H.A.L.O - Hall-effect Array for Live Observation

Final Review Presentation - Group 18

Brogan Dougherty
Electrical Engineering

Aidan Kennedy
Electrical Engineering

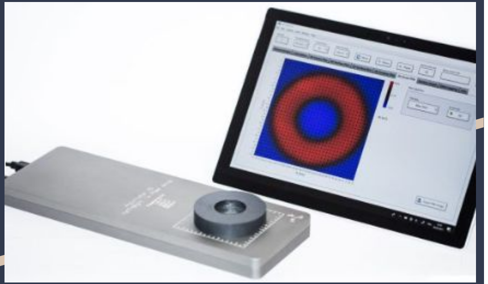
Gabriel Martin
Computer Engineering

Ryan McCord
Electrical Engineering

Motivation and Background

- H.A.L.O. utilizes hall-effect sensors to image magnetic fields.
- Magnetic field measurement technology was developed in the early 1800’s.
- Modern usages include power system monitoring, object detection, and medical applications such as MRI.
- Most of these systems are large, heavy, and unwieldy.
- Our motivation is to build a system that can image magnetic fields effectively while maintaining a size and weight that allows for better maneuverability and mobility.

MRI machine



Metrolab HallinSight

Goals and Objectives

Goal

- Create a system capable of measuring and visualizing magnetic fields

Objectives

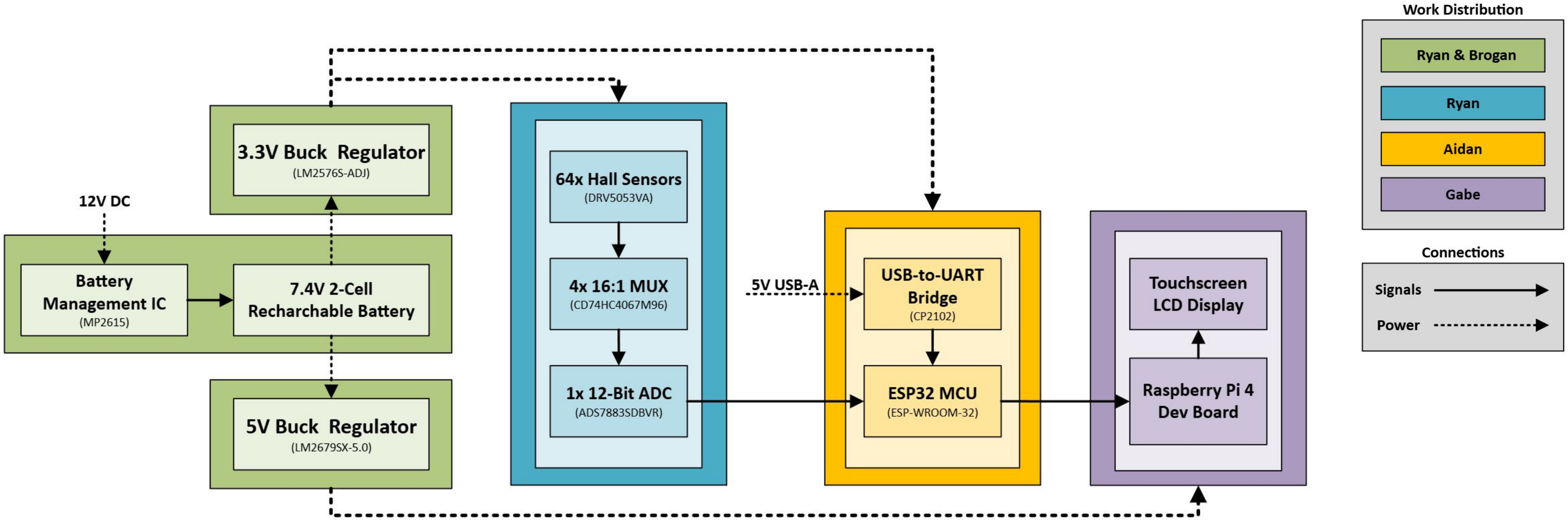
- Use an 8x8 grid of sensors to capture magnetic field measurements across a region of space
- Use multiplexers and an ADC to convert the sensor data into manipulable output
- Process the data with an ESP 32
- Transmit processed data to Raspberry Pi to display
- Visualize the system in an intuitive manner.
  - Heat Map mode where lighter colors corresponds to larger magnitudes, darker colors correspond to lower magnitudes, red corresponds to positive, and blue corresponds to negative
  - Raw Data mode where the user can select between mT or G

Engineering Specifications

Parameter	Purpose	Value
Number of Tiles in Display	Magnetic field visualization	≥ 64 tiles
Field Size Detected	Magnetic field visualization	≥ 64 cm^2
Field Intensity Range Detected	Magnetic field visualization	[-8, 8] mT
Detection Distance	Accurate measurements	[1 cm, 2 cm]
Screen Refresh Rate	Expedient display of data	≥ 15 Hz
Length of Recording	Value storage for analysis	≥ 30 s
Power Consumption	Small energy footprint	< 5 W

Parameter	Purpose	Value
Accuracy of Detection	Accurate measurements	± 1 mT
Power Efficiency	Small energy footprint	> 75% efficiency
Operating Temperature	Convenience	< 140° F
Latency	Expedient data transmission	< 200 ms
Weight	Convenience	< 5 lbs
Size	Convenience	< 1 square foot
Cost	Affordability	< $250

Hardware Block Diagram



Power Requirements

- Important to consider current draw across system components
    - System components operate on either 3.3V or 5V
    - CP2102 requires 5V supplied via USB-A
    - A separate 12V input required for battery charging (MP2615)
- Selected a two-cell (7.4V) Li-io battery pack with 5000mAh
- Buck regulators step down voltage to 3.3V and 5V

	Voltage Requirement	**Current Requirement**
Hall Sensor (64x)	3.3 V	230.4 mA
Multiplexer (4x)	3.3 V	32 uA
ADC	3.3 V	3 mA
ESP32	3.3 V	240 mA
Raspberry Pi w/ LCD	5 V	3 A
Totals	3.3 - 5 V	~ 3.5 A

	Voltage Requirement	**Current Requirement**
CP2102 (USB-A)	5V	20mA
MP2615 (12V Jack)	12V	2A

Voltage Regulators

3.3V Regulator - LM2576

- Handles up to 3A Load
- Simple packaging
- Good efficiency at low frequency

5V Regulator - LM2679

- Handles up to 5A
- Higher efficiency

Regulator	Output Current	Efficiency	Input Voltage Range	Footprint Size	Switching Frequency
LM2576S-ADJ	3A	~75%	4 - 40V	Large	~52 kHz
LM1117-3.3	0.8A	~50%	< 20V	Small	N/A (Linear)
LM2679-ADJ	5A	~85%	8 - 40V	Large	~260 kHz
TPS5430	3A	~85%	5.5 - 36V	Small	500 kHz

Hall-effect Sensor

Technology	Resolution	Sensitivity	Size	Cost
Hall-effect Sensor	poor	poor	miniscule	cheap
Magneto-Resistive Sensor	fair	fair	fair	fair
Fluxgate Sensor	great	great	massive	expensive

Component	Sensitivity	Temperature Range	Power Consumption	Cost	Risk of EMI
DRV5053VA	± 9 mV/mT	-40°C to +125°C	Low	Moderate	Medium
DRV425RTJT	± 2 mT	-40°C to +125°C	Moderate	High	High
A1324LLHLX-T	2.5 mV/G	-40°C to +125°C	Low	Low	Low
SS495A1	3.125 mV/G	-40°C to +150°C	Low	Low	Medium

Hall-effect sensor was selected because are resolution and sensitivity needs are very minor, so better options would be unnecessary.

DRV5053VA was chosen because it has a high sensitivity while maintaining a moderate cost and consuming low power.

Analogue to Digital Converter

SAR

- Balance between speed and resolution
- Most common in consumer electronics making them accessible

ADS7883SDVR

- High Speed
- Low Cost/Availability
- Sufficient Resolution

ADC Type	Max Resolution	Max Sample Rate
SAR	18 bits	10 MHz
Delta-sigma	32 bits	1 MHz
Flash	12 bits	10 GHz

ADC	Resolution	Sample Rate	Communication Protocol	Cost
ADS7883SDBVR	12 bit	3 MSPS	SPI	$4.73
AD7940BRMZ	14 bit	100 kSPS	SPI, DSP	$16.90
ADC081C021CIMKX/NOPB	8 bit	188.9 kSPS	I2C	$1.13

Comparison and Selection of Hardware

MCU Requires

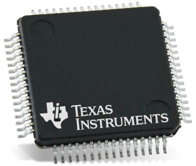
9 GPIO: S0-S3, EN0-3, CS

UART: RX, TX

SPI: MISO, SCKL

MCU

Feature	MSP430FR6989	ESP32	STM32
Architecture	16-bit MSP430	32-bit Xtensa LX6	32-bit ARM Cortex-M7
Clock Speed	16 MHz	< 240 MHz	< 550 MHz
Flash Memory	128 KB FRAM	4 MB (external)	64 KB - 2 MB
SRAM	2 KB	520 KB	16 KB - 1024 KB
GPIOs	40	34	100+
ADC Resolution	12-bit (16 channels)	12-bit (18 channels)	12 to 16-bit (24 channels)
Operating Voltage	1.8V - 3.6V	2.2V - 3.6V	1.8V - 3.6V
Wireless Capability	None	Wi-Fi, Bluetooth	None
Interfaces	I2C, UART, SPI, ADC, Timer, PWM	I2C, UART, SPI, ADC, DAC, PWM, CAN	I2C, UART, SPI, USB, CAN, Ethernet, LTDC



Development Board

Feature	STM32H735G-DK	STM32H747I-DISCO	Raspberry Pi 4B
MCU	STM32H735IGK6	STM32H747XIH6 (Dual-core: Cortex-M7 + M4)	Broadcom BCM2711
Core(s)	Cortex-M7 @ 550 MHz	Cortex-M7 @ 480 MHz + Cortex-M4 @ 240 MHz	Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
Flash/RAM	1 MB Flash / 564 KB RAM	2 MB Flash / 1 MB RAM	Micro SD card slot for loading operating system and data storage / 8GB LPDDR4-3200 SDRAM
Display	4.3" 480x272 TFT LCD (with capacitive touch)	4.3" 480x272 TFT LCD (with capacitive touch)	3.5” 480x320 TFT LCD (with resistive touch)
Connectivity	USB OTG HS (micro-B), Ethernet, microSD	USB OTG HS (micro-B), Ethernet, microSD, Wi-Fi expansion	2 USB 3.0 ports; 2 USB 2.0 ports. Raspberry Pi standard 40 pin GPIO header, 2 × micro-HDMI®, 1 MIPI DSI

Display

Display Type	RGB LED	LCD - IPS	LCD - TFT
Resolution	Low	High	Moderate
Refresh Rate	High (120 Hz)	Moderate-High (60-120 Hz)	Low - Moderate (30-60 Hz)
Power Consumption	High	Moderate	Low

Display	Hosyond 5” IPS LCD	Kuman 3.5" TFT LCD	Elegoo 2.8" TFT LCD
Resolution	800x480	480x320	320x240
Refresh Rate	~60 Hz	~5 Hz	~45 Hz
Interface	MIPI DSI	SPI	RGB888
Cost	$35	$20	$15

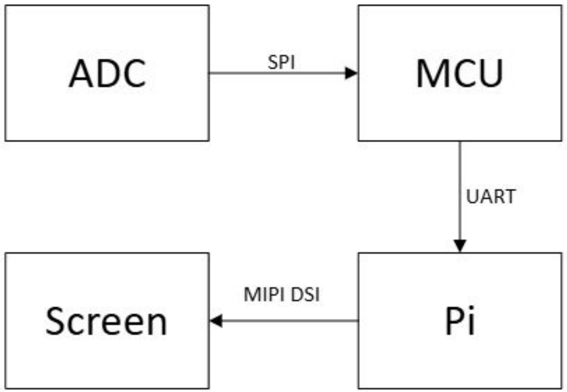
Comparison and Selection of Communication Protocol

Communication protocols required

SPI: For ADC to ESP

UART: For ESP to Pi

MIPI DSI: For Pi to Screen



Spec	I2C	SPI	UART	MIPI DSI
Speed	Slow	Fast	Slow	Fast
Distance	Long	Short	Short	Short
Clock	Synchronous	Synchronous	Asynchronous	Synchronous
Configuration	Half duplex	Full duplex	Simplex, Half Duplex, or Full Duplex	High-Speed Serial
Connectable devices	Multi	Multi	Single	Single
Lines	2	4	2	4-6

Comparison and Selection of Software

Bare-Metal vs. Operating System

Platform	**Approach**	**Pros**	**Cons**
ESP32	Bare-Metal	- Fast startup - Low memory usage - Tight timing	- Manual scheduling - Slower development
ESP32	FreeRTOS	- Task prioritization - Abstraction drivers	- More complexity - Steeper learning curve
Raspberry Pi	Bare-Metal	- Fast Startup - Full control	- Extreme development difficulty - No display / touch drivers
Raspberry Pi	Raspbian, FreeRTOS	- Easy driver support - High level programming	- Slower boot - Higher latency

Comparison and Selection of Software

Operating System Options for Raspberry Pi

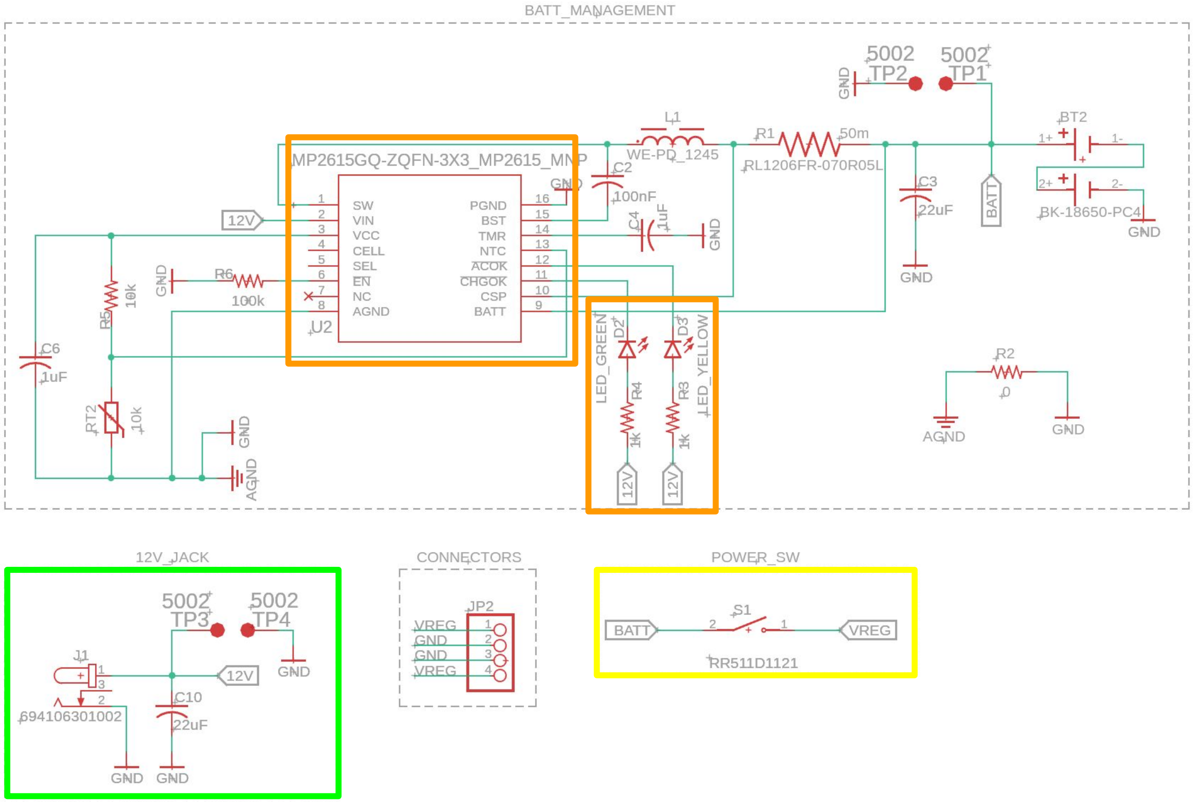
Feature	Raspberry Pi OS (Linux)	FreeRTOS
Display Driver Support	Drivers for framebuffers and SPI LCD interface	Manual configuration
Touchscreen Support	Drivers and calibration tools	Manual configuration
Package & Library Support	Linux pip (easy installation)	Less commonly used, lacking support
Development Tools	IDEs, ssh, file systems, near-complete stacks	Lower-level programming, limited abstraction

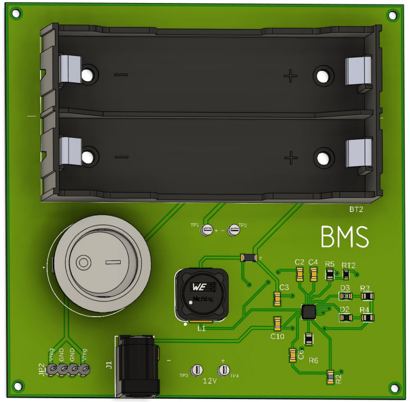
Comparison and Selection of Software

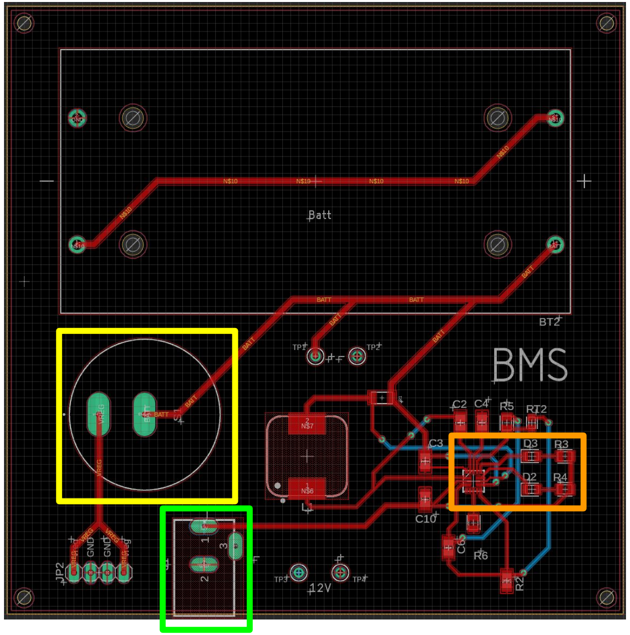
GUI Development Frameworks

Framework	**Language**	**Capability**	**Ease of Use**
Tkinter	Python	Basic widgets and raster graphics	Easy
PyQt	Python	Full Qt GUI Toolkit	Moderate
GTK	C / Python	High-quality Linux-native UI	Steep learning curve

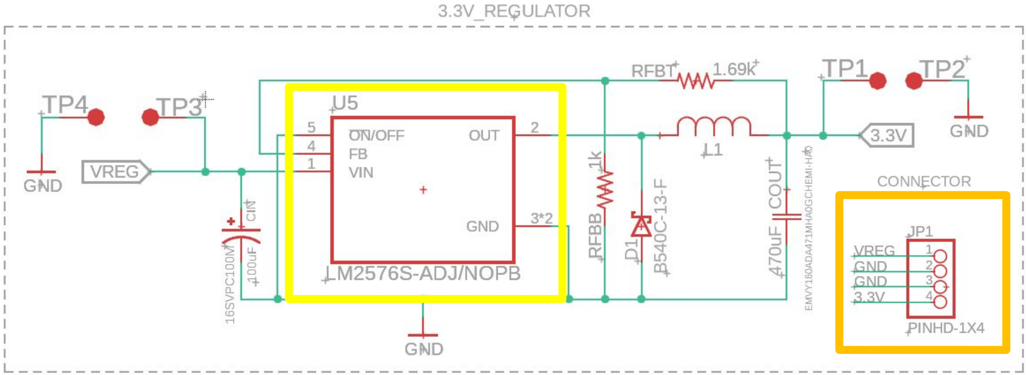
BMS Schematic/PCB

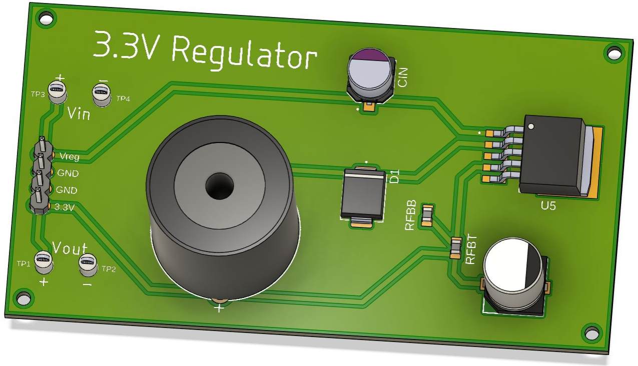


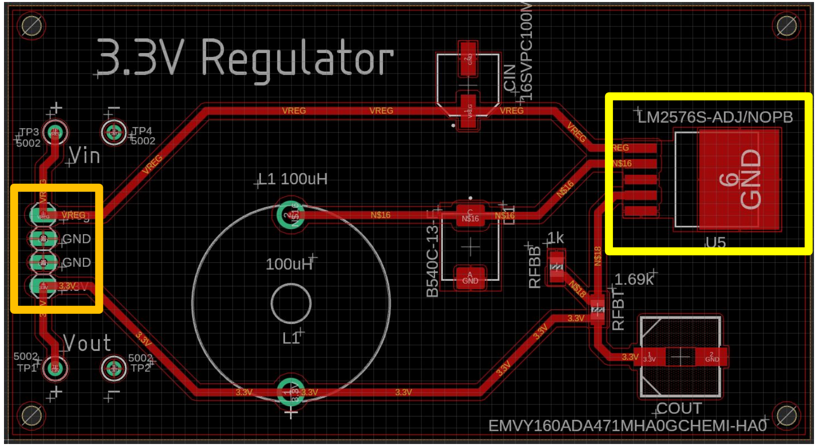




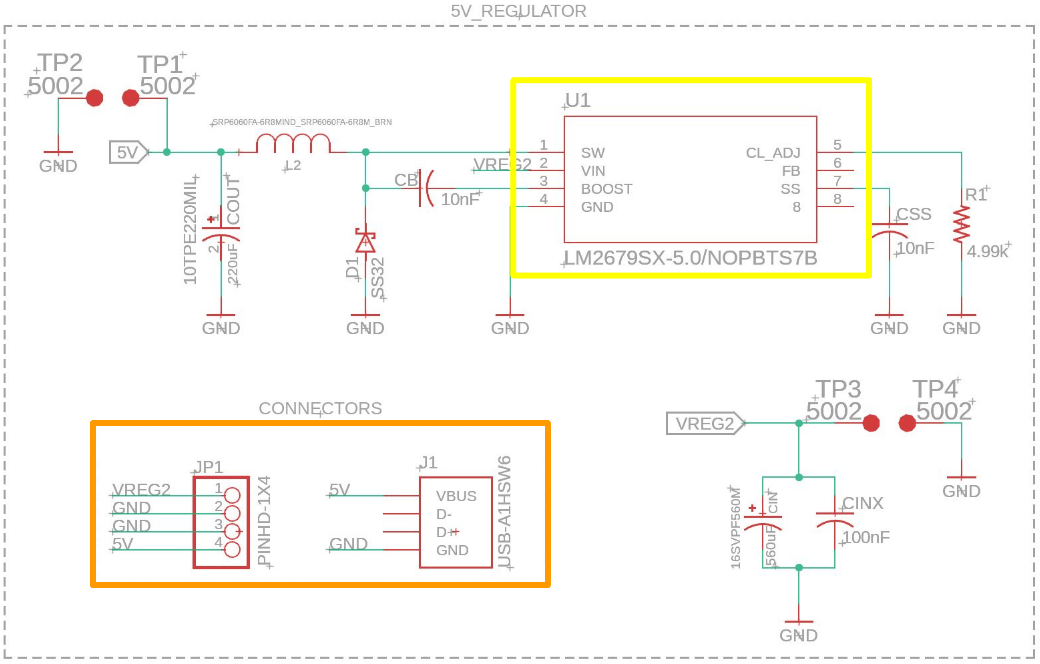
3.3V Regulator Schematic/PCB

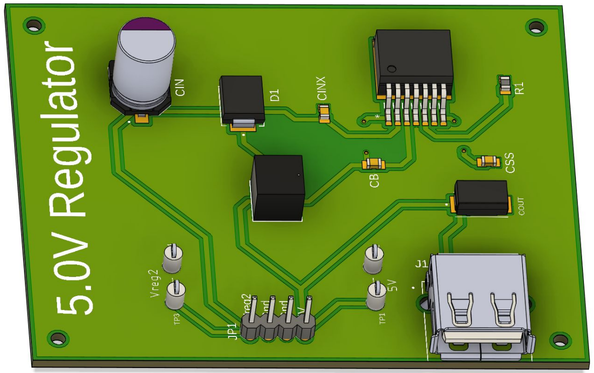


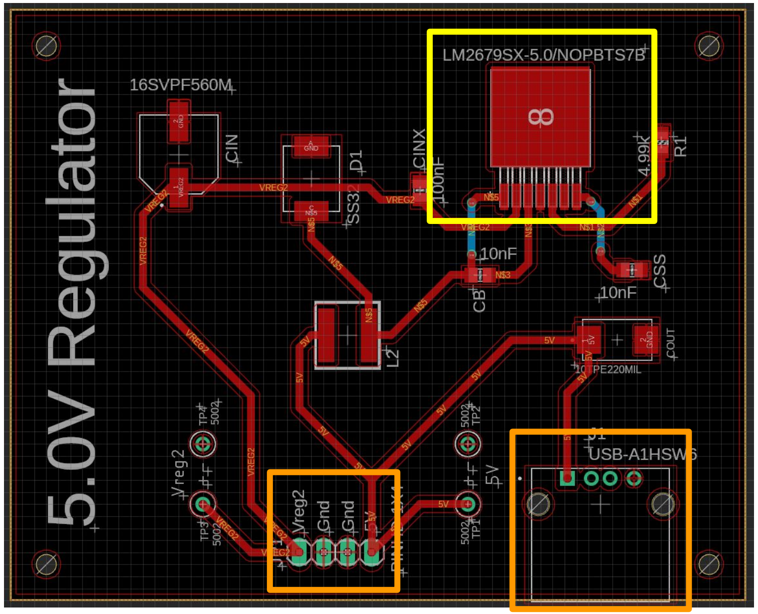




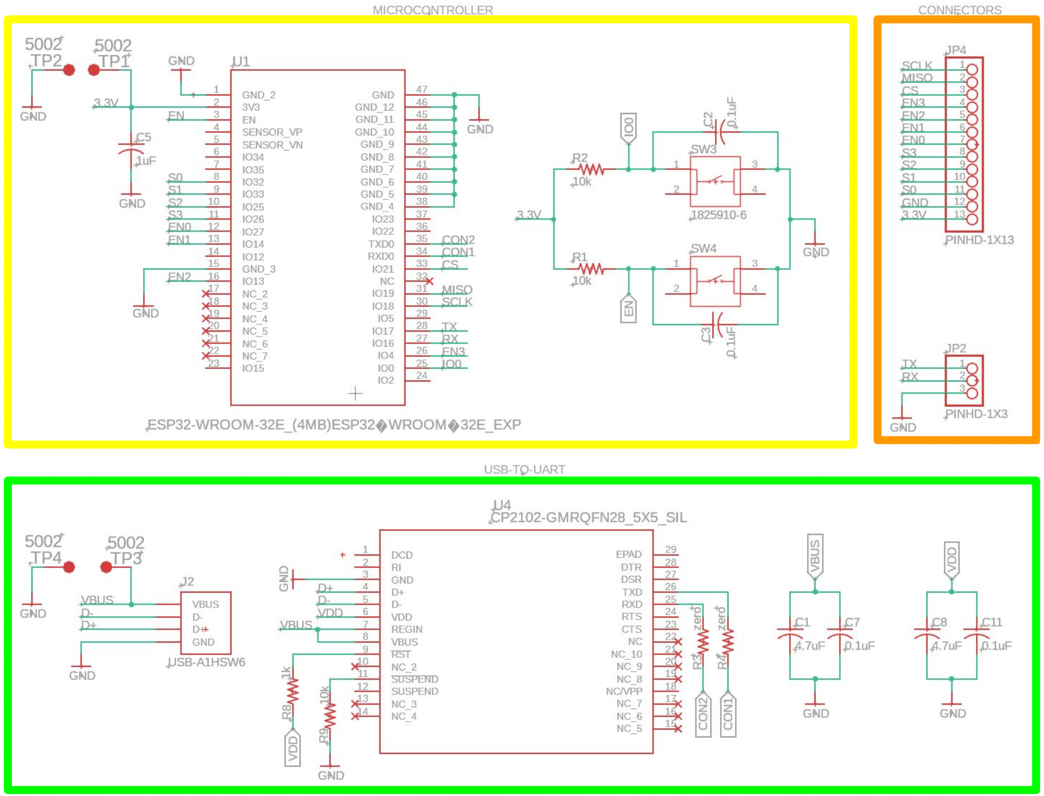
5V Regulator Schematic/PCB

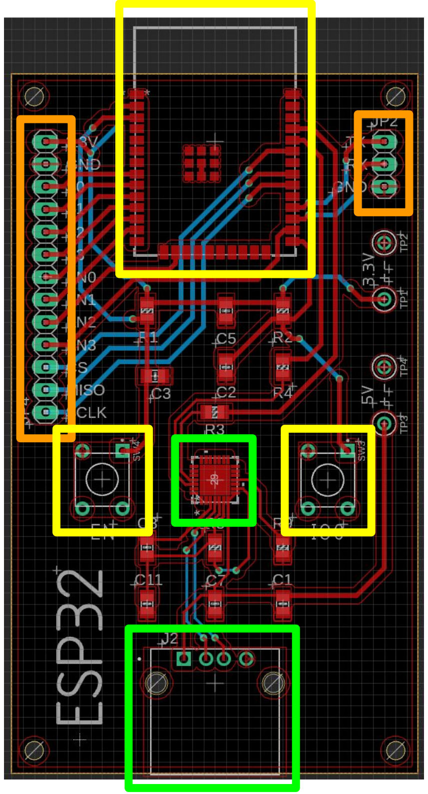


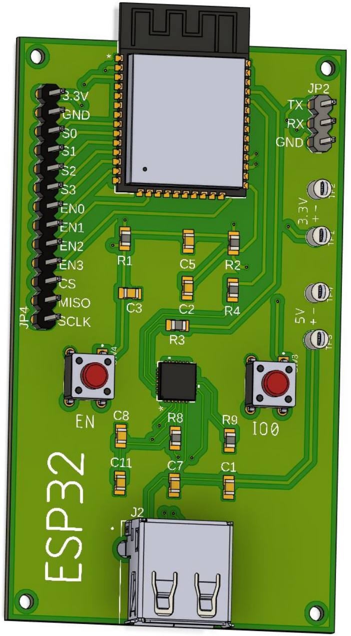




Microcontroller Schematic/PCB

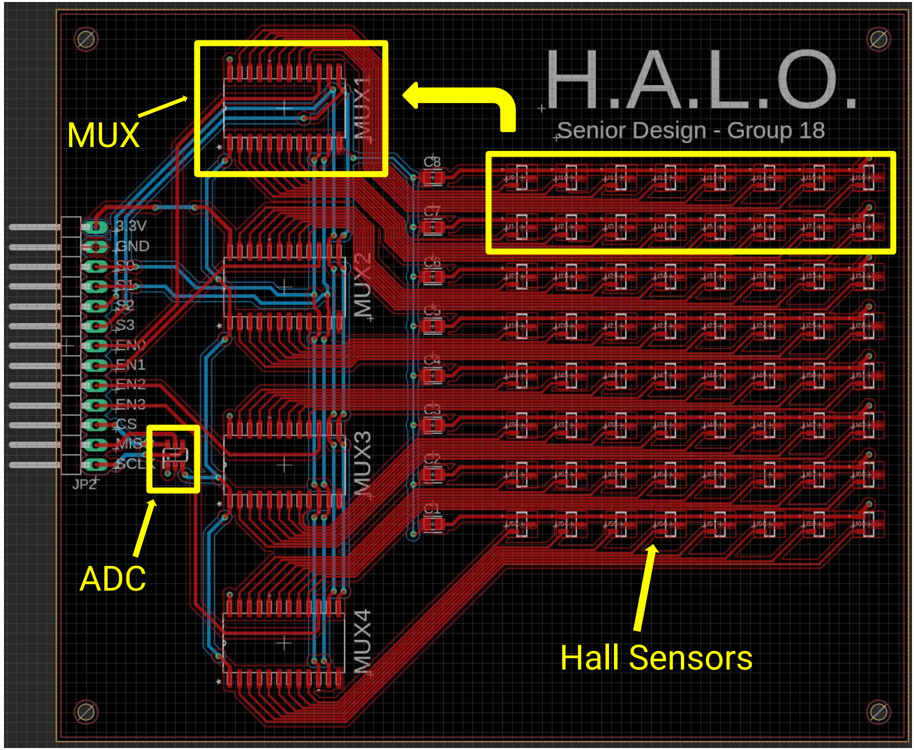




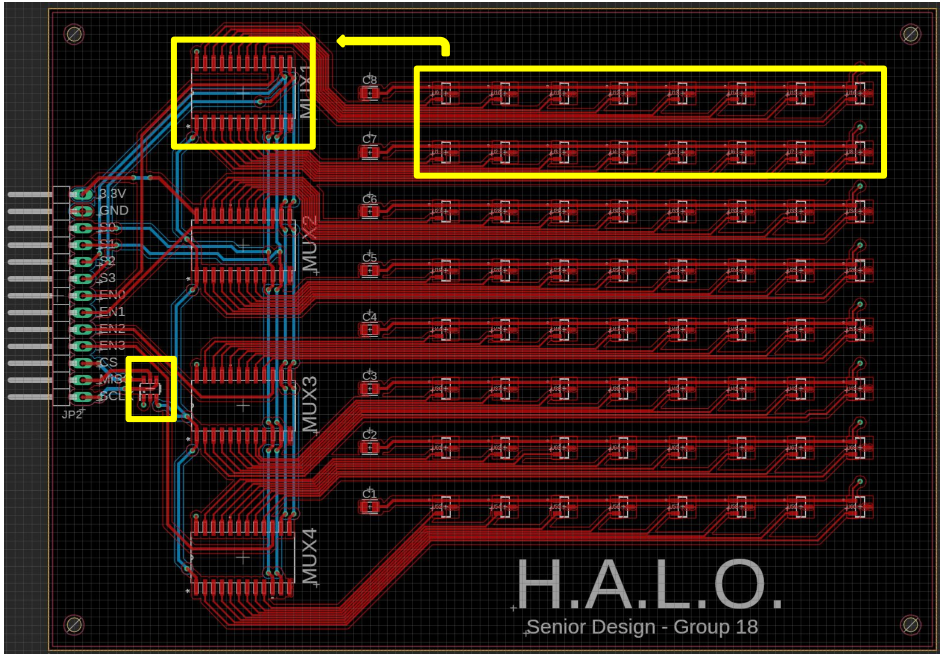


Sensor Array Schematic

Sensor Array PCBs

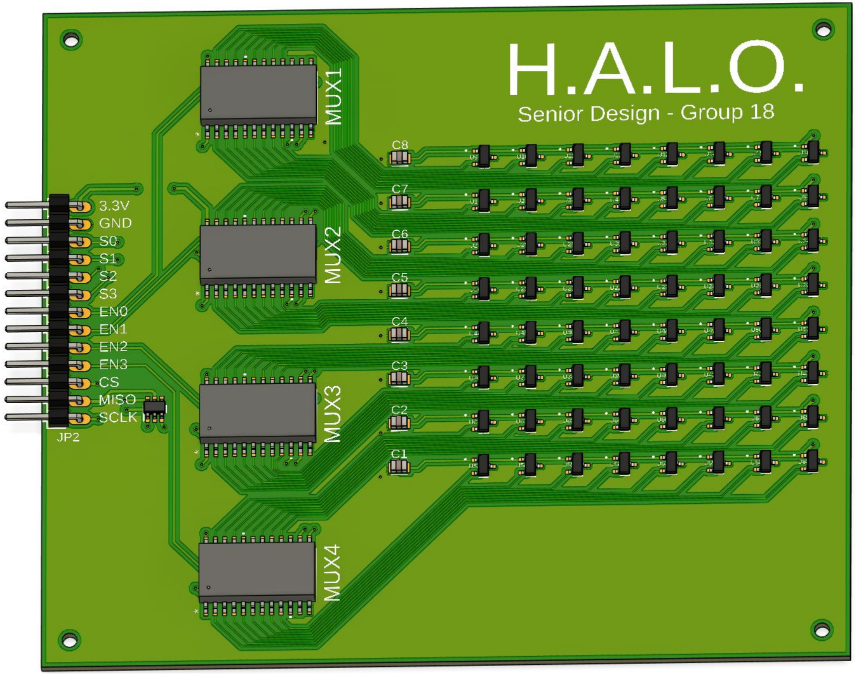


(Compact Layout)

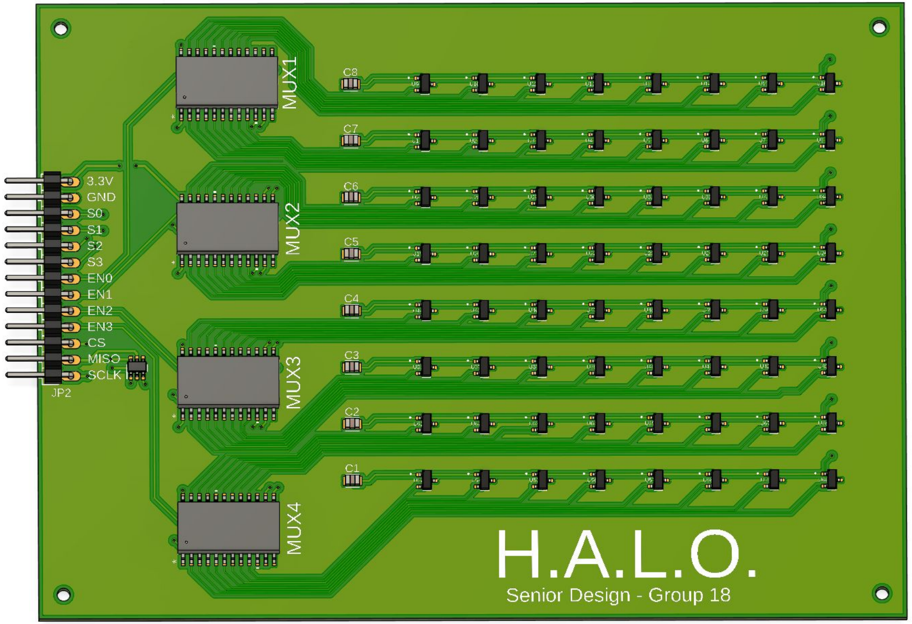


(Expanded Layout)

Sensor Array Models

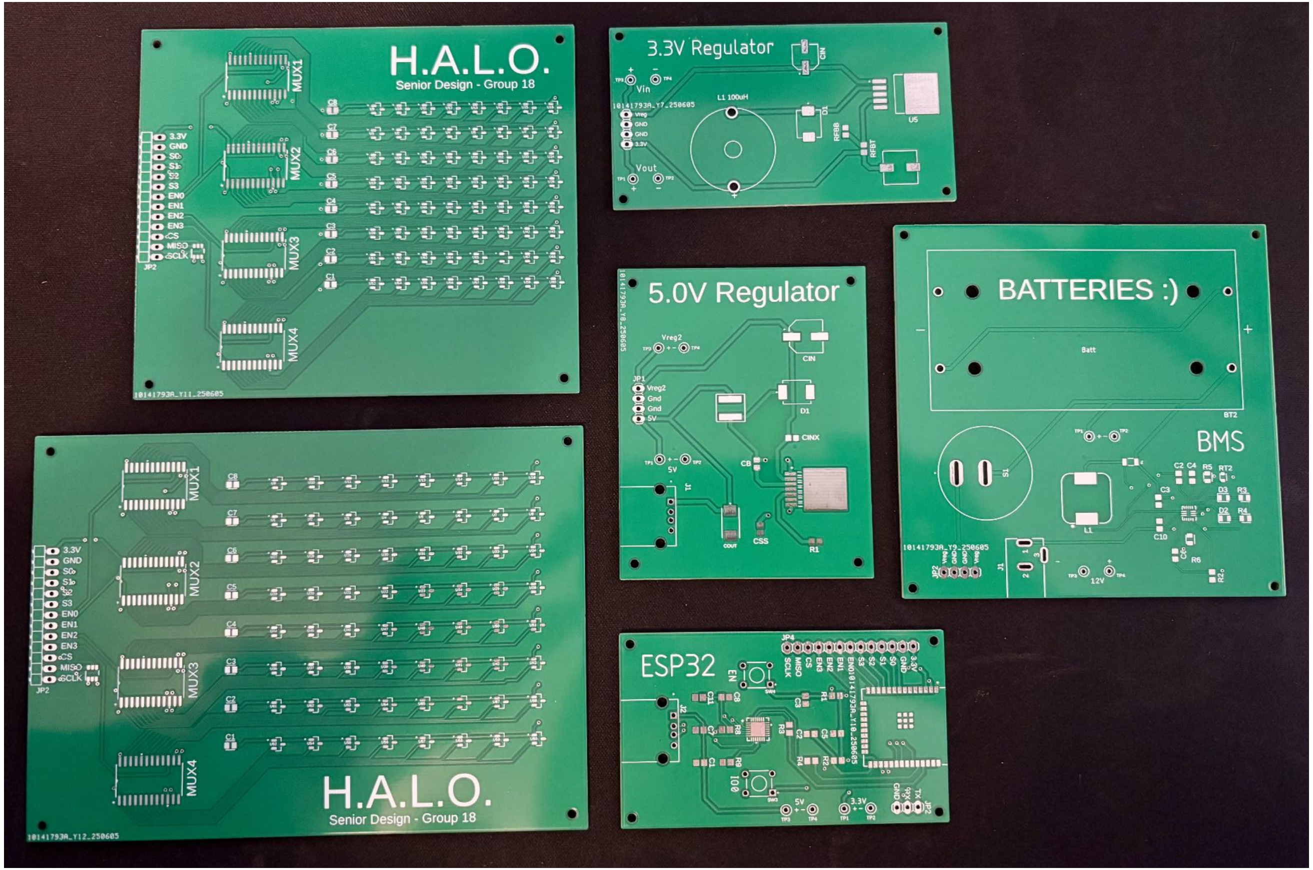


(Compact Layout)

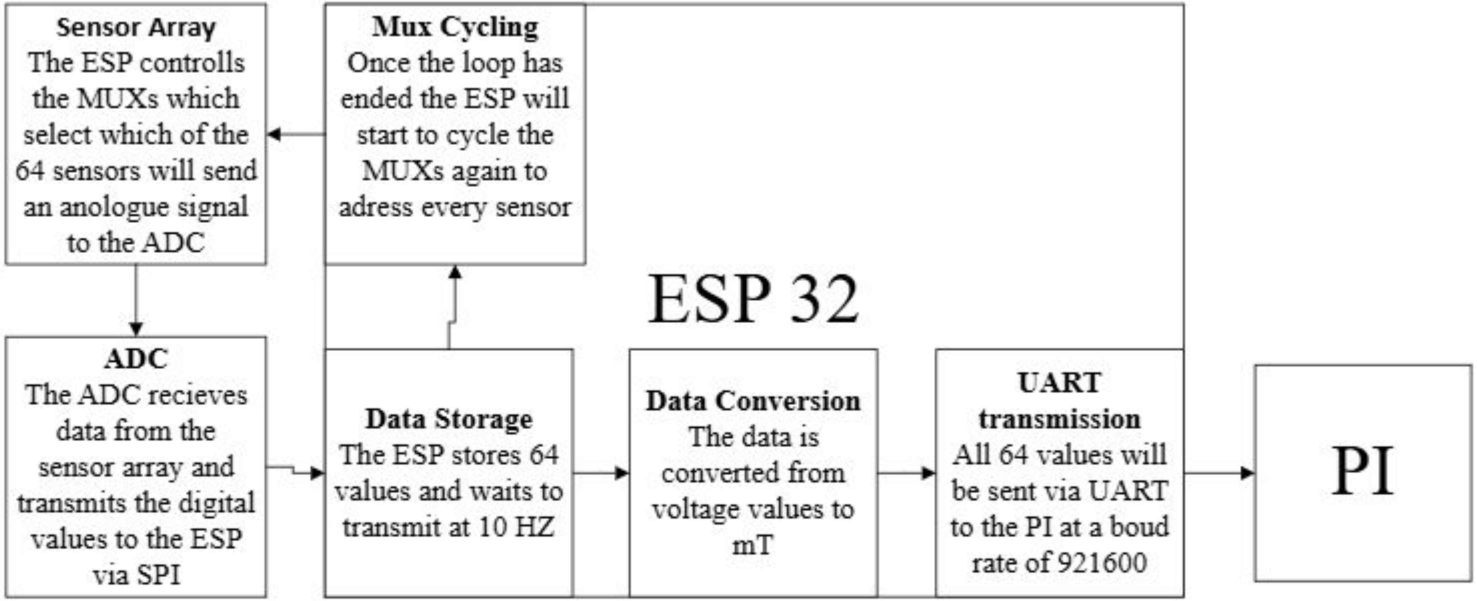


(Expanded Layout)

PCB Fabrication



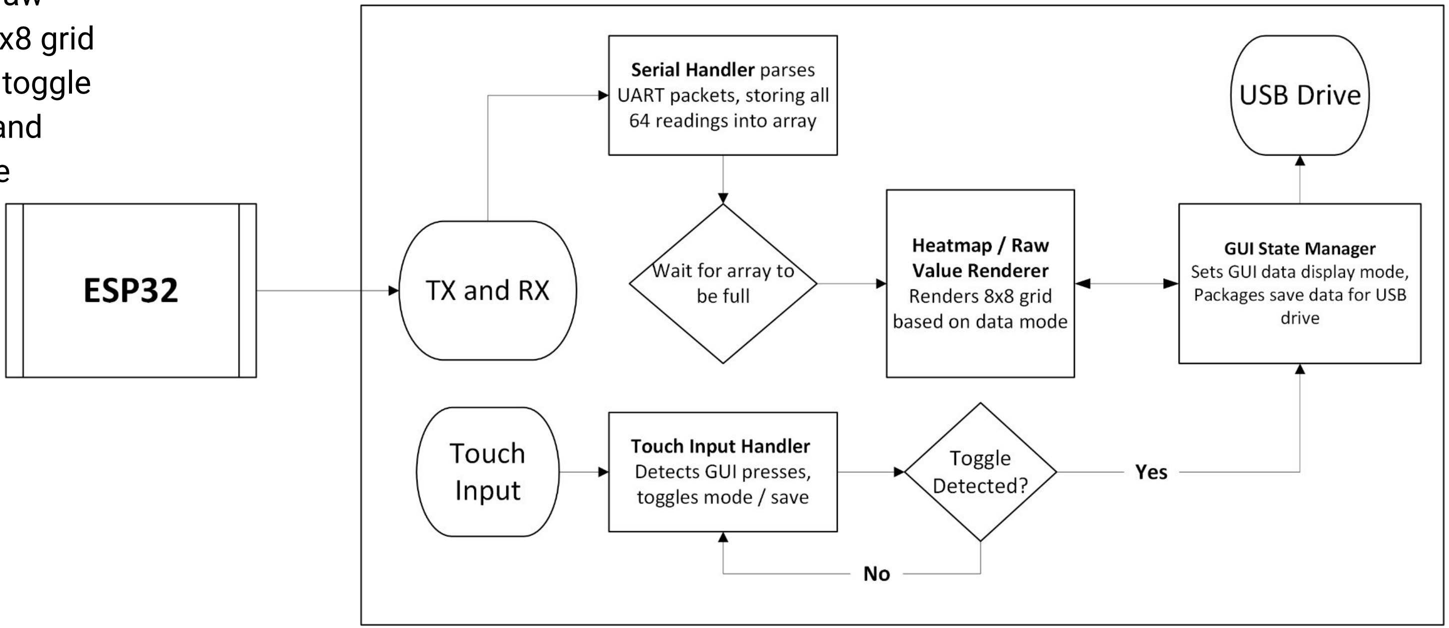
Software Design - ESP-32



Software Design - Raspberry Pi 4

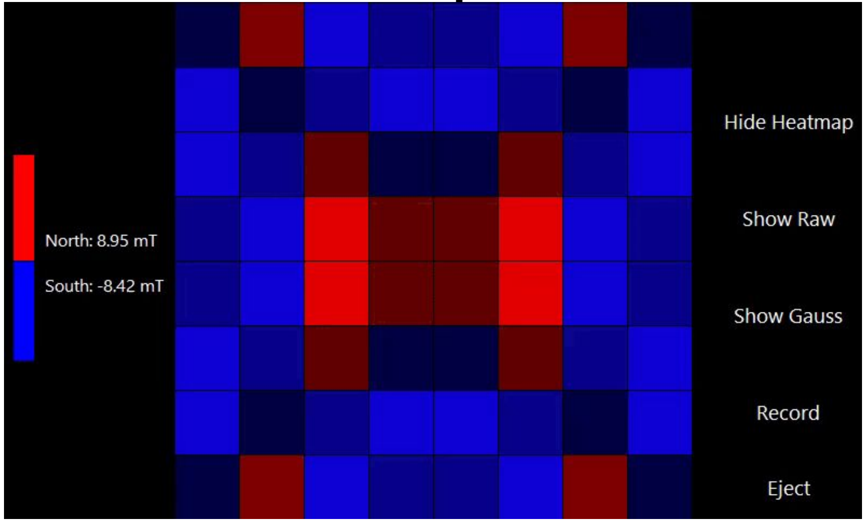
- Receive and parse UART data packets from ESP32
- Render heatmap and raw sensor values on an 8x8 grid
- Handle touch input to toggle between GUI modes, and save data to USB Drive

Raspberry Pi 4B



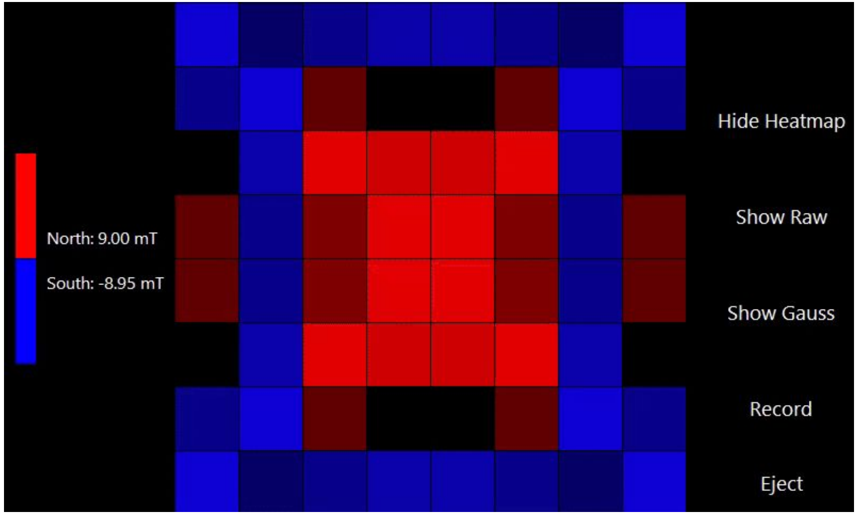
Software Design - Display GUI

Heatmap GUI



- Field intensity displayed with color mapping
- Distributed into 18 bins (colors)
- Various GUI Display control options

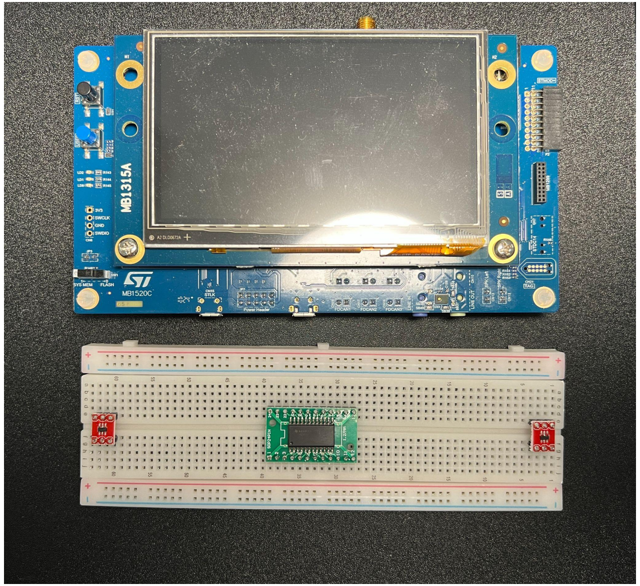
Raw Value GUI



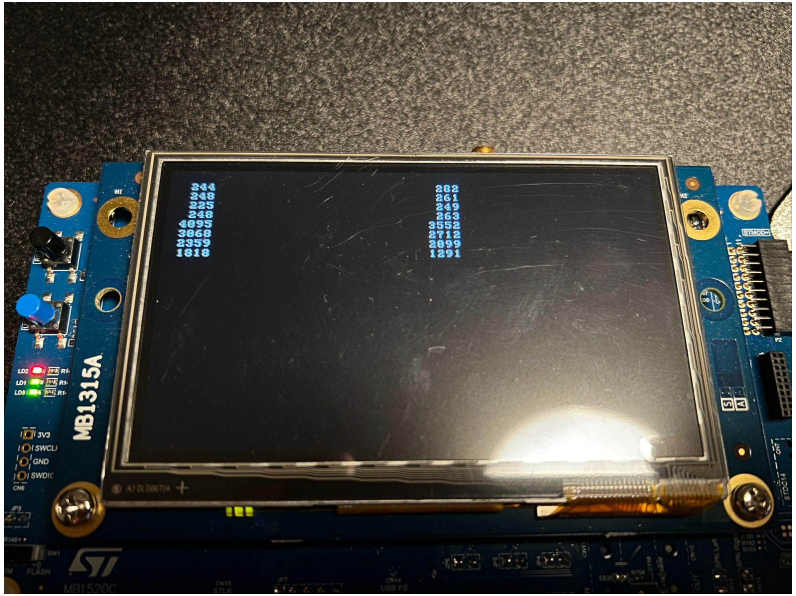
- Displays field intensity values numerically
- Cells contain readings from each sensor
- Calibrated to within 1 mT accuracy
- Can display Gauss or milliTesla

Prototyping and Testing - Display Driver & MUX

Verifying analog multiplexing scheme and initial RGB888 display interface

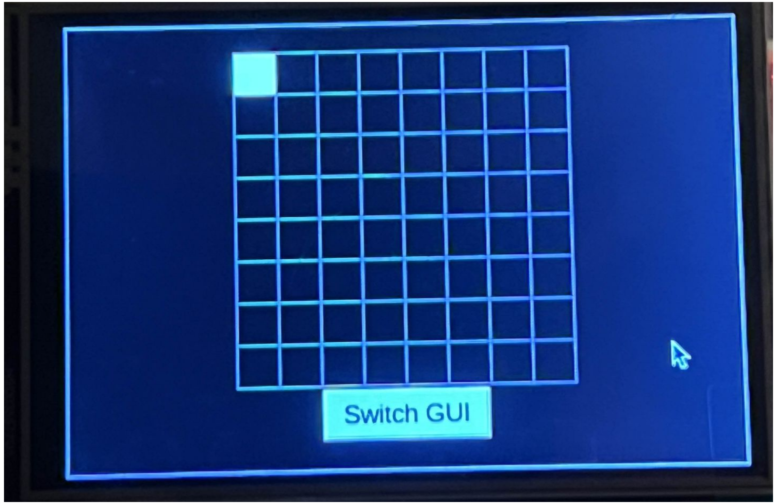


Development on STM hardware was arduous and led to redesign

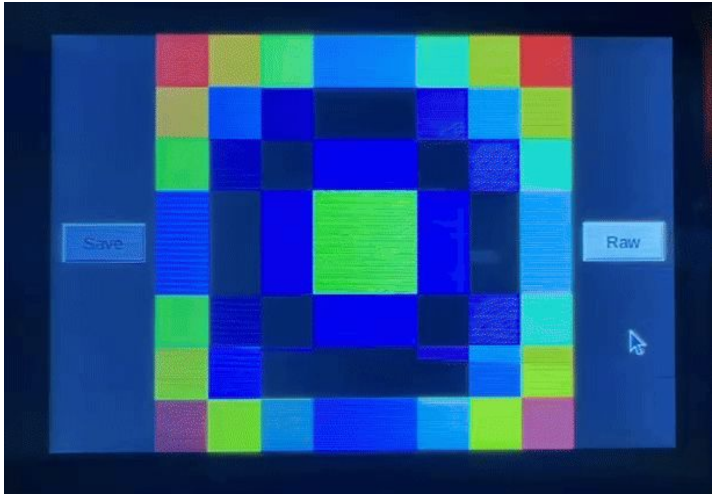


Prototyping and Testing - Display Driver & GUI

Initial Proof of Concept GUI on static display



GUI Simulation running on early driver Implementation



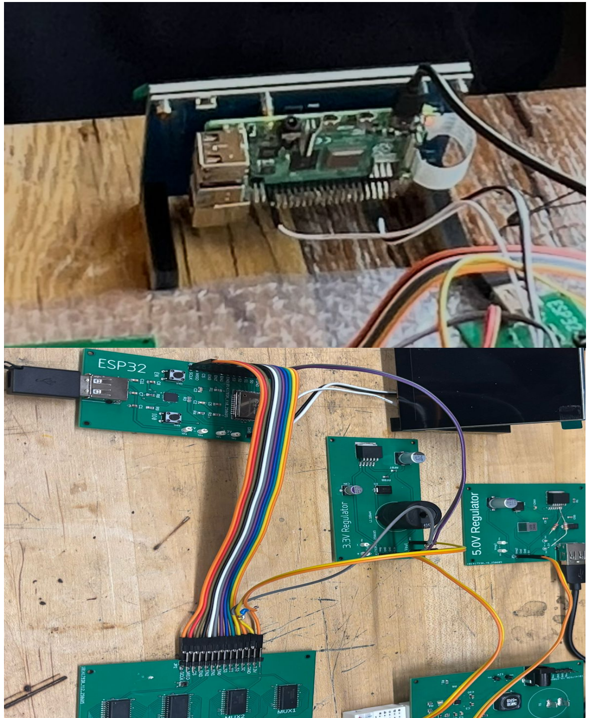
Prototyping and Testing - Screen Integration

Old Screen had input delay and required too many jumpers

New screen has better response time and a larger screen for better viewing

Old Resistive touch screen

New Capacitive touch screen



Prototyping and Testing - Sensor Array Issues

Second board had defect which led to an esp being fried

Components where resoldered onto a different board to resolve the issue

A faulty sensor was also identified on the first board


```
A fatal error occurred: Could not open COM3, the port is busy or doesn't exist.
(could not open port 'COM3': FileNotFoundError(2, 'The system cannot find the file specified.', None, 2))

Hint: Check if the port is correct and ESP connected

Failed uploading: uploading error: exit status 2
```

Prototyping and Testing - ADC Issue

The display was showing the incorrect ADC values showing 2600 for 1V

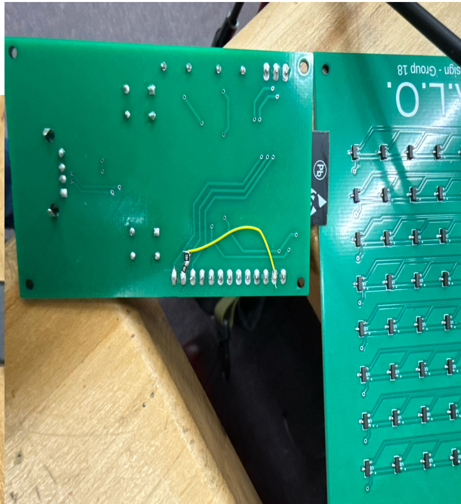
The correct value with a 3.3 reference should be about 1200

At 1.5V the value would exceed 4095 which is the max ADC and roll over back to 0

A voltage divider made of 1.8k and 3.3k resistors with a 5V reference was attempted to solve the issue before a software fix was found

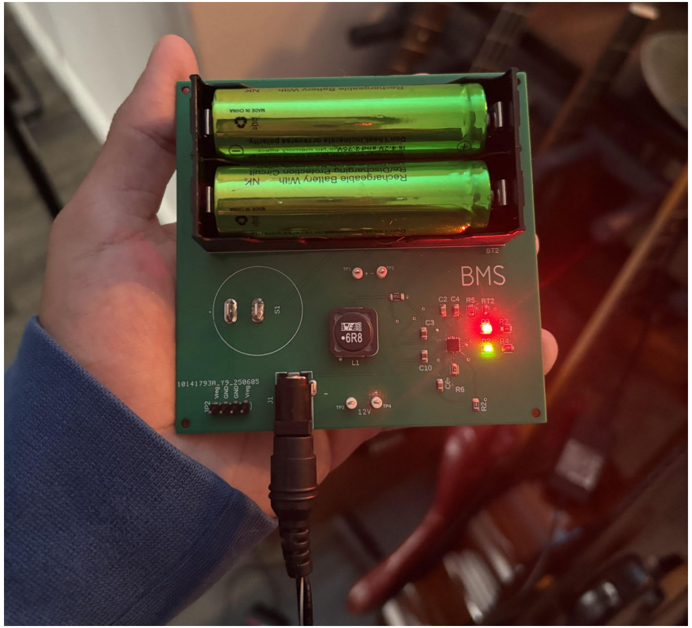
Incorrect ADC value for 1 Volt

Voltage divider on ESP



Prototyping and Testing - BMS Issue

- Identified Issue: Battery voltage dropped rapidly during charging
- Observed that charge indicator LEDs stayed on after removing the 12V input
- Likely Cause: Internal short on the MP2615
- Proposed Solutions: Test system with new MP2615 or switch to single-cell charging PCBs



Budget

Quantity	Component	Cost
64	Hall-effect sensor	$24.44
1	ADC	$4.73
4	MUX	$2.32
2	Raspberry Pi	$180
1	3.3 V regulator	$4.32
1	5 V regulator	$6.60
1	Battery management chip	$3.19
	Miscellaneous components	$90.94
	Obsolete Parts	$129.38
	Extras	$44.41
	Shipping, taxes, and tariffs	$38.20
Total		$528.53

Work Distribution

Aidan	Brogan	Gabe	Ryan
• Embedded Code • Logic Circuitry	• Power system design • Enclosure design	• GUI design • Software • Website management	• Schematic and PCB design • Board assembly and troubleshooting