Slow-Wave Sleep Detector

ECE 445 – Senior Design Laboratory

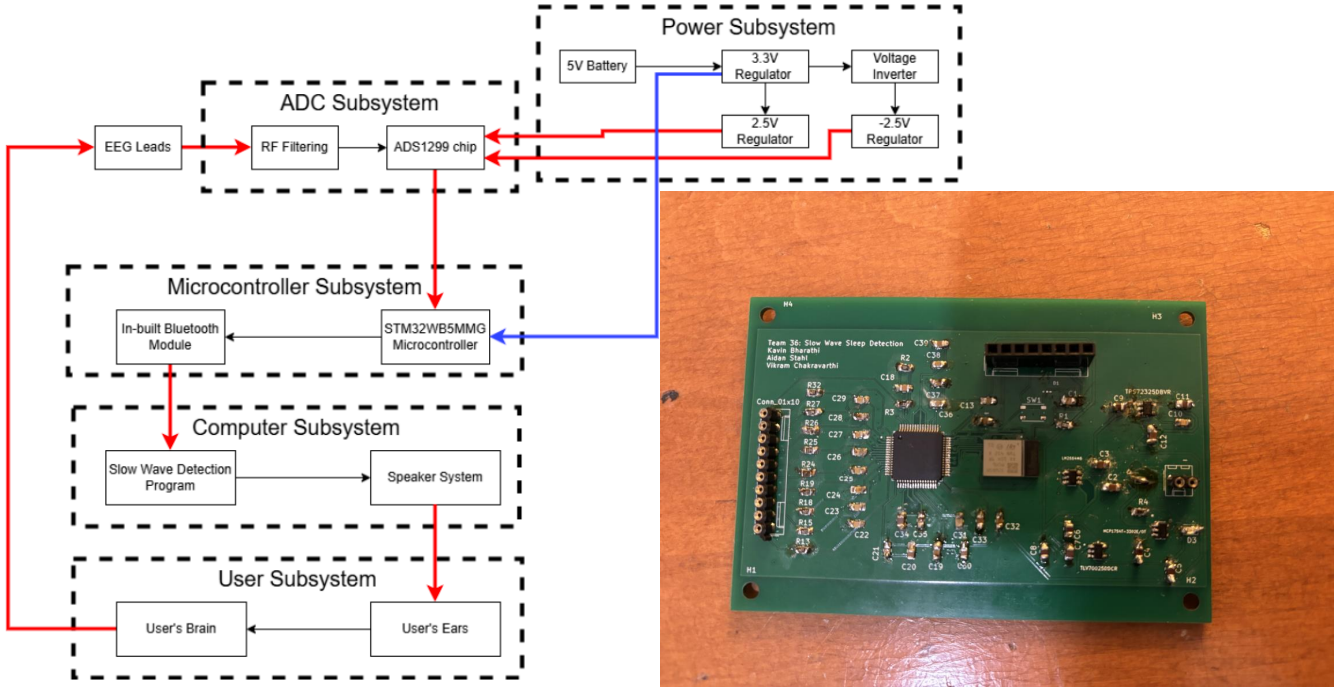
By: Kavin Bharathi, Aidan Stahl, and Vikram Chakravarthi

May 5, 2026

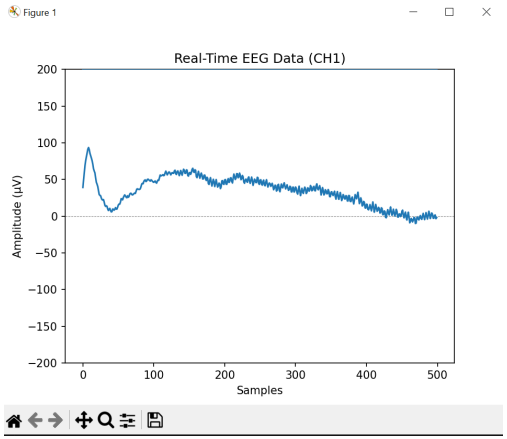
Objective

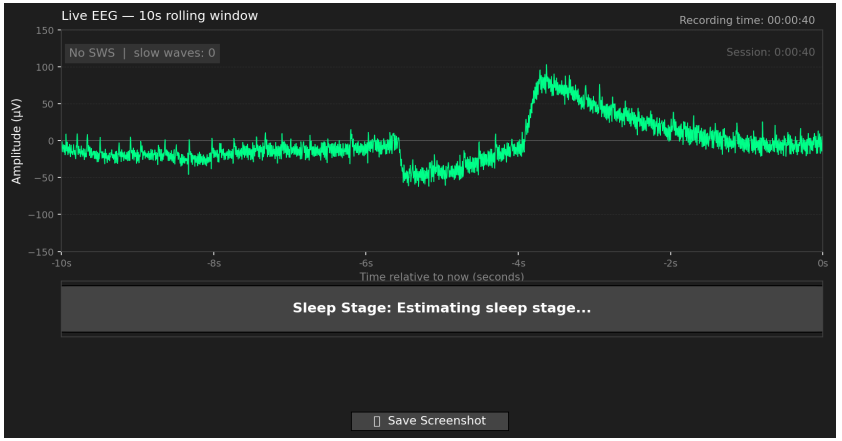
Enhancing sleep quality to combat common sleep-related disorders and memory consolidation issues by measuring EEG brain waves and playing pink noise while the user is in slow-wave sleep

Block Diagram



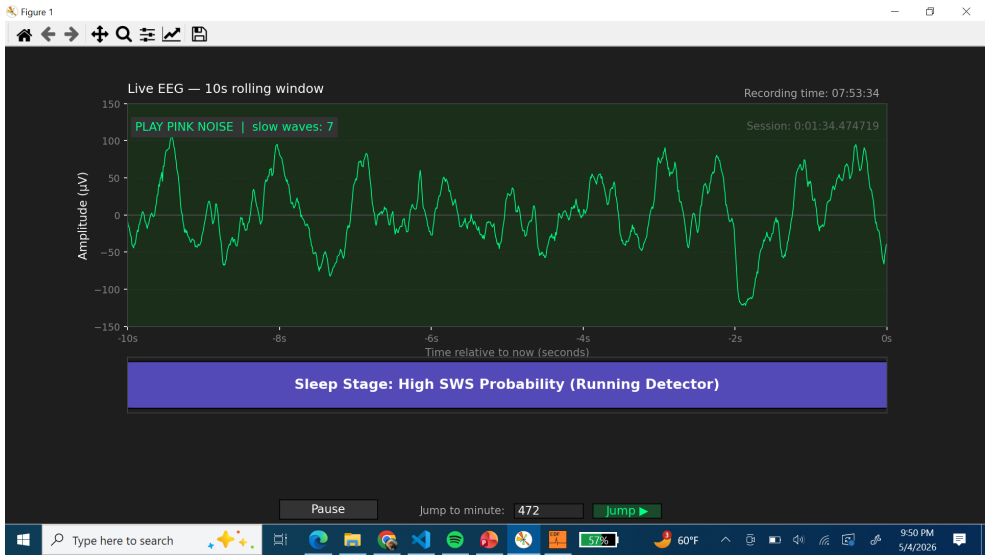
Finalized Plotting Window





Screenshot of Live Data from our PCB vs Cyton Board (Alternate PCB for EEG signal transmission)

Finalized Plotting Window



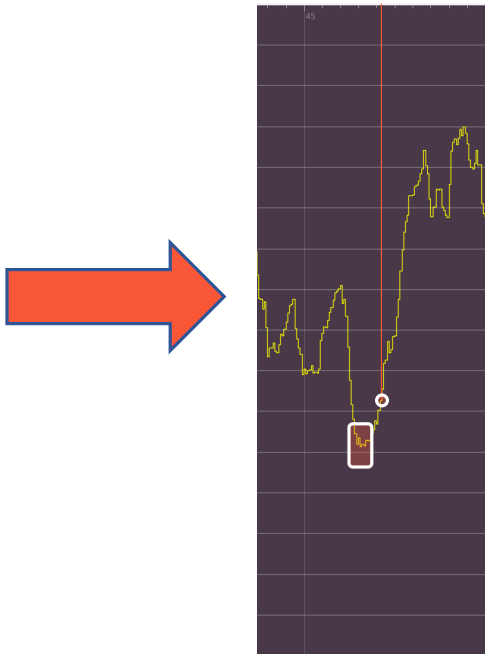
Detecting Slow Wave Sleep in publicly available EEG data

High Level Requirements

- **Play "Pink Noise" within 300 ms of detecting slow-wave sleep**
- **Support 10 hours of consecutive sleep, meaning the battery should last at least 10 hours**
- **Average comfort rating of the headset should be a 4/5**

Play Pink Noise Within 300 ms of Detecting SWS

```
07:30:14.114  LiveSOStimulator started
07:30:20.183  Filter warm-up
07:30:25.232  Detection active. Initial threshold: -80.0 µV
07:30:35.331  SO #1 played - thresh=-80.0 µV - played within 300 ms.
07:30:35.331  Pink noise played - detection #1
07:30:45.410  SO #2 played - thresh=-80.0 µV - played within 300 ms.
07:30:45.410  Pink noise played - detection #2
07:30:45.410  SO #3 played - thresh=-80.0 µV - played within 300 ms.
07:30:45.410  Pink noise played - detection #3
07:31:16.074  SO #4 played - thresh=-80.0 µV - played within 300 ms.
07:31:16.074  Pink noise played - detection #4
07:31:16.074  LiveSOStimulator stopped. Detections this session: 4.
```



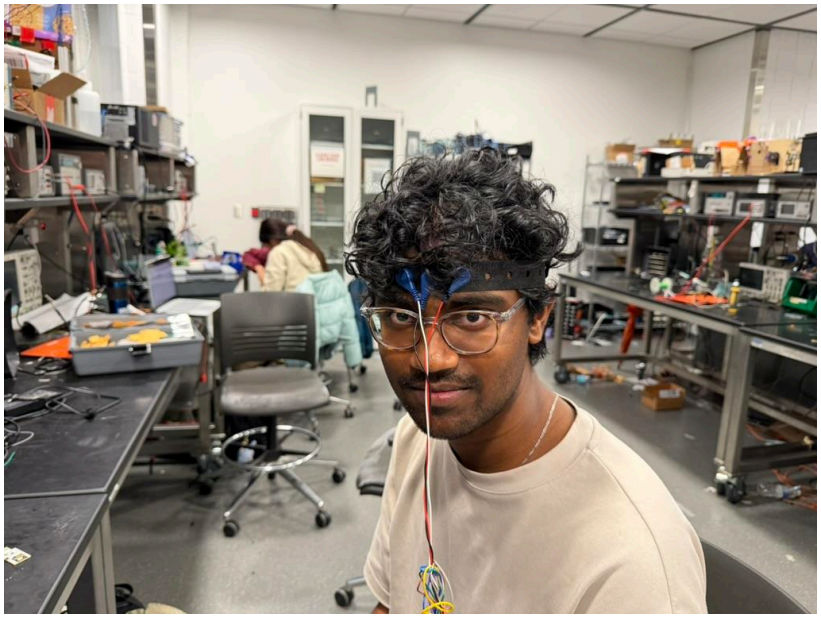
Battery Should Last At Least 10 Hours

We used a 9-volt battery with an estimated capacity of 600 mAh

Our circuit drew roughly 30 mA of current while in use

Based on this, battery life should be around 18-20 hours

Average Comfort Rating of 4/5



Requirements and Verifications

Requirements	Verifications
• Microcontroller and ADC must receive between 1.8V and 3.6V (ideally around 3.3V)	• Power the regulators with a 6V lithium-ion battery and measure the voltage received at the Vdd input pin of the MCU and the DVDD pins of the ADC. Confirm that the values are around 3.3V and within the required range of 1.8V and 3.6V.
• The analog positive power supply pins (AVDD) must receive a voltage between 2.475V and 2.625V (ideally around 2.5V).	• Power the regulators with a 6V lithium-ion battery and measure the voltage received at all AVDD pins of the ADC. Confirm that the values are around 2.5V and within the required range of 2.475V and 2.625V.
• The analog negative power supply pins (AVSS) must receive a voltage between -2.475V and -2.625V (ideally around -2.5V).	• Power the regulators with a 6V lithium-ion battery and measure the voltage received at all AVSS pins of the ADC. Confirm that the values are around -2.5V and within the required range of -2.475V and -2.625V.

Table 1: Power Subsystem Requirements and Verifications

Requirements	Verifications
• At low-frequency inputs, the ADS must output the correct digitized measured voltage values within a standard error of ±1%.	• Use a signal generator to generate a low-frequency signal in the range of 1-100 Hz with a very low voltage and ensure the ADC's digitized output corresponds with the voltage set on the signal generator within a ±1% standard error. • Test with multiple frequencies in the range of 1-100 Hz and multiple voltages to ensure accuracy remains.
• High-frequency signals, such as bluetooth's 2.4 GHz frequency, should be filtered out.	• Use a signal generator to generate a high-frequency signal in the range of 2-3 GHz with a low voltage similar to what an expected EEG voltage would be. Ensure the ADC's digitized output is very close to zero with at least a suppression of - 90 dB or greater (for frequencies >= 2.4 GHz). • Test with multiple frequencies in the range of 2-3 GHz and multiple voltages to ensure signals remain suppressed.
• The sampling frequency should be greater than or equal to 200 Hz meaning the Nyquist frequency is at least 100 Hz, which ensures that there is no aliasing for signals less than or equal to 100 Hz.	• Use a signal generator to generate a sinusoidal wave. Test inputs at various frequencies from 10 Hz to 100 Hz and ensure the digitized output is the correct frequency within a ±1% standard error. • Plot the ADS's output while changing the voltage at a constant frequency and make sure that the output graph matches the frequency and voltages expected with a standard error of ±1. ○ Test with several frequencies to ensure accuracy is maintained.

Table 2: ADC Subsystem Requirements and Verifications

Requirements and Verifications

Requirements	Verifications
• Able to see digital signals corresponding to EEG on a receiver with sampling rate of 250 Hz and resolution of 24 bits.	• Use a signal generator to generate a specific frequency within the range of 1-100 Hz. Set the voltage at a low range, similar to what would be expected from an EEG lead. • Check amplitude and received signals to see if they are the same as the signal from our SG within a ±1% standard error. • Confirm that the sampling rate is within ±1% standard error of actual sampling rate of 250 Hz. • Validate that amplitude corresponds to gain scaling. • If Bluetooth communication is not working, try to see if EEG signals are received through wired UART connection. • Test with multiple different frequencies from the SG within a range of 1-100 Hz and change the amplitudes as well to simulate representing multiple different voltage peaks.
• SPI communication works as expected at 8 MHz with no missed DRDY interrupts, no bit errors.	• Check if the digital information at the DRDY pin corresponds to the EEG received on the microcontroller output. • Check if proper MISO/MOSI transitions are made, SPI clock frequency matches. • Make sure that DRDY pins toggle at the correct interrupt times.
• Bluetooth operates with latency below 100 ms with packet loss rate less than 2% and throughput of 54 kbps.	• Timestamp the transmission and reception to compute actual latency. • Measure packet loss by streaming known data streams for a certain fixed time.

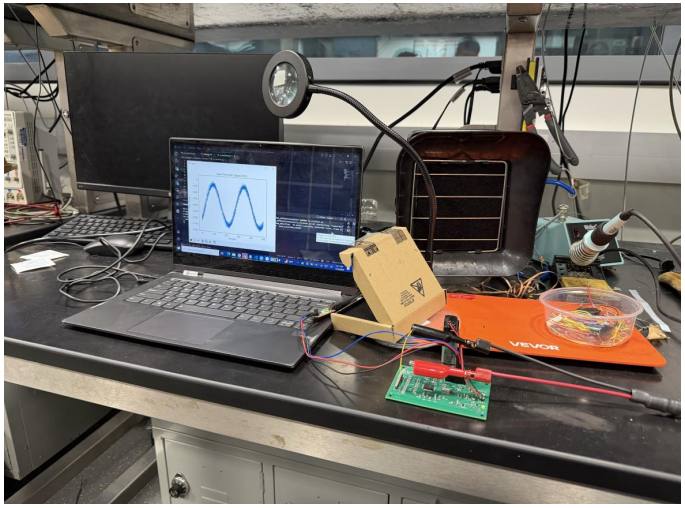
Table 3: Microcontroller Subsystem Requirements and Verifications

Requirements	Verifications
• Slow-wave sleep is detected correctly.	• Use sample EEG data to test if the algorithm is successful at identifying slow-wave sleep. If the algorithm identifies slow-wave sleep during the provided slow-wave sleep sample data and successfully identifies other sleep cycles or not sleeping, when non-slow-wave sleep sample data is provided, the algorithm is working as intended.
• Pink noise is played within 150 ms or less after the up-state of a slow-wave sleep wave is correctly identified.	• Use sample EEG data to test if the software correctly identifies when a slow-wave sleep up-state is occurring by making the program print a line indicating the up-state wave has been detected and look at the EEG sample data at this specific time to determine if an up-state is truly occurring. • After the up-state of the wave has correctly been identified, make sure the pink noise MP3 file is played within 150 ms of this detection.

Table 4: Computer Subsystem Requirements and Verifications

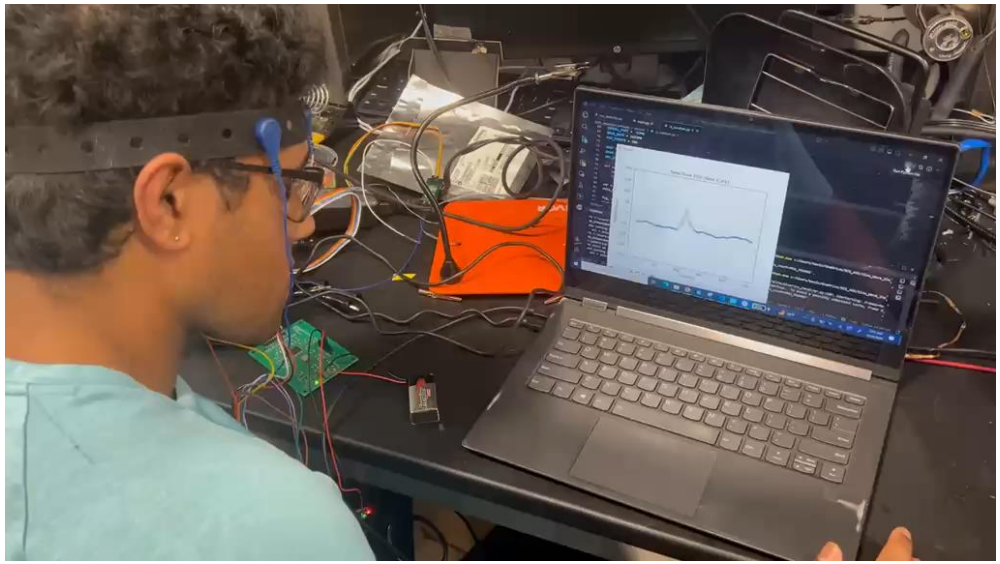
Requirements and Verifications

- **Most requirements and verifications were satisfied by demonstrating specific aspects of our project functioning**



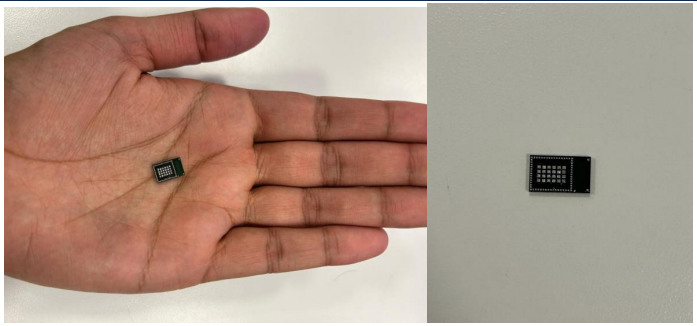
Successes

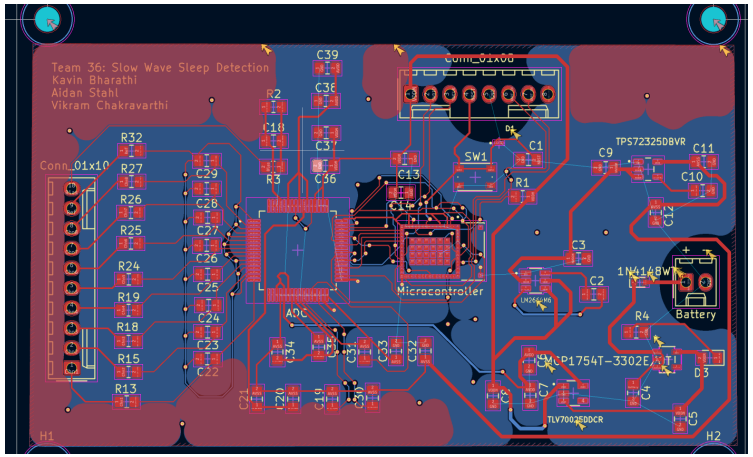
- Working PCB that could transmit EEG signals to computer interface
- Pink noise played correctly within specified time as shown in high level requirements

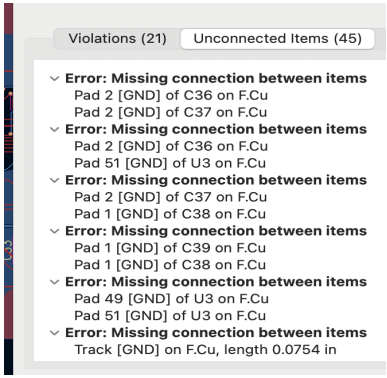


Challenges

- Soldering small pad spaces
- Grounding in PCB design
- STM32CubeIDE not user-friendly







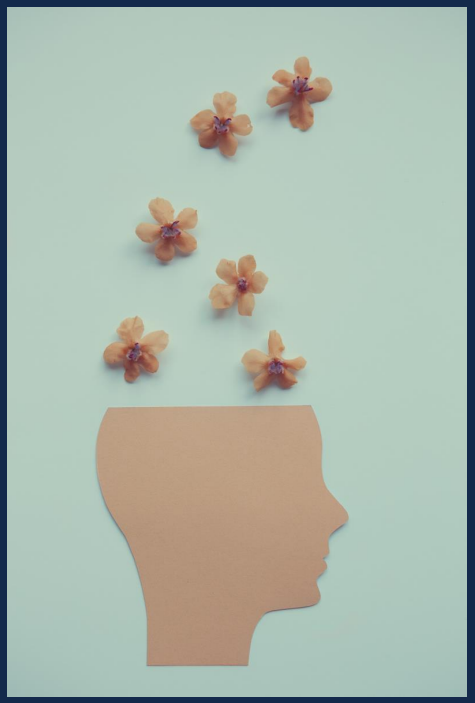
Conclusion

What did we learn?

- Sensitivity of measuring signals microvolts
- Paying attention to all documentation and datasheets
- Good soldering and PCB design practices
- Programming STM microcontroller and getting accustomed to IDE

What would we do diffferently?

- Use ESP32 instead of STM32
- Correct placement of antenna for BLE
- Understanding the solder space for soldering
- Using electrodes that are less sensitive and susceptible to noise



Recommendations for Further Work

Expanding Slow Wave Detection Software

- Analyzing multiple channels of data when deciding the time to play “pink noise”
- Minimize calibration time when estimating sleep stage and detecting slow waves
- Testing on actual sleeping patients

Addressing PCB Design Issues

- Correcting the placement of STM Microcontroller antenna
- Minimizing noise both analog and digitally

Thank You!

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN