Design and Implementation of a Smart Floor with Topology Reconstruction

Blaž Jerman

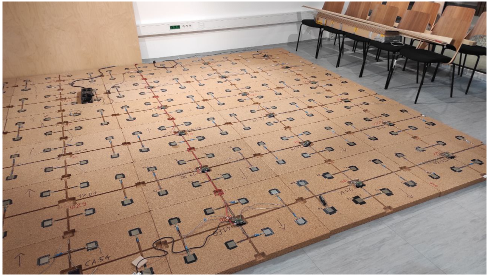
Mentor: Aleksandar Tošić

What Are Smart Floors?

Smart floors are modular floor systems that detect force, movement, and position of people or objects on the surface.

Basic idea:

- the floor is built from multiple sensor nodes
- each node measures force at several points
- data is collected by a central computer
- the system can analyze position and movement



The original concept was proposed by **Dr. Aleksandar Tošić** at InnoRenew.

Why Smart Floors?

Smart floors can be used wherever movement on a surface needs to be detected or analyzed.

Example applications:

- fall detection for elderly people
- analysis and health monitoring
- sports training and movement analysis
- interactive gyms or sports halls
- interactive games and smart environments

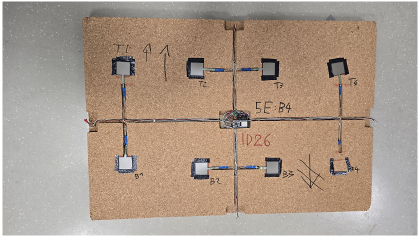
Existing System

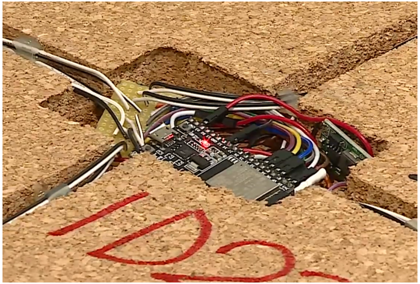
The existing system is built from modules called **nodes**.

Each node:

- has size **65 cm × 95 cm**
- contains **8 force sensors**
- uses an **ESP32 microcontroller**
- sends data to a central computer

In the original implementation, communication was done over **Wi-Fi**.





Motivation

The existing system works, but it has several limitations.

Main problems:

- data collection is too slow
- data from different nodes does not arrive at the same time
- many Wi-Fi connections can cause collisions
- node positions have to be entered manually
- power distribution becomes harder with more nodes

The goal was therefore to design a faster, more reliable, and easier-to-use system.

Project Goal

The goal of this project is to develop a new smart floor architecture.

Main improvements:

- wired communication instead of Wi-Fi
- faster and more reliable data transfer
- improved power distribution
- automatic topology reconstruction
- graphical interface for testing and visualization

Topology Reconstruction

Topology reconstruction means that the system can automatically determine how the nodes are connected.

This allows the system to:

- detect neighboring nodes
- reconstruct the floor layout
- avoid manual position entry
- make setup faster and less error-prone

This is important because the floor can be rearranged or expanded more easily.

Work Plan

No.	Task	Start	End
1	Literature review	12 Nov 2024	20 Dec 2024
2	ESP-IDF environment setup	03 Dec 2024	20 Dec 2024
3	Prototype with sensors	23 Dec 2024	24 Jan 2025
4	Protocol implementation	30 Jan 2025	21 Feb 2025
5	Graphical user interface	01 Feb 2025	28 Feb 2025
6	Prototype and protocol testing	07 Feb 2025	07 Mar 2025
7	Circuit design based on prototype	22 Apr 2025	09 May 2025
8	Circuit production	12 May 2025	20 May 2025
9	Final product testing	21 May 2025	20 Jun 2025
10	Result analysis	14 Mar 2026	14 Mar 2026
11	Final report preparation	16 Apr 2026	16 Apr 2026

Prototype and Protocol

First prototype

- ESP32 microcontroller dev board
- force sensors connected to the node
- basic wired communication setup
- used for early testing and debugging

Protocol implementation

- implemented in ESP-IDF using C
- sensor data transfer between nodes
- designed for wired communication
- prepared for topology reconstruction

The prototype allowed testing before designing the final circuit.

First prototype

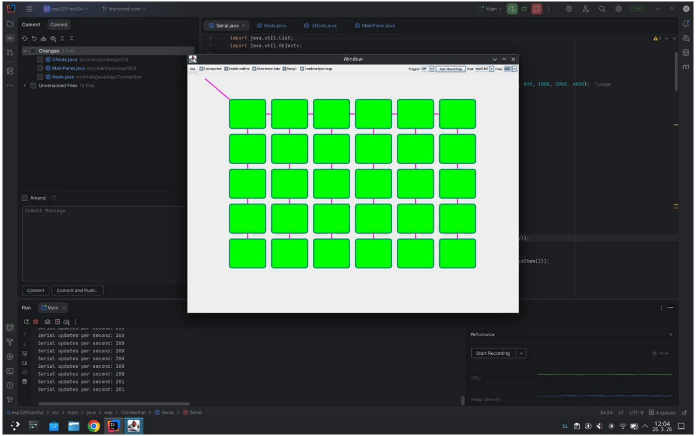
Graphical Interface and Testing

Graphical interface

- implemented in Java Swing
- displays sensor values
- helps with debugging
- used for floor layout visualization

Testing

- sensor readings
- communication reliability
- protocol behavior
- power stability
- comparison between prototype and final circuit



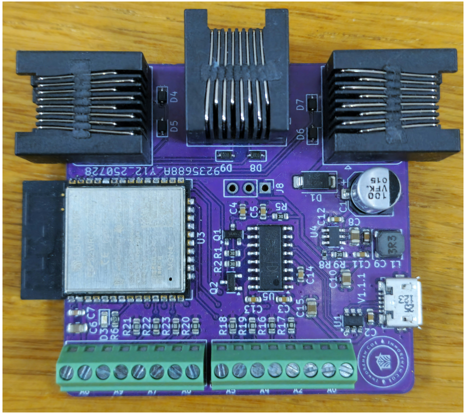
Graphical interface

Final Hardware

After the prototype was tested, a custom circuit was designed in KiCad and manufactured in China.

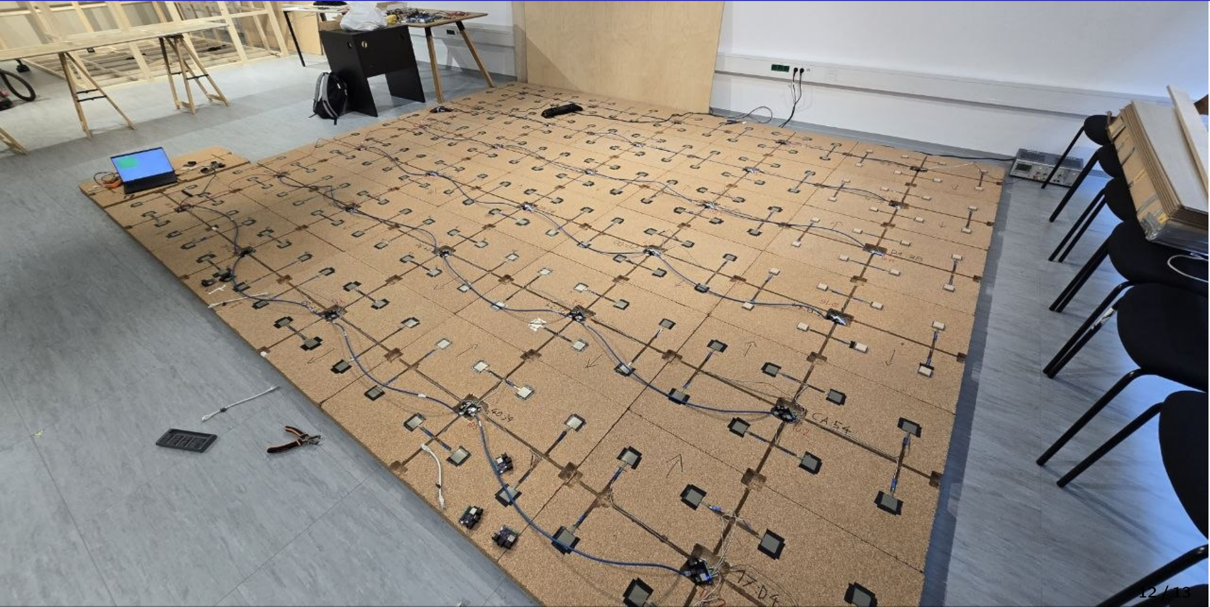
The final hardware includes:

- ESP32 microcontroller
- connectors for force sensors
- connectors for wired communication
- power regulation components
- USB to UART converter
- a header for an LED strip, which is planned for future work



Final PCB

Demonstration



Conclusion

- A working prototype was built and tested
- The communication protocol was implemented on the ESP32
- A graphical interface was developed for visualization and testing
- A custom circuit was designed and produced
- The final system was tested as a complete product

Questions?