Field Trainer v5.2 - Circuit Training System

A distributed circuit training management system with mesh networking capabilities for outdoor athletic training.

Architecture Overview

The system has been separated into modular components for better organization and flexibility:

Core Components

- `field_trainer_core.py` - Core device management and TCP server
  - Device registry and status tracking
  - TCP heartbeat server for device communication
  - Course deployment and management logic
  - Gateway status monitoring
- `field_trainer_web.py` - Web interface and REST API
  - Flask web application with Bootstrap UI
  - REST API endpoints for course management
  - Real-time device monitoring dashboard
  - System logging interface
- `field_trainer_main.py` - Combined application launcher
  - Single entry point that starts both components
  - Simplifies deployment and management

Quick Start

Option 1: Run Complete System (Recommended)

```
bash

python3 field_trainer_main.py
```

This starts both the TCP server and web interface together.

Option 2: Run Components Separately

Terminal 1 - Core TCP Server:

```bash
python3 field_trainer_core.py
```

Terminal 2 - Web Interface:

```bash
python3 field_trainer_web.py
```

System Requirements

- Python 3.7+
- Flask
- Linux system with mesh networking capabilities
- `iwconfig`, `batctl`, and `ip` commands available

Network Configuration

The system expects:

- **wlan0**: Mesh network interface (BATMAN-adv)
- **wlan1**: Internet connection interface
- **TCP Port 6000**: Device heartbeat communication
- **HTTP Port 5000**: Web interface

Device IP Ranges

- **192.168.99.100**: Controller/Gateway (Device 0)
- **192.168.99.101-105**: Training devices (Device 1-5)

Features

Circuit Training Management

- Deploy training courses to connected devices
- Real-time device status monitoring
- Action-based circuit training with Device 0 loop
- Automatic device discovery and registration

Web Dashboard

- Live device status with ping, battery, and sensor data
- Course deployment and activation controls
- Gateway mesh network status
- System event logging

Device Communication

- TCP heartbeat protocol for reliable device connectivity
- JSON message format for course deployment
- Automatic offline detection and recovery

Course Configuration

Courses are defined in `courses.json` with the following structure:

```json
{
  "courses": [
    {
      "name": "Course A",
      "description": "6-station circuit training loop",
      "stations": [
        {
          "node_id": "192.168.99.100",
          "action": "lunge",
          "instruction": "Welcome! Do 10 lunges, then sprint to Device 1"
        },
        {
          "node_id": "192.168.99.101",
          "action": "sprint",
          "instruction": "Sprint to Device 2",
          "distance_yards": 40
        }
      ]
    }
  ]
}
```

Development and Customization

Adding New Device Types

1. Extend the `NodeInfo` dataclass in `field_trainer_core.py`
2. Update the heartbeat protocol to handle new device capabilities
3. Modify the web interface to display new device information

Custom Course Actions

1. Add new action types to your course configuration
2. Update device firmware to handle new action commands
3. Modify the web interface to display action-specific information

API Integration

The web component exposes REST endpoints that can be used by external systems:

- `GET /api/state` - Current system status
- `GET /api/courses` - Available courses
- `POST /api/deploy` - Deploy a course
- `POST /api/activate` - Activate deployed course
- `POST /api/deactivate` - Deactivate current course

Troubleshooting

No Devices Connecting

- Check mesh network status in the web dashboard
- Verify BATMAN-adv is running: `sudo batctl if`
- Check device IP assignments: `ip addr show bat0`

Web Interface Not Loading

- Ensure port 5000 is not blocked by firewall
- Check Flask is binding to all interfaces (0.0.0.0)
- Review system logs for Python errors

Device Communication Issues

- Verify TCP port 6000 is open

- Check network routing between controller and devices
- Review device logs for connection errors

License

This project is designed for athletic training applications. Please ensure proper safety protocols when deploying in outdoor training environments.