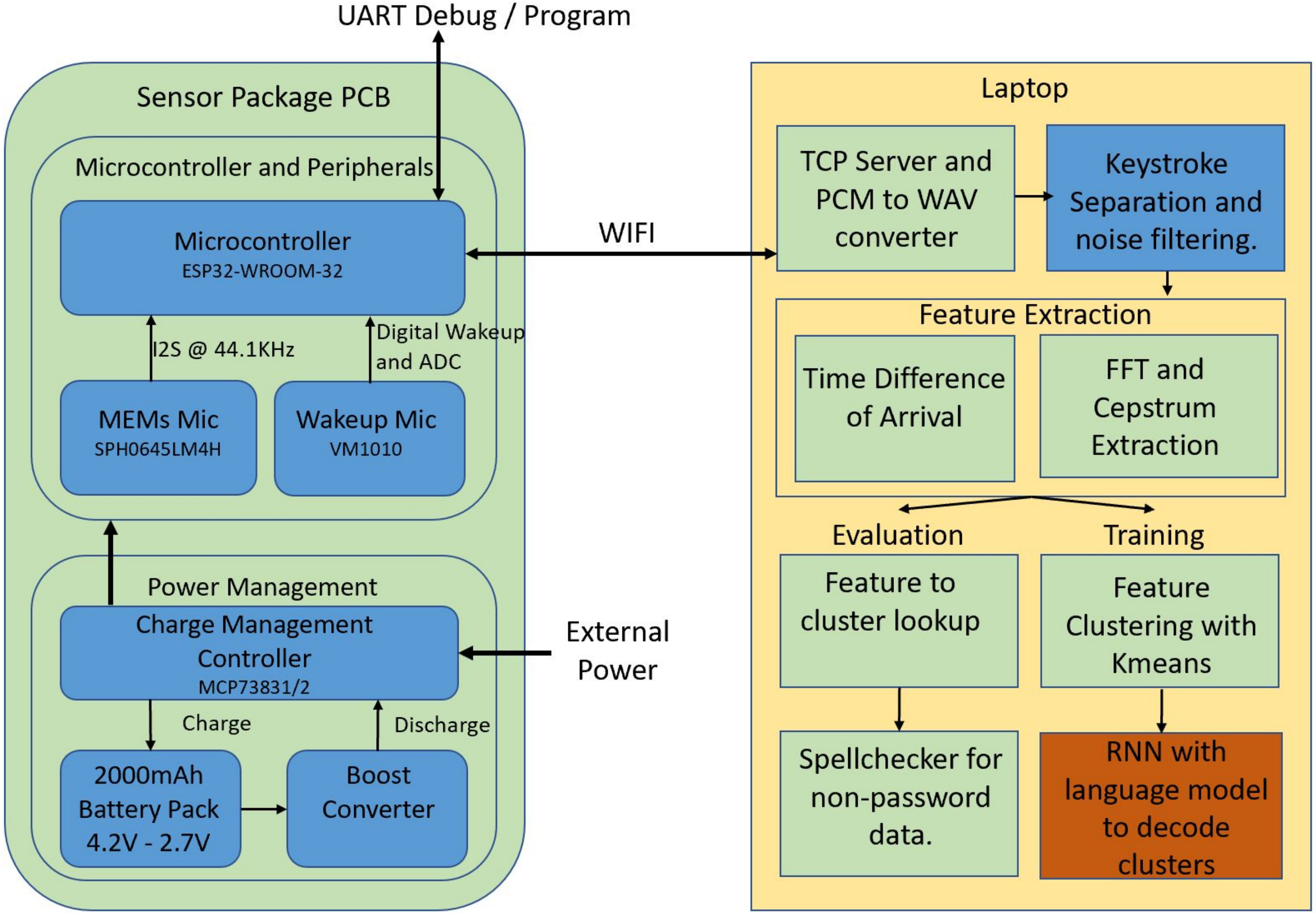
# Project L.A.K.E.

## Logging of Acoustic Keyboard Emanations

Team A2: Ronit Banerjee, Kevin DeVincentis, James Zhang

# Using Sound as a Keylogger

- Determine what a person is typing based on the sound of their keystrokes
- Exploit small differences in key sounds
- Ultimate goal: determine passwords from recordings of typing



UART Debug / Program

Sensor Package PCB

Microcontroller and Peripherals

Microcontroller
ESP32-WROOM-32

I2S @ 44.1KHz

Digital Wakeup
and ADC

MEMs Mic
SPH0645LM4H

Wakeup Mic
VM1010

Power Management

Charge Management
Controller
MCP73831/2

Charge

Discharge

2000mAh
Battery Pack
4.2V - 2.7V

Boost
Converter

External
Power

WIFI

Laptop

TCP Server and
PCM to WAV
converter

Keystroke
Separation and
noise filtering.

Feature Extraction

Time Difference
of Arrival

FFT and
Cepstrum
Extraction

Evaluation

Training

Feature to
cluster lookup

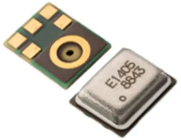
Feature
Clustering with
Kmeans

Spellchecker for
non-password
data.

RNN with
language model
to decode
clusters

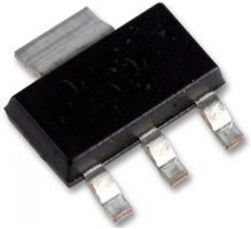
# ESP32 and Peripherals

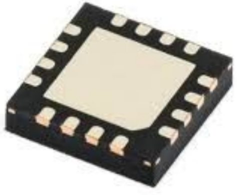
- ESP32
  - Built-in-wifi
  - Low power modes
  - Lot’s of support
- MEMS Microphone
  - SNR: 64 dB
  - Cheap
  - Nothing Exotic
  - I2S compatible
- Wake-Up Microphone
  - Ultra-low power
  - Digital and analog signals



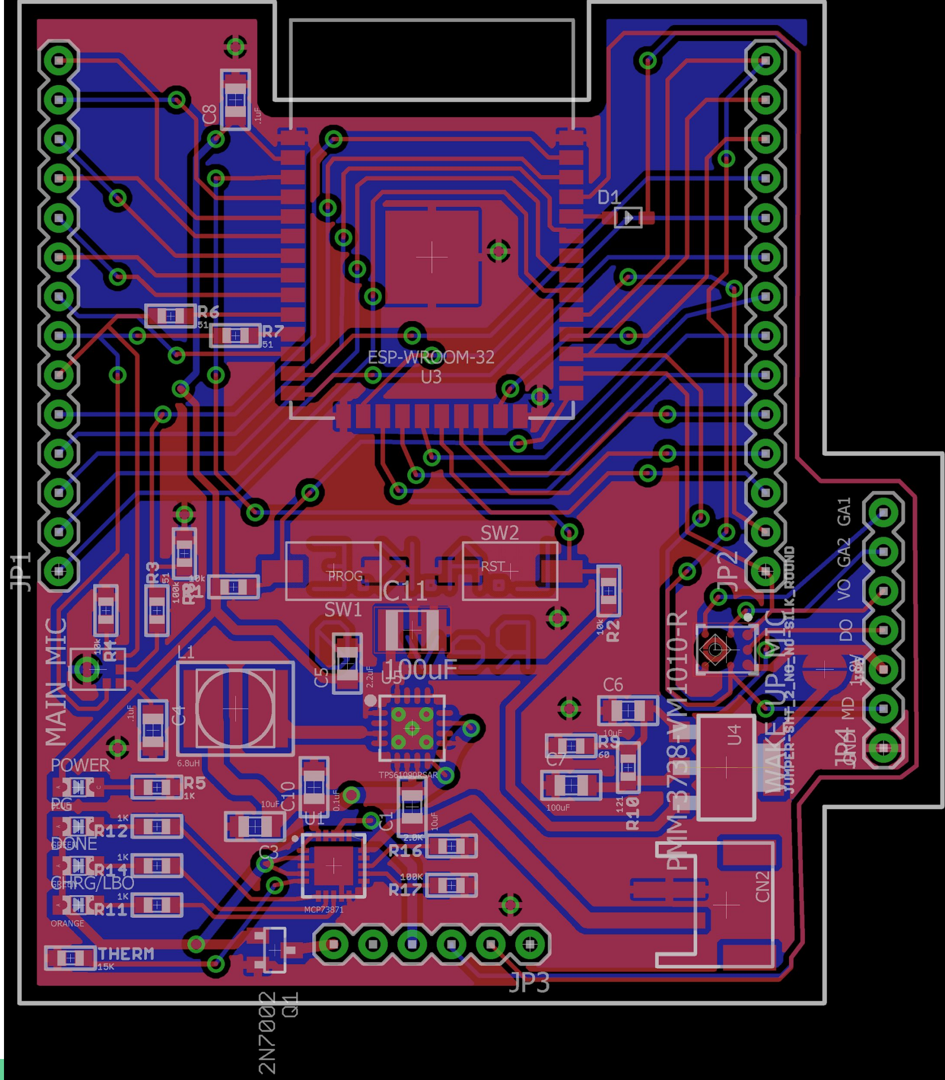
# PCB Power Management

- Charging battery on PCB
  - Self-contained unit
  - Convenient
- Doesn’t require battery while programming/debugging
- Boost converter needed when battery voltage drops
- Linear voltage regulator for 1.8V line





# PCB Layout and Routing

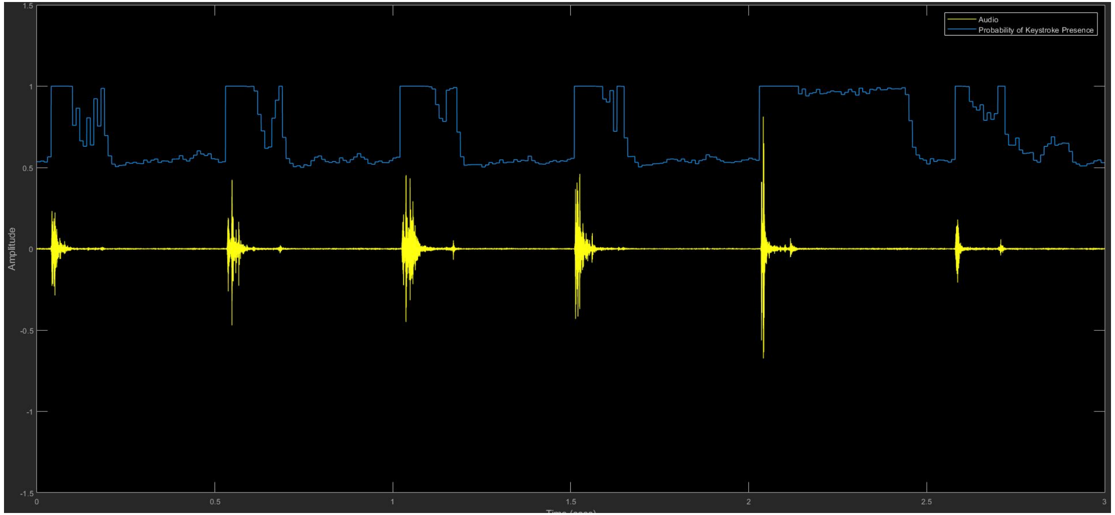


# Free RTOS, I2S, ADC, and Wifi

- Free RTOS + Espressif IoT Development Framework (ESP-IDF)
- Debugging over UART
- Inter-IC Sound Bus (I2S)
- DMA
    - Multiple buffering
- TCP Throughput requirements
    - 512kB of SRAM
    - 44.1kHz sample rate
    - 32 bit data width
    - 172kB/s of data generation

# Keystroke Isolation and Feature Extraction

- Bandpass filter from 400Hz to 12kHz
- Matlab Voice Activity Detector
- Features
    - FFT
    - Cepstral
    - TDoA



# Keystroke Clustering and Classification

- Clustering
  - K-means
  - Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
    - No pre-set number of clusters
  - NN
- Cluster-to-Key Classification
  - RNN
  - Brute force
- Spell Checker
  - Substitutions
  - Frequency vs Hamming Distance

# Metrics and Validation

- Accuracy
  - Goal: Design practical approach to match accuracy of research studies conducted in contrived situations
  - 80% of 10-character random passwords in 75 tries or less
- Power Consumption
  - Last 1 day, with at least 4 hours of acoustic activity, on a 2000mAh battery pack
- Other metrics
  - Password accuracy in 3 guesses

# Testing

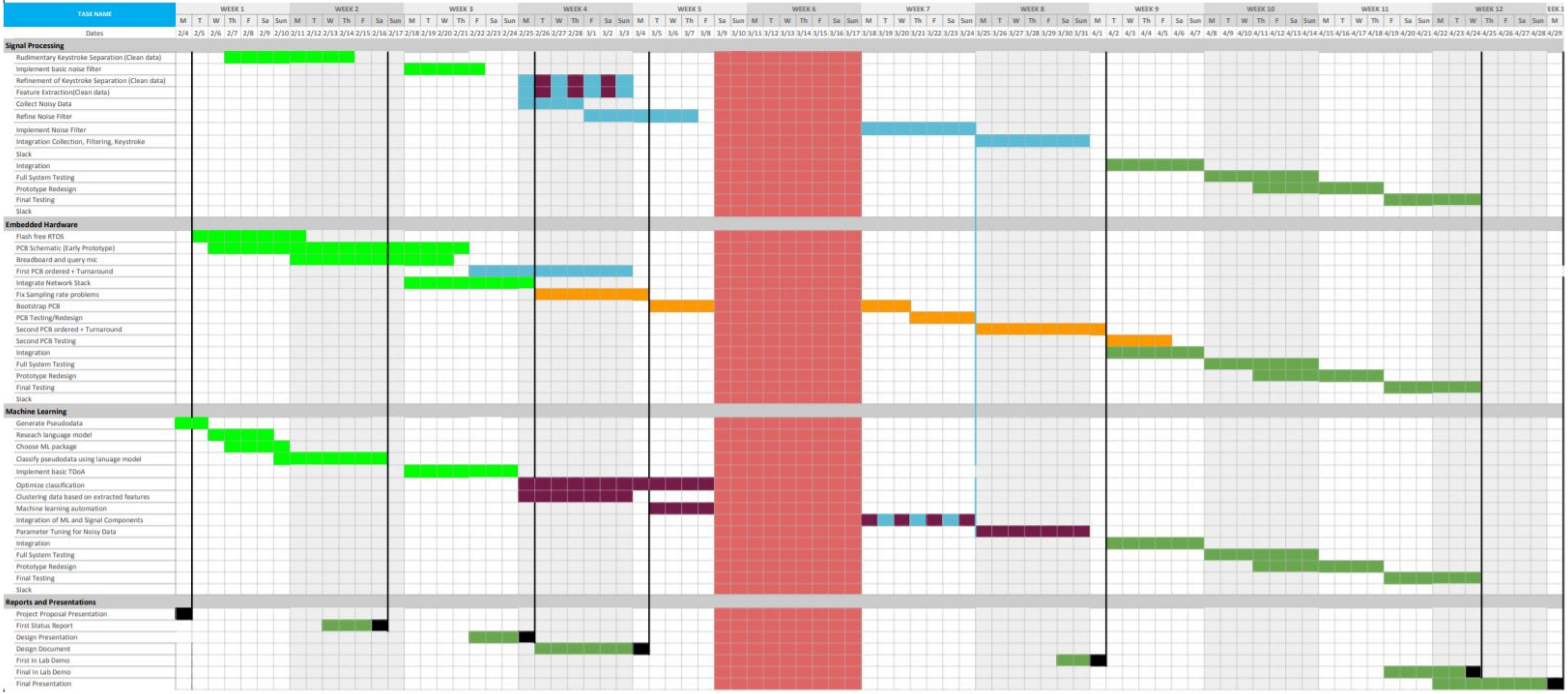
- Accuracy
  - Place device within 6” of a keyboard. User types a predetermined article, 400 to 600 words
  - Data is collected, then trained on
  - User types 20 random 10 letter strings, all lowercase
- Power Consumption
  - Measure current draw in active/sleep modes
  - Stress test in real environment (HH1303) for 24 hours, with no real data collection

## Unit Testing

- Measure packet loss over wifi
- Measure accuracy of TDoA algorithm with sound source of known position
- Clustering/classification accuracy with labeled data

# L.A.K.E Project Timeline

⬛ Deadline 🟦 Kevin 🟧 Ronit 🟪 James 🟩 Group 🟩 Task Complete



**Signal Processing**

- Rudimentary Keystroke Separation (Clean data)
- Implement basic noise filter
- Refinement of Keystroke Separation (Clean data)
- Feature Extraction(Clean data)
- Collect Noisy Data
- Refine Noise Filter
- Implement Noise Filter
- Integration Collection, Filtering, Keystroke
- Slack
- Integration
- Full System Testing
- Prototype Redesign
- Final Testing
- Slack

**Embedded Hardware**

- Flash free RTOS
- PCB Schematic (Early Prototype)
- Breadboard and query mic
- First PCB ordered + Turnaround
- Integrate Network Stack
- Fix Sampling rate problems
- Bootstrap PCB
- PCB Testing/Redesign
- Second PCB ordered + Turnaround
- Second PCB Testing
- Integration
- Full System Testing
- Prototype Redesign
- Final Testing
- Slack

**Machine Learning**

- Generate Pseudodata
- Reseach language model
- Choose ML package
- Classify pseudodata using lanuage model
- Implement basic TDoA
- Optimize classification
- Clustering data based on extracted features
- Machine learning automation
- Integration of ML and Signal Components
- Parameter Tuning for Noisy Data
- Integration
- Full System Testing
- Prototype Redesign
- Final Testing
- Slack

**Reports and Presentations**

- Project Proposal Presentation
- First Status Report
- Design Presentation
- Design Document
- First In Lab Demo
- Final In Lab Demo
- Final Presentation