**A black background with blue text

Description automatically generated**

IOT-Barcode checker

Project report

**Students:**

Muhammad Biadsy

Omar Sharafy

**Supervised By:**

Mony Orbach

\*Add a photo of the pcb

**Spring** 2024

Contents

[**1.** **Overview** 4](#_Toc182042062)

[**2.Introduction** 5](#_Toc182042063)

[**2.1 Device Objective** 5](#_Toc182042064)

[**2.2 Device Components** 5](#_Toc182042065)

[**2.2.1 ESP-32** 5](#_Toc182042066)

[**2.2.2 Display ILI9341** 6](#_Toc182042067)

[**2.2.3 Barcode scanner Grow Gm810** 6](#_Toc182042068)

[**2.2.4 RTC** 6](#_Toc182042069)

[**2.2.5 SD Card** 6](#_Toc182042070)

[**2.2.6 Usb Charger – type C** 7](#_Toc182042071)

[**2.3 Development Environments** 8](#_Toc182042072)

[**2.3.1 Arduino IDE** 8](#_Toc182042073)

[**2.3.1 OrCad** 8](#_Toc182042074)

[**2.3.1 GrebV** 8](#_Toc182042075)

[**3.Flow Diagram** 9](#_Toc182042076)

[**4.Hardware Block Diagrams** 10](#_Toc182042077)

[**4.1 Power Management Unit** 10](#_Toc182042078)

[**4.2 Communication Map** 11](#_Toc182042079)

[**5.Layout** 12](#_Toc182042080)

[**5.1 Top View** 12](#_Toc182042081)

[**5.2 Bottom View** 12](#_Toc182042082)

[**5.3 Considerations taken in layout:** 12](#_Toc182042083)

[**7.UI Manual** 13](#_Toc182042084)

[**8.** **Summary and results** 13](#_Toc182042085)

[**8.1** **Conclusions** 13](#_Toc182042086)

[13](#_Toc182042087)

[**8.2** **Future improvements** 14](#_Toc182042088)

[ Revise the electrical engineering (E.E.) schematic to provide distinct connections for the SPI and DC/DL components. 14](#_Toc182042089)

[ Add an option to remove the SD card, instead of overwriting its data when full. 14](#_Toc182042090)

[ Make the device more user-friendly. 14](#_Toc182042091)

[**9.Appendix** 15](#_Toc182042092)

[**9.1 E.E. schema** 15](#_Toc182042093)

[**9.2 Layout schema** 19](#_Toc182042094)

[**9.3 Refernces** 19](#_Toc182042095)

1. **Overview**

This project is a collaboration with Bnai Zion Medical Center aimed at addressing a common issue in hospital and medical center laboratories: the mismanagement of patient samples.

Currently, when samples are collected from patients, they are labeled with barcode stickers containing relevant patient information, such as name and ID. Upon arrival at the lab, these samples are checked against the patient's details in the computer system. Typically, staff label each sample with the patient’s name/ID and a unique barcode specific to the patient.

However, problems arise when multiple samples from the same or different patients arrive simultaneously. Manual handling can lead to confusion and mix-ups, making it crucial to verify at every stage that the samples belong to the correct patient and are intended for the designated laboratory.

In this project, we aim to design a low-cost, reliable barcode checker device that ensures all samples labeled as belonging to a patient (via name/ID) are correctly labeled with the matching barcode unique to the patient (a.k.a. "Golden Barcode").

# **2.Introduction**

## **2.1 Device Objective**

The goal of this project is to develop a compact and portable device that ensures all samples labeled with a patient’s name/ID have barcodes matching the patient's unique "Golden barcode."

This device will feature a 2D barcode scanner capable of reading the barcodes on the samples, providing immediate feedback on its screen and via sound to indicate whether the barcode on the sample matches the patient’s Golden barcode.

Additionally, the device will connect to a computer via Wi-Fi to display a list of the scanning results, ensuring comprehensive and accessible sample verification.

Designed to be user-friendly, cost-effective, and compatible with existing lab equipment and software, the device will be small in size, chargeable via USB-Type C & Showing results Via sound and Screen to avoid any confusion.

## **2.2 Device Components**

For the objectives this project aims to achieve, we used a list of components to help with our design.

### **2.2.1 ESP-32**

ESP32 is a series of low cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth.

Cost: 1$

### **2.2.2 Display ILI9341**

3.2-inch display that is used to display

important information about the barcode matching.

Resolution: 320X240 pixels

Comm. Protocol: SPI

Cost: 6.5$

### A black square with blue and white lights Description automatically generated**2.2.3 Barcode scanner Grow Gm810**

5.6 cm wide barcode scanner uses UART protocol.

Scans the barcode and provide us with important information about it.

Comm. Protocol: UART

Cost: 18$

### **2.2.4 RTC**

A real time clock with high accurce, will be use to recored the time of each scanning and comparing operation and save it in order to allow backtrace.

Cost: 1$

### **2.2.5 SD Card**

A high-accuracy real-time clock will be used to record the time of each scanning and comparison operation, ensuring precise timestamping. This data will be saved to facilitate backtracking and auditing of all activities.

Comm. Protocol: H-SPI.

Cost: 1$

### **2.2.6 Usb Charger – type C**

A USB Type-C charger delivers the necessary power to efficiently charge the battery.

Cost: 0.5$

The overall cost of the components and sensor for the device sums up to about 28$ US Dollars, in addition to other components we will talk about and manufacturing cost, we can sum up to about 45$ US Dollars which is a noticeably a low price in comparison to other devices in the market.

## **2.3 Development Environments**

### **2.3.1 Arduino IDE**

Used to write and upload the code to ESP-32 control board.

### A close up of a logo Description automatically generated **2.3.1 OrCad**

Used for designing the electric schematic and it’s blocks.

### **2.3.1 GrebV**

Used for viewing the Greber files.

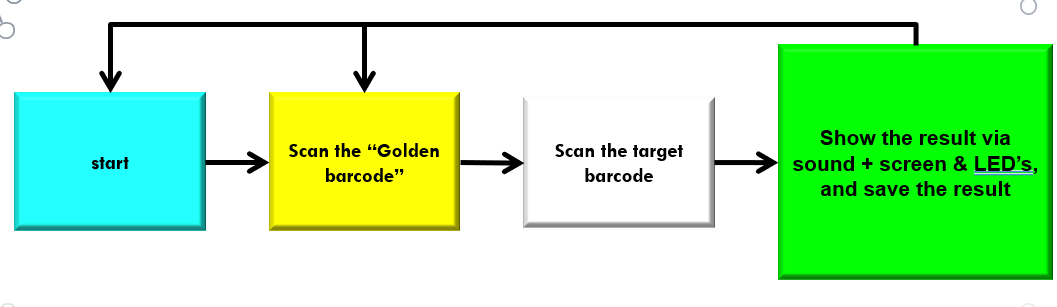
# **3.Flow Diagram**

As described before, the device functionality starts by clicking a button to start the device.

When the user wants to start a series of comparing a pool of barcodes from sample with a “golden barcode”, they must press the golden switch and scan the golden barcode.

Then they can scan each barcode and get the results via screen and sound and save the result into the SD card.

When the user wants to change the golden barcode – he must press the golden switch again and scan the desired barcode.



# **4.Hardware Block Diagrams**

A top-view block diagram of the Bar-code checker components and connection both internally between the sensors and microprocessor, and externally with the outside world.

A diagram of a block diagram

Description automatically generated

## **4.1 Power Management Unit**

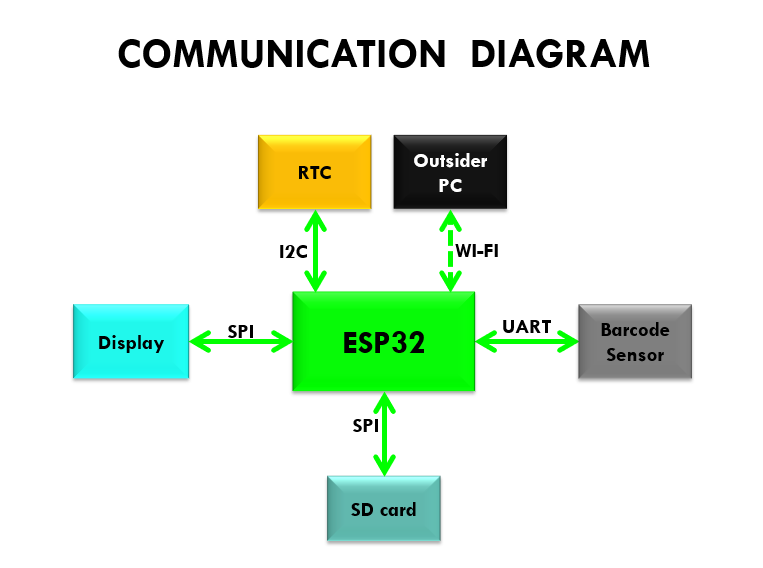
This unit is responsible for supplying the voltages for the different components of the design, It includes USB connection, The USB connection supplies power to the battery charger, which powers the system while simultaneously charging the battery. This feature enables the system to run without a battery or with defect battery.

Note that the Decision on whenever a certain component is working and gets its required voltage is managed in the Control Unit.

A diagram of a power management system

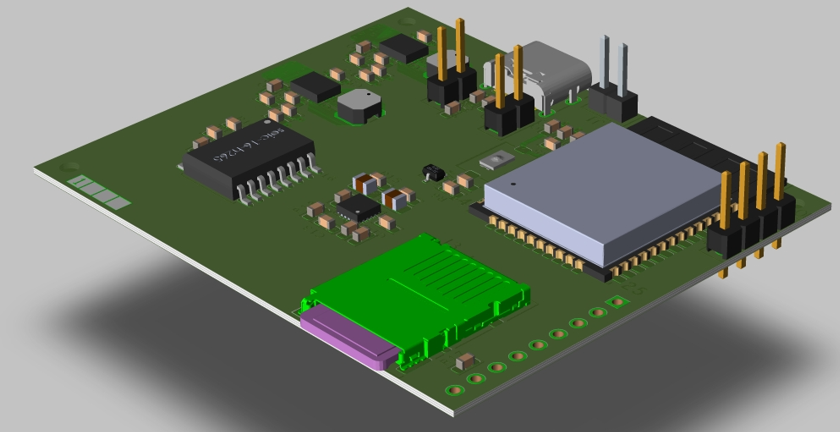
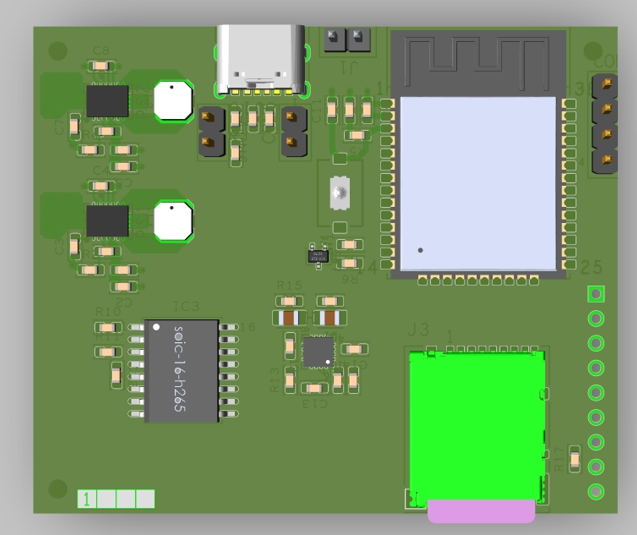
Description automatically generated

## **4.2 Communication Map**

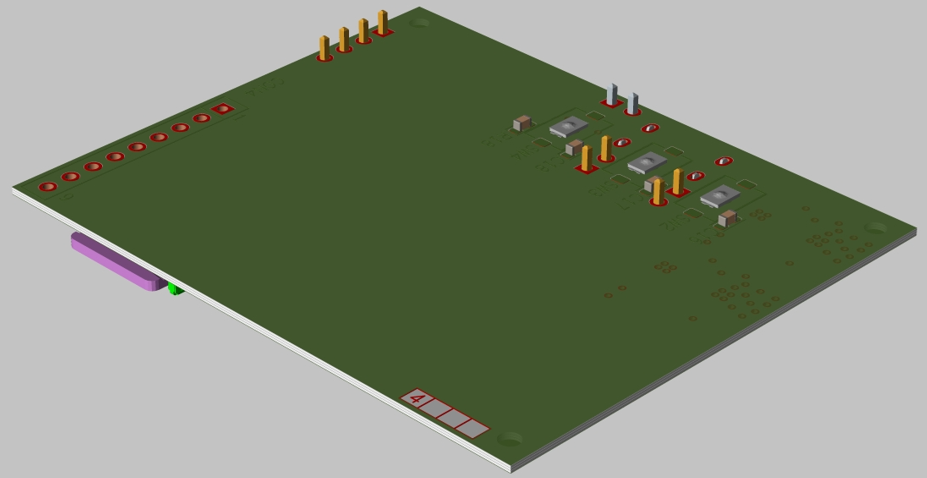
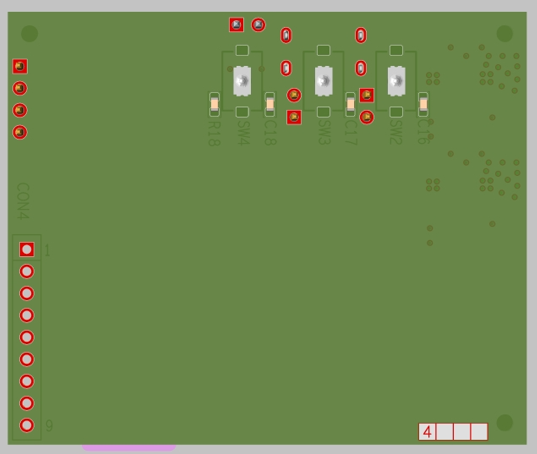


# **5.Layout**

## **5.1 Top View**



## **5.2 Bottom View**



The top of the PCB contains Our ESP32, SD card, two DC to DC converters- TPS63020 and battery charger- BQ24072. It also contains UART connector to the Barcode Reader.

The bottom of the PCB contains the Connector to the Screen, the LEDs and the 3 switches we will use.

## **5.3 Considerations taken in layout:**

The size of the screen forced us to put it in the opposite size of most other components, to allow us to interact easily with the other components.

**Top:** the DC-to-DC converters contain inductors. This fact Forced us to place it far enough from the other components.

Then we placed the ESP32, which is the biggest component. The USB and battery connectors are close to the edge, as well as the SD card, so they can be connected easily.

**Bottom:** we placed the screen connector on the edge, in order to allow the user to interact with the LEDs and switches placed on the bottom.

# **7.UI Manual**

# **8.** **Summary and results**

In this project, we were able to designee a low cost & friendly user device with low cost & small size efficient components.

Our project was able to provide an accurate result when comparing Bar-codes, with a system of alarms that get the attention of the user if such a difference is deducted.

## **8.1** **Conclusions**

We made full end to end process and achieved most of our goals. While working on the project we met a few difficulties.

There were several components that didn't work as expected (SD card which does not save the data and the Screen)

This was due to human error when working on the screen, the fact we used a lot of components which needs special connection to the esp32 made us miss some things and connect things wrongly on the E.E Scheme

the connections and the fixes for the screen are shown below:

# 

In addition, we had to re connect the SD such that it’s SPI connections are the same as the screen (expect the chip select), this added some complexity as we had to make sure both are not selected at the same time, thus making it impossible to save things to the SD while the screen is on …

In conclusion, we gained experience in design schematics and netlists, coding in Arduino and implement layout.

## **8.2** **Future improvements**

## Revise the electrical engineering (E.E.) schematic to provide distinct connections for the SPI and DC/DL components.

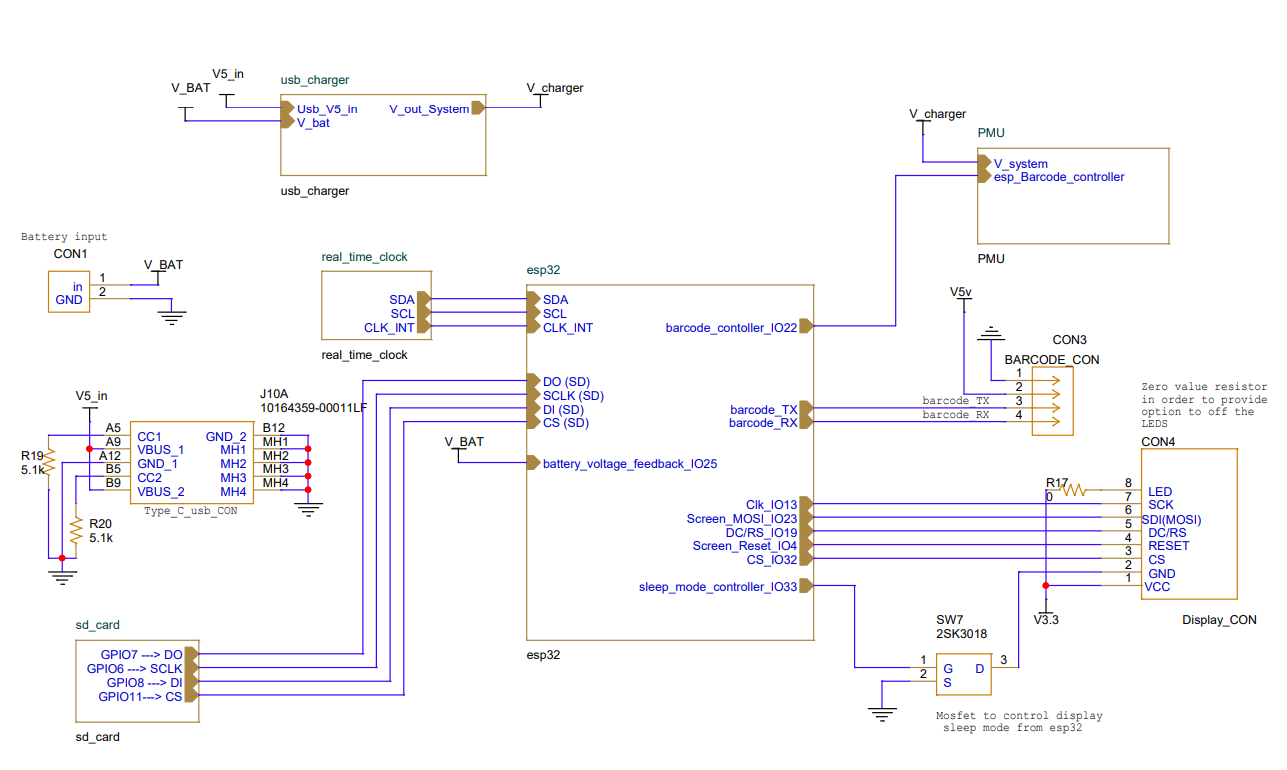
## Add an option to remove the SD card, instead of overwriting its data when full.

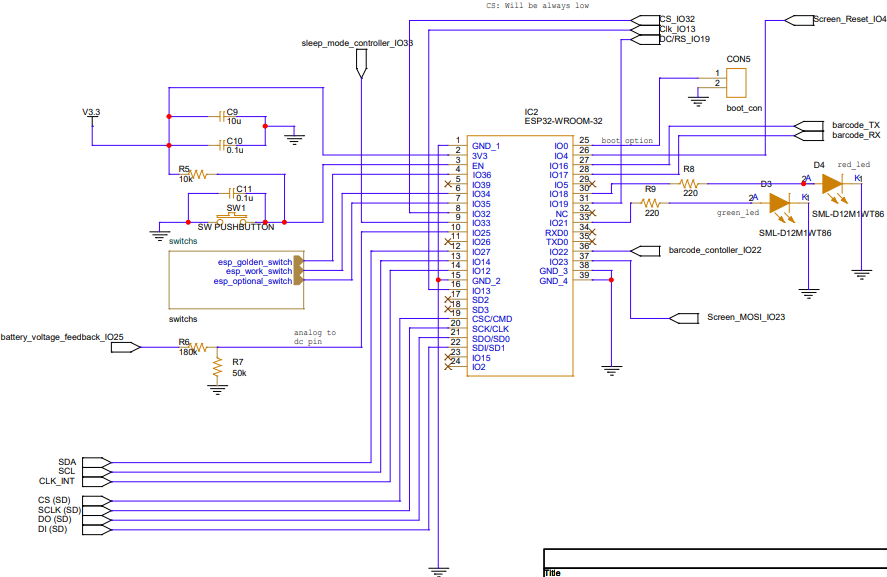
## Make the device more user-friendly.

# **9.Appendix**

## **9.1 E.E. schema**

Top level:



ESP32: 

DC-TO-DC Converters (the 3.3v, the 5v is different slightly only in the resistor R3 value)

A diagram of a circuit

Description automatically generated

The Power Unit:

A computer screen shot of a diagram

Description automatically generated

The SD Card:

A diagram of a circuit board

Description automatically generated

The USB Charger:

A diagram of a charger

Description automatically generated

The switches:

A screenshot of a computer program

Description automatically generated

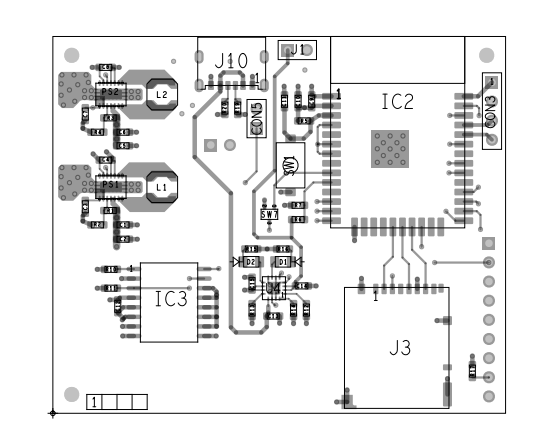
The RTC:

A diagram of a circuit

Description automatically generated

## **9.2 Layout schema**

TOP:



Bottom:

A grey and white computer screen

Description automatically generated

## 

## **9.3 Refernces**

We would like to express our gratitude to our supervisor, Mony, for his invaluable guidance and support. Without his assistance, we would not have completed this project. We have included some of the references Mony directed us to, which were essential for our progress.

We also extend our thanks to former HSDLS Lab students, whose reports provided valuable insights that helped us complete our project.

And a special thanks to all the lab staff.

Esp32 –

<https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>

Barcode –

<https://www.dropbox.com/scl/fo/87hz5h82k25j3p9k5u603/AJfkL6iYDATRGkLYJjuhUJE?e=1&preview=GM810+Series+Barcode+reader+module+User+Manual-V1.2.1.pdf&rlkey=2fyvdir15kb1kj2ada1zkadqt&spm=a2g0o.detail.1000023.1.5cf0OrwfOrwfmn&st=x3ic3hkk&dl=0%E2%80%8B>

Screen –

<http://www.lcdwiki.com/2.8inch_SPI_Module_ILI9341_SKU:MSP2807>

Real time clock –

<https://www.mouser.co.il/datasheet/2/609/DS3231-3421123.pdf>

USB type – c connector –

<https://cdn.amphenol-cs.com/media/wysiwyg/files/drawing/10164359.pdf>

Charger –

<https://www.ti.com/lit/ds/symlink/bq24072.pdf?ts=1730900206995&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ24072%252Fpart-details%252FBQ24072RGTT>

SD slot –

<https://www.mouser.co.il/datasheet/2/185/DM3BT_DSF_PEJS_CL0609_0029_9_00_2DDrawing_00009199-1614465.pdf>

Dc to dc –

<https://www.ti.com/lit/ds/symlink/tps63020.pdf?ts=1730817589484&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS63020>

Mosfet –

<https://www.mouser.co.il/datasheet/2/258/2SK3018_SOT_323_-3364975.pdf>