

# Design Document: MicroBadge

## Overview

**MicroBadge** is an interactive embedded application built for the BBC micro:bit v2. It is written in **Rust** using the **Embassy** async embedded runtime. The project functions as a **conference name badge** that acts as a professional and technical icebreaker at events.

The application framework includes: \* A **menu system** for switching between apps. \* A **name scroller** for displaying your name or custom message. \* A **Snake game** for fun and interactivity. \* A **NFC business card** app for sharing contact info. **Still in progress**

## Goals

- Provide a custom and memorable name badge experience.
- Showcase embedded Rust development using async and no-std.
- Enable fun interactions and sharing via NFC.
- Serve as a technical portfolio piece for conferences, meetings, and interviews.

## Architecture

### 1. Core Framework

- Built on **Embassy** for asynchronous concurrency.
- Uses hardware abstraction layers (HALs) for GPIO, timers, display, and buttons.
- Implements a lightweight app-switcher system with a menu UI.

### 2. Applications

#### Name Scroller

- Scrolls a name or message across the LED matrix.
- Configurable for speed and repeat mode.
- Useful for identifying the wearer at a glance.

#### Snake Game

- Classic Snake game rendered on the 5x5 LED display.
- Uses **Button A** and **Button B** for turning.
- Includes food spawning, score tracking, and difficulty selection.
- Uses PWM for different intensity led lights.

#### NFC Business Card (WIP)

- Leverages the onboard NFC peripheral.
- Intends to share vCard/contact data over NFC.

- Targeted at mobile phones for quick transfer of contact info

## User Interaction

- Navigation is handled via micro:bit's **Button A**, **Button B**, and long-press **Start** gesture
- LED matrix provides visual feedback for game states and menu navigation
- Apps are selected from a simple menu UI
- Snake game is tuned to a slower speed (~4Hz) for usability

## Development Environment

- Written entirely in **Rust**, `#![no_std]`
- Uses **Embassy** for async device access and task scheduling
- Debugging output via `defmt`
- Built and flashed using `probe-rs` and `cargo-embed`

## Future Improvements

- Finalize NFC vCard transmission
- Add configuration storage via flash memory
- Create a desktop companion app for editing settings
- Support BLE for data sync or configuration

## Summary

MicroBadge combines a fun hardware platform with modern Rust development. It serves not only as a name badge but also as a way to demonstrate embedded skills, share contact info, and invite technical discussion at events.