txt"

while True:
    try:
        distance = pyperclip.paste().strip()
        if distance and distance.replace(".", "").isdigit():  #
Sprawdź, czy to liczba
            with open(file_path, "w") as file:
                file.write(distance)
            print(f"Dystans zapisany: {distance} m")
    except Exception as e:
        print(f"Błąd: {e}")
```

```
time.sleep(1)  # Aktualizacja co sekundę
```