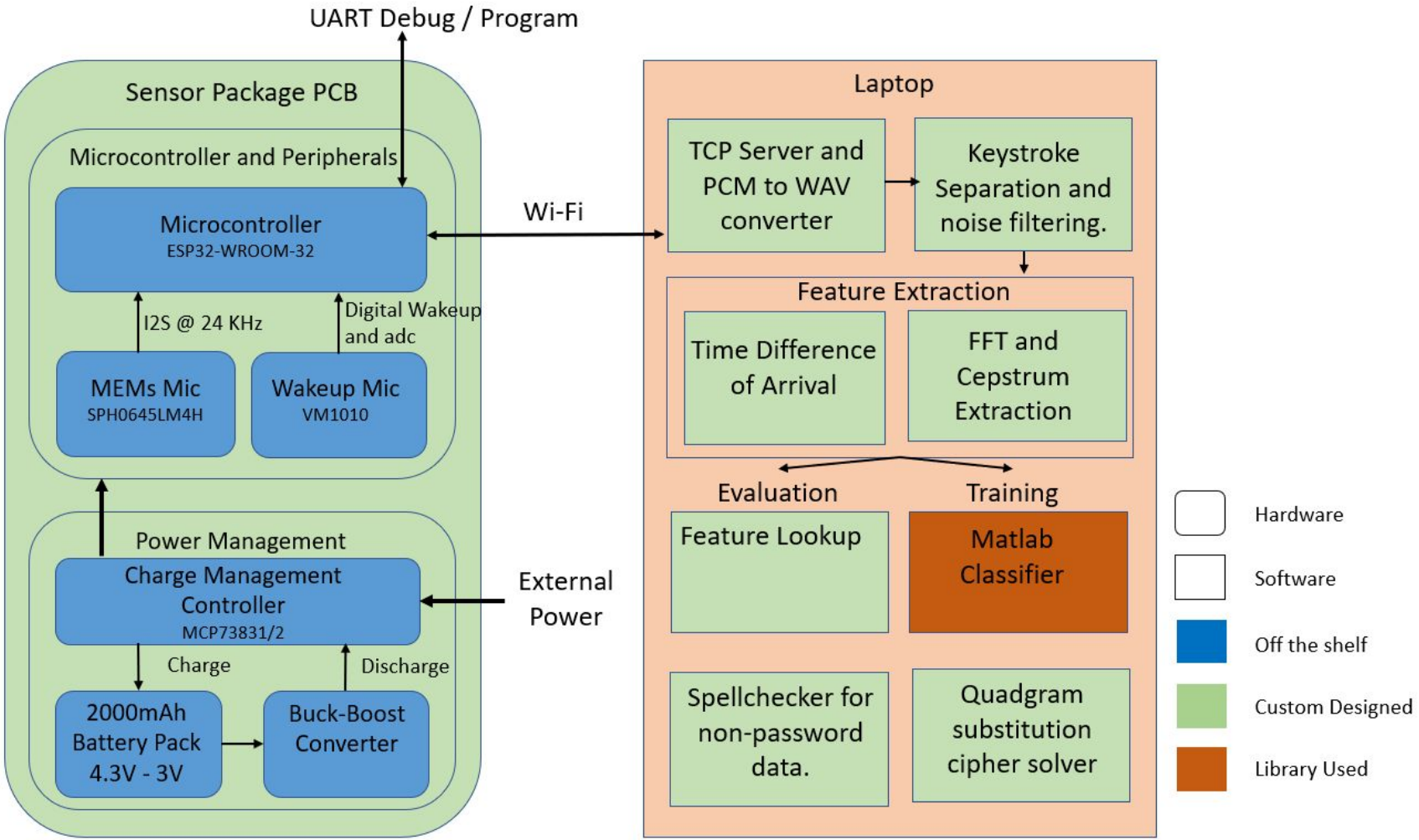
Project L.A.K.E.

Logging of Acoustic Keyboard Emanations

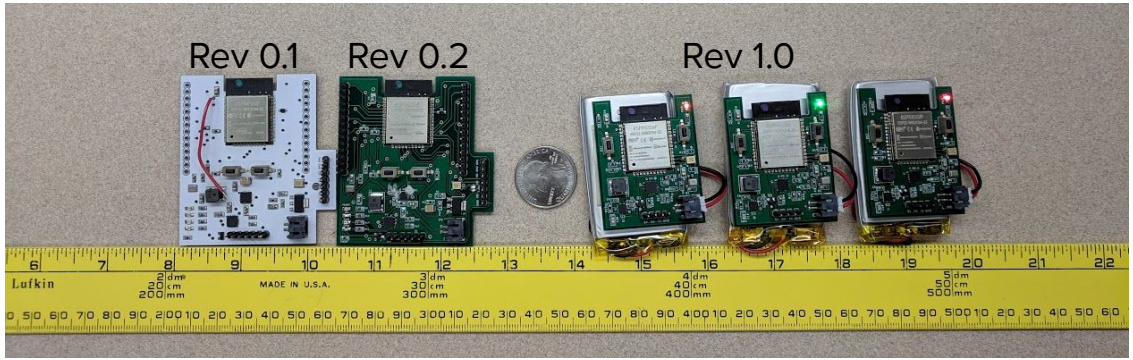
Using Sound as a Keylogger

- 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 @ 24 KHz
Digital Wakeup and adc
MEMs Mic
SPH0645LM4H
Wakeup Mic
VM1010
Power Management
Charge Management Controller
MCP73831/2
Charge
Discharge
2000mAh Battery Pack 4.3V - 3V
Buck-Boost Converter
Wi-Fi
External Power
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 Lookup
Matlab Classifier
Spellchecker for non-password data.
Quadgram substitution cipher solver
Hardware
Software
Off the shelf
Custom Designed
Library Used

Final Approach

- 3 Small PCBs to record audio
- Surround keyboard to get TDoA data
- Extract keystrokes and classify offline
- For Demo:
  - Keyboard surrounded by sound-absorbing foam
  - Use pre trained keyboard
  - Attempt to guess what user typed solely based on sound



PCB Specifications

- Goal: Last 1 day, with 4 hours of acoustic activity, on a 2000mAh battery pack
- Normal Mode: 120mA - 140mA
- Deep Sleep: 0.71mA - 0.77mA
- Can be in normal mode up to 70% of the time (16.8hr)
- Charging time: 8 hours

- Goal: 2 inches x 3 inches
- 1.5 inches x 1.9 inches

Metric: Keystroke Extraction

- Amplitude Thresholding
- Automated Finding of Threshold
- Very accurate in constant noise background (HVAC)
- Needs extra noise reduction in louder environments.

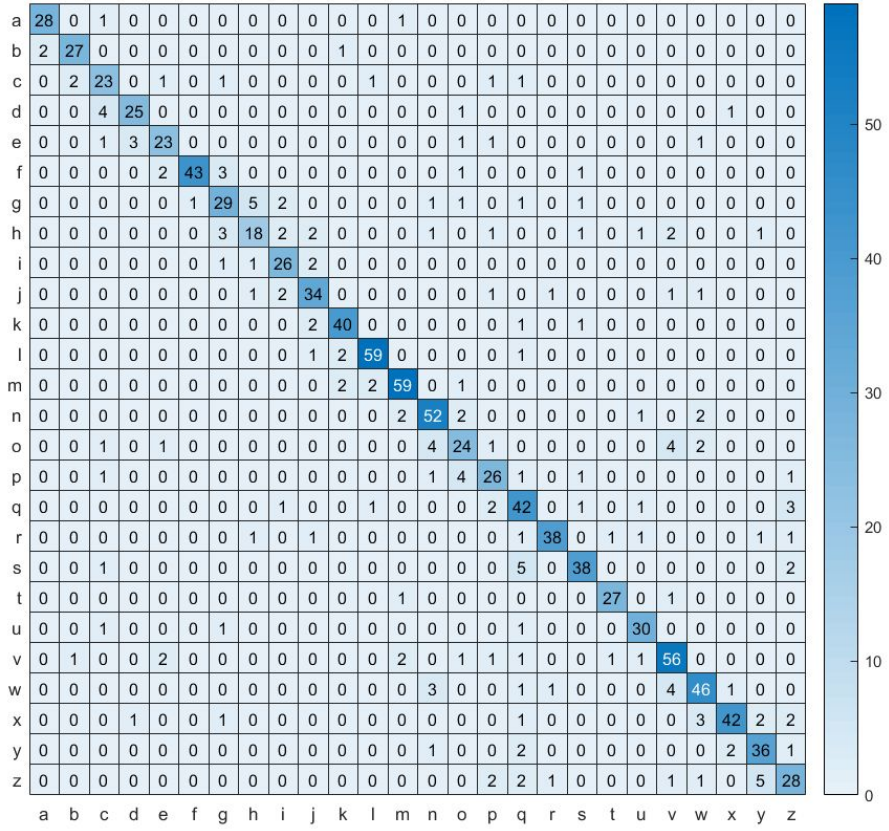
Noise Level	40dB (constant)	45dB	55dB
False Positive	0%	4%	9%
False Negative	0%	3%	1%

Clustering, TDoA, Machine Learning

- Clustering
  - FFT and Cepstral Features
  - K-means, gaussian mixture model
  - Dimensionality reduction via PCA
    - Noise was largest variance
  - Unable to successfully cluster
- 3-way TDoA
  - Issues with dropped samples
- Frequency analysis using English quadgrams from practicalcryptography.com
  - TION, THER, INTH, INGA
  - Fast and accurate
  - Resistant to noise
  - Word boundaries not needed

Metric: Classifier Accuracy

- Linear discriminant analysis
- Leave-One-Out Cross Validation
  - Error Rate: 16.9% (N = 1107)



Metric: Password Accuracy

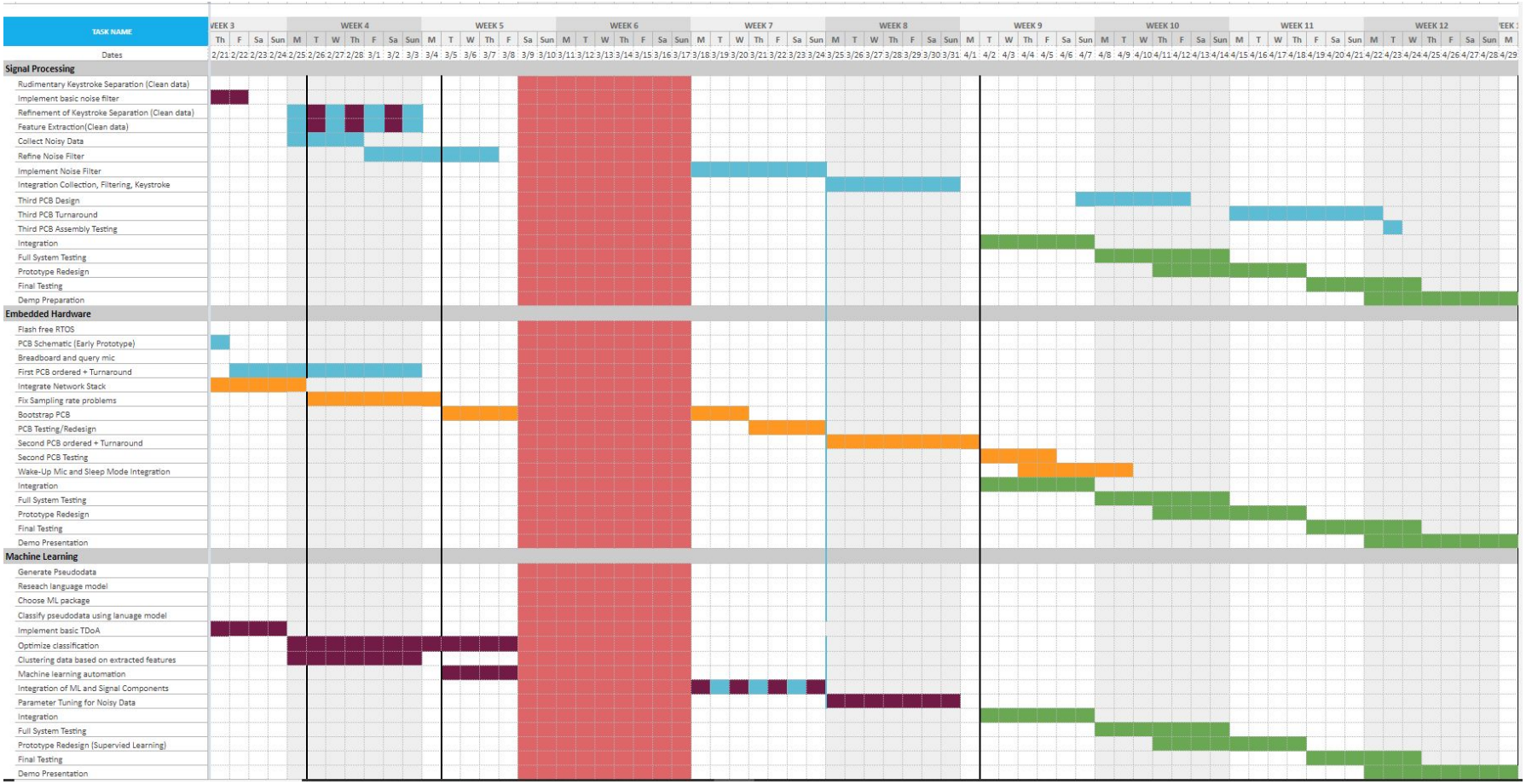
helloworld	ndlckahelu	jvmboplakc
helloworld	nduckahelu	yhmbhppaac
delloworld	nduckahelu	yvebhppaac
hulloworld	nhuckahelu	yvmhhppaac
heploworld	ndlckahelu	yvmboppaac
helioworld	ndufkahelu	yvmbhopaac
hellpworld	nduccahelu	yvmbhpoaac
helloyorld	nduckehelu	yvmbhpppac

- Target: 80% of 10-character random passwords in 75 tries or less
- Result:
  - 60% within 75 tries

Summary of Metrics

	Specifications	Actual
Size	2 inches x 3 inches	1.5 inches x 1.9 inches
Power	Last 1 day, with 4 hours of acoustic activity, on a 2000mAh battery pack	17 hours of acoustic activity
Processing Time	1 hour	10 minutes
Accuracy	80% of 10-character random passwords in 75 tries or less	60% of 10-character random passwords in 75 tries or less

Schedule


TASK NAME
Dates
WEEK 3
WEEK 4
WEEK 5
WEEK 6
WEEK 7
WEEK 8
WEEK 9
WEEK 10
WEEK 11
WEEK 12
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
Third PCB Design
Third PCB Turnaround
Third PCB Assembly Testing
Integration
Full System Testing
Prototype Redesign
Final Testing
Demp Preparation
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
Wake-Up Mic and Sleep Mode Integration
Integration
Full System Testing
Prototype Redesign
Final Testing
Demo Presentation
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 (Supervied Learning)
Final Testing
Demo Presentation

Lessons Learned

- Noise reduction is hard
- If something doesn’t work as well as you wanted, don’t just throw it away
- Don’t be afraid to ask professors/other students for help
- Pick something within your area of expertise