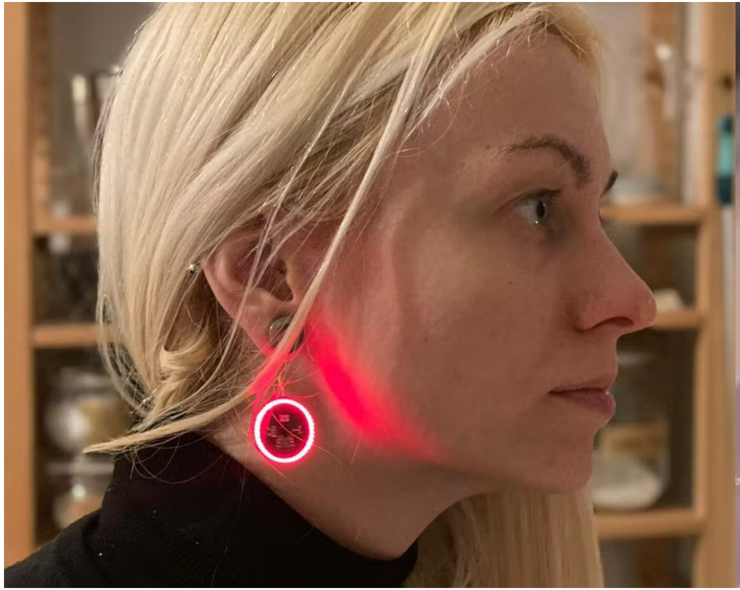
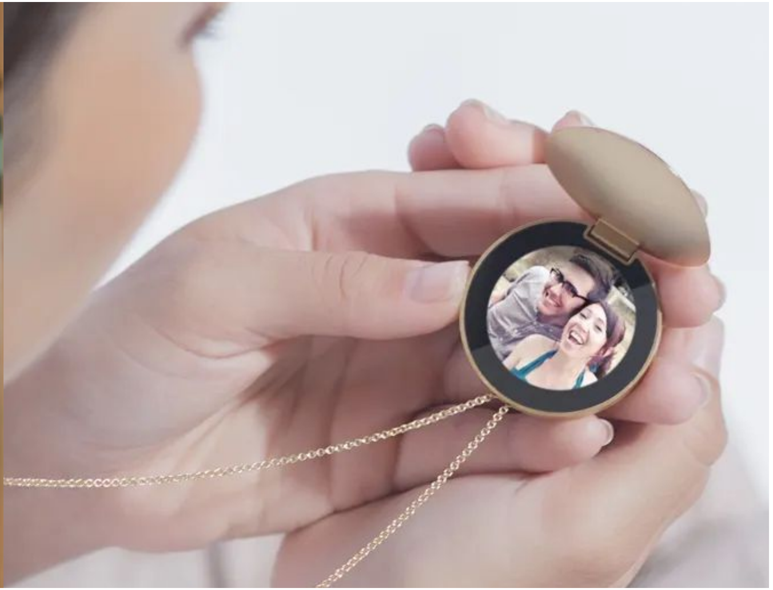
Use Case/Application



HALO-90 open source earrings: uses 90 single-color red 0402 diodes, STM8L151G4 microcontroller, CR2032 coin battery, microphone & button



Purple locket by Artefact (prototype): digital locket, connects via BLE, theoretically uses photos from social media (no specs available)

Quantitative Design Requirements

Physical Design

Spec	Requirement
weight	< 20 g per earring
temperature	< 40 C device temperature
material	earring post must be skin-safe, hypoallergenic

Functionality

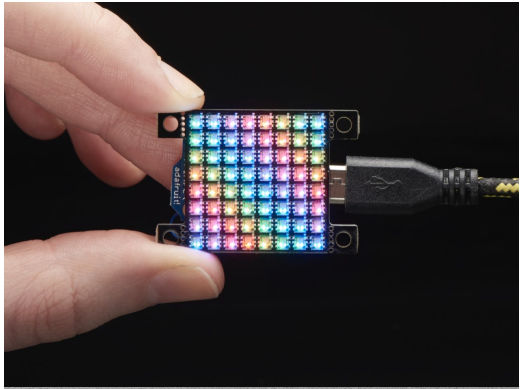
Spec	Requirement
update speed	< 1500 ms to update design
setup time	< 90 s to scan/pair device and set custom pattern
battery life	3 hr for RGB matrix 45 min for LCD

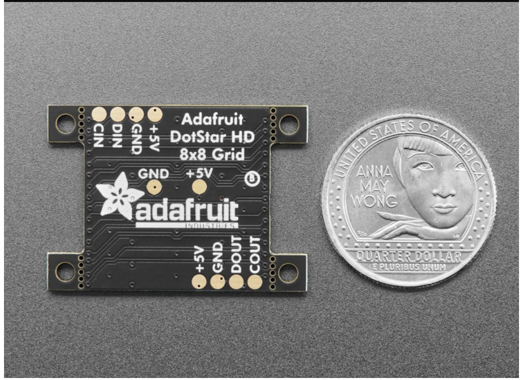
Solution Approach

We aim to develop a highly customisable and aesthetic driven wearable for **fashion forward** individuals **aged 18-35** seeking new avenues of self expression

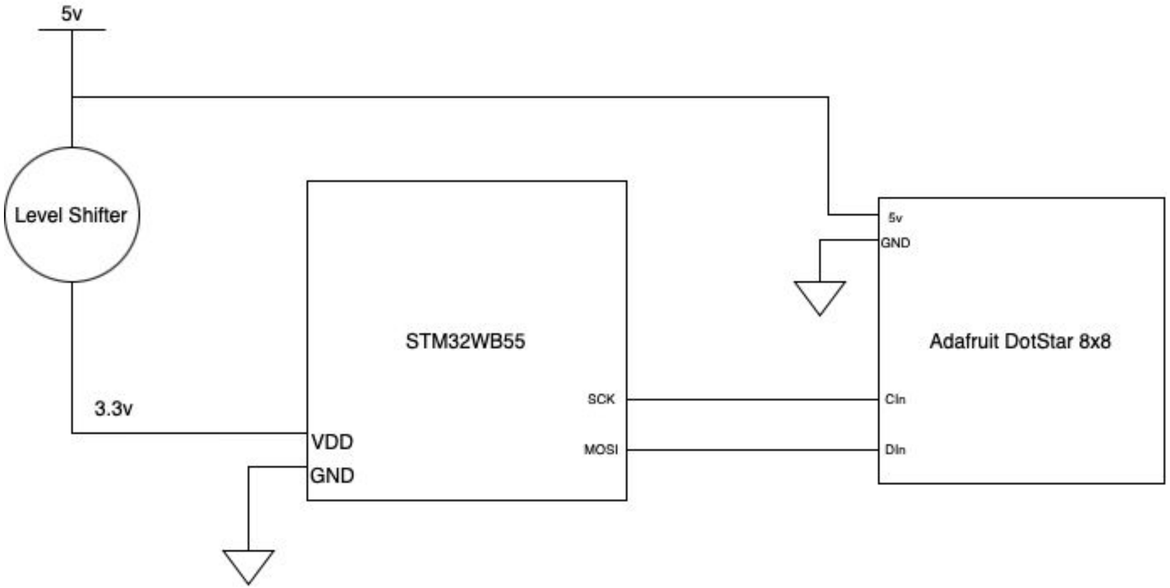
- The wearables market is expected to reach $USD 186.1 billion by 2030
- There is a global shift in the fashion industry towards digitisation
- Our adaptable fashion solution promotes sustainability via reduced consumption
- Younger consumers trend towards fashion that enables personalisation

System Specification/Block Diagram: MVP (LED Matrix)





Size: 25.4mm x 25.4mm
Update rate: 400HZ
Easy to set up: Only needs two IO pins
Each LED: 24 bits RGB values (8 bits for R, G, B)
Potential violation of temperature requirement: small board
Remedy: Cap all the RGB values to 128

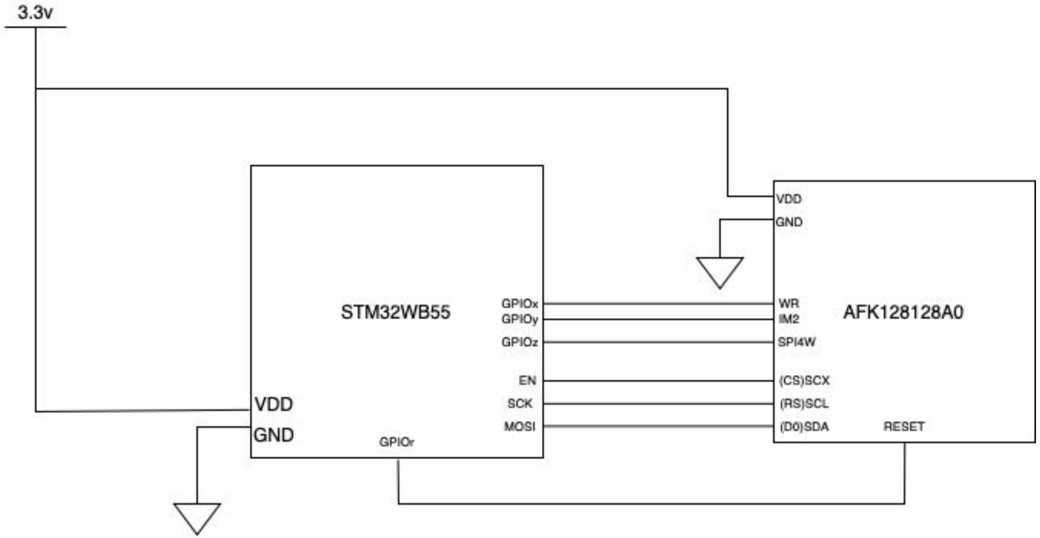


System Specification/Block Diagram: Final (Screen)

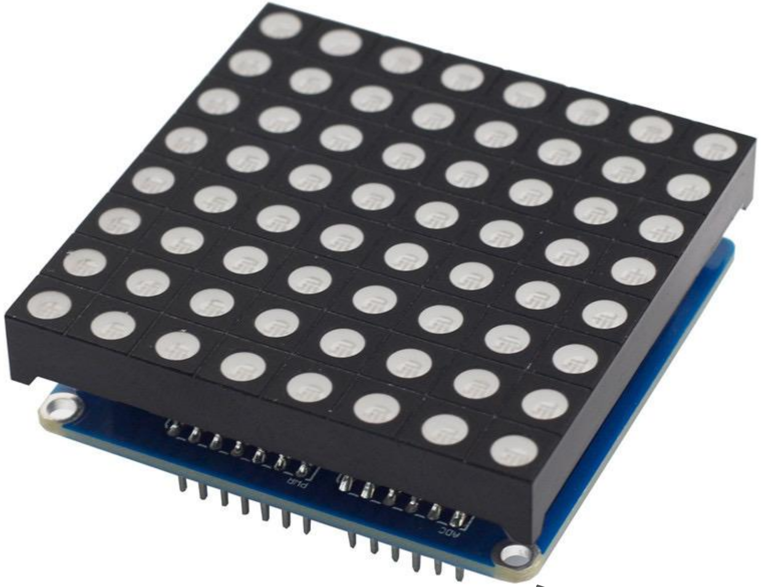
Size: 30mm x 35mm x 2.6mm

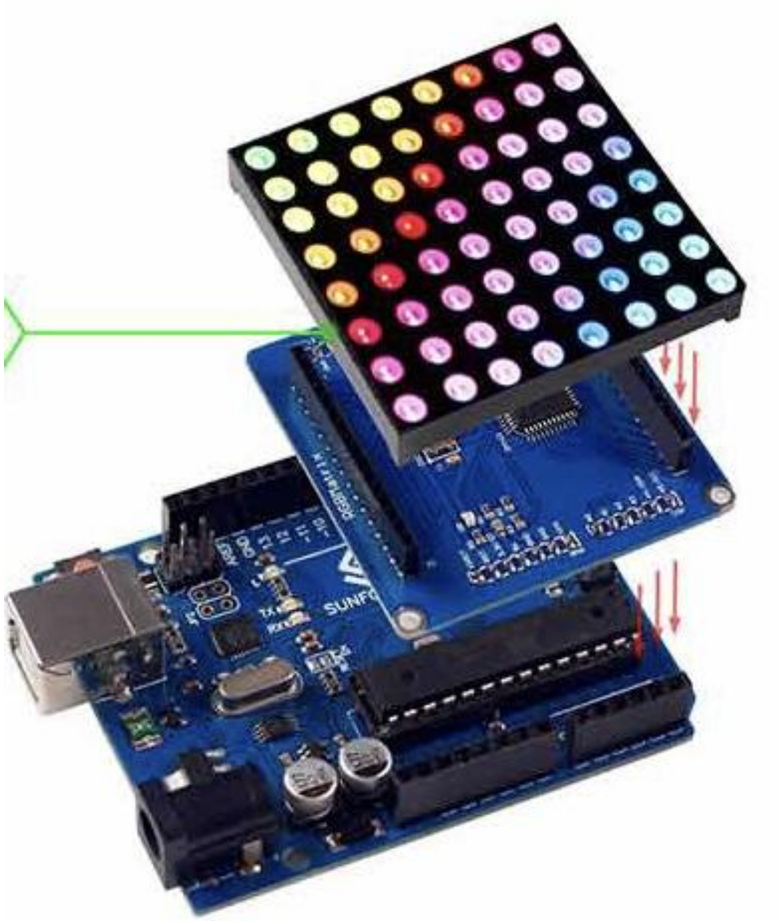
Connection: Full 3-line SPI peripheral (SCX, SCL, SDA)

Better Resolution: 128x128 pixels, allow image update



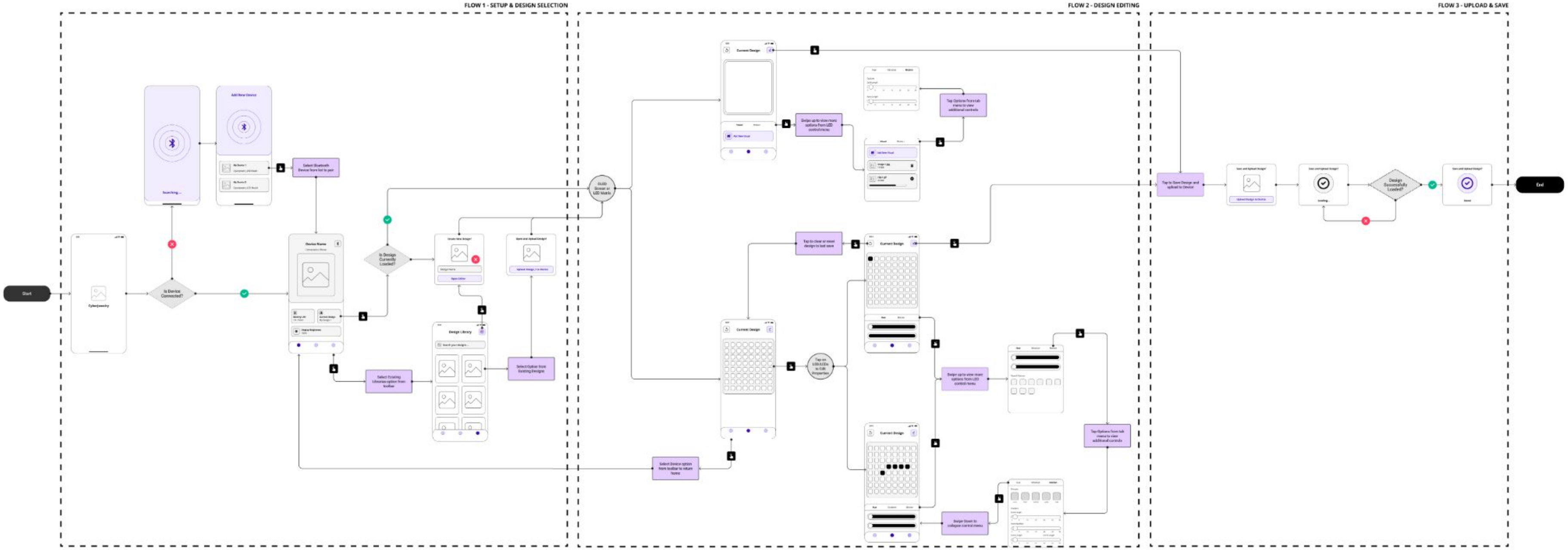
Physical Design



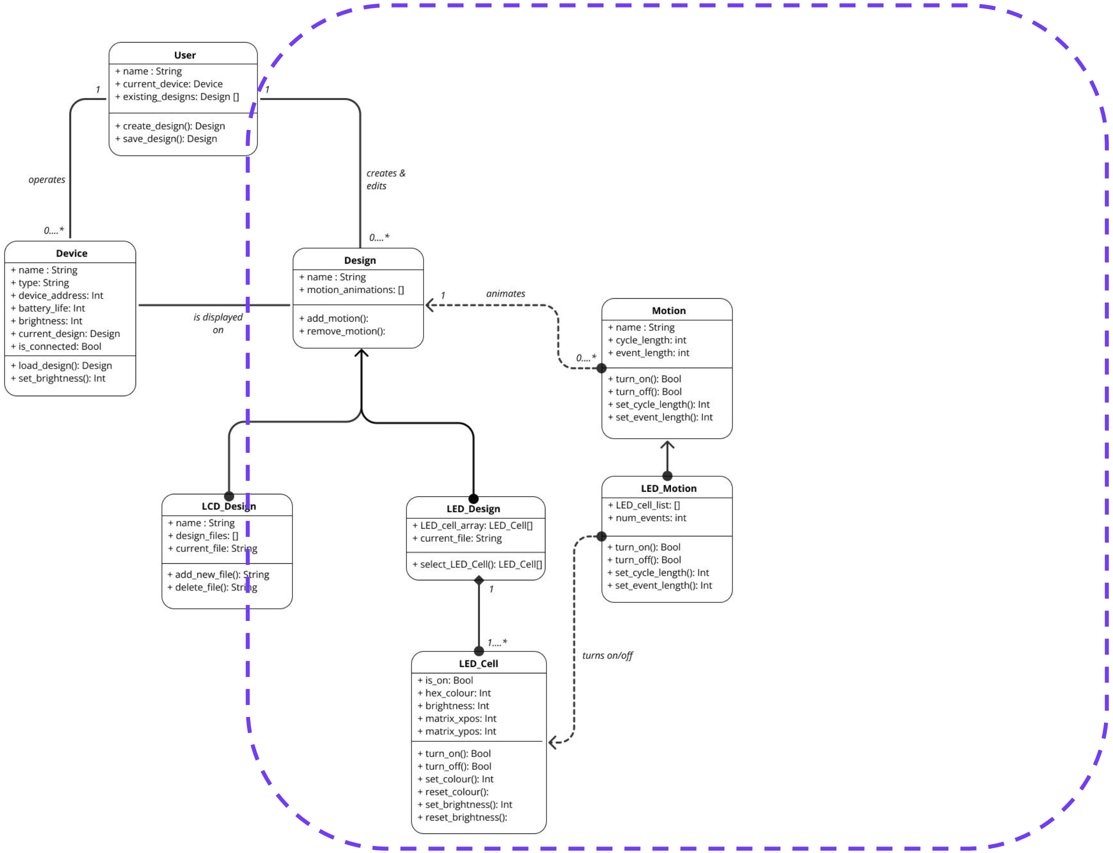


Connect earring here

System Specification/Block Diagram: iOS UI/UX


FLOW 1 - SETUP & DESIGN SELECTION
FLOW 2 - DESIGN EDITING
FLOW 3 - UPLOAD & SAVE
Start
Cyberjewelry
Is Device Connected?
Searching ...
Add New Device
Select Bluetooth Device from list to pair
Device Name
Is Design Currently Loaded?
Create New Design?
Open Editor
Save and Upload Design?
Design Library
Search your designs ...
Select Existing Libraries option from toolbar
Select Option from Existing Designs
OLED Screen or LED Matrix
Current Design
Swipe up to view more options from LED control menu
Tap Options from tab menu to view additional controls
Tap to clear or reset design to last save
Tap on LED/LEDs to Edit Properties
Select Device option from toolbar to return home
Swipe Down to collapse control menu
Tap to Save Design and upload to Device
Upload Design to Device
Loading...
Design Successfully Loaded?
End

System Specification: iOS Class Diagram



User
+ name : String
+ current_device: Device
+ existing_designs: Design []

+ create_design(): Design
+ save_design(): Design

Device
+ name : String
+ type: String
+ device_address: Int
+ battery_life: Int
+ brightness: Int
+ current_design: Design
+ is_connected: Bool

+ load_design(): Design
+ set_brightness(): Int

Design
+ name : String
+ motion_animations: []

+ add_motion():
+ remove_motion():

Motion
+ name : String
+ cycle_length: int
+ event_length: int

+ turn_on(): Bool
+ turn_off(): Bool
+ set_cycle_length(): Int
+ set_event_length(): Int

LCD_Design
+ name : String
+ design_files: []
+ current_file: String

+ add_new_file(): String
+ delete_file(): String

LED_Design
+ LED_cell_array: LED_Cell[]
+ current_file: String

+ select_LED_Cell(): LED_Cell[]

LED_Motion
+ LED_cell_list: []
+ num_events: int

+ turn_on(): Bool
+ turn_off(): Bool
+ set_cycle_length(): Int
+ set_event_length(): Int

LED_Cell
+ is_on: Bool
+ hex_colour: Int
+ brightness: Int
+ matrix_xpos: Int
+ matrix_ypos: Int

+ turn_on(): Bool
+ turn_off(): Bool
+ set_colour(): Int
+ reset_colour():
+ set_brightness(): Int
+ reset_brightness():

Implementation Plan

Copying:
STM32WB55 PCB Schematic (SnapEDA)
AFK128128A0-1.44N6NTM PCB Schematic (SnapEAD)
Adafruit DotStar 8x8 RGB LED Matrix PCB Schematic (Github)

Downloading:
STM32W HAL Peripheral Library (Github)
CircuitPython Libraries (Github)

Developing on our own:
Customized PCBs
BLE and SPI Firmware
User App (iOS)
Bare-metal Embedded Software

Testing, Verification and Metrics

Spec	**Requirement**	**Testing/Verification Method**
Weight	< 20g	Use a scale
Temperature	< 40 degrees	Run the device for 40 min and record maximum temperature reached
Update Speed	< 1500ms	1. Update speed and be inferred from SPI peripheral clock frequency 2. Measure the update speed and make sure it’s < 1500ms
Setup Time	< 90 s to scan/pair device and set custom pattern	Run scan/pair experiment 10 times and record the average latency
Battery Life	3 hr for RGB matrix 45 min for LCD	Put the device in real-life scenario and record the battery life

We also plan to collect user feedback by letting them wear the device and fill out a survey to estimate overall user satisfaction

Risk factors and unknowns

Risks:

- PCB not working as expected
    - Parallel development. We believe developing the system prototype and PCB at the same time can save time and guide our PCB design

- Can’t get BLE to work
    - On the hardware side, we can consult peripheral code provided by ST from cubeIDE.
    - On the user side, we can first develop a Python bluetooth script

Upcoming Tasks: Hardware	Madi	Shize	Saniya
Writing the basic interface to interact with the DotStar Matrix		🟦	
Creating the second level of functions: static preset patterns, selective color changes, color shifts along the spectrum, simple moving patterns		🟦	🟩
Download generic BLE driver code onto STM32 and customize for our application (configure IO for our devices)		🟦	🟩
Receive single-color signal over BLE		🟦	🟩
Design encoding schema for complex pattern data (setting all 64 pixels)		🟦	🟩
Upcoming Tasks: Software			
Create bare bones Swift app: single-color selection interface for entire matrix, if time: individual pixel color selection	🟪		
Integrate Swift Core BLE and configure STM32 connection	🟪		
Use BLE data transfer to transmit single-color selection signals over Bluetooth to device	🟪		🟩
Add preset pattern options and encode pattern data as decided to transmit to STM32	🟪		