



INTELLIDATE

FINAL PRESENTATION

GROUP 27

KYLE DENNIS DAT TRAN

TYLER CLAITT KORY MARKS

Project Responsibilities

Kyle Dennis (CpE):

- Team Lead
- Web Application
- Panel [Desktop] Application
- System Communication



Tyler Claitt (CpE):

- PCB Component Soldering
- Identified System Components



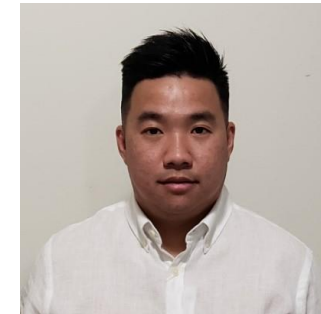
Kory Marks (EE):

- PCB Design
- Part Procurement/Treasury

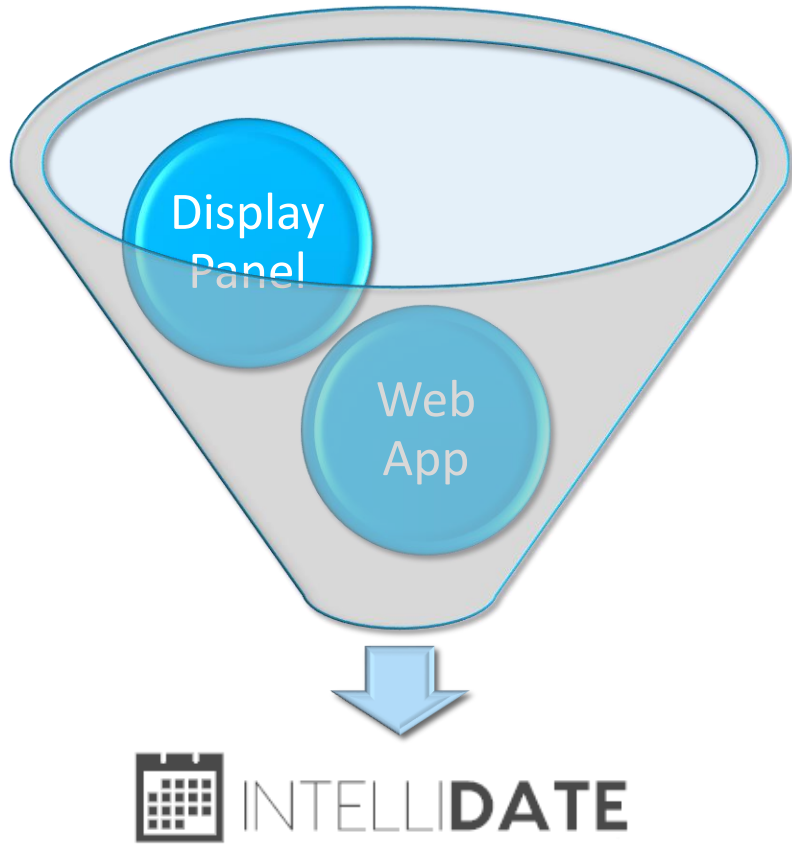


Dat Tran (EE):

- PCB Design
- System Enclosure Design



Administrative Introduction



Consumer product

Digitally updateable calendar

Increases efficiency and feasibility of organizing and maintaining a physical calendar

Problem



Utilizing a dry-erase/paper calendar keeps a user's schedule present in their everyday life

- Maintaining the calendar is a manual chore that can lead to confusion and mistakes

Software applications (Google Calendar, Outlook, Apple Calendar, etc.) have made scheduling events much more efficient, sharable, and organized

- No affordable product exists that combines the efficiency offered by scheduling software with the physical presence offered by dry-erase/paper calendars

Solution



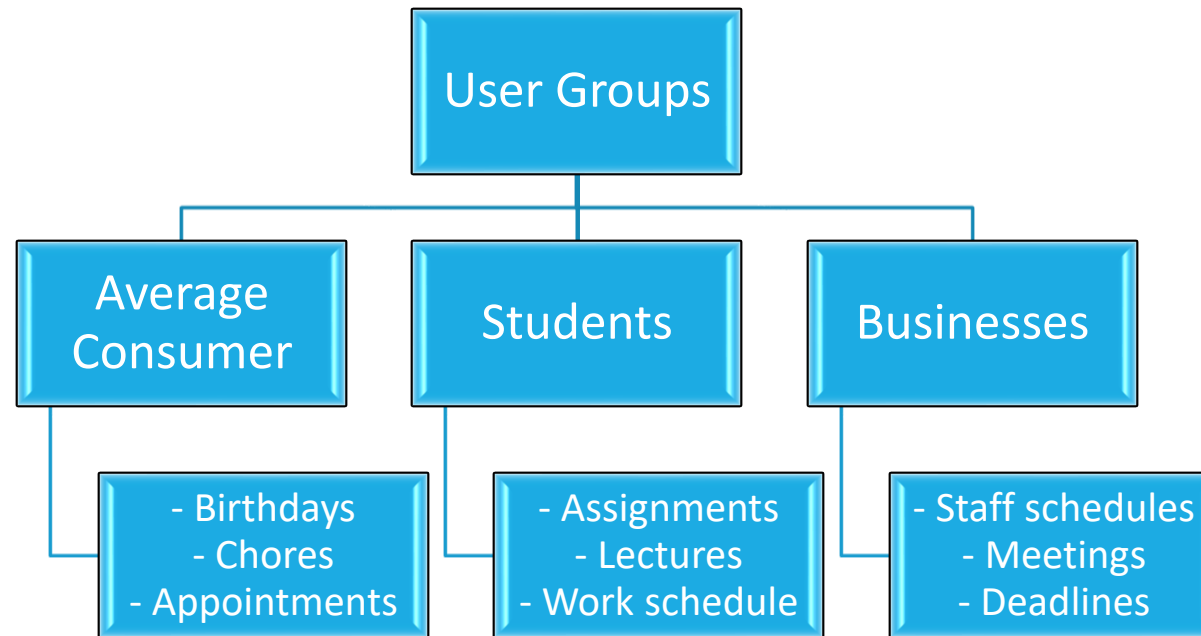
IntelliDate combines a display panel with a scheduling application to produce a low-cost product that offers the software benefits of event-scheduling with the physical presence of a dry-erase/paper calendar.

One product of this nature currently exists (DAKboard), but the product costs \$399.95 and requires a monthly service fee, starting at \$5/month.

- The IntelliDate display panel required a production cost of \$155 and provides the customer with a software application with no service fee.



Target Audience & Possible Applications



Goals and Objectives



Develop IntelliDate software application to communicate with IntelliDate display, allowing users to:

- Create/edit/delete events and notes
- Add a display panel to the user's account (via unique serial ID)
- Change calendar view of display (monthly, weekly, daily)
- Lock/unlock visibility of IntelliDate display contents

Modify traditional computer monitor to act as IntelliDate display, allowing the monitor to:

- Display a monthly, weekly, and daily calendar view
- Reflect calendar contents and user settings, from software application, automatically

Other Selling Points



- IntelliDate provides an organized, efficient, and precise method of maintaining and displaying a calendar display that is more affordable than any other existing products
- The contents of the calendar can be updated from any location, so long as the user has a device capable of accessing the website with an active internet connection
- The IntelliDate display can also be locked/unlocked, from the website, introducing a novel layer of privacy to their calendar display, previously unattainable with dry-erase/paper calendars

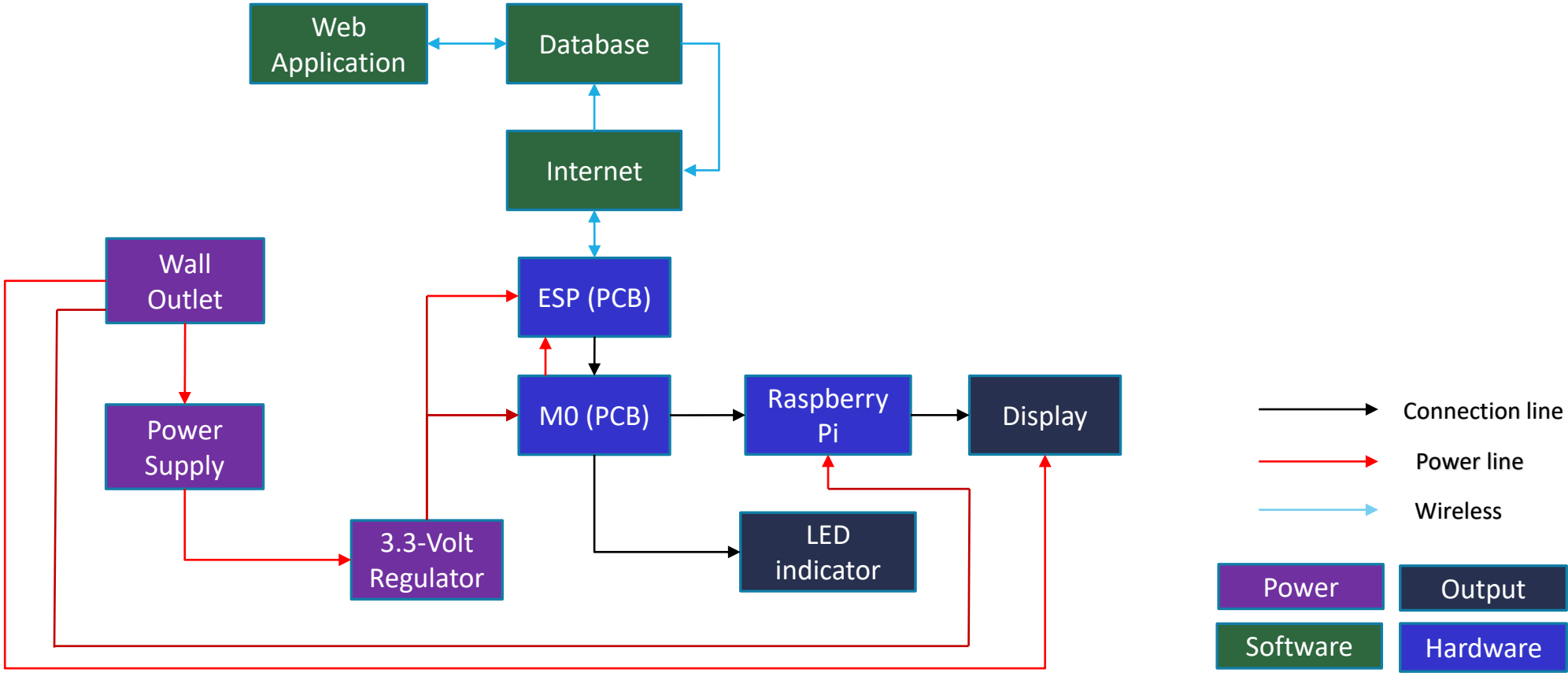
Engineering Requirements



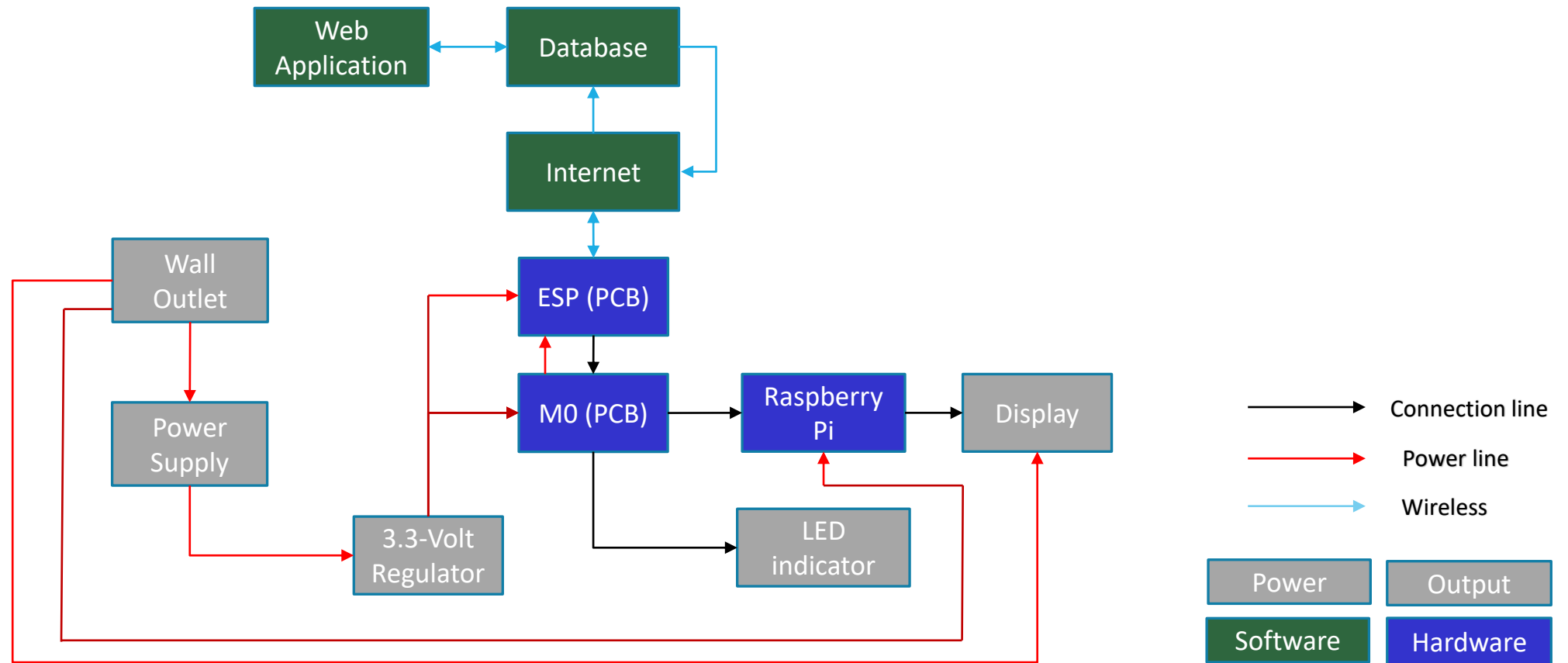
Component	Parameter	Design Specification
Display	HDMI	Up to 1080p
Communication between Panel and Web Application	Range	20 ft
Web Application/Database HTML Requests	Update Time	No longer than 5 seconds
Panel Application	Refresh Time	No longer than 30 seconds
Cost	Production Cost	Less than \$300
Physical Design	Mounting Capabilities	Monitor can be mounted on wall or stand on flat surface



Overall Block Diagram



Software



Web Application



Enables communication between user and Panel from any location

Can be accessed on any device with a web browser and active internet connection

Developed with the MERN Stack (MongoDB, Express.js, React.js, Node.js)

Frontend was deployed with Firebase; Backend was deployed with Heroku

Even without an attached Panel, can be used as a scheduling software



Web Application Features

Create IntelliDate user account

Add/remove IntelliDate Panel to/from account

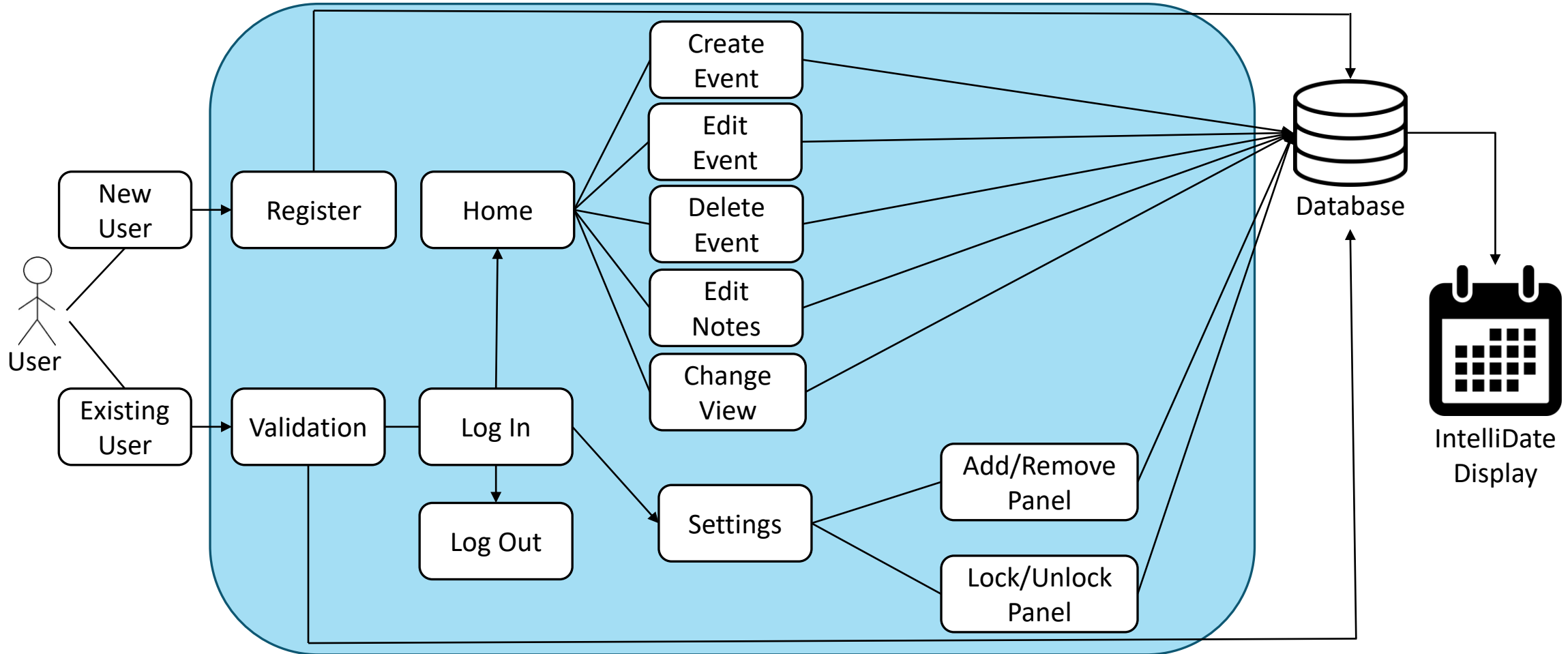
Configure lock/unlock status of Panel

Configure calendar-view of Panel (monthly, weekly, daily, agenda)

Create, edit, and delete calendar events and notes that are reflected on the IntelliDate Panel



Web Application Use Case Diagram



Panel Application



Desktop application, developed with Electron.js

Executes on Raspberry Pi

Reads string from a local text file as JSON document and converts to JavaScript object

Text file includes: Notes, Events, Lock Status, Selected Date, and Calendar View

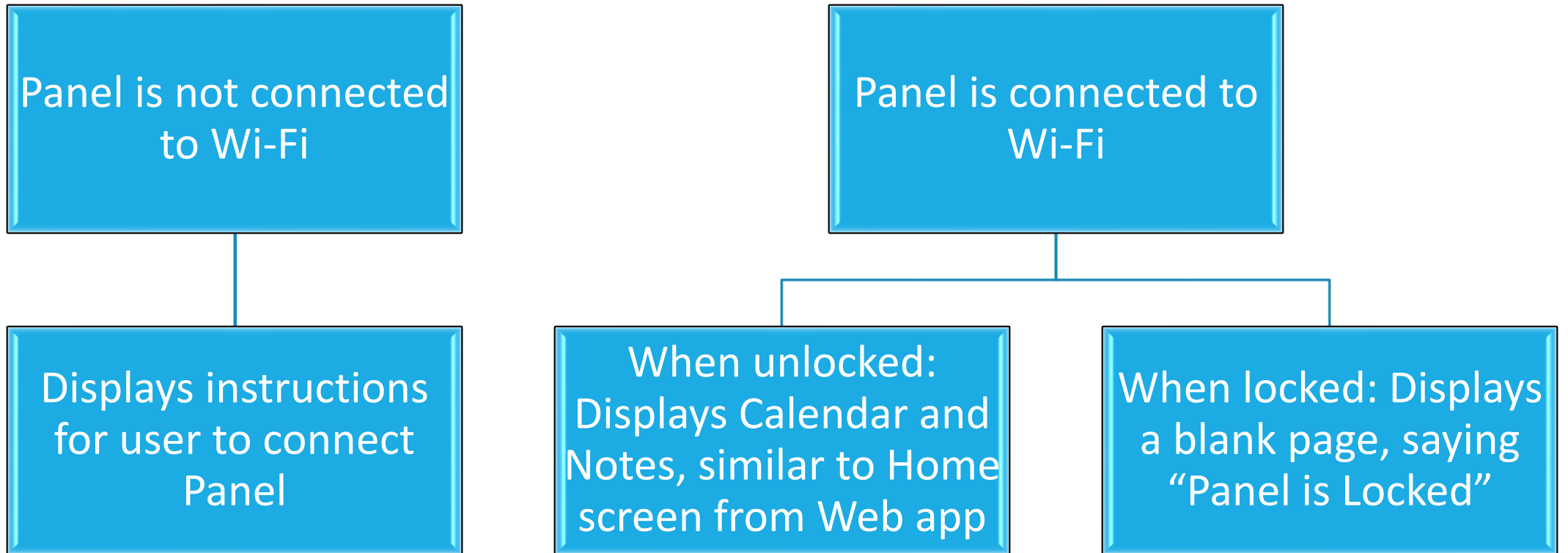
Reads text file and updates contents accordingly once per second

```
{ "isLocked": "false", "notes": "- Wash the car\n- Take out trash\n- Sweep garage\n", "view": "month", "currentDate": "Tue Apr 06 2021 19:09:02 GMT+0000 (Coordinated Universal Time)", "events": [{"_id": "60655636aa1f4e0015787f0e", "title": "Physics Lecture", "start": "2021-04-02T15:00", "end": "2021-04-02T16:15", "creator": "60652bc0aa1f4e0015787f05", "__v": 0, "id": "60655636aa1f4e0015787f0e"}, {"_id": "606c9a8490c5970015cf0cd2", "title": "Board Meeting", "start": "2021-04-02T10:00", "end": "2021-04-02T11:00", "creator": "60652bc0aa1f4e0015787f05", "__v": 0, "id": "606c9a8490c5970015cf0cd2"}, {"_id": "606c9aa190c5970015cf0cd3", "title": "Gym Workout", "start": "2021-04-02T17:00", "end": "2021-04-02T18:00", "creator": "60652bc0aa1f4e0015787f05", "__v": 0, "id": "606c9aa190c5970015cf0cd3"}, {"_id": "606c9ad190c5970015cf0cd4", "title": "Camping Trip", "start": "2021-04-03T10:30", "end": "2021-04-04T18:30", "creator": "60652bc0aa1f4e0015787f05", "__v": 0, "id": "606c9ad190c5970015cf0cd4"}]}
```

Example of local text file contents



Panel Application Features



System Communication



ESP

- Requests information from database
- Sends information to M0

M0

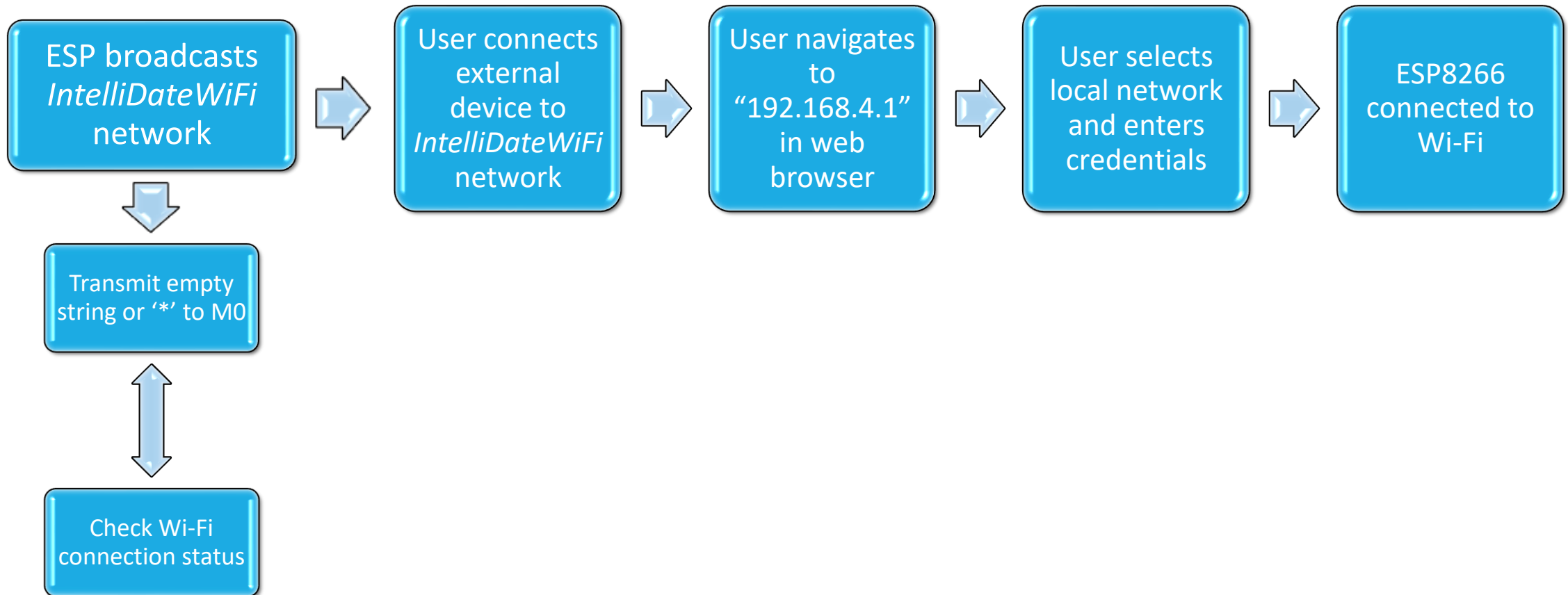
- Receives information from ESP
- Sends information to Raspberry Pi

RbPi

- Receives information from M0
- Writes information to local text file

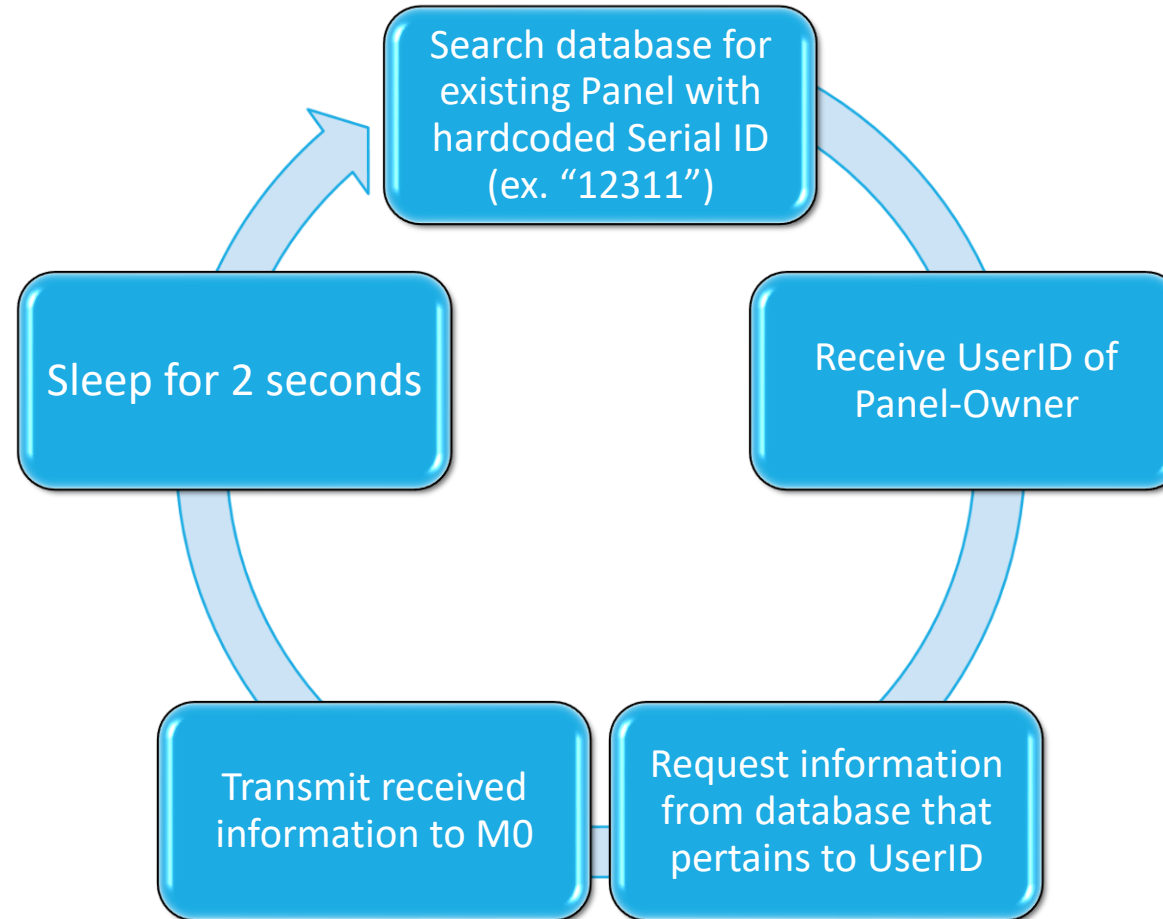


ESP8266 Program Structure (Not Connected to Wi-Fi)



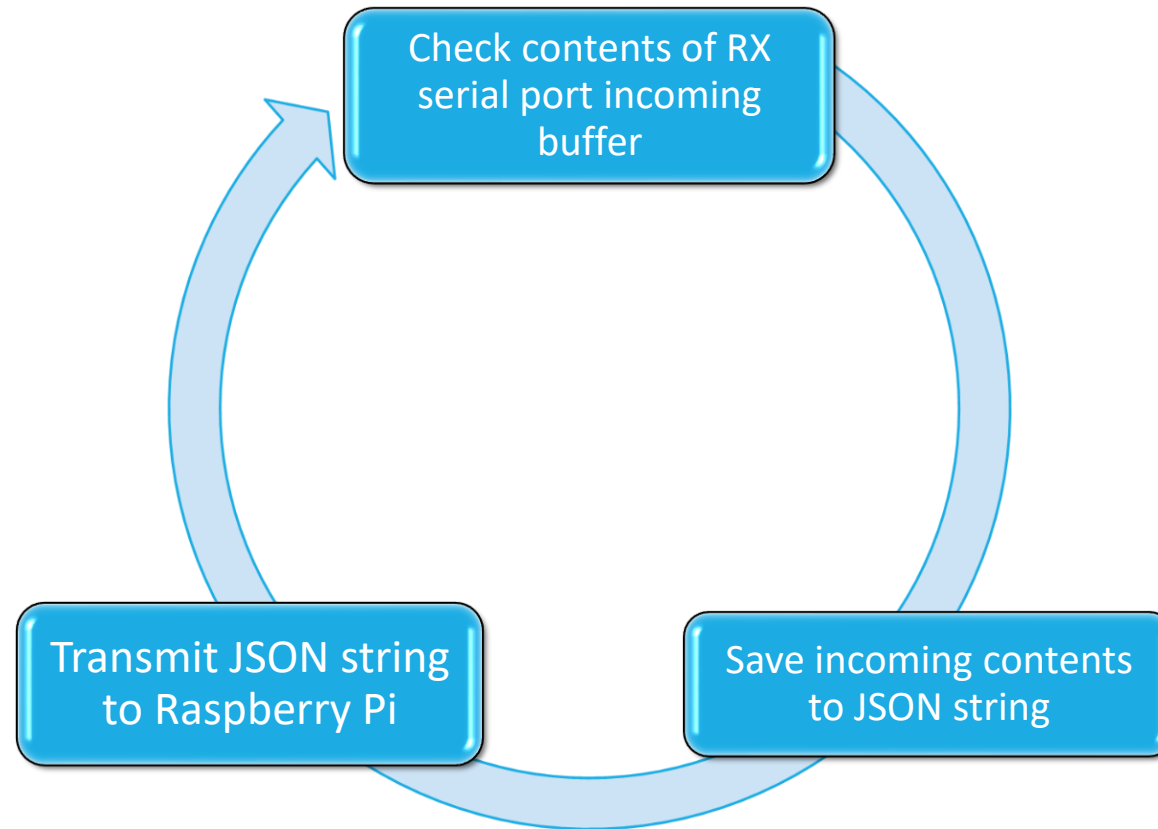


ESP8266 Program Structure (Connected to W-Fi)



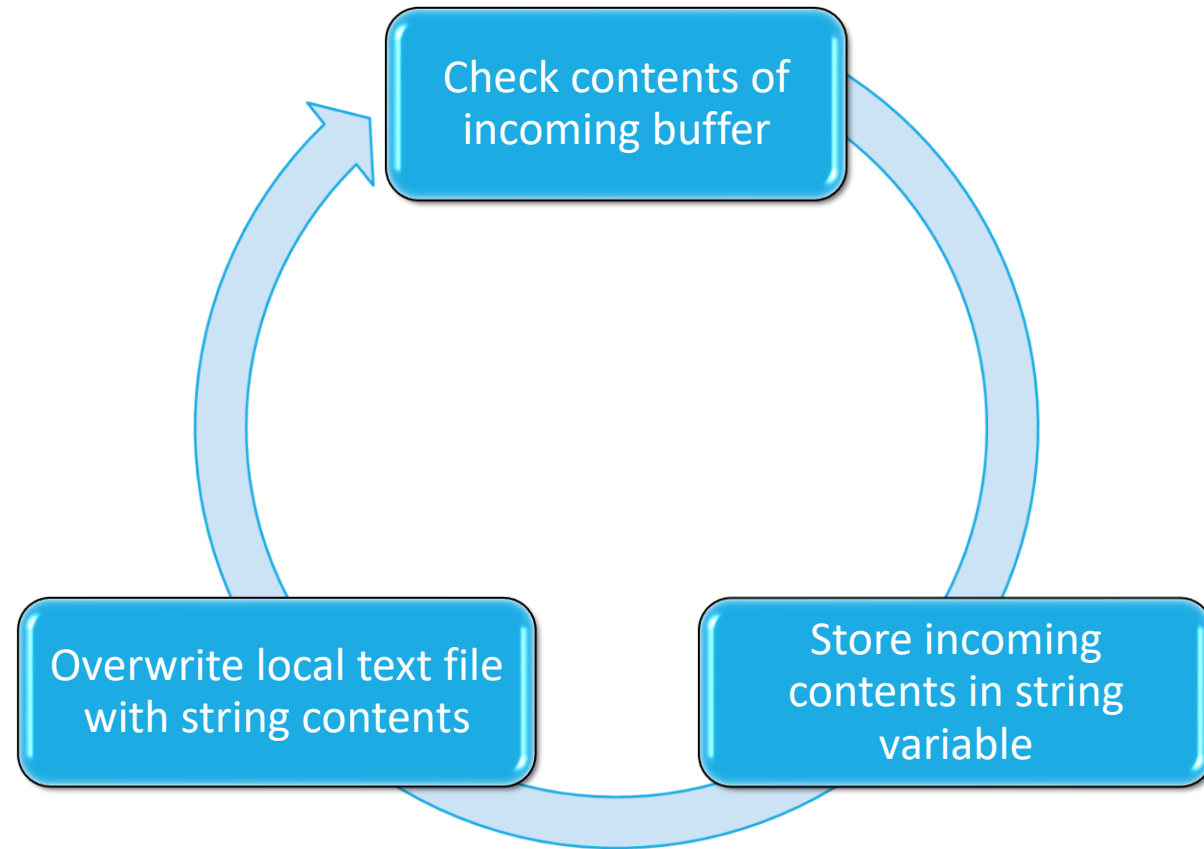


Arduino MD Program Structure



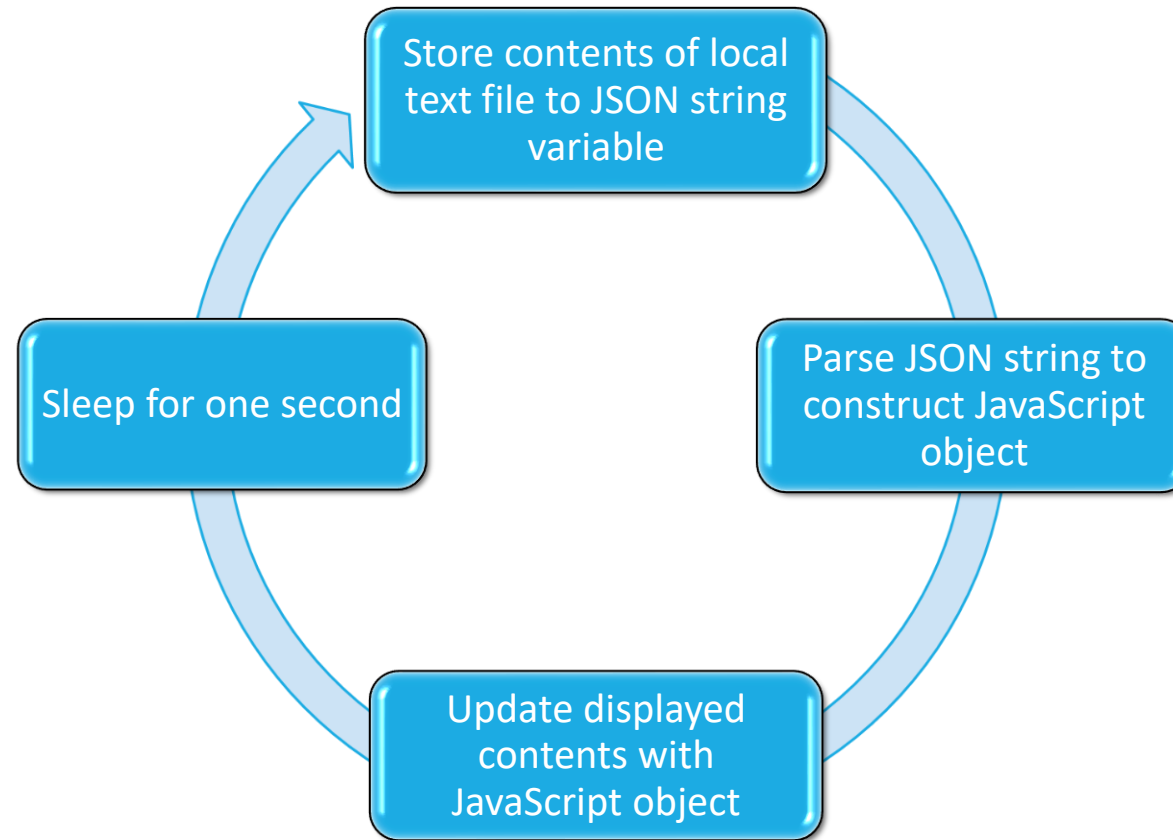


Raspberry Pi Python Script Structure

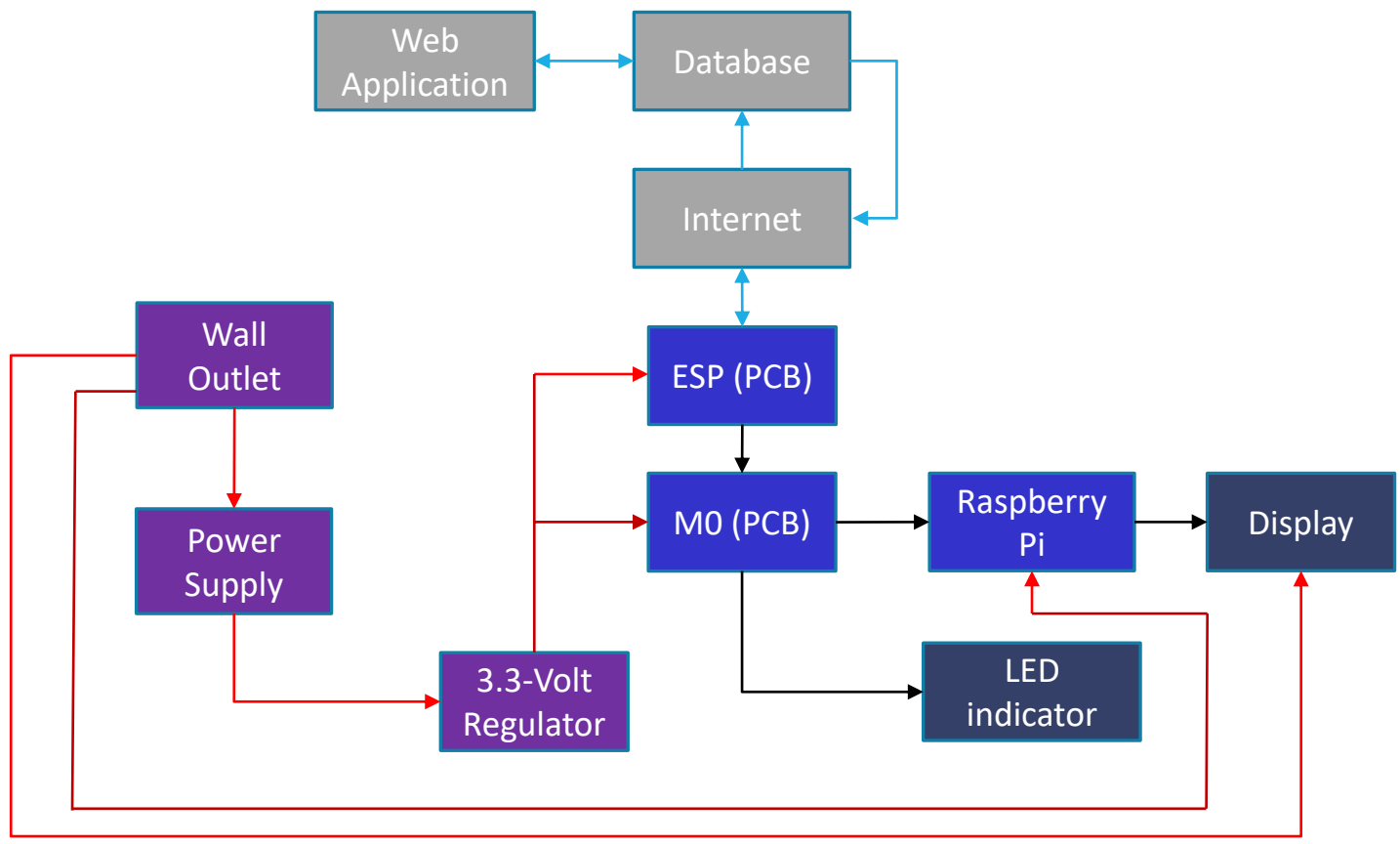




Raspberry Pi Panel Application Structure



Hardware Components



Legend for the diagram:

- Black arrow: Connection line
- Red arrow: Power line
- Light blue arrow: Wireless

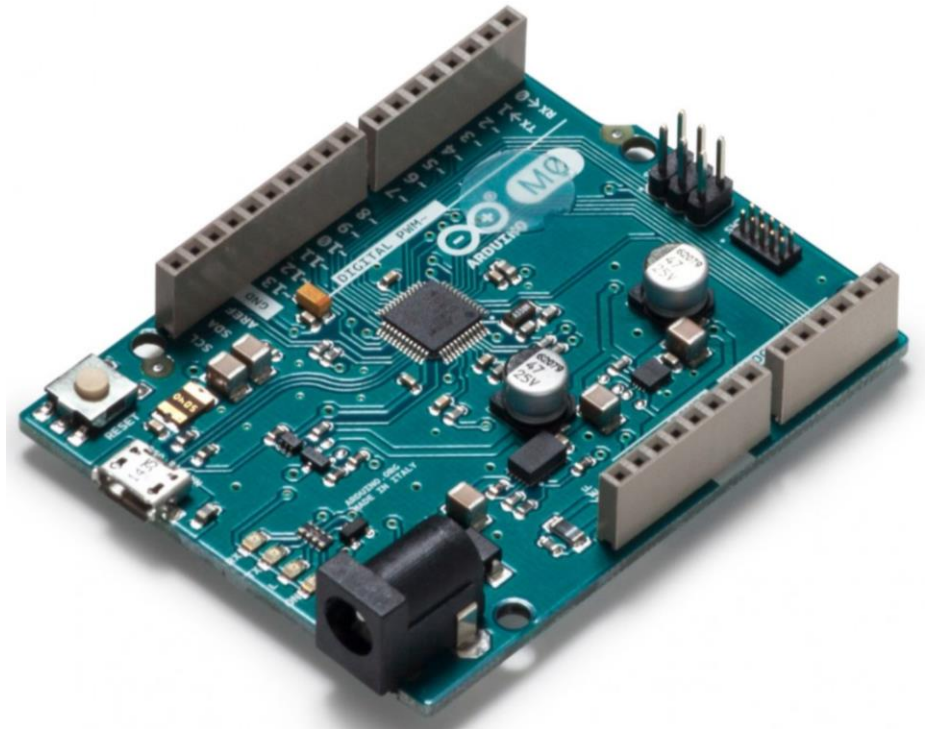
Power	Output
Software	Hardware

Arduino MD Microcontroller



Specification

- Atmel's SAMD21 MCU
- 32-bit Cortex
- 48-pin
- 32 KB SRAM
- 48 MHz Clock Speed
- 3.3V Operating Voltage
- 256 KB Flash Memory
- 14 Digital I/O Pins (12 PWM, 2 UART)

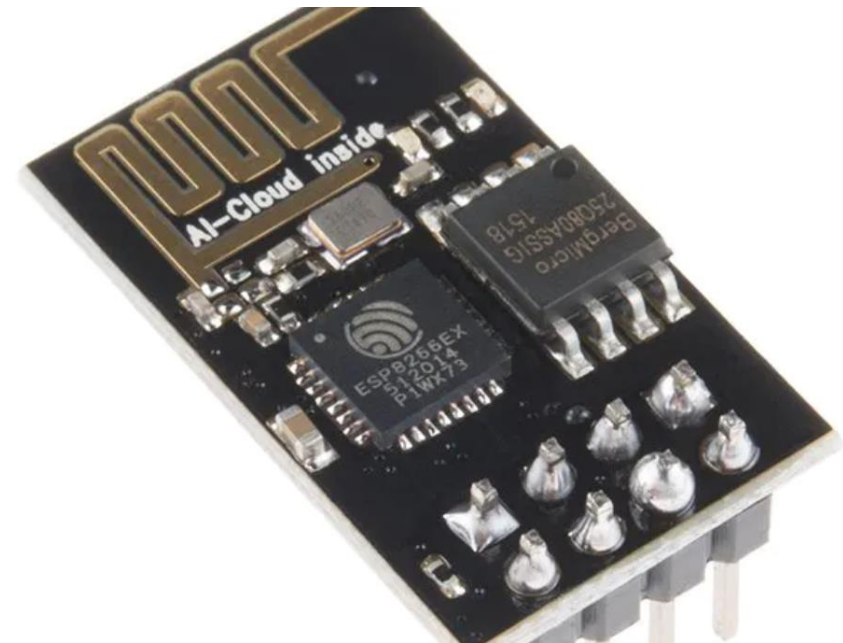


ESP8266-01 Wi-Fi Module



Connects system to Wi-Fi network

Requests information from database for Panel Application to read

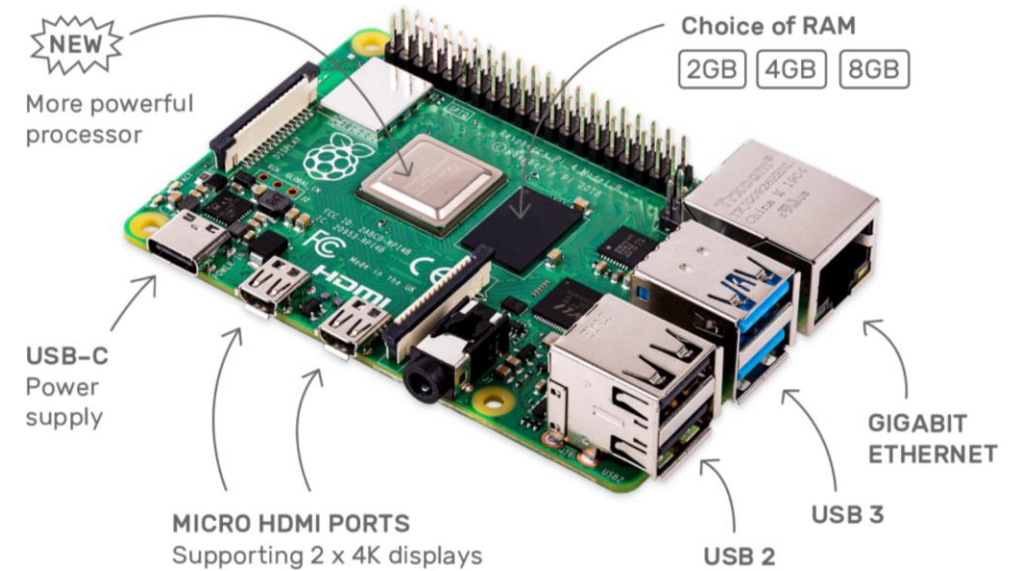


Raspberry Pi 4

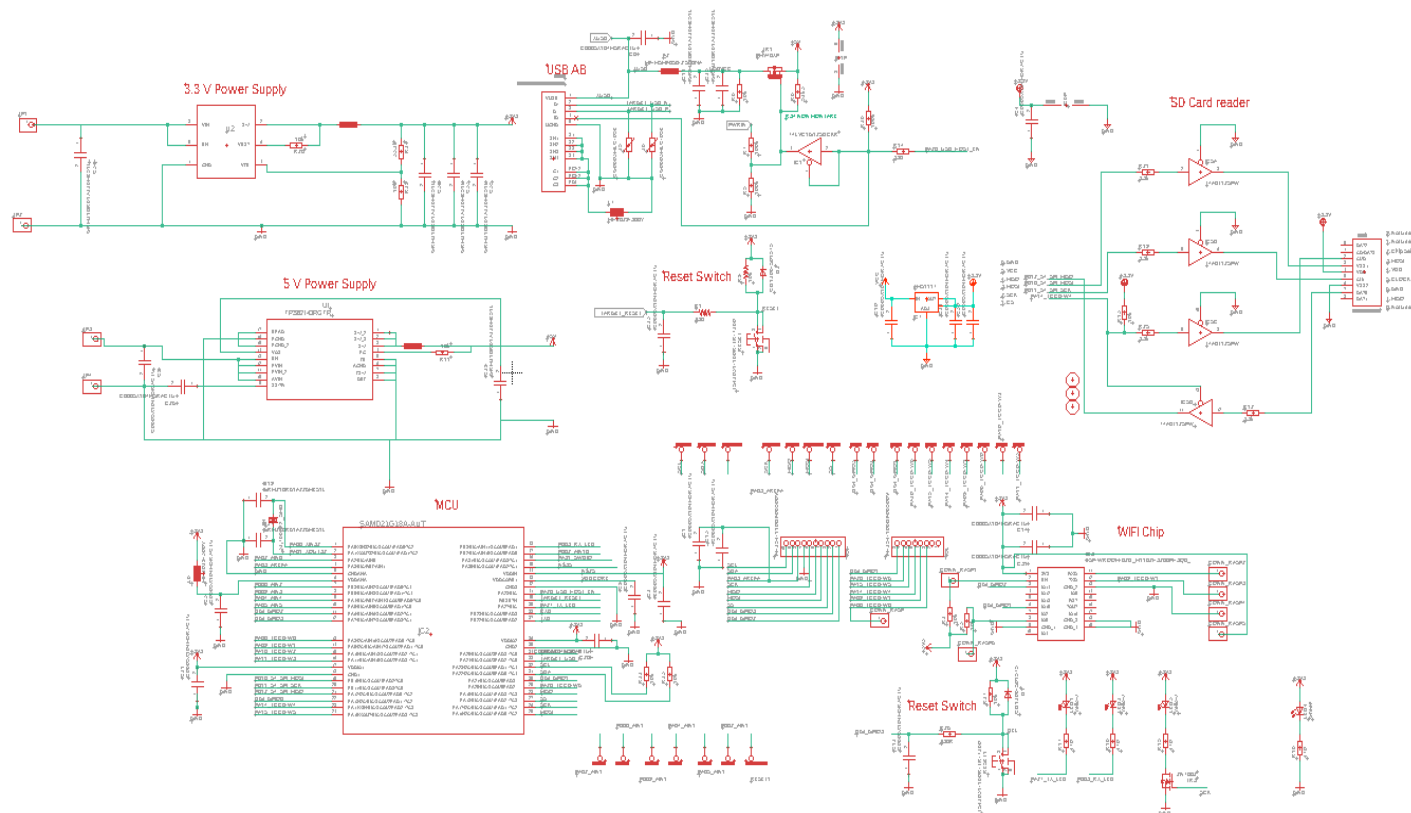


Specification

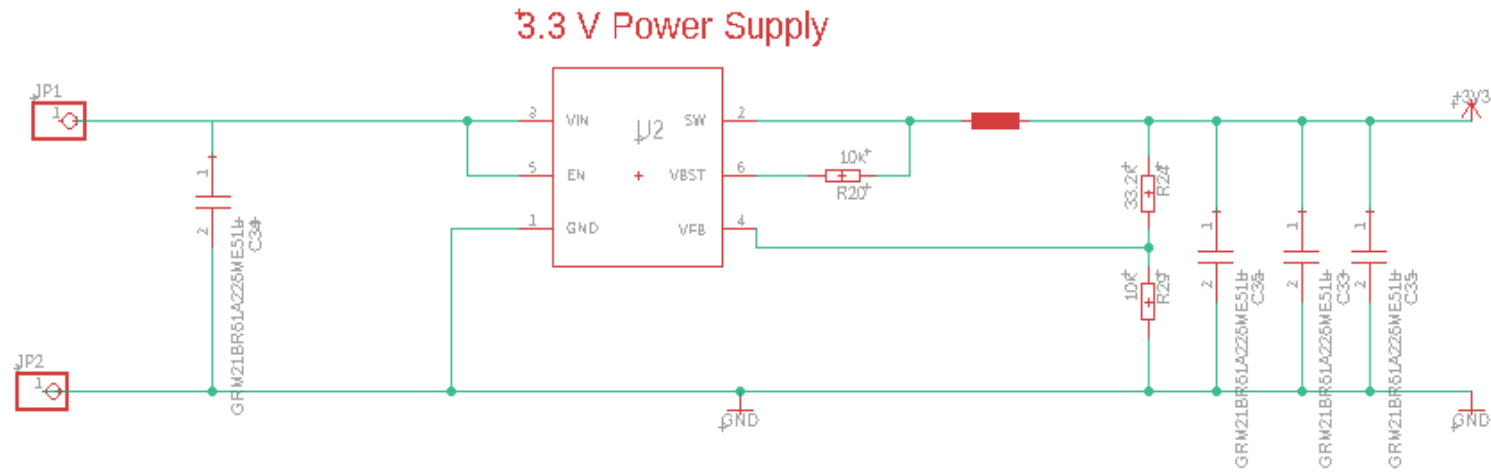
- Broadcom BCM2711, Quad core: 64-bits 1.5Ghz
- 1GB SDRAM
- 2x USB 3.0 ports
- 2x USB 2.0 ports
- 2x Micro HDMI ports
- 5V DC (USB and pin headers)
- Wi-Fi/Ethernet Capabilities
- 5V, 2.5A USB-C Power supply



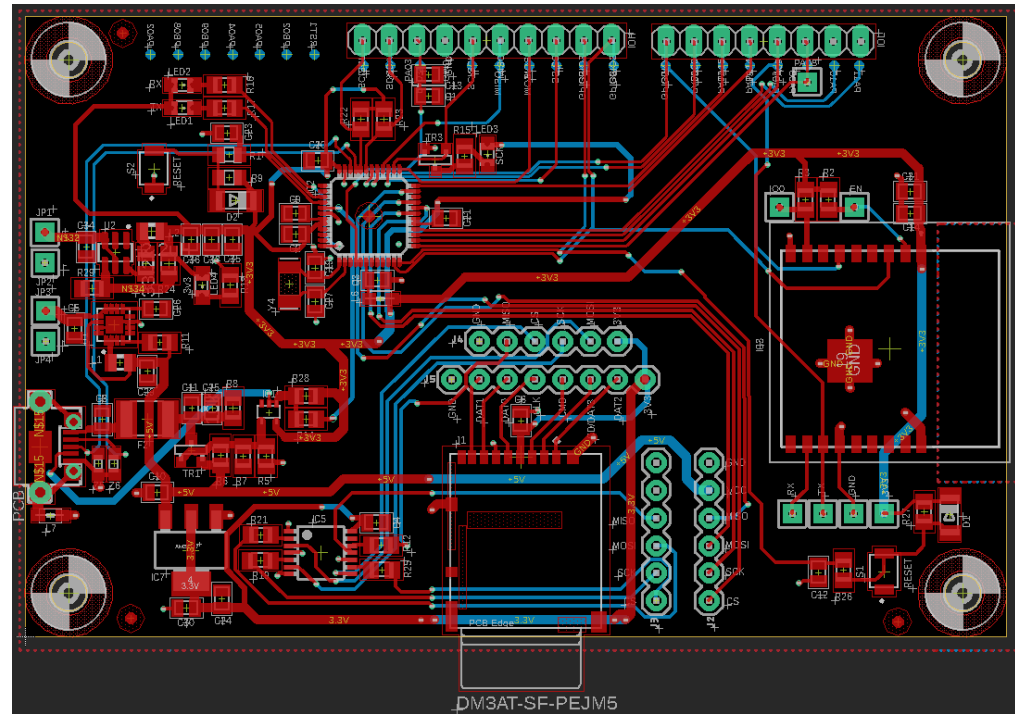
Printed Circuit Board Schematic



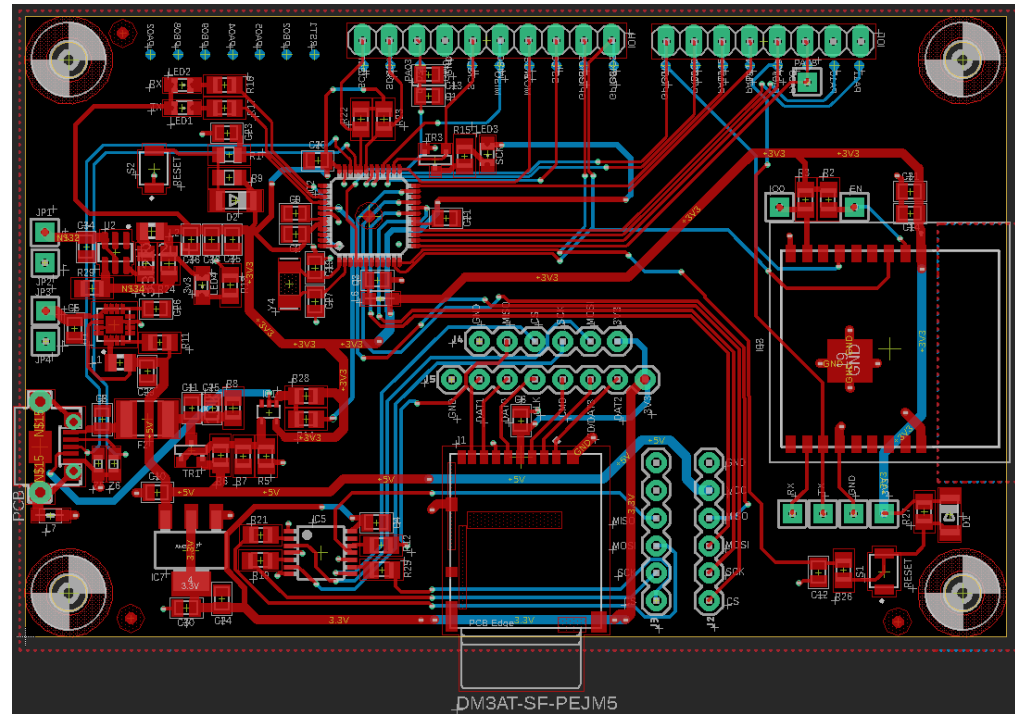
Power Supply Schematic (3.3V)



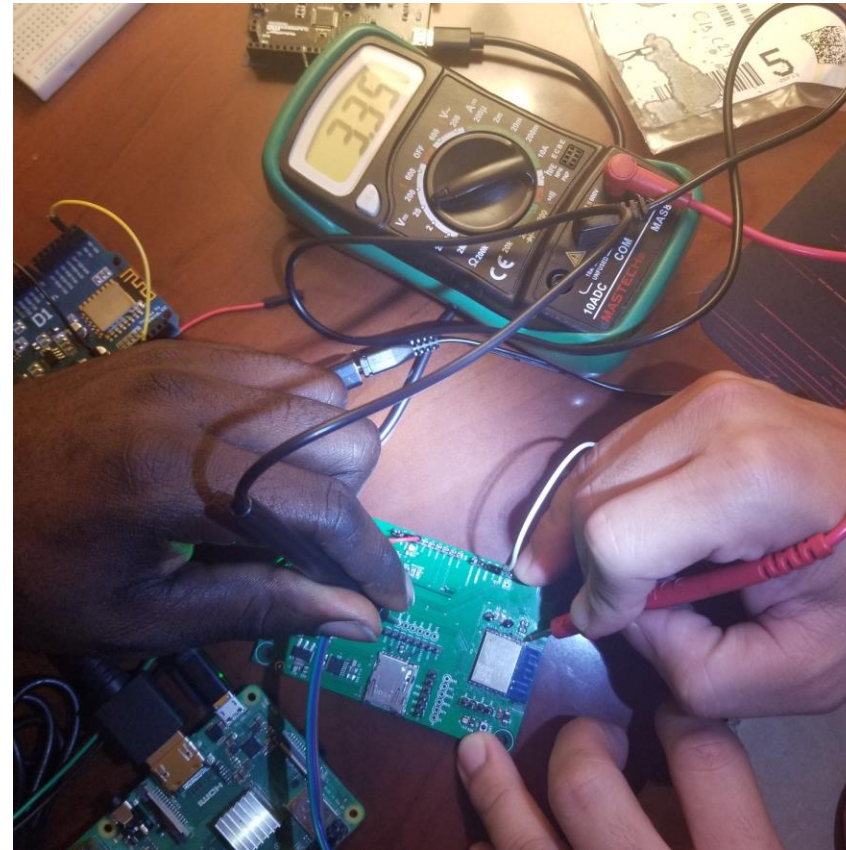
Printed Circuit Board Layout



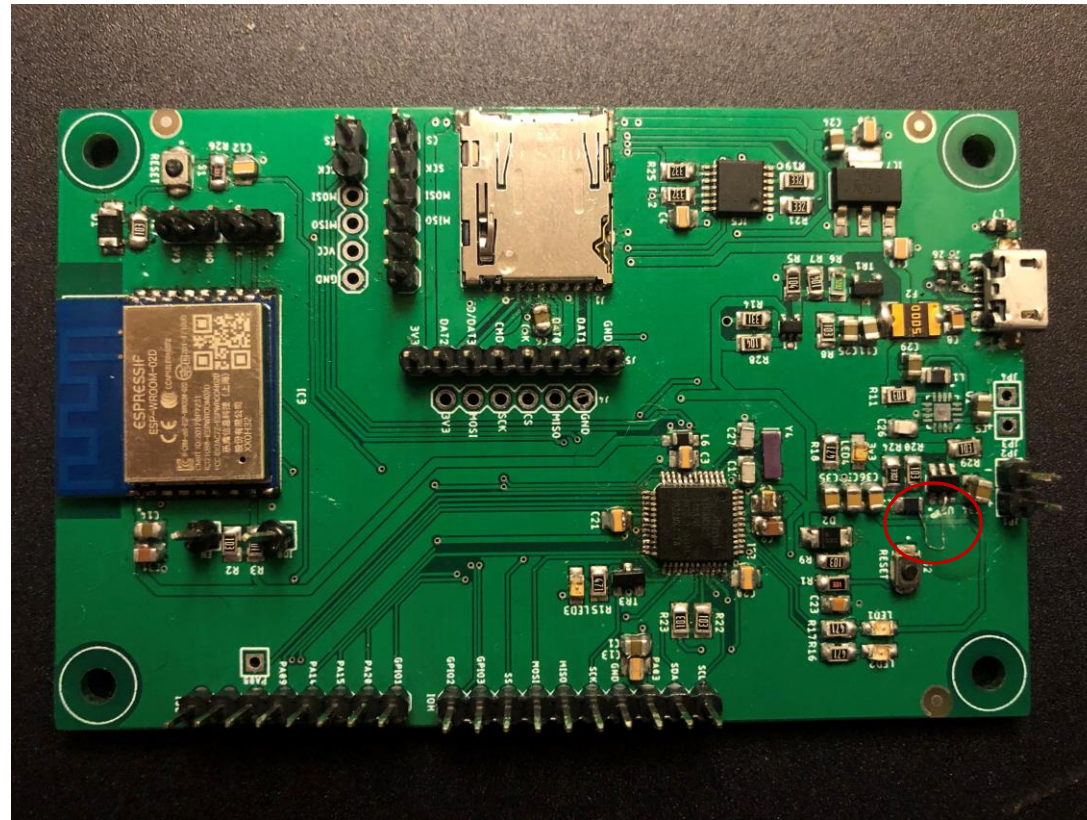
Printed Circuit Board - Challenges



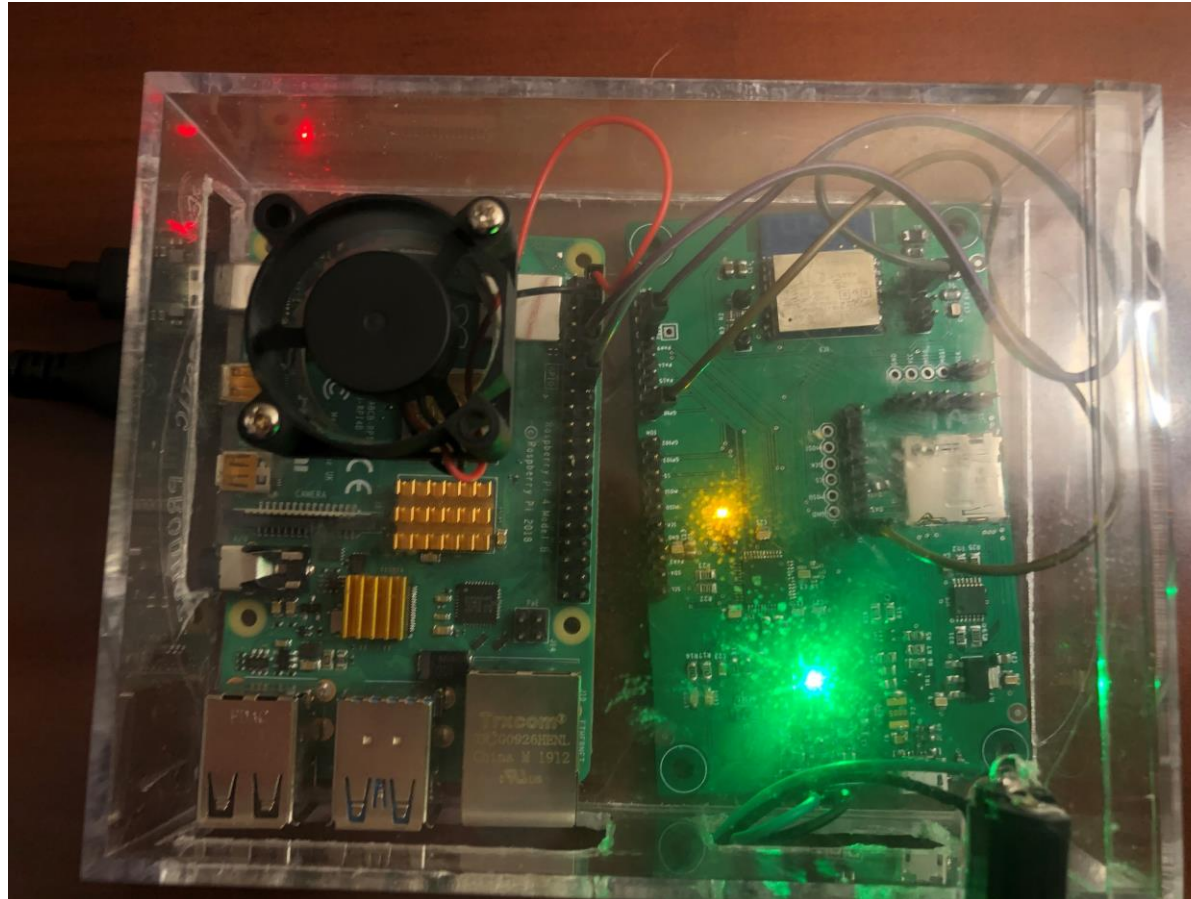
Printed Circuit Board - Challenges



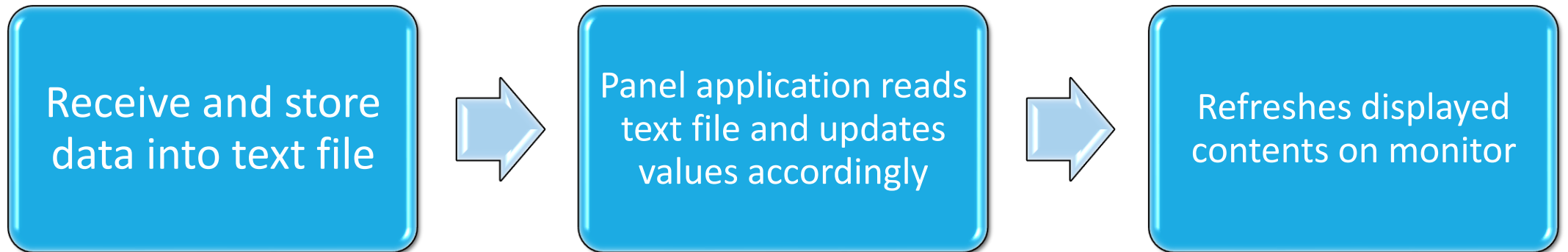
Printed Circuit Board – Soldering Components



System Enclosure



Raspberry Pi to Monitor



Constraints and Restrictions



- Self-funded (restricted budget)
- Shipping delay for online purchases
- Physically meeting with group members (geographical distance)
- System communication buffer size limit (64 bytes)
- Needed monitor with HDMI port

Budget



Item	Quantity	Price
Monitor	1	-
Printed Circuit Board	1	\$20.00
Raspberry Pi 4	1	\$40.00
PCB Components	82	\$60.00
Arduino M0	1	\$10.00
ESP8266	1	\$10.00
System Enclosure	1	\$15.00
Total Price		\$155.00

Broader Impacts of IntelliDate



IntelliDate could serve as an extremely helpful tool for the elderly and disabled, as it can keep users' lives organized and visually present on a large screen with large readable font, and its contents can be updated with a keyboard, rather than having to write.

- The feature of updating the calendar's contents from any location (via the website) allows for a caretaker to add/edit events and notes for someone else, in a separate physical location, to read.