

# **Technical documentation – Beacon Electronics**

Capstone 2021S23 Future of Health  
Singapore University of Technology And Design

Created: 15th August 2021  
Updated: 17th August 2021

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Electronics</b>	<b>3</b>
2.1	Circuit Diagram . . . . .	3
2.2	Electronic Components . . . . .	3
<b>3</b>	<b>Functional Description</b>	<b>4</b>
3.1	BLE . . . . .	4
3.2	E-Paper . . . . .	4
3.3	Source File Structure . . . . .	5
3.4	E-Paper . . . . .	6
3.5	Handling Prototype . . . . .	7
<b>4</b>	<b>Resources</b>	<b>7</b>
4.1	GitHub Repository . . . . .	7
4.2	Tutorials . . . . .	8
	<b>References</b>	<b>8</b>

# 1 Overview

The Beacon is a device that provides responses through a change of visual display and playing of tunes upon connecting with the app on a mobile device as shown in figure 1a and 1b. It is placed at the locations noted in and mapped by the application.

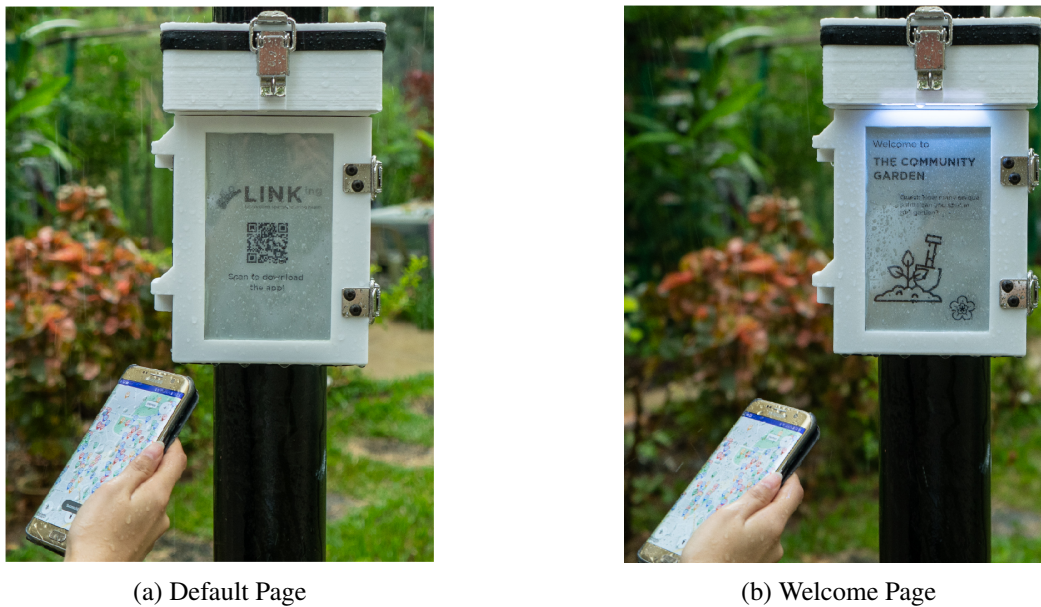


Figure 1: Beacon responses from a to b as user approaches

The beacon comprises two main parts: the electronics and the casing. The electronics consist of 6 components: the System-On-Chip, visual display, Light-emitting Diode (LED) lights strip, speaker module, battery power source, and solar power source. The casing consists of the main body, e-paper holder/ door flap, LED/solar panel casing, and several accessory components.

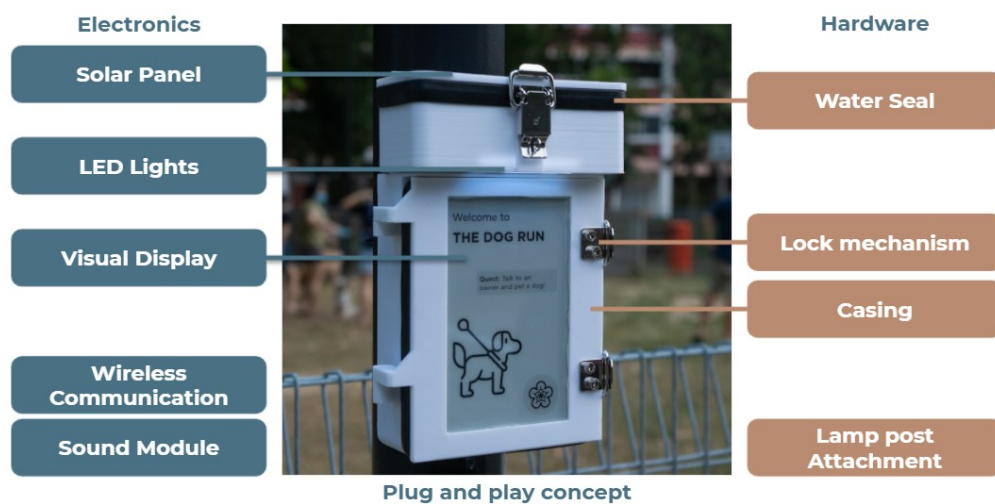


Figure 2: Overview of Beacon

## 2 Electronics

### 2.1 Circuit Diagram

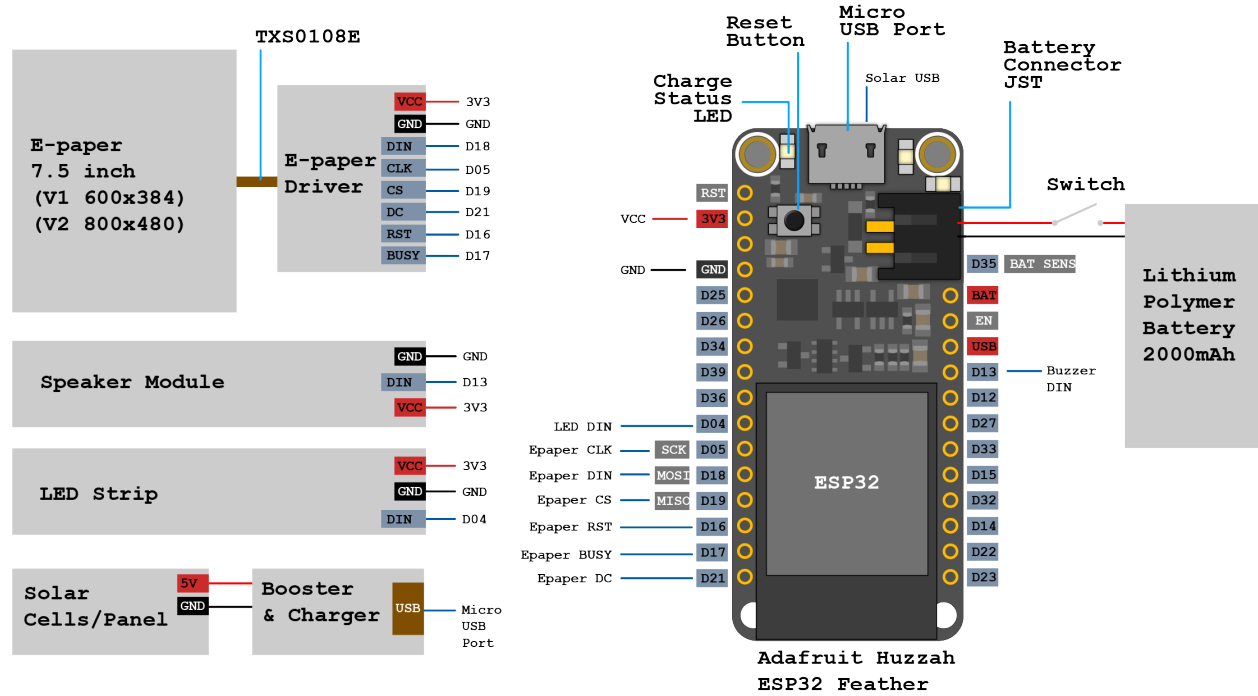


Figure 3: Circuit Diagram

### 2.2 Electronic Components

No.	Component	Item
1	Battery Power Source	Polymer Lithium Ion Battery - 2000mAh 3.7V [8]
2	Solar Power Source	Solar Cell/Panel 5V 250mA (1.25W) with Wires [9]
3	Display Module	800x480, 7.5inch E-Ink display HAT for Raspberry Pi, SPI interface   WF0583CZ09 [1]
4	Speaker module	Gravity: Digital Speaker Module-DFRobot [5]
5	LED light	Grove - RGB LED Stick (10 - WS2813 Mini) [6]
6	Chipset	Espressif ESP32 [3]
7	Development Board	Adafruit HUZZAH32 – ESP32 Feather Board [3]

Table 1: Electronics components

### 3 Functional Description

#### 3.1 BLE

No.	Component	Item
1	Transmission Power	Set as ESP_PWR_LVL_N14 (-12 dBm) The lowest transmission power for the ESP32 chipset.
2	Characteristic Value	“BeaconS23 Batt BatteryLevel” Battery level value is converted directly into a String.
3	Advertising Interval	Set at 800 900 ms
4	Protocol	Identifier check: <ul style="list-style-type: none"><li>• 36,31 or “61”</li></ul> If the first two bytes in the received packet do not match the identifier, the Beacon will not process the packet. Received Message Commands in Bytes: <ul style="list-style-type: none"><li>• 36,31,00,00 - Trigger Beacon</li><li>• 36,31,00,01 - Reset image to default</li><li>• 36,31,00,02 - Show the image</li><li>• 36,31,00,03 - Send image over but does not change display</li><li>• 36,31,00,04 - Send image over and change the image</li></ul>

#### 3.2 E-Paper

No.	Component	Description
1	Resolution	800 x 480 pixels
2	Colours	Black and white only

### 3.3 Source File Structure

1. Resources (include schematics, flowchart, guides)
2. src
  1. assets (Contains individual c files for each image with char variable defined with size 48000)
  2. lib
    1. epaper
      1. utility
        1. EPD\_7in5\_V2.cpp
        2. EPD\_7in5\_V2.h
      2. DEV\_Config.cpp
      3. DEV\_Config.h
      4. EPaper.h
      5. EPD.h
    2. Adafruit\_NeoPixel.cpp
    3. Adafruit\_NeoPixel.h
    4. battery.h
    5. ble.h
    6. buzzer.h
    7. esp.c
    8. ESP32AnalogRead.cpp
    9. ESP32AnalogRead.h
  10. led.h
  11. settimer.h
  12. Sleep.cpp (unused)
  13. Sleep.h (unused)
  14. util.h
3. main.cpp
4. main.h

### 3.4 E-Paper

No.	Component	Description
1	Platform	PlatformIO (Arduino)
2	Languages	C/C++
3	Dependencies	<ol style="list-style-type: none"><li>1. Espressif 32 (PlatformIO)</li><li>2. ESP32Servo</li><li>3. esp32-waveshare-epd</li><li>4. ESP32AnalogRead</li><li>5. Adafruit_NeoPixel</li></ol>
4	Development	<ol style="list-style-type: none"><li>1. Clone the project from the github repository or open the downloaded project source code</li><li>2. Install Visual Studio Code (VSC) [10]</li><li>3. Install PlatformIO IDE in VSC [7]</li><li>4. Create a PlatformIO project<ol style="list-style-type: none"><li>1. Select Framework as Arduino</li><li>2. Select Board as Adafruit ESP32 Feather</li><li>3. Install Espressif 32 platform (if its not installed)</li></ol></li><li>5. Import source code into the project</li></ol>
5	Image Generation	<ol style="list-style-type: none"><li>1. Import to Adobe Photoshop</li><li>2. Crop the image to 800x480 pixels</li><li>3. Convert to grayscale</li><li>4. Convert to bitmap</li><li>5. Rotate to landscape (if image is in portrait)</li><li>6. Export or Save as gif</li><li>7. Install Image2Lcd program [4]</li><li>8. Open image in Image2Lcd</li><li>9. Set width to 800</li><li>10. Set height to 480</li><li>11. Uncheck include head data</li><li>12. Set output file as C array</li><li>13. Set scan mode as horizontal mode</li><li>14. Set BitsPixels as monochrome</li><li>15. Check reverse color</li><li>16. Save c file</li><li>17. Check that the file is in the correct format: <i>* const unsigned char new_image[48000] = { /*(header hex array)*/ 0XFF, 0XFF, ... }</i></li></ol>

6	Updating code to display new image	<ol style="list-style-type: none"> <li>1. After generating image, move the .c file into the src\assets folder</li> <li>2. In ImageData.h (in src\assets folder), add <i>extern const unsigned char image_name[]</i>; at the bottom before <i>#endif</i></li> <li>3. In EPaper.h (in src\lib\epaper folder), under <i>void epaper_printDisplay(char imgSelect)</i> function, under <i>switch(imgSelect)</i>, under <i>case 1</i> in the <i>else:</i> statement, change code to <i>EPD_7IN5_V2_Display(new_image);</i></li> </ol>
6	Flowcharts	<p>Refer to Resource folder in the cloned repository or downloaded project source code</p> <ol style="list-style-type: none"> <li>1. epaper_flowchart</li> <li>2. LED_flowchart</li> <li>3. sound_flowchart</li> </ol> <p>The flowchart depicts the logical flow of the code to facilitate the functions of the Beacon.</p>

### 3.5 Handling Prototype

#### Charging

To charge the battery in the Beacon, use a micro USB cable and connect it to the micro USB port as shown in the circuit diagram. The Adafruit Huzzah32 ESP32 Feather board has an on-board charging circuitry that handles the charging of the battery. In the prototype, the board is found in the top compartment of the Beacon.

When the battery is disconnected, the charging led light will flash if micro USB is connected. The charging led will stop flashing and display a solid light when the battery is connected. However, when the micro USB is not connected while the battery is connected, the led light will not switch on.

#### Switch on and off

To switch on and off the Beacon, there is a switch connected in series on the power line of the battery as shown in the circuit diagram. In the prototype, the switch is found in the top compartment of the Beacon.

When the Beacon is switched on, the epaper display should flash and display the default page with the LINKing logo and a QR code.

## 4 Resources

### 4.1 GitHub Repository

Project source code	<a href="https://github.com/Qiftan/Capstone2021_S23_Beacon">https://github.com/Qiftan/Capstone2021_S23_Beacon</a>
---------------------	---



## 4.2 Tutorials

Component	Link
Adafruit ESP32 Feather	<a href="https://learn.adafruit.com/adafruit-huzzah32-esp32-feather">https://learn.adafruit.com/adafruit-huzzah32-esp32-feather</a>
Waveshare Epaper	<a href="https://www.waveshare.com/wiki/7.5inch_e-Paper_HAT">https://www.waveshare.com/wiki/7.5inch_e-Paper_HAT</a>
DFRobot Speaker	<a href="https://wiki.dfrobot.com/DFRobot_Speaker_v1.0_SKU_FIT0449">https://wiki.dfrobot.com/DFRobot_Speaker_v1.0_SKU_FIT0449</a>
Grove - rgb led stick	<a href="https://github.com/adafruit/Adafruit_NeoPixel">https://github.com/adafruit/Adafruit_NeoPixel</a>
Bluetooth Low Energy	<a href="https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/">https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/</a>

## References

- [1] *800×480, 7.5inch e-ink display hat for raspberry pi*, Waveshare. [Online]. Available: <https://www.waveshare.com/7.5inch-e-Paper-HAT.htm>.
- [2] *Adafruit huzzah32 – esp32 feather board (pre-soldered)*, Adafruit. [Online]. Available: <https://www.adafruit.com/product/3591>.
- [3] *Esp32-wroom-32 datasheet*, Espressif. [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf).
- [4] *File:image2lcd.7z*, Waveshare. [Online]. Available: <https://www.waveshare.com/wiki/File:Image2Lcd.7z>.
- [5] *Gravity: Digital speaker module*, DFRobot. [Online]. Available: <https://www.dfrobot.com/product-1401.html>.
- [6] *Grove - rgb led stick (10 - ws2813 mini)*, Seeedstudio. [Online]. Available: [https://wiki.seeedstudio.com/Grove-RGB\\_LED\\_Stick-10-WS2813\\_Mini/](https://wiki.seeedstudio.com/Grove-RGB_LED_Stick-10-WS2813_Mini/).
- [7] *Platformio*, PlatformIO. [Online]. Available: [https://platformio.org/?utm\\_source=platformio&utm\\_medium=piohome](https://platformio.org/?utm_source=platformio&utm_medium=piohome).
- [8] *Polymer lithium ion battery - 2000mah 3.7v*, Amicus. [Online]. Available: [http://amicus.com.sg/index.php?route=product/product&product\\_id=6238&search=battery+2000mAh](http://amicus.com.sg/index.php?route=product/product&product_id=6238&search=battery+2000mAh).
- [9] *Solar cell/panel 5v 250ma (1.25w) with wires*, Cytron. [Online]. Available: <https://sg.cytron.io/p-solar-cell-panel-5v-250ma-1.25w-with-wires>.
- [10] *Visual studio code*, Microsoft. [Online]. Available: <https://code.visualstudio.com/>.