April 15th, 2020

THE BARTENDER BUTLER BOT

University of Central Florida

Senior Design II - Final Presentation

Group 13

Yianni Babiolakis	yi657277	Electrical Engineering
Rachael Caskey	ra231875	Electrical Engineering
Edward Nichols	ed891610	Electrical Engineering
Corey Scott	co079592	Electrical Engineering

Overview

Motivations &
Technical Objectives

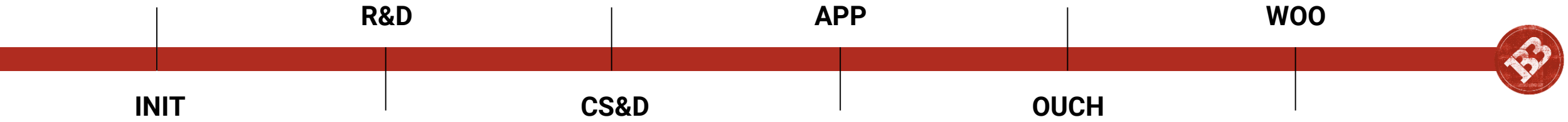
Or what are we doing & what's the point?

Component Selection &
System Design

Or what is the *fastest* route to a functional prototype?

Challenges Overcome

Or how have our efforts panned out thus far and what is left for us to do?



Research &
Design Plan

Or what are the practical boundaries of the project?

System Integration &
The User Interface

Or what will the overall system look like and how is it tied together?

Administrative Details

Budgets, schedules, roles & responsibilities - as per ABET.

What is the B3?

Automated Beverage Mixer

A personal mixologist for classy and exotic cocktails

+

Autonomous Delivery Bot

A dedicated drink-running bot for ready, continuous and convenient service

+

System Integration Software & UI

A seamless user-interface for easy event-hosting

Bartender Butler Bot

An integrated luxury home-appliance for tireless party hosts, posh socialites, and robotics enthusiasts alike

Motivations

45%

The Core Problem is Relatable

The best and worst aspects of being a good and dutiful partymaking host resonate with us:

- Taking care of guests is work
- Missing out on conversations, games, or moments for dutiful partymaking errands
- “I’ll fetch us another round!”
- “Oh, what did I miss?”

40%

The Technical Challenges are Alluring

The core problem demands a comprehensive multi-faceted technical approach which yields a bounty of open-ended design questions:

- Automation of beverage services
- Autonomous package delivery
- System integration for simple & seamless user experience
- Packaging for functionality vs. manufacturability vs. aesthetics

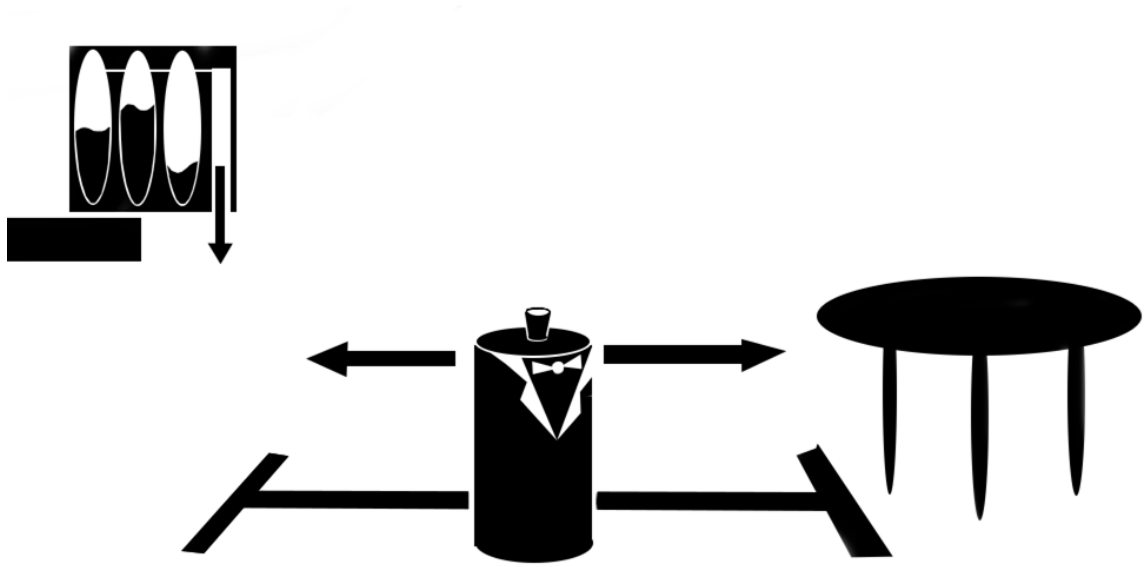
15%

The Commercial Space is Enticing

The prototype & development journey to and beyond commercial validation of a minimally viable product is itself intrinsically exciting:

- Bespoke designs vs. modular approach to scaling
- New & empirically derived market-ready design objectives
- New & improved technical challenges

Overview - B3



Overall

- **Automate** the multistep process of **mixing and delivering a beverage** across a predefined path

Corollaries

- **Maximize** the number of **off-the-shelf** or open-source subsystems
- **Minimize** the number of **mechanical** subsystems
- **Minimize** the overall **cost** of the system

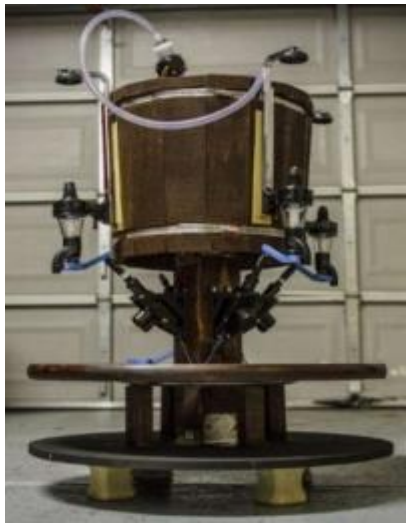
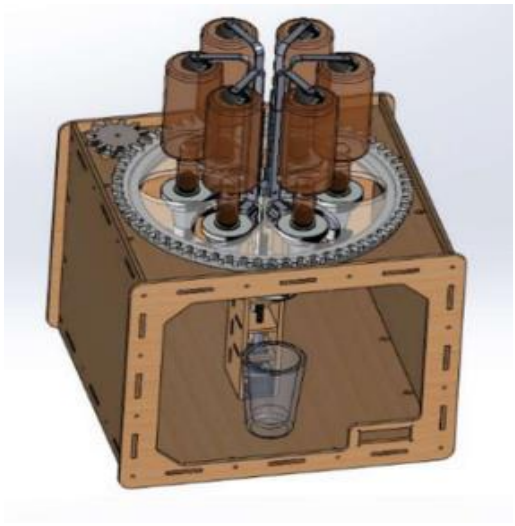
Research

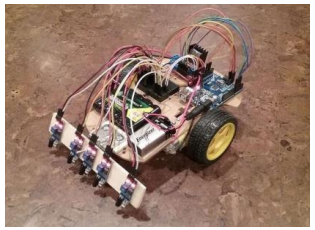
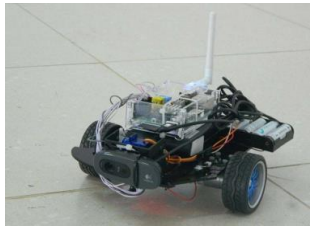
Bartendro
- Raspberry Pi
- Peristaltic Pumps

SirMixaBot
- Peristaltic Pumps

RaspRobot
- OpenCV
- Object Tracking Technique

Arduino Elliot
- IR Sensor Array
- PID Line Following

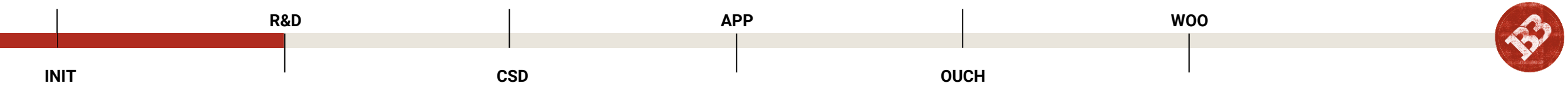



INIT | R&D | CSD | APP | OUCH | WOO

Requirement Specs

		Unit	Requirement	Achieved?	
1	**Max. Avg. Bartender Configuration Time**	Minutes	< 10	✓	
2	**Ordering Time**	Minutes	< 1	✓	
3	**Beverage Mixing Time**	Minutes	< 2		
4	**Delivery Time**	Minutes	< 5		
5	**Min. Capacity for Beverage Ingredients**	Ingredients	> 3	✓	
6	**Drink Accuracy**	% Target Vol.	< 5%		
7	**Delivery Range**	Meters	> 10	✓	
8	**Charge Lifespan**	Minutes	> 90	✓	
9	**Obstacle Collisions**	Num/journey	< 3	✓	
10	**Bartop Appliance Dimensions Target**	cm^3	< 50,000	✓	
11	**Total Appliance Cost**	USD	< $1500	✓	

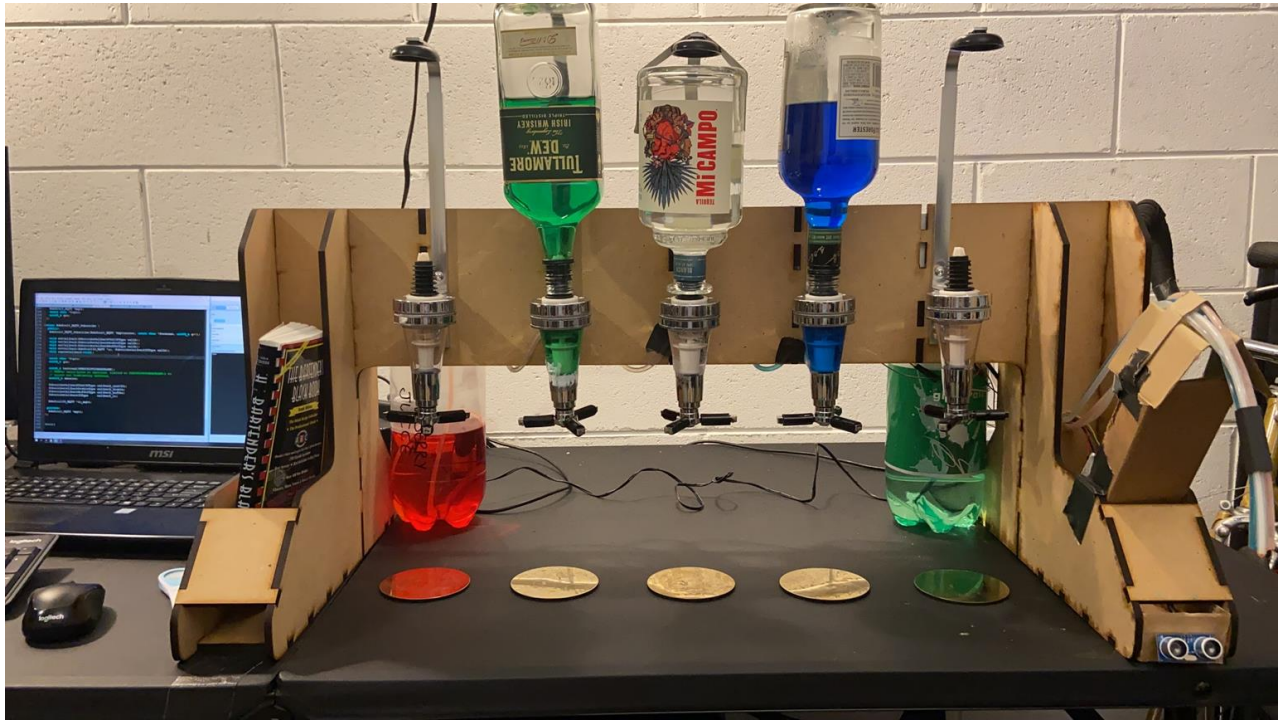


Minimum Capacity for Bev. Ing.

Original Objective:
- **At least** 3 ingredients total

Current Benchmark:
- **Up to** 5 mountable ingredients
- **Up to** 3 backup/bulk ingredients
- Dual pumps option

Achievable Hypotheticals:
- Modular expansion to pump system
- Stronger/faster pumps
- Easy-clean pumps



Minimum Charge Lifespan

Original Objective:
- **Greater than** 90 minutes total

Current Benchmark:
- **6+ Hours** total lifespan
- 36+ Round Trips (Low Distance)

Achievable Hypotheticals:
- Converge on single battery-pack
- Tap into iRobot Power

Minimized Unit Cost

Original Objective:
- **Less than $1500** (US 2019) total CoGS

Current Benchmark:
- Bartender: **~$300**
- Butler: **~$570**
- Application **~$110**
- Misc/Gen: **~$320**
- Total: **$1300**

Achievable Hypotheticals:
- Focus on design-for-manufacturability
- Scale production of custom PCB
- Bulk purchases on components
- Etc.

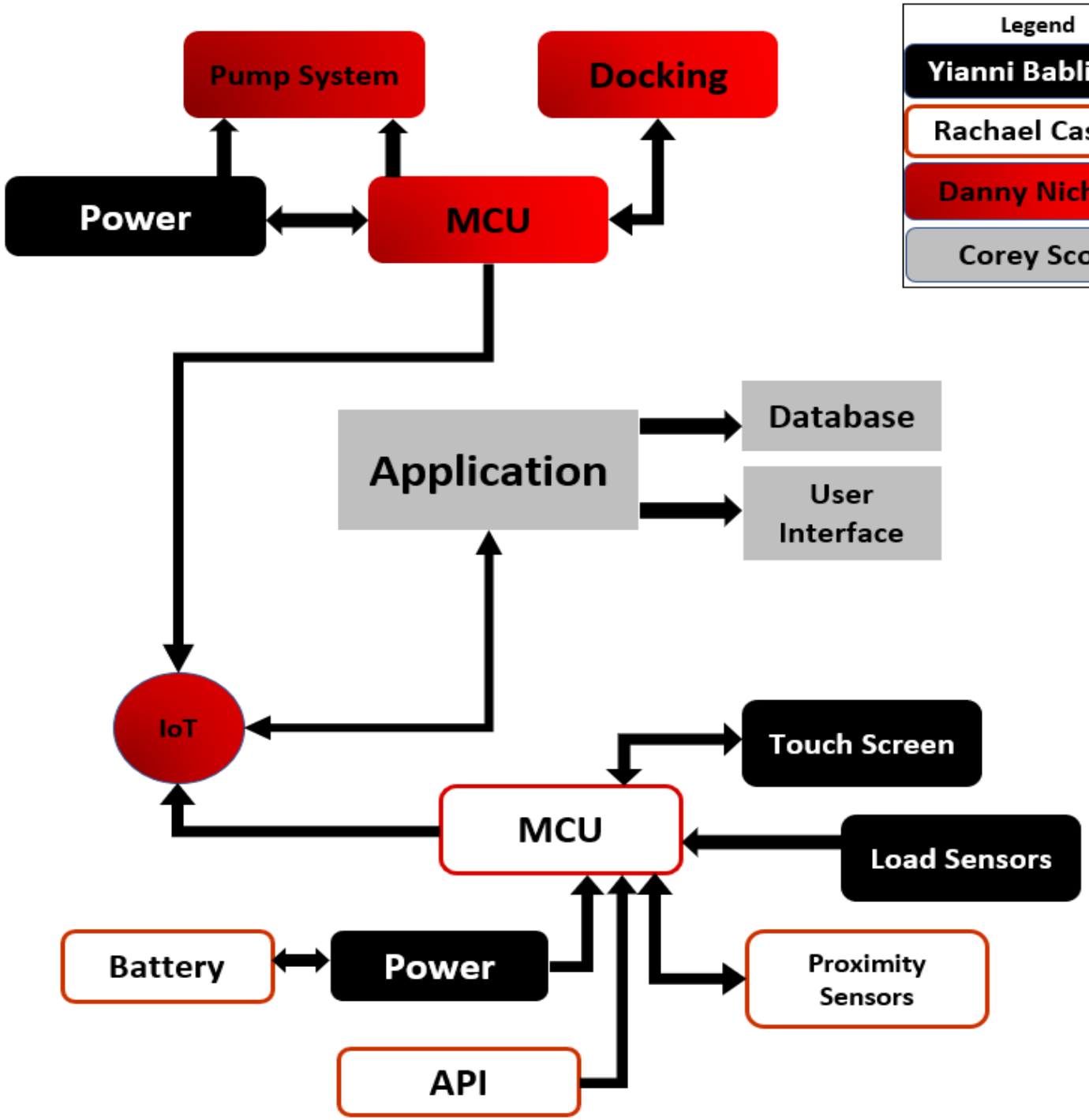
Part No. and description	Size (mm)	Qty	Unit price (USD)	Amount (USD)	Shipping charge (USD)	Bank fee (USD)	Total (USD)
Printed Circuit Board: FR-4 2layers No. :W302224AS1K1 b3 butler- rev00 - gerbers	68*108	10	11.5	115.0	2.0	6.0	123.0
Printed Circuit Board: FR-4 2layers No. :W302224ASY3 b3 bart - rev01 - gerbers	101.6*63.5	5	20.8	104.0	2.0	5.0	111.0
SMT No. :T-1K2W302224A (W302224AS1K1) Components cost:$20 Assembly cost:$30		0		50.0	1.0	3.0	54.0
SMT No. :T-Y4W302224A (W302224ASY3) Components cost:$64 Assembly cost:$88		0		152.0	1.0	6.0	159.0

Summary	
Total products amount (USD)	$421.0
Total Freight cost (USD)	$6.0
Bank fee (USD)	[illegible]
Total (USD)	$447.0

PCBWay.com Bank information

OUCH!

INIT | R&D | CSD | APP | OUCH | WOO



SYSTEM DIAGRAM

Responsibilities Highlight

ADMINISTRATION:
Roles & Responsibilities

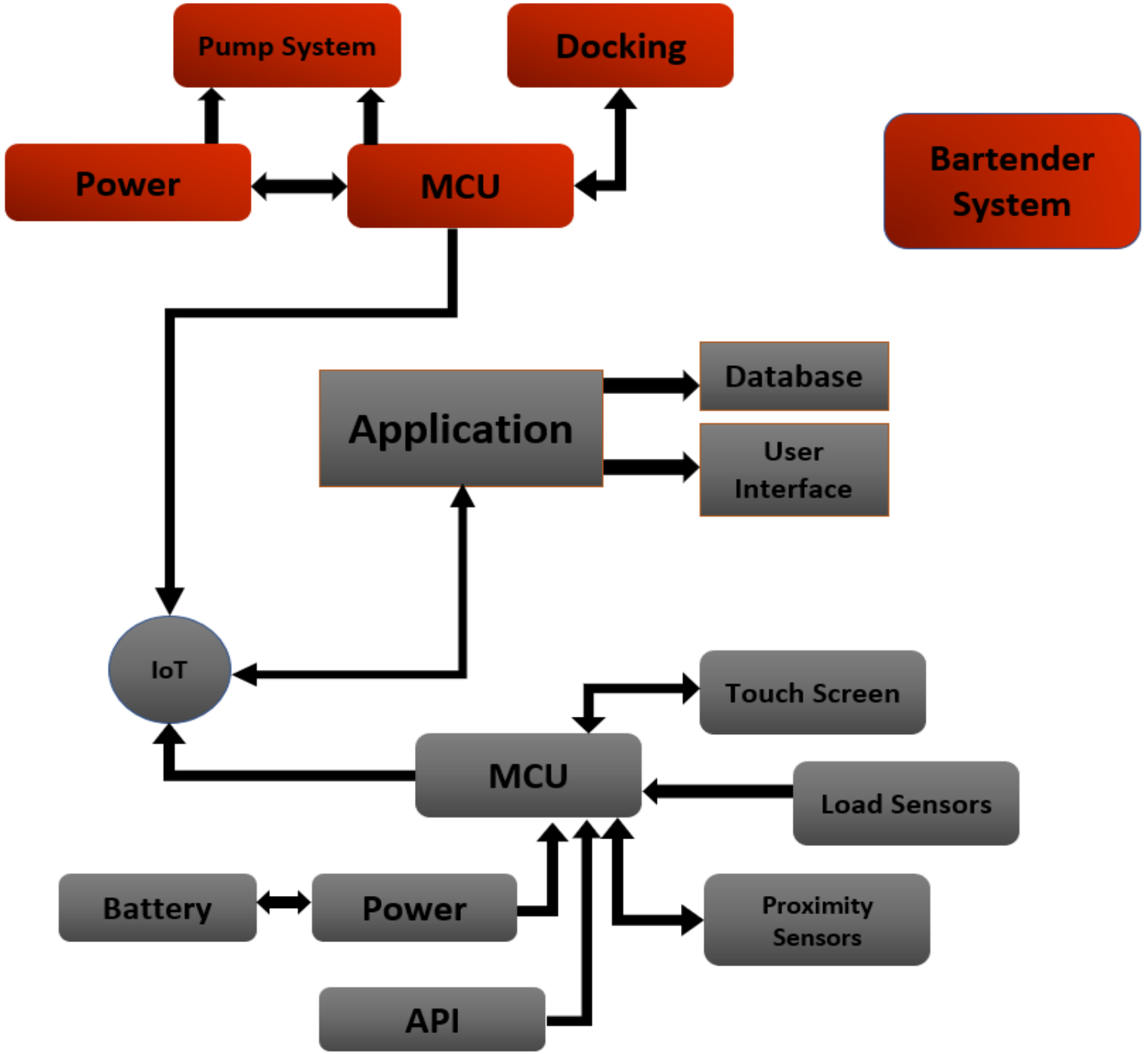
Main: ✓ Assisted: 

Task	Pump System	Bartender MCU	Internet of Things Platform	Application	User Interface	Database	Butler MCU	Butler Proximity Sensors	Butler API	Butler Battery System	Butler Power System	Bartender Power System	Butler Load Sensors	Butler Touch Screen
Yianni Babiolakis											✓	✓	✓	✓
Rachael Caskey							✓	✓	✓	✓				
Danny Nichols	✓	✓	✓											
Corey Scott				✓	✓	✓								

R&D APP WOO

INIT CSD OUCH

System Breakdown



THE BARTENDER

Sub-System Highlight

BART: Hardware Comparison

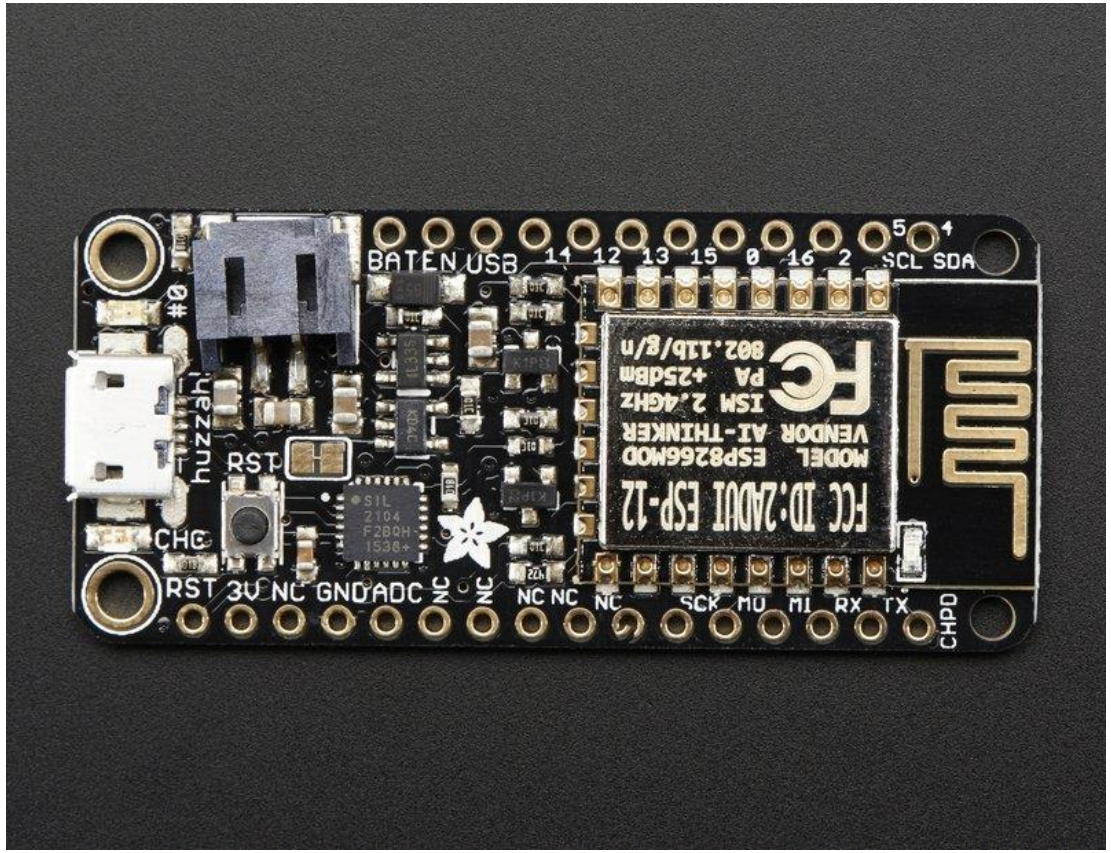
	MT ARM 7697 (Media Tek)	ESP 32 Dev (Espressif)	ESP 8266 HUZZAH (Espressif + Adafruit)
Price	~$40	~$15	~$17
GPIO Pins	28	34	17
I2C | SPI	✔	✔	✔
802.11 b/g/n/ WiFi	✔	✔	✔
Hardware Libraries		✔	✔
Community Rating (1 - 5)	**1**	**4**	**5**
Accessibility Rating (1 - 5)	**2**	**3**	**5**

BART: Hardware Selection

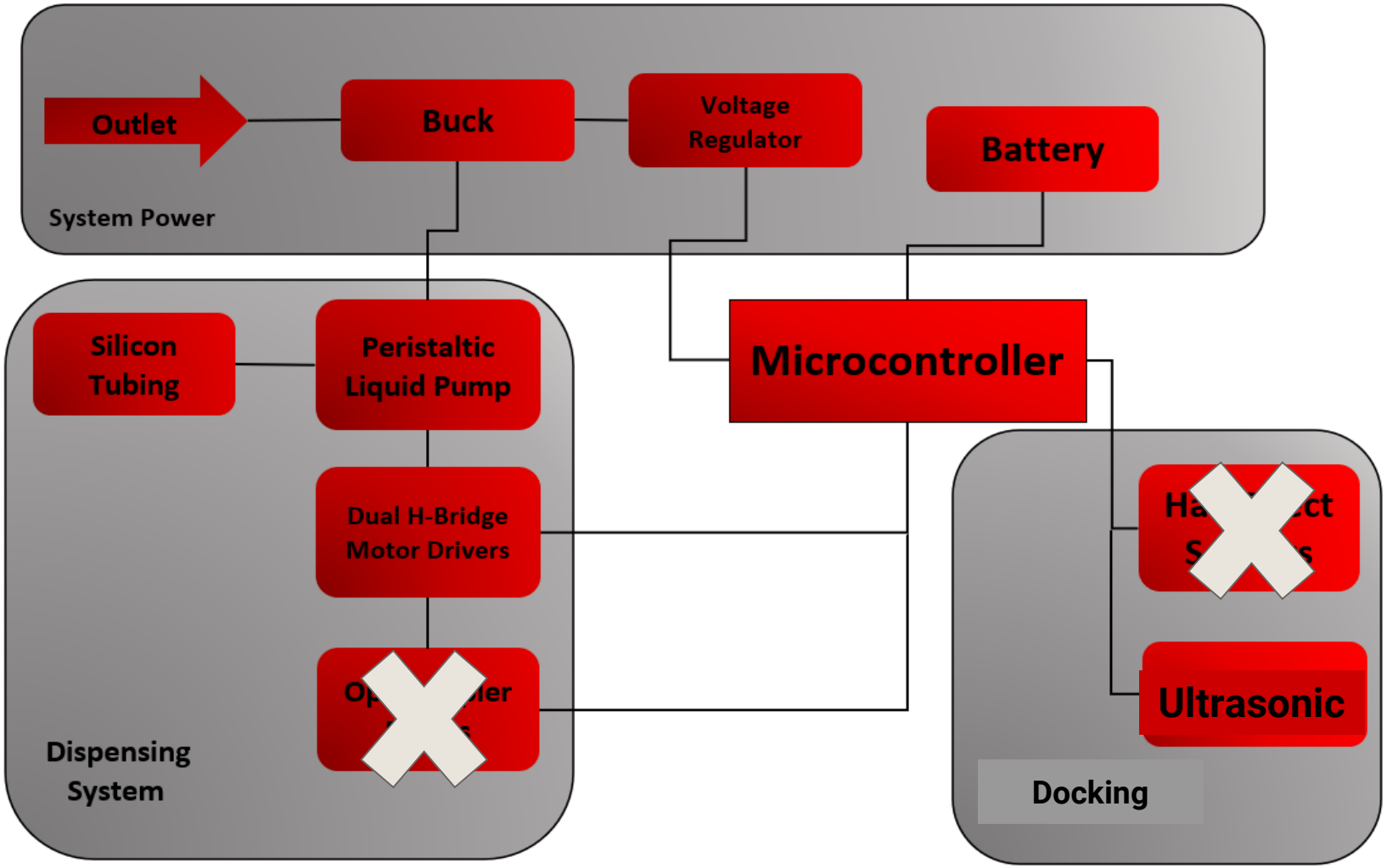
Microcontroller - comparison
- HUZZAH Feather (Adafruit) based on…
- ESP8266 (Espressif)

Dispensing Recipe
- Peristaltic Pumps (Adafruit)
- Silicon Tubing (McMaster-Carr)
- ~~Opto-Coupled Relays (Songle)~~
- DC Motor Driver (Texas Inst.)
- 74HC4051 8-Channel (Texas Inst.)

Docking/Alignment Check
- HC-SR04 UltraSonic Proximity (ElecF)



BART: Design Diagram

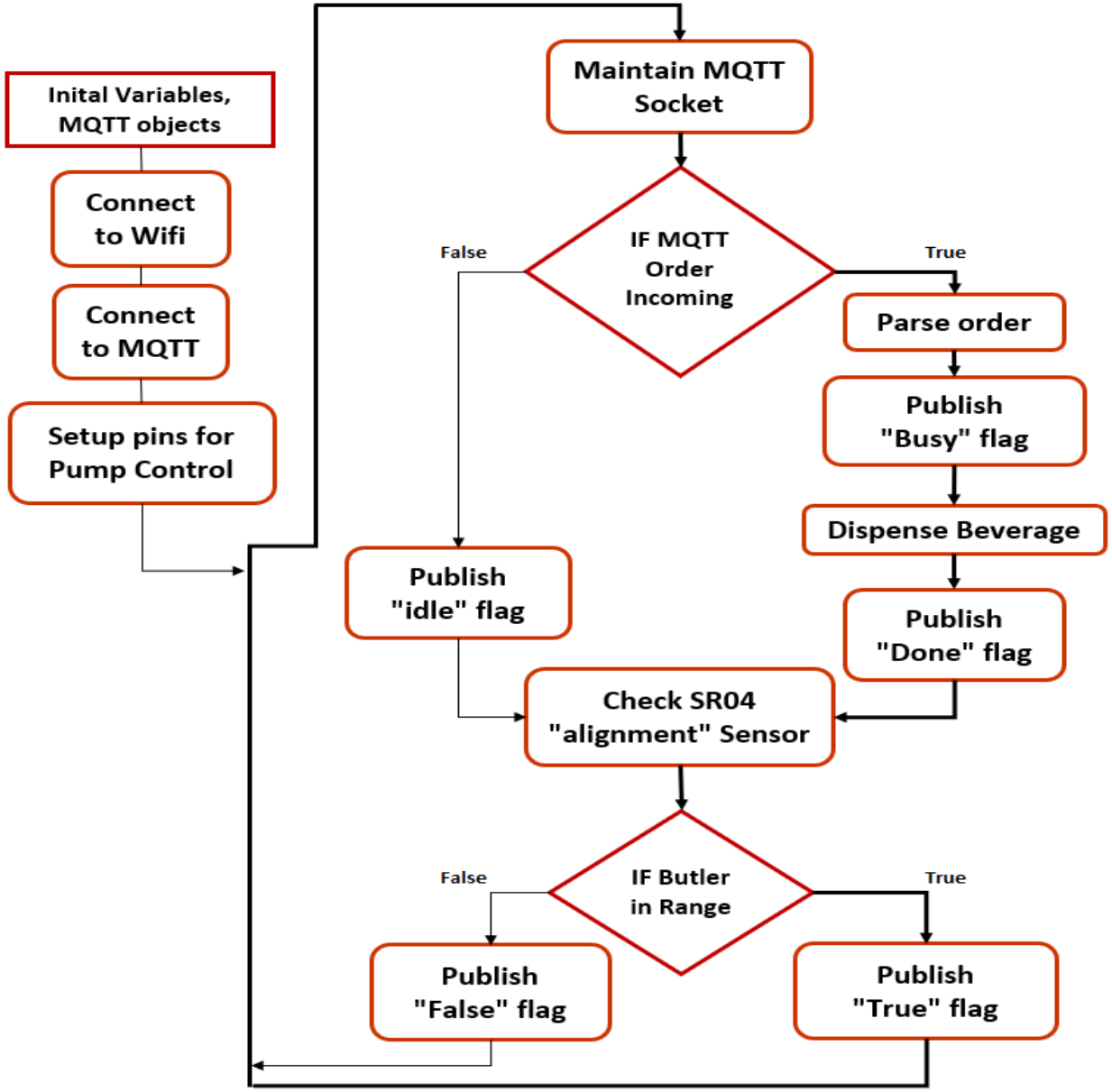


BART: Firmware

Connect to IoT Hub
- DHCP IP assignment (WiFi)
- Connect to broker & subscribe to data topics (MQTT)

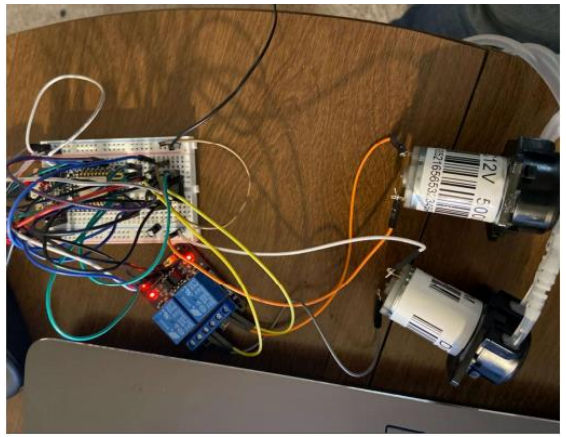
Readiness Checks (Loop)
- Validate container alignment with nozzle (ADC)

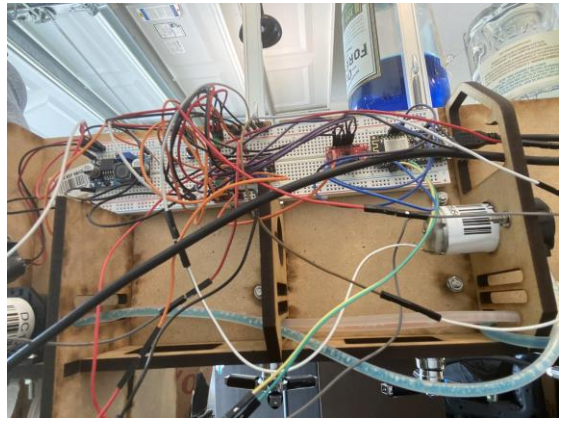
Dispensing Recipe
- Receive beverage order (MQTT)
- Sequential pump action (PWM)
- Release for delivery (MQTT)



BART: Testing

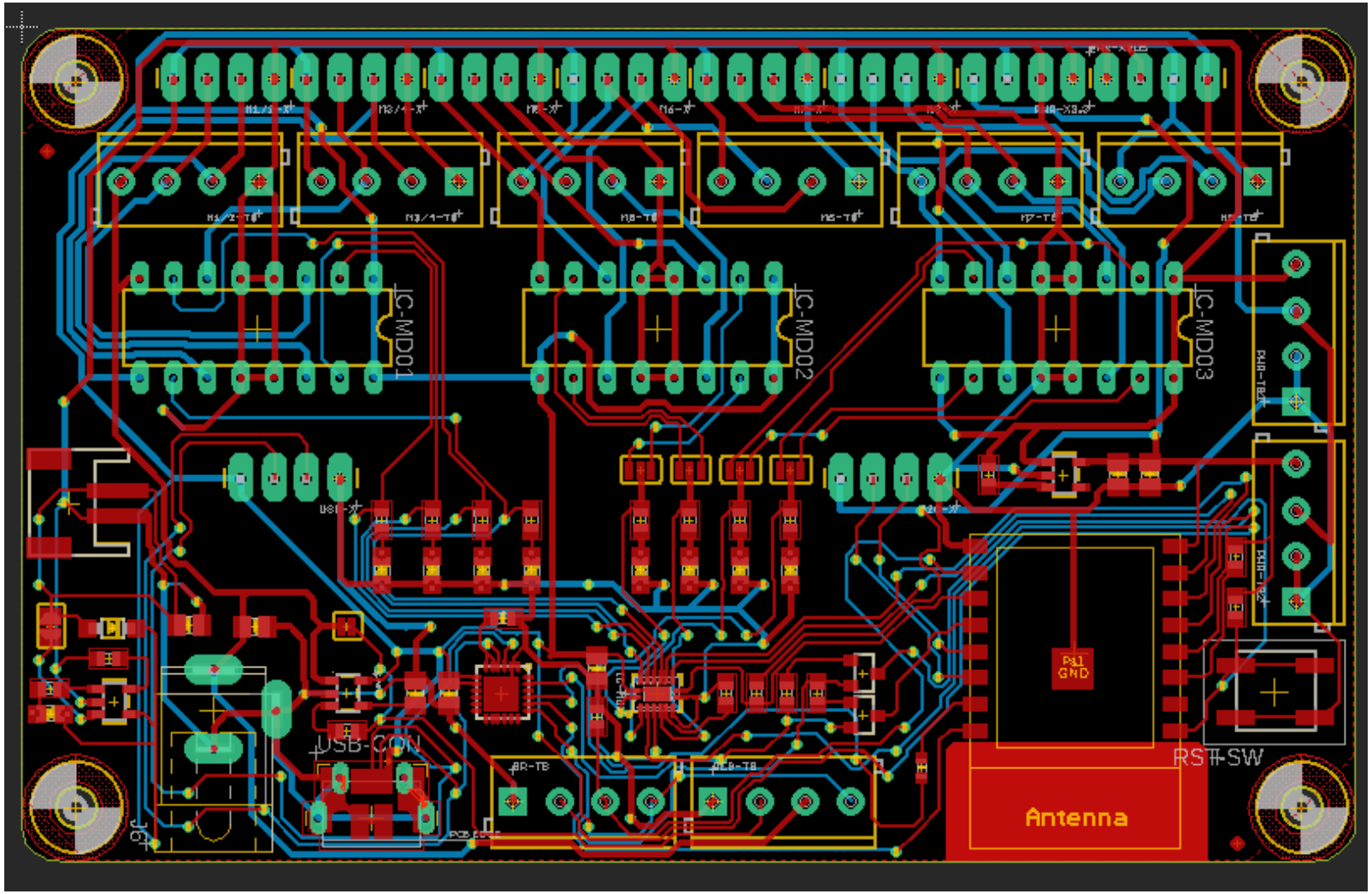
- **WiFi**
  - WPA Authentication
  - Reconnect on fail
- **Pump Control**
  - Selection via MUX
  - Arbitrary control
  - Temperature / Power
- **Proximity Sensor**
  - Readings & Sensitivity
- **MQTT**
  - Maintain connection
  - Publish to broker
  - Subscribe to topics





R&D | APP | WOO

INIT | CSD | OUCH



Bartender PCB Design

- 2-Layers of 1-oz Cu on FR-04
- 10 mil STD / 18 mil MAX
- Through-Hole Motor Drivers for easy replacement
- 60 unique SMD components

- Manufacturing and Assembly: PCBWAY in Hangzhou, China

ETA: ~~May 3rd, 2020~~

April 29th, 2020

BART: Issues Encountered

MQTT Libraries

- **Adafruit_MQTT library failed** to reliably subscribe to topics
  - No response during “ProcessPackets()”, and library definitions not useful
  - Required for basic functionality
  - **switched to PubSubClient**
- Rewrote swathes of the core firmware related to MQTT
  - Misalignment of MQTT tags
  - Also **opportunity to redress logic**, for modularity and general cases

PCB Supply Chain Disruptions

- Factory **delays from prolonged labor shortage &**
- **increased lead times** in component procurement
- **Expedited shipping**

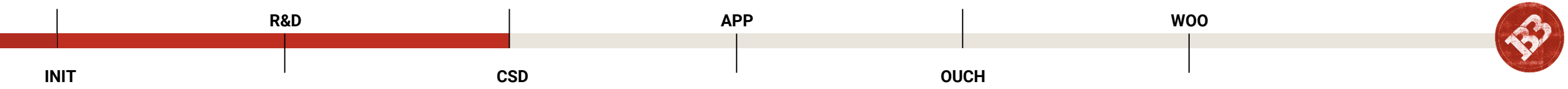
BARTENDER: Status

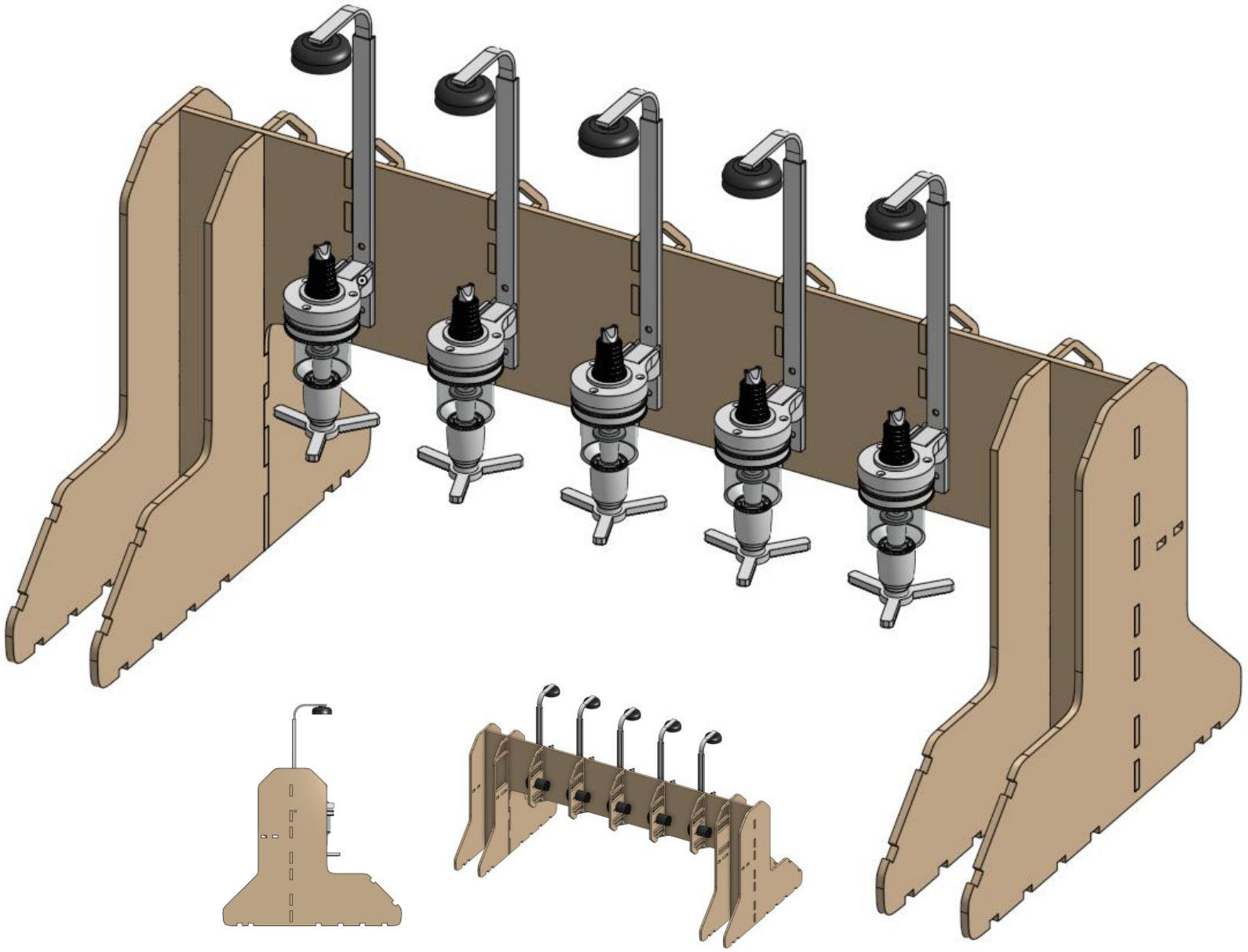
(Breadboard) Prototype/Firmware
- Power, relays, & motor drivers **OK**
- Connection to WiFi & MQTT **OK**
- Pump control via PWM **OK**
- Arbitrary recipe fulfillment via MQTT **OK**
- Logging & Troubleshooting Output **OK**

PCB Design
- Module schematics & footprints **OK**
- SMD component vendor sourced **OK**
- Schematic **OK**
- Layout **OK**
- Fabrication / Assembly **NO**
- Received Shipment **PEND**

Structural
- CAD for base components **OK**
- MVP CAD prototype **OK**
- Components sourced/ordered **OK**
- Structure manufacturing **OK**

Overall Integration
- MQTT tags validated **OK**
- MQTT integration live demo **OK**
- Repeatable/reliable response **OK**

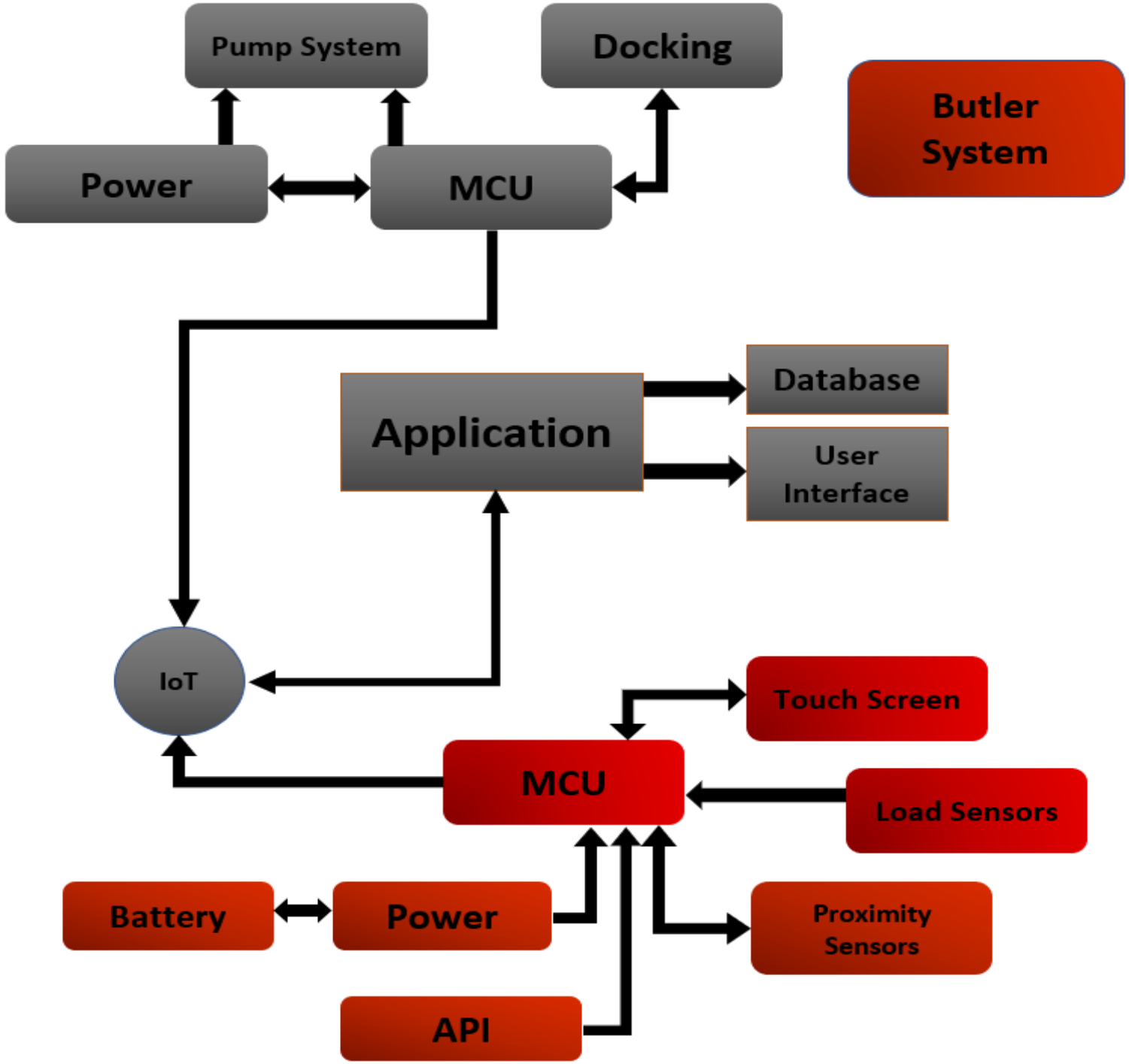




THE BARTENDER

LATEST 3D MODEL

INIT | R&D | CSD | APP | OUCH | WOO



THE BUTLER

Block Diagram Highlight

BUTLER: Hardware Comparison

	iRobot Create 2.0	Bare Motorized Base	Custom Build
Price	~$200	$30-$50	~$200+
Precise Alignment	Built-in Docking	None	Custom
Structural Support Rating (1-5)	5	3	3
Development Time Rating (1- 5)	**5**	**5**	**2**
Accessibility Rating (1 - 5)	**5**	**3**	**2**

BUTLER: Hardware Comparison

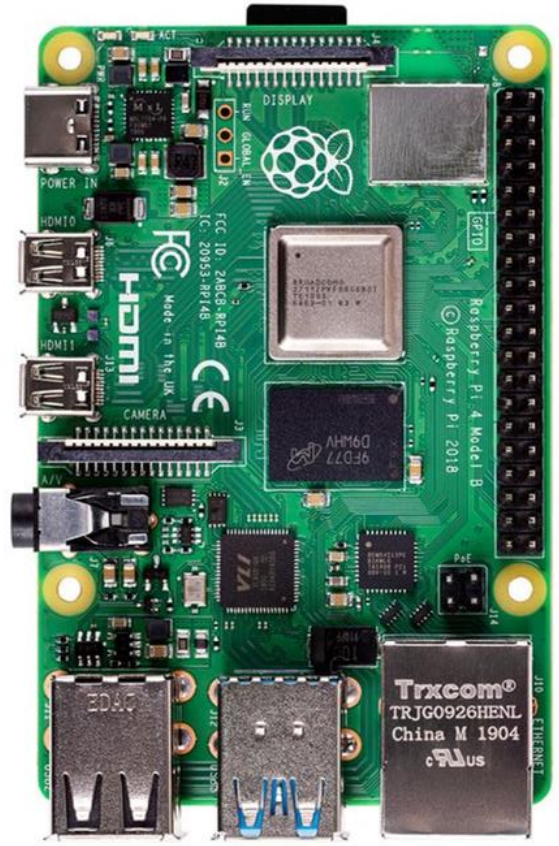
	Raspberry Pi 4.0	Intel NUC	Huawei Hikey 4
Price	~$30	~$150	~$250
GPIO Pins	40	None	40
OS	Raspbian Lite (Linux)	Windows/Ubuntu	ASOP/Linux
Power Requirements	5V @2.5A	19V @3.5A	12V @2A
Functionality Rating (1- 5)	**4**	**5**	**4**
Accessibility Rating (1 - 5)	**5**	**4**	**4**

BUTLER: Hardware Selection

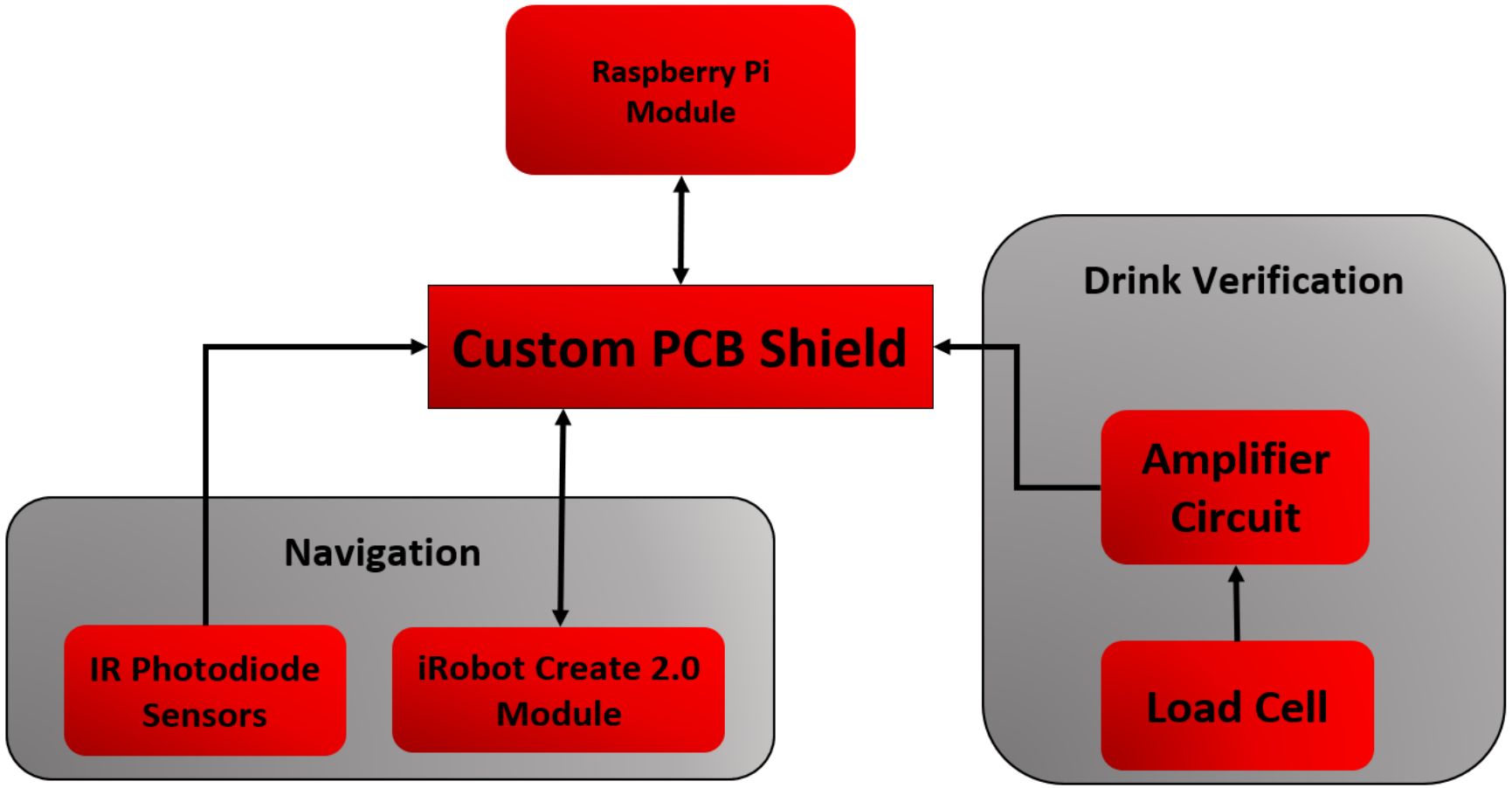
Motorized Platform
- Create 2.0 (iRobot)

Navigation “Firmware”
- Raspberry Pi 4.0 + Raspbian Lite
- IR photodiode (Generic)
- TAL221 Load Cell (HTC-Sensors)
- HX711 24-Bit ADC (Avia)

Structure
- ¼” MDF (HomeDepot)
- Aluminum T-slot (8020 Inc.)



BUTLER: Design

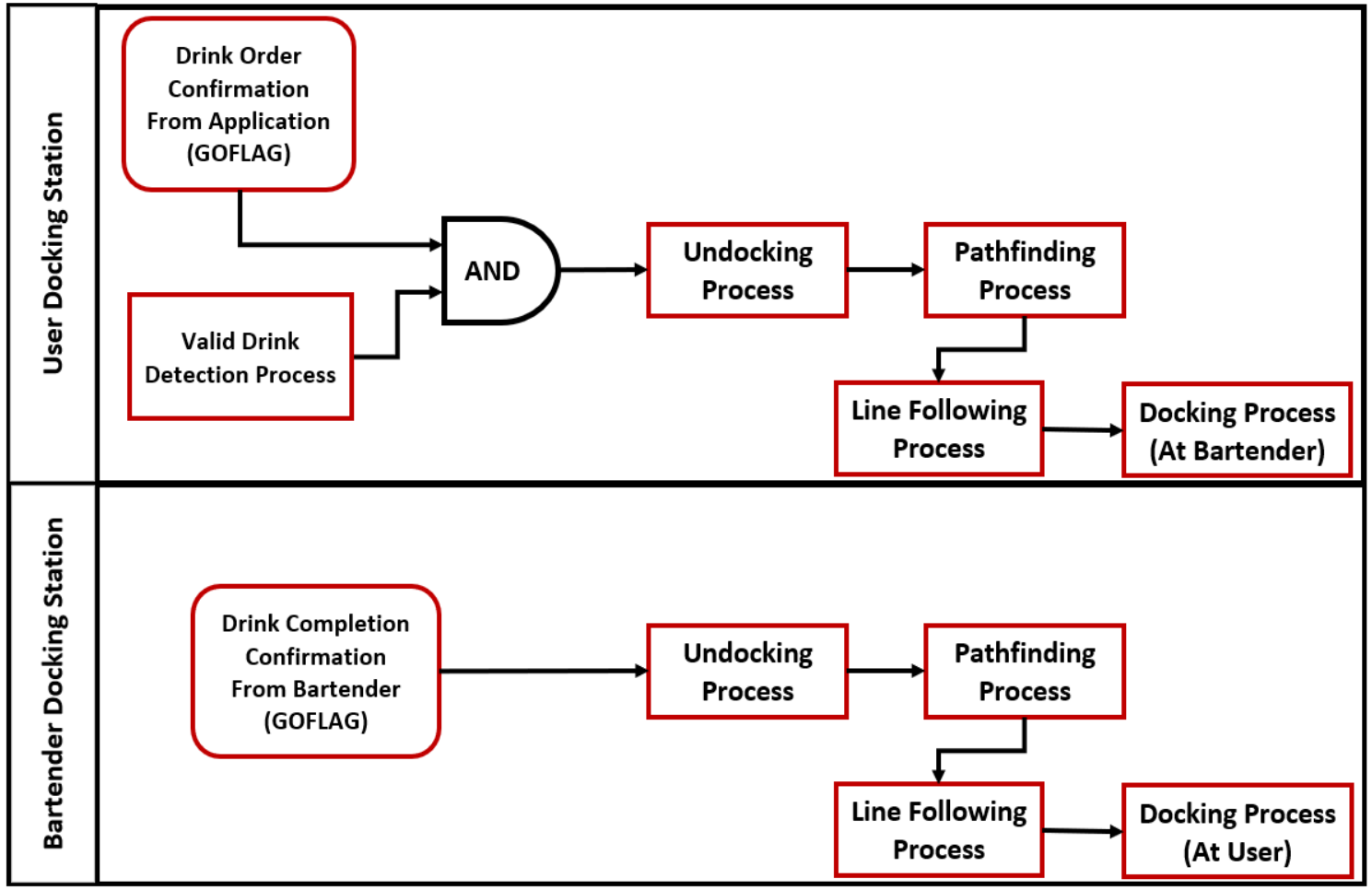


BUTLER: Firmware

Navigation
- Line following with PID controller
  - IR Sensors
  - Raspberry Pi + Python
- Built-in docking feature

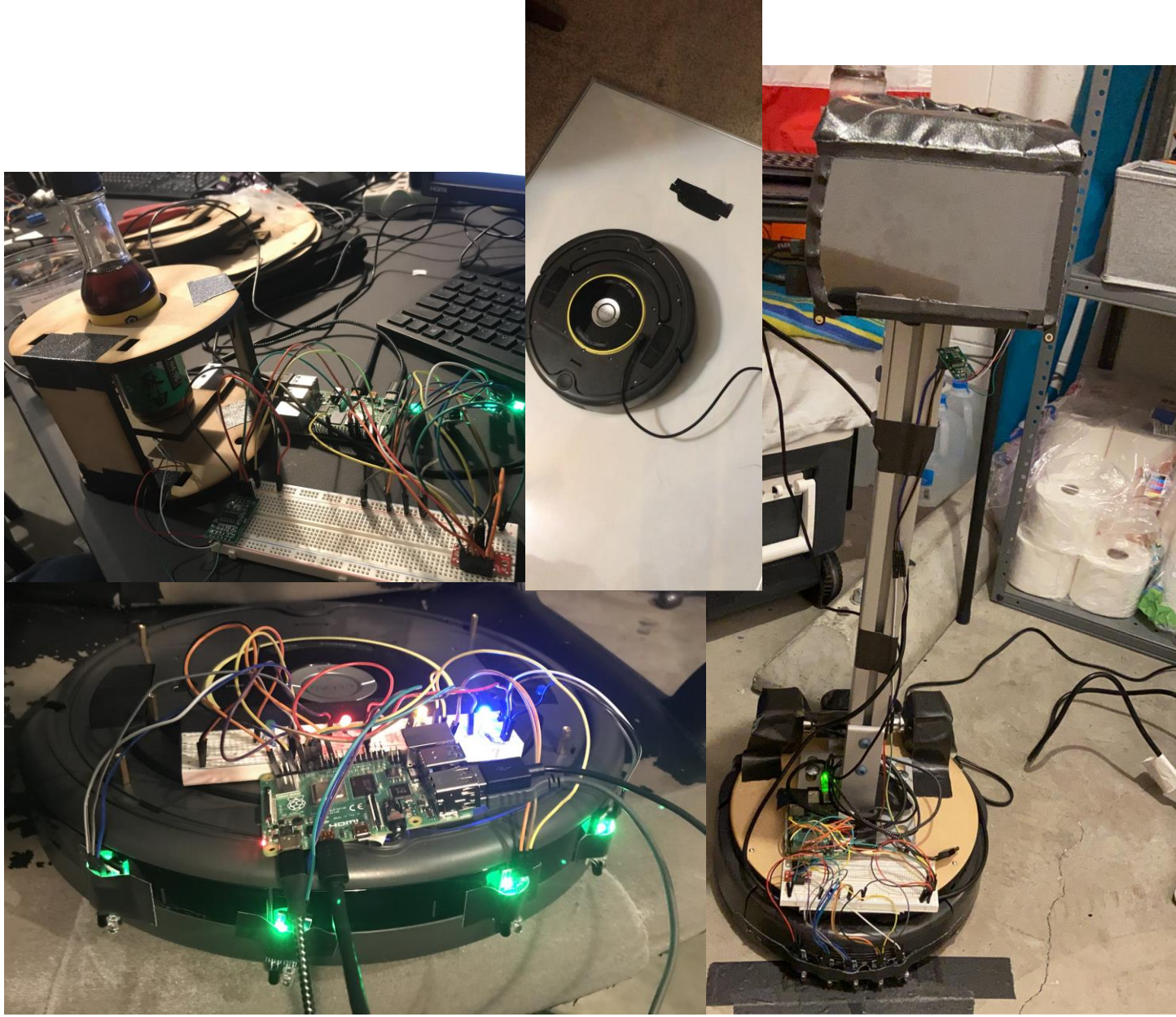
Cup Detection (Override Interrupt)
- Load cell monitoring

Communication with MQTT
- Start/End Navigation
  - Docking at endpoints
- Cup Presence + Empty/Full

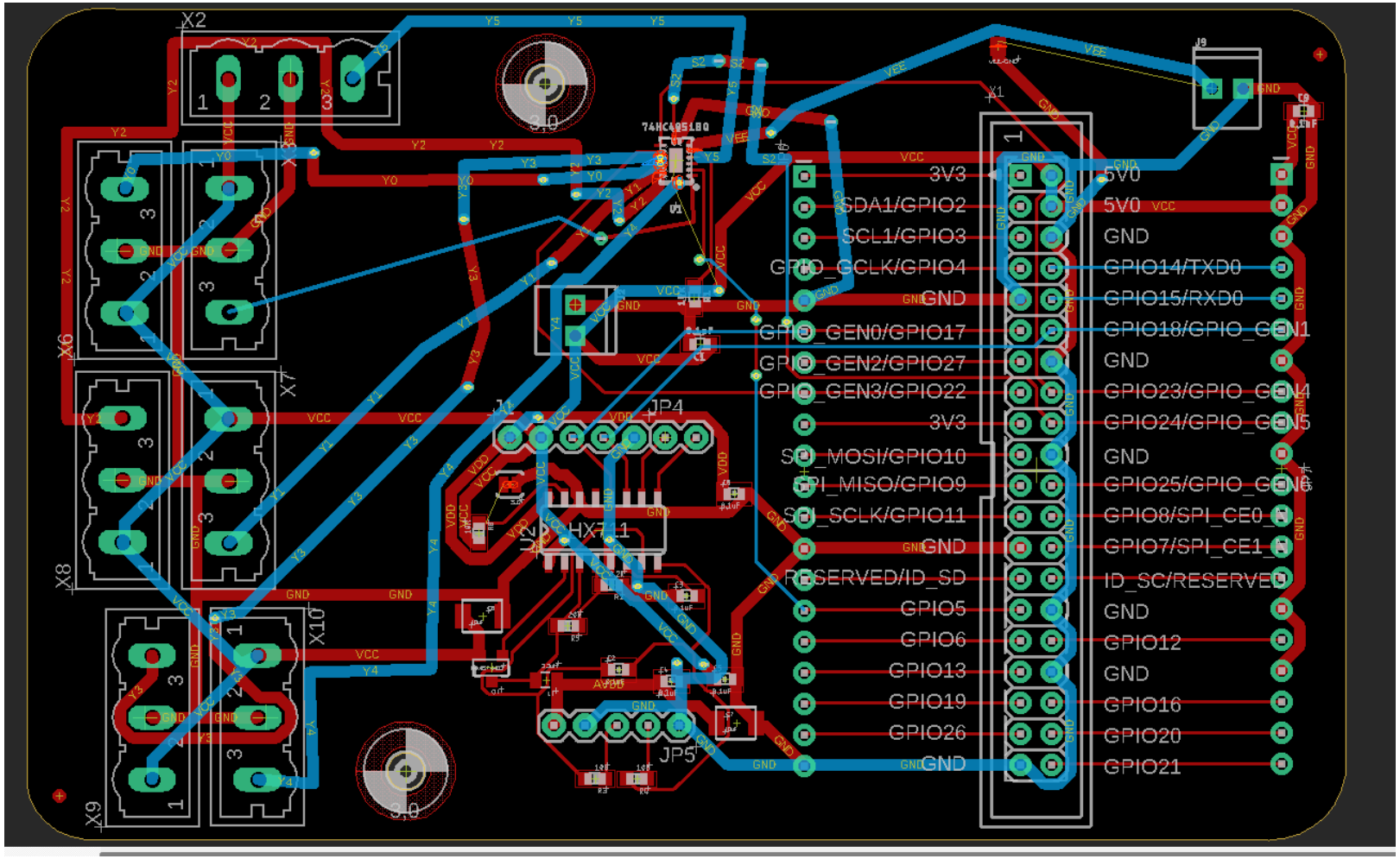


Butler- Testing

- **iRobot Serial Commands**
  - Wheel Control
  - Docking
- **IR Sensors**
  - Sensor input
  - Multiplexer
  - Mounting Locations
- **Load Sensor**
  - Sensor input
  - Fine Tuning/Tare
- **MQTT**
  - Receive from Application (Loop)
  - Publish to Application
- **Integration**



PCB Design for the Butler completed in Eagle



Butler PCB Design

- 2-Layers of 1-oz Cu on FR-04
- 10 mil STD / 18 mil MAX

- Manufacturing and Assembly: PCBWAY in Hangzhou, China

ETA: ~~May 3rd, 2020~~

April 29th, 2020

BUTLER: Issues

PCB Design
- Completed PCB Design and found vendor
- Extended lead times prevent integration

Battery Life
- Successfully tapped into Motor Driver
- Motor Driver overwritten during docking
- External Battery Purchased

IR Sensor Mounting
- Precise spacing was found
- Docking Procedure inhibited
- IR sensor damage
- Back-mounting with 180-degree turn

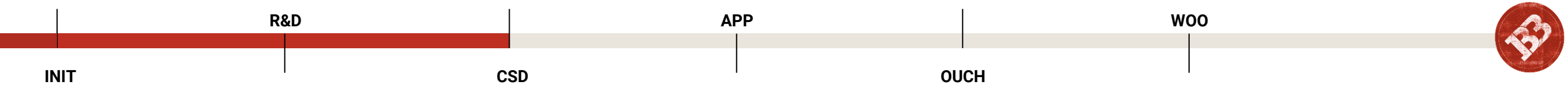
BUTLER: Status

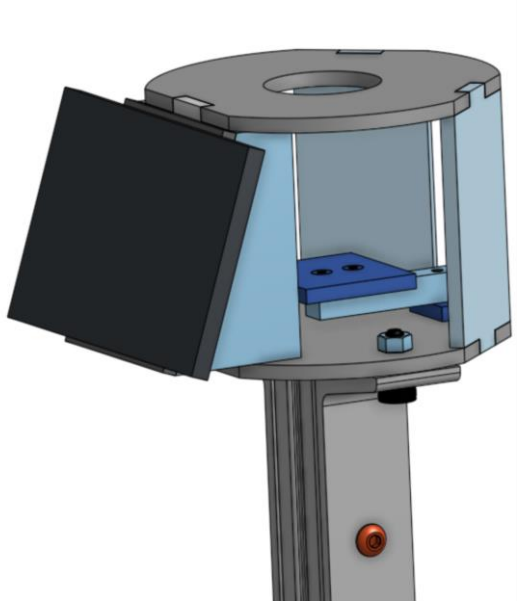
Prototype/Firmware
- Load cell / IR sensor validation **OK**
- IR sensor array integration via Python **OK**
- iRobot PID arbitrary line-following **OK**
- iRobot line-end docking **OK**
- Continuous uninterrupted travelling **OK**

PCB Design
- Module schematics & footprints **OK**
- SMD component vendor sourced **OK**
- Schematic **OK**
- Layout **OK**
- Fabrication / Assembly **PEND**
- Received Shipment **NO**

Structural
- CAD for base components **OK**
- MVP CAD prototype **OK**
- Components sourced/ordered **OK**
- IR Sensor mounting plate **NO**
- Overall Aesthetics **NO**

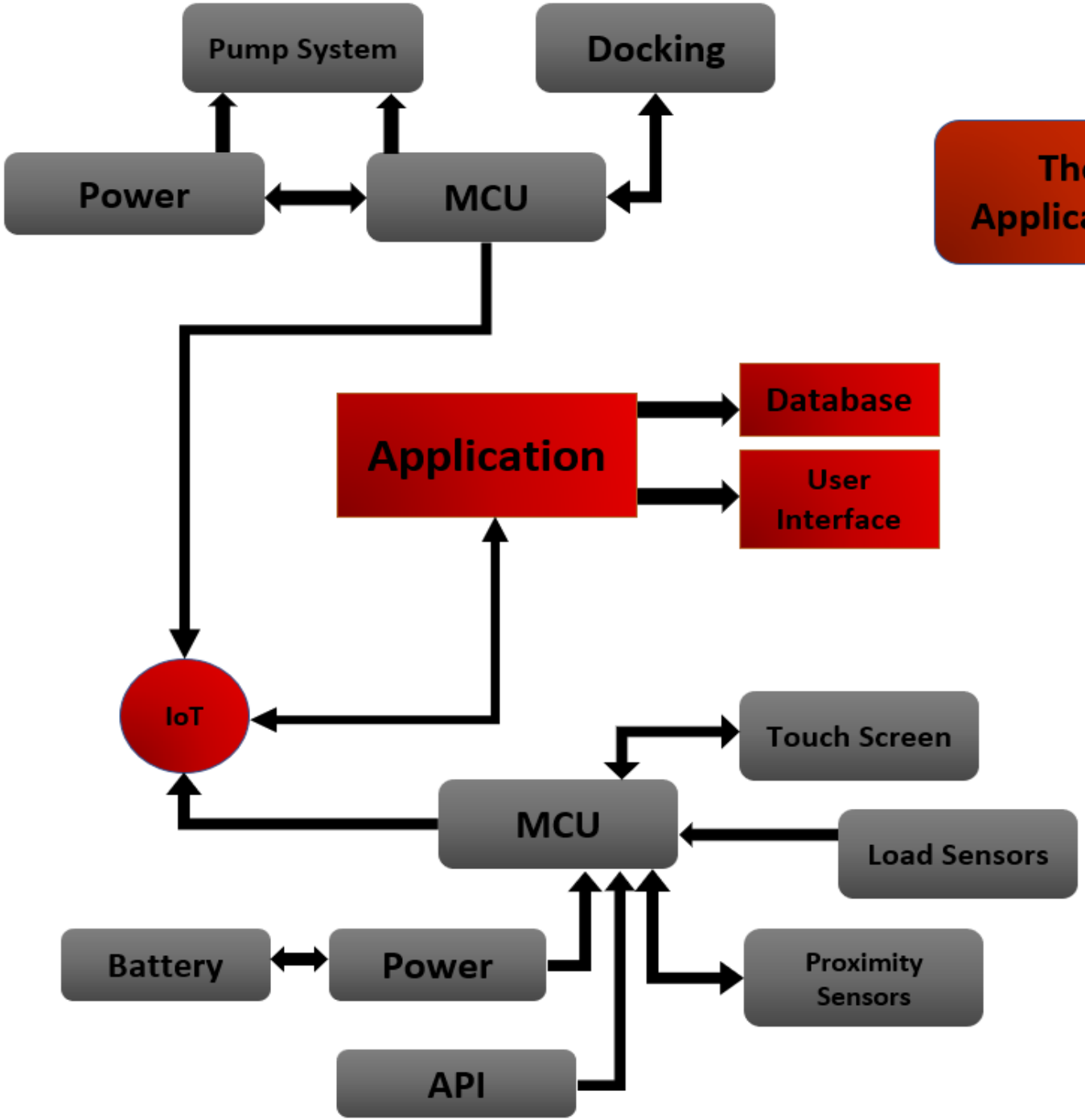
Overall Integration
- MQTT tags pre-defined **OK**
- Transition development to Pi **OK**
- MQTT tags validated, live demo **OK**
- Subscription and Publishing **OK**





SUB SYSTEM 2 THE BUTLER

3D Model vs Actual



THE APPLICATION

Block Diagram Highlight

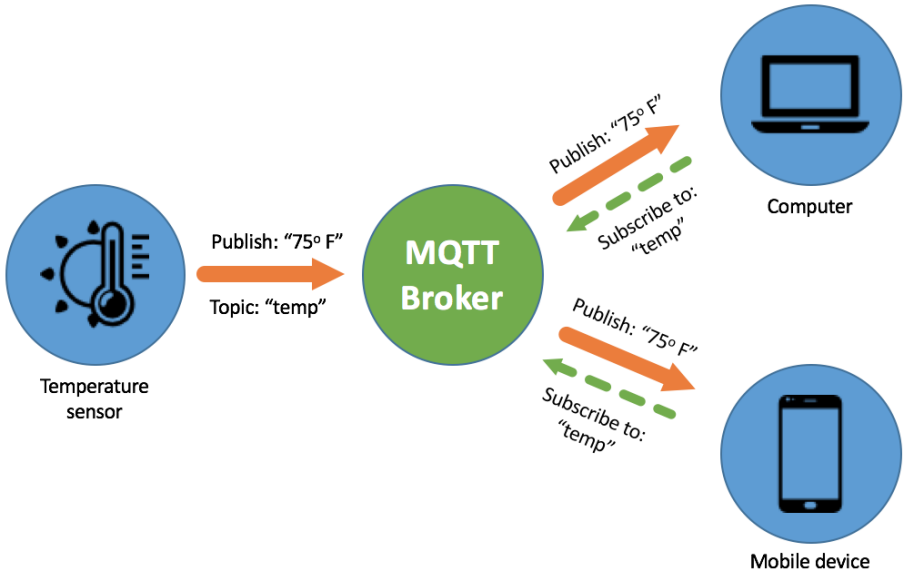
APPLICATION: Building Blocks

Operating System / Environment
- Linux Raspbian Lite (bare bones)

“Internet of Things” Platform
- Eclipse Mosquitto (MQTT)

Persistent Database
- SQLite3

Programming Language
- Python 3.7.4
  - PyQT5 (GUI) lib
  - Paho (MQTT) lib
  - Sqlite3 lib

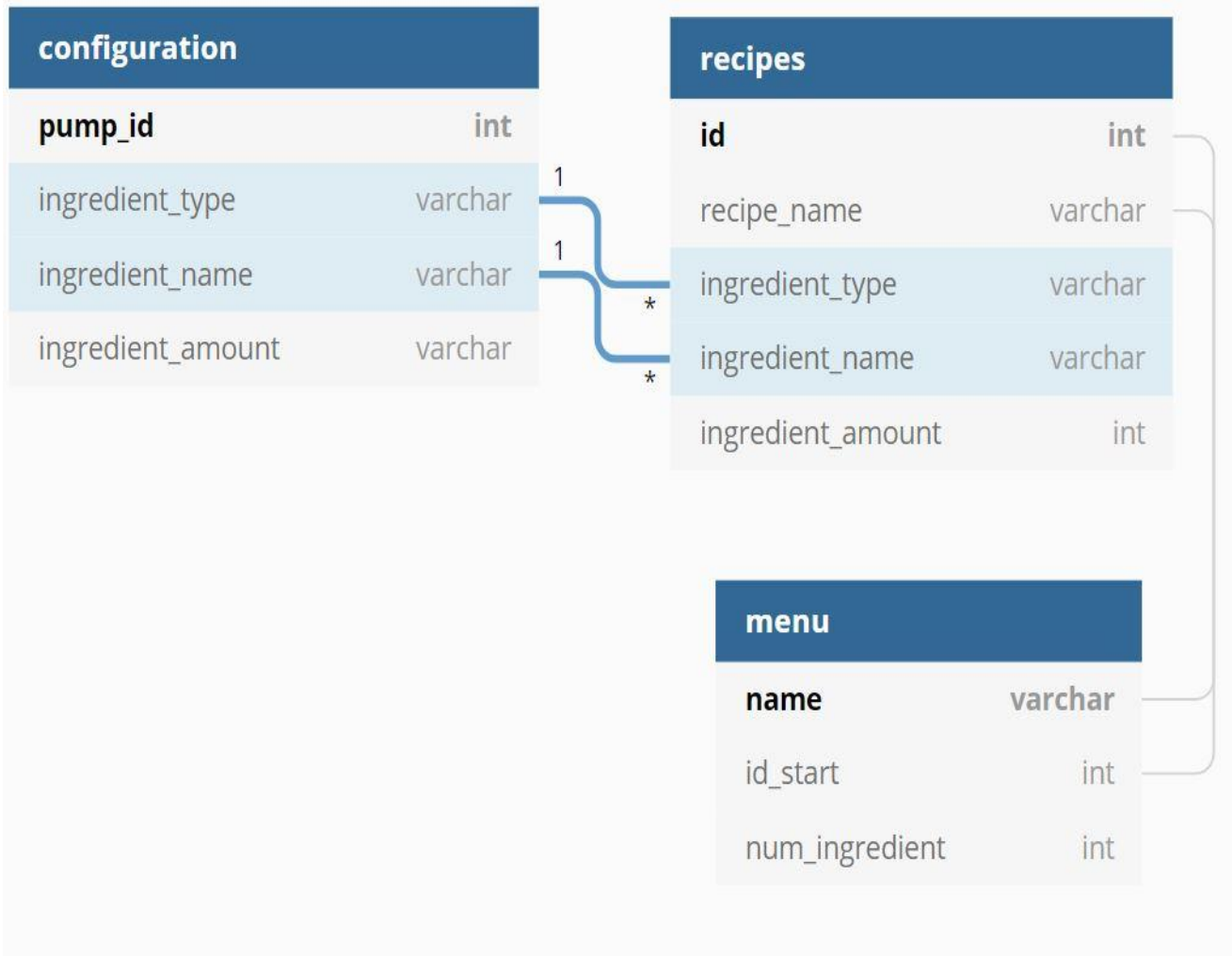


APP: Status of Backend Env.

Operating System & Runtime Environment
- Flash OS, strip to OS bare essentials, enable GPIO functions **OK**
- Install Python, libraries, configure Git repos **OK**

MQTT Broker Configuration
- Install, enable service, configure for maximum QoS **OK**
- Validate errorless machine-to-machine data transfer between subsystems **OK**

SQL Database Configuration
- Install, enable service **OK**
- Format tables, populate tables with basic entries, validate core query structures **OK**



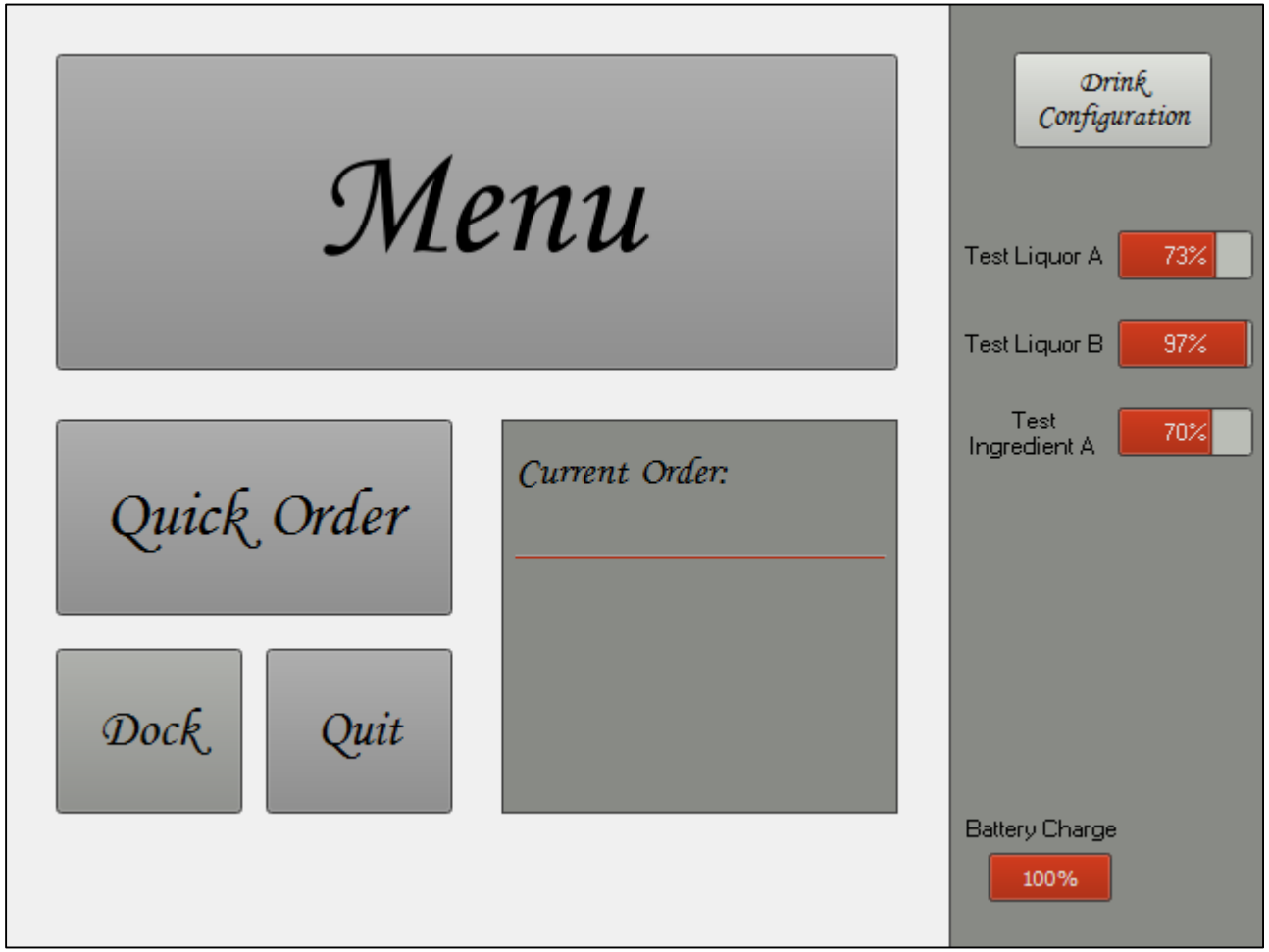
APP: Status of GUI

Graphical User Interface

- Formalize visual design & generate UI basis files from development toolkit **OK**
- Translate/Port to Python & PyQT5 **OK**
- Link basic button functionalities
  - i.e. “exit all processes” **OK**
  - i.e. confirmation dialogues **OK**
- Instantiate primary Window & actions **OK**
- Instantiate menu shell & actions **OK**
- ...

Core Logic

- Send/receive data from MQTT **OK**
- Send/receive data from SQLite **OK**
- Live GUI refresh on data updates **OK**
- Subsystem management logic **OK**
- ...

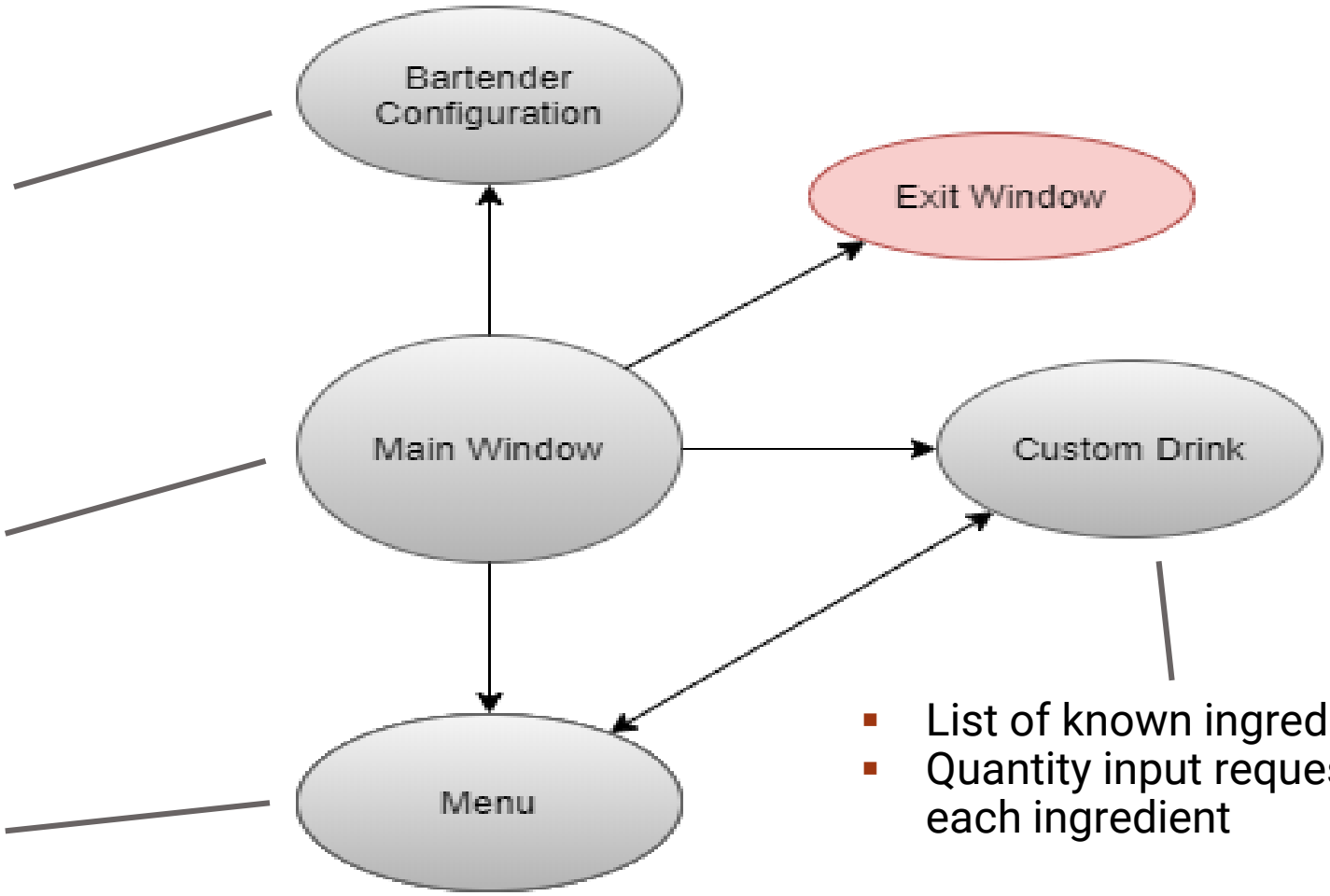


APPLICATION: GUI Breakdown

- Link to menu window
- Quick order last order
- Basic system information and options
- Ingredient inventory

- Small display of current ingredient inventory
- Core Change button
- Return link to main window

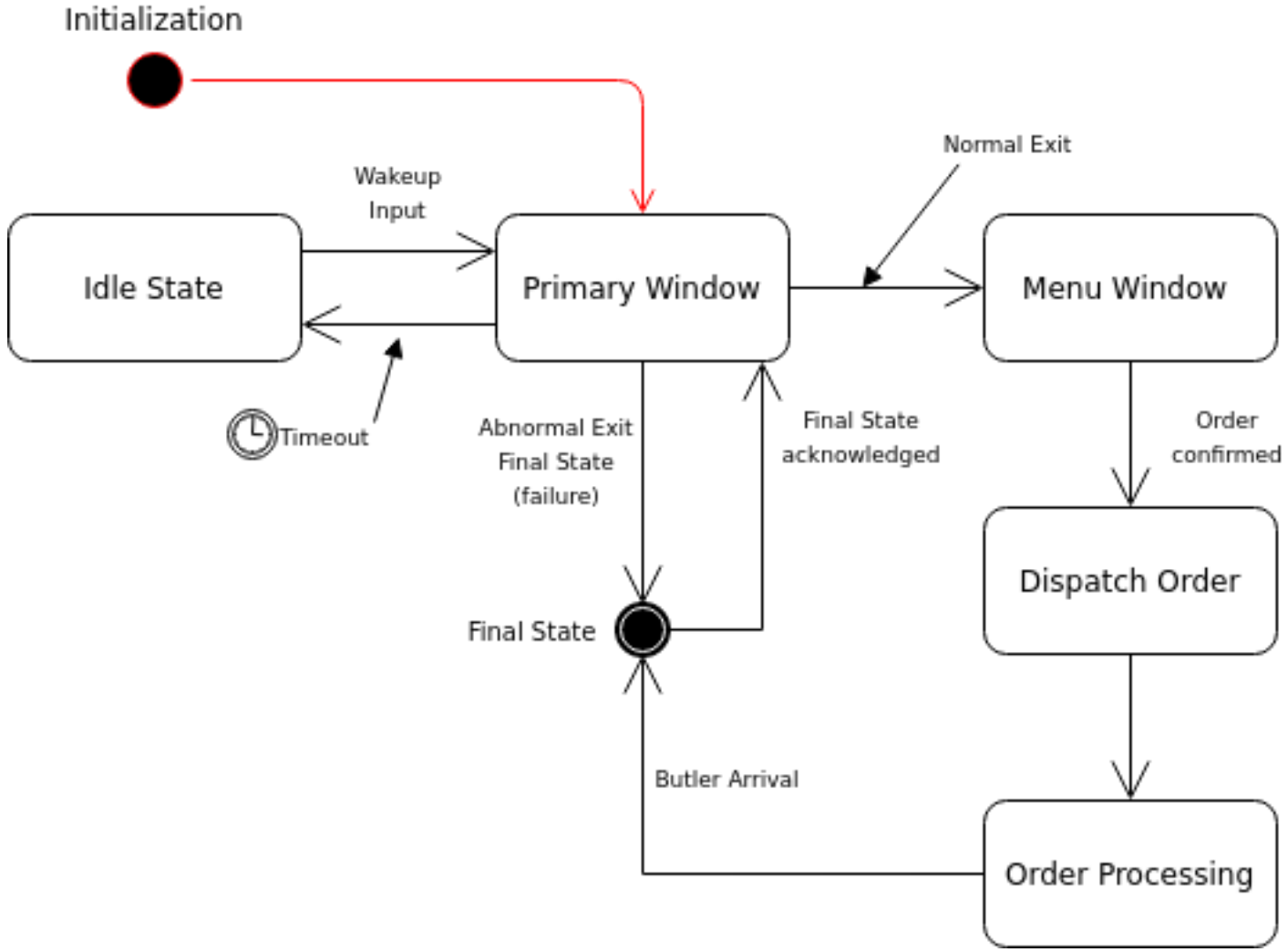
- List of available drinks and their components



- List of known ingredients
- Quantity input requests for each ingredient

APPLICATION: State Diagram

Idle State: Lack of user input or relevant system processes
Primary Window: Constantly checks various system states
Menu Window: Queries SQL database to offer relevant information to the user
Dispatch Order: Received confirmation from user. Initiates Butler movement.
Order Processing: Constantly monitors Butler and Bartender processes until end of delivery.



Application - Testing

- **Test:** MQTT client able to publish and subscribe
  - Set up command windows on other computers to publish and subscribe to test topics

- **Test:** SQL interactions linked to User input
  - Linked “submit” button press to lineEdit fields allowing that information to be used in any SQL function

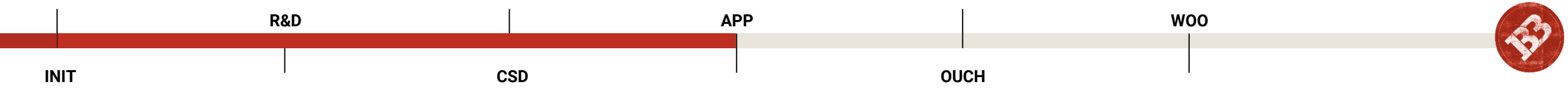
- **Test:** Able reset/refresh multiple parts of the GUI upon any information update, either from user input or MQTT
  - Tested various inputs and edge cases from Bartender, Butler, user input, and command line inputs

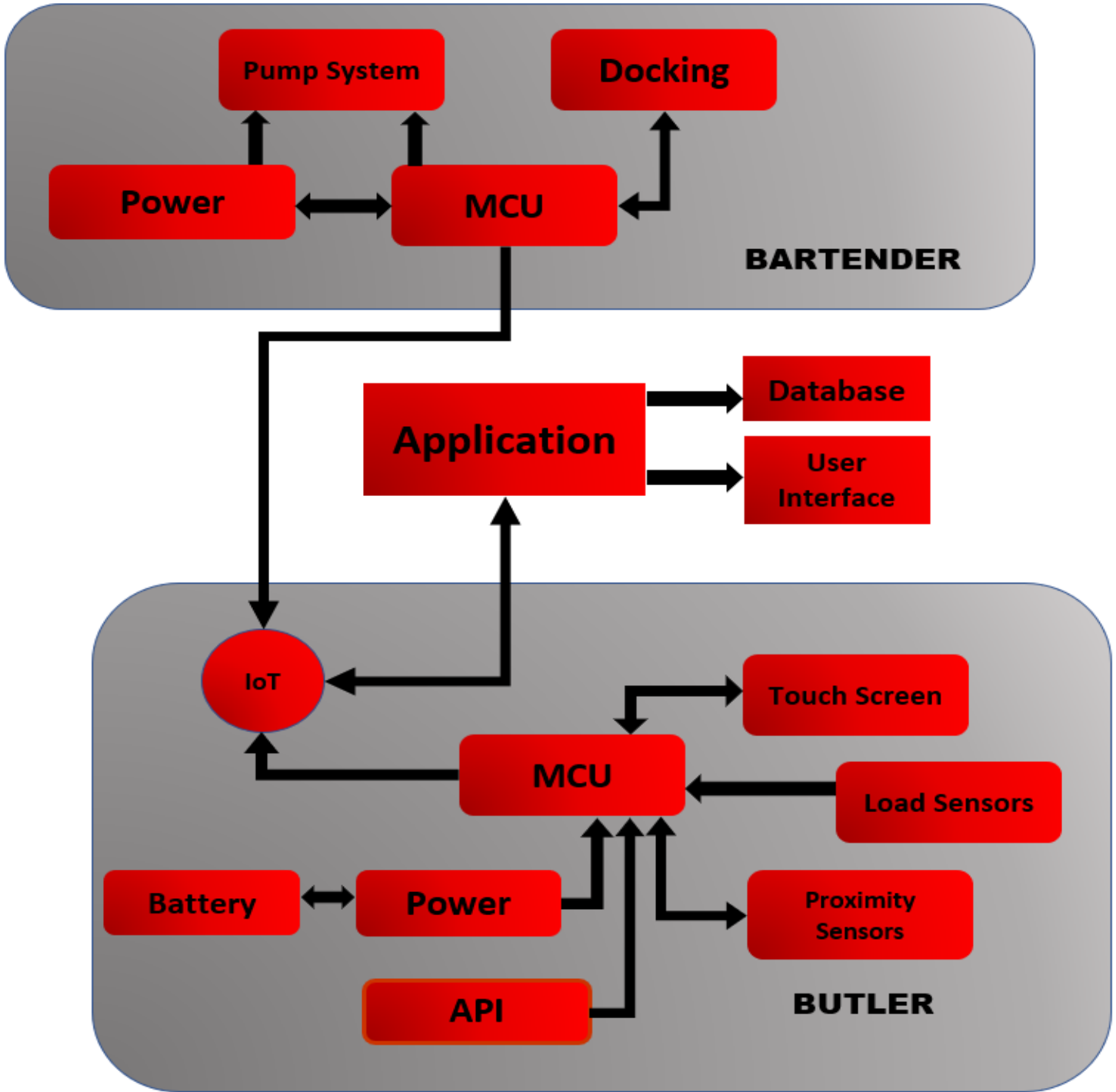
Application: Issues

- **Integration:** With all subsystems being developed simultaneously, there were slight **discrepancies in the exact outputs meant to be sent via MQTT**. This led to creation of several useless functions and improper logic based off of misinterpretation of flags

- **Refreshing GUI:** Despite sounding like a simple task, various **edge cases** and oversights led to confusion on where the problem lies when **updating various widgets** within the GUI

- **Adding Functions:** As the application grew more and more complex, functions that interacted between multiple pages, MQTT signals, and SQL executions became increasingly complex as well. Estimated time to complete an objective increased as the project neared completion





ADMINISTRATIVE SUMMARY

& Notes on Integration

RELEVANT STANDARDS

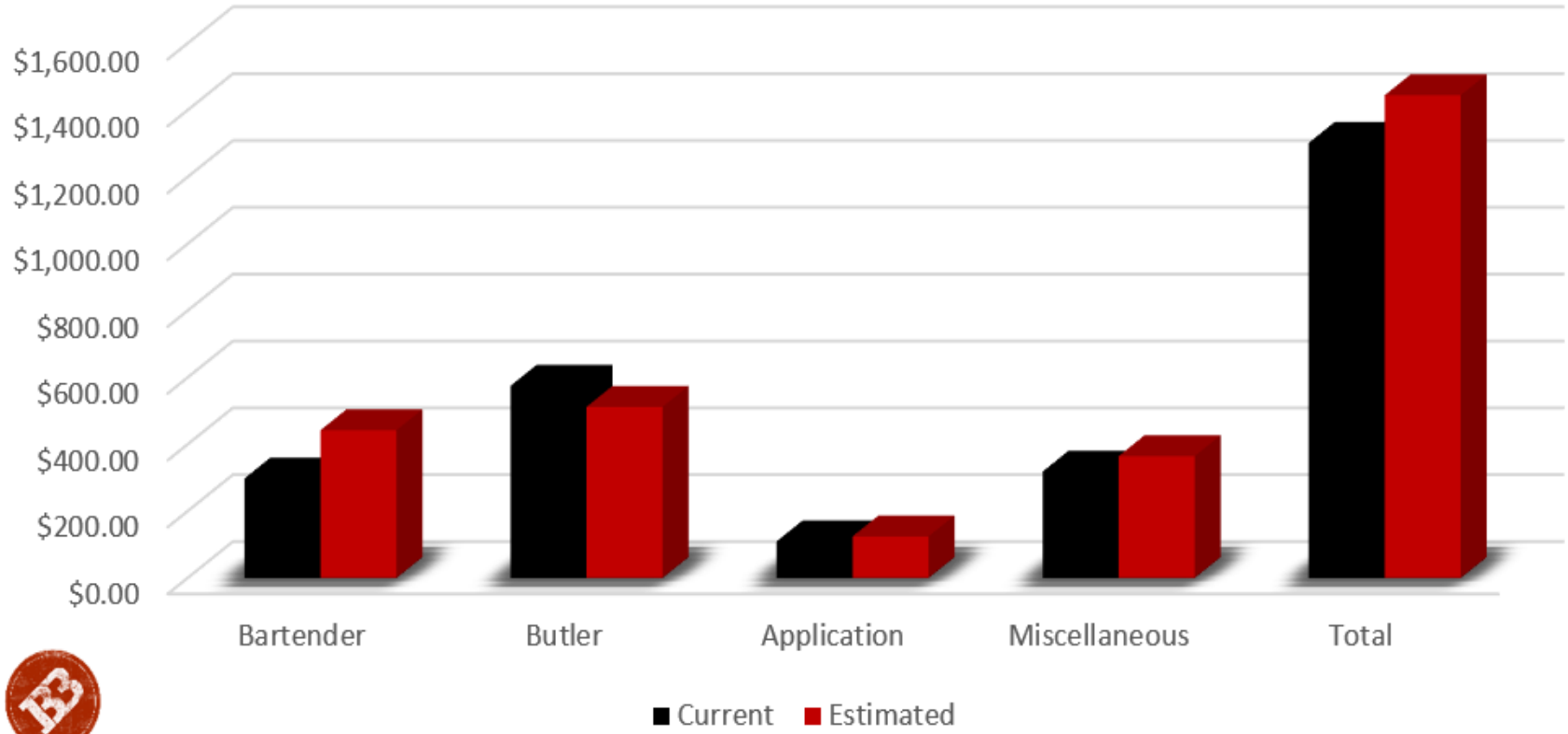
Digital Standards

- Wireless Communication - IEEE 802.11
- MQTT v5.0 standards - ISO/IEC PRF 20922

Health and Safety Standards

- **Food Safety - NSF/ANSI 25**
  - **The purpose of this standard is to establish the minimum food protection and sanitation guidelines**
- Drinking Water - NSF/ANSI 61
  - the standard that covers drinking water system components
- Fire Safety and Emergency - NFPA
  - the fire safety standard that provides symbols used to effectively communicate fire safety, emergency, and associated hazards information.

Budget



Estimated Budget

- **$1,445.00**

Total Cost

- **$1,302.36**

INIT R&D CSD APP OUCH WOO

Thank You