Design Document

<u>EpiCap - A Wearable EEG</u>

September 30, 2021

ECE 445 - Fall 2021

Team 9

Shiru Shong (shirus2)

Qihang Zhao (qihangz2)

Casey Bryniarski (bryniar2)

TA: Melia Josephine

With help from Jennifer Cortes and Kenny Leung

1. Introduction

1.1 Problem Statement

Electroencephalograms (EEGs) are procedures that measure electrical activity at the very perimeter of the brain. Physicians use the results of these tests to diagnose and determine courses of treatment for abnormal brain behavior, such as epilepsy. A typical EEG test, however, presents difficulties to the patient and physician. Patients can be admitted to a hospital, occupying an inpatient bed, while sleep-deprived and off medication, in hopes that a seizure occurs and can be assessed; which leads to an increase in healthcare resource utilization [1]. Epilepsy or convulsion diagnoses have led to more than 1 million emergency department visits and 280,000 hospital admissions [2]. As the average hospital stay for epilepsy patients is 3.6 days, the accumulated hospital costs for epilepsy annually totaled approximately $2.5 billion [2]. Additionally, a survey showed that the average annual cost of epilepsy per person was $15,414 [3]; which includes outpatient, inpatient, ED, and treatment costs.

Ambulatory options, which are more cost-effective than outpatient/inpatient treatments, do exist. Patients are sent home with equipment and are tasked with wearing a very visible net of electrodes, head stocking that protects the electrodes and their leads, and the measurement is performed by a device that fits in a fanny pack. However, as many of these ambulatory types of equipment are bulky, not portable, or discreet, patients are forced to stay home and surrender important responsibilities that impact the patients' normal everyday activity such as work. People with a history of epilepsy were observed to have a lower annual income and a higher probability of employment [4].

1.2 Solution

If the present ambulatory technology was further miniaturized, we can create a device as discreet as a baseball cap that allows the wearers to carry on about their day while remaining monitored for EEG activity. Such a device would eliminate the bulky, inconvenient present ambulatory systems, and draw less attention in public. EEG patients would have the ability to wear it to work and would be able to record important EEG data if a seizure was to occur. Children who are suspected or diagnosed with the condition can wear something that does not interfere with their self-esteem or most daily activities.

We propose a discrete ambulatory EEG that can monitor patients, come off standby when an event is occurring, and measure brain activity while also utilizing a camera to further record muscle activity (or inactivity), essential data to arrive at a diagnosis and severity of an event. The benefit of having a seizure captured on video is the seamless synchronization to the patient's brain waves recorded from the EEG data. Moreover, the device will include on-board storage that will save EEG data and camera footage from a seizure event. We can then use the onboard SD cards to have the EEG data and camera footage can be viewed on a physician's PC.

1.3 Visual Aid

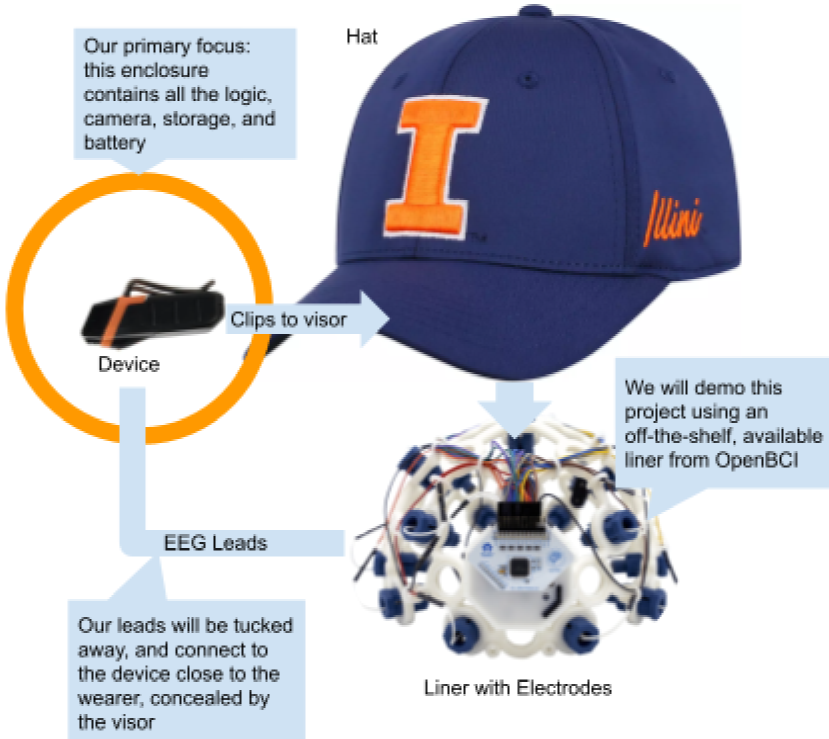


Figure 1. Pictorial Representation of the EpiCap

1.4 High-Level Requirements

- The EEG cap must be discreet and all the main devices components must be within the cap and cap visor (enclosure volume = 72 mm x 36 mm x 25 mm).
- Record EEG data at 240 +/- 5% Hz sampling rate for at least 24 hours and be able to store EEG data– electrical activity of the brain during a seizure on the flash storage.
- The EEG cap will track the patient’s eye and arm movement to shoulder height by using the wide-angle camera (minimum 240p) located in the cap visor.

2. Design

The EpiCap requires the following subsystems in order to operate successfully: power subsystem, microcontroller/processing subsystem, wireless interface, on-board flash storage, and a camera module as shown in Figure 2. The EpiCap would also require a software platform that allows certified physicians and the patient themselves to view EEG data reports. The power supply would provide the proper 3.3V to the board EpiCap to ensure that the system can be running for at least 24 hours. The microcontroller subsystem would be the central processing unit of the system and would be dealing with commands such as saving EEG traces and saving camera footage to on-board storage when detecting seizures. The wireless interface consists of a GSM chip and SIM card that would send emergency messages to the wearer's emergency contacts during a seizure. The on-board flash storage would store EEG data and eye/arm movement camera footage. Lastly, the camera subsystem consists of a wide-angle camera in order to track the patient's eye and arm movements. By implementing each of these design specifications, we can ensure that all of the high-level requirements would be satisfied.

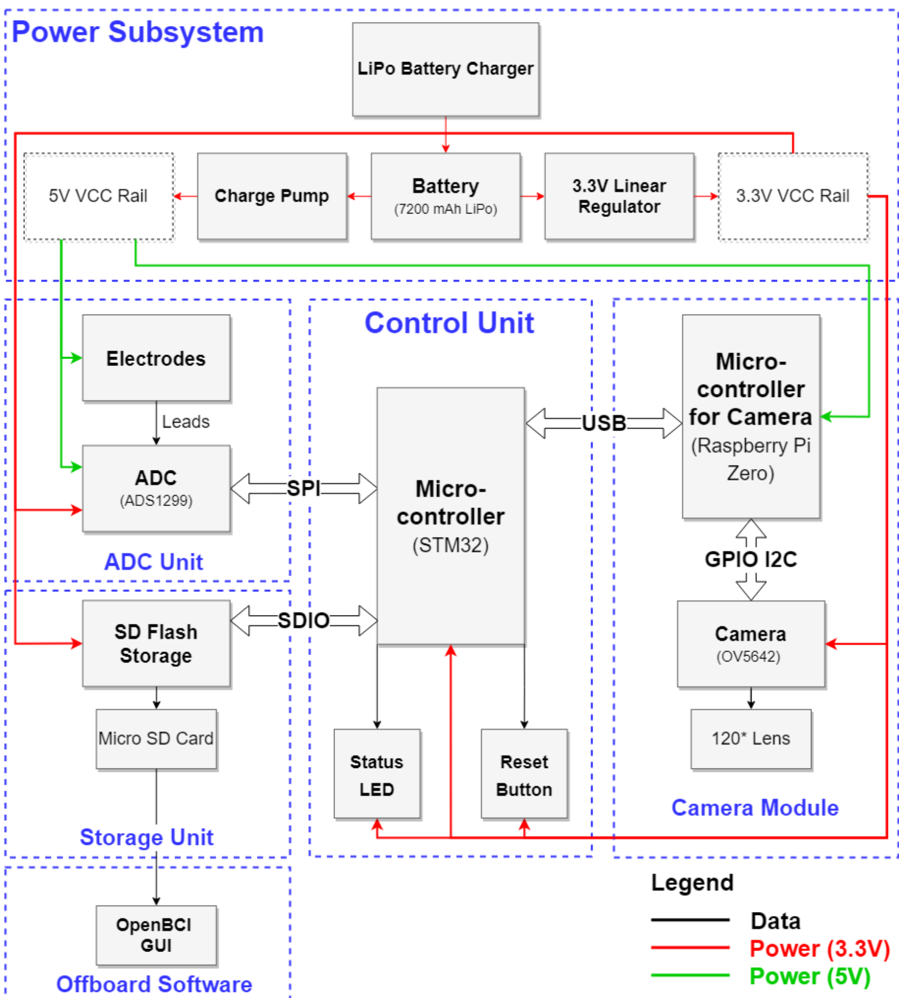


Figure 2. EpiCap High Level Block Diagram

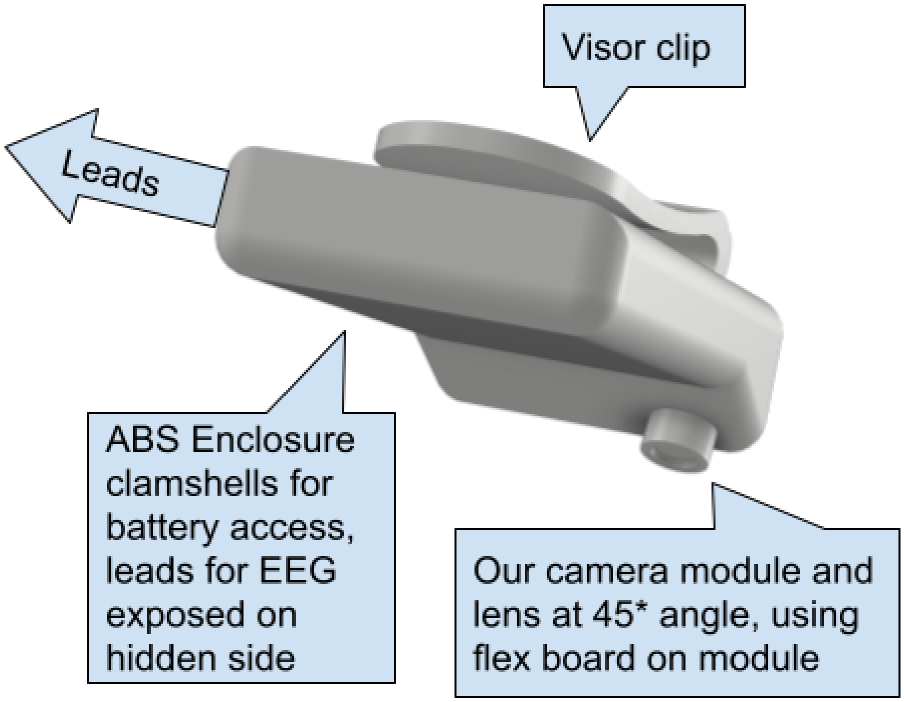


Figure 3. EpiCap Enclosure's Physical Design Mockup

2.1 Power Subsystem

The power subsystem provides a constant 3.3 V supply to the board when the EpiCap is in use. The rechargeable battery provides the only supply of power while the low drop voltage regulator ensures that the power delivered to the rail is at 3.3V. The 3.3V power rail supplies power to the camera, status LED, ADC, and the SD flash storage. This design also requires another charge pump that boosts the LiPo battery voltage to a constant 5V voltage for the 5V power rail. The components that require the 5V rail include the camera microcontroller, electrode leads, and ADC.

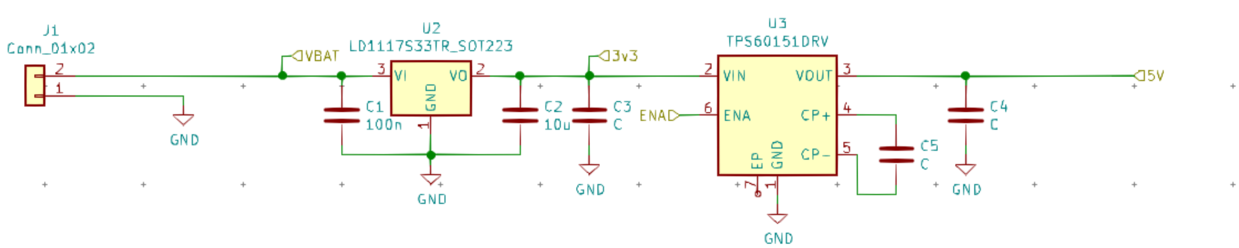


Figure 4. Schematic of Power Rail: Battery, Linear Regulator, and Charge Pump

2.1.1 Rechargeable 7200 mAh Lithium Polymer Battery

The board will be powered by a rechargeable LiPo battery. The battery will have enough to power all the components for at least 24 hours. As the microcontroller draws the maximum current (150 mA), the whole board requires an average power of 4800 mA. We chose the LiPo battery for its small size and low cost and it meets the power specifications for the board.

Table 1. Requirements and Verification Table for Rechargeable Lithium Polymer Battery

Requirements	**Verification**
1. Provide reasonably low noise (<µV).	1. Take oscilloscope measurements to measure the output voltage ripple signal is less than 1 µV.
2. Must be readily rechargeable, and not provide too much weight (<220g) or bulk to the whole device.	2. a. Discharge the Li-Po battery entirely to 3.2V and charge the battery to 4.2V in order to ensure the battery cell voltage drops between 3.5-3.7V. b. Weigh the battery using a weighing scale to ensure the battery is less than 220g.
3. During discharging at maximum current and voltage, the temperature of the Li-Po battery would be less than 27℃.	3. a. Charge the Li-Po battery entirely to 4.2V to ensure that the cell voltage is at its highest. b. Discharge the battery and use an IR thermometer to ensure that the battery does not discharge at a temperature greater than 27℃.
4. Must be able to power the board at full capacity for at least 24 hours.	4. a. Connect the fully charged LiPo battery to the board with all the devices connected. b. While connected to the board, discharge the LiPo battery at an operating voltage of 3.5-4.2V for 24 hours. c. Use a voltmeter to ensure that the

	battery cell voltage drop is greater than 3.2V (preferably between 3.5-3.7V).

2.1.2 Linear Voltage Regulator

The low-drop voltage regulator ensures that the power rail is being supplied with 3.3V voltage from the 3.5-3.7V LiPo battery. This regulator needs to be able to regulate the voltage ranging from 3.2-4.2V and at a peak current draw of 1A.

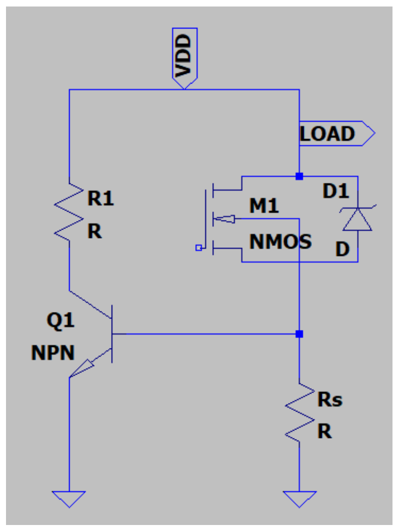


Figure 5. Constant Current Test Circuit

Table 2. Requirements and Verification Table for Linear Voltage Regulator

Requirements	**Verification**
1. Provide 3.3V+/- 5% from a 3.5-4.2V source.	1. Measure output voltage using an oscilloscope to ensure that the output voltage stays within 5% of 3.3V
2. Can operate at currents within 0-1A	2. a. Use a constant-current test circuit (figure 4) and connect the output of the voltage regulator to the VDD node. b. Adjust the Rs (potentiometer) value to deliver 1A to the load, measured by the multimeter.

	c. Use an oscilloscope to measure the output voltage to make sure it stays within 5% of 3.3V.
3. Maintain a temperature range of 20℃ to 27℃.	3. Use the IR thermometer to make sure the voltage regulator stays in the temperature range of 20℃ to 27℃.

2.1.3 Charge Pump with Integrated LDO

The charge pump with integrated LDO, chosen to be LM27762, would step up the 3.5-3.7V input voltage from the battery to supply the 5V power rail required by both the ADC (ADS1299) and the leads of the electrodes. The charge pump, similar to the voltage regulator, needs to be able to regulate the voltage ranging from 3.2-4.2V and at a peak current draw of 1A.

Table 3. Requirements and Verification Table for Charge Pump with LDO

Requirements	**Verification**
1. Step up to 5V+/- 5% from a 3.5-4.2V source.	1. Measure output voltage using an oscilloscope to ensure that the output voltage stays within 5% of 5V
2. Can operate at currents within 0-1A	2. a. Use a constant-current test circuit (figure 4) and connect the output of the voltage regulator to the VDD node. b. Adjust the Rs (potentiometer) value to deliver 1A to the load, measured by the multimeter. c. Use an oscilloscope to measure the output voltage to make sure it stays within 5% of 3.3V.
3. Maintain a temperature range of 20℃ to 27℃.	3. Use the IR thermometer to make sure the voltage regulator stays in the temperature range of 20℃ to 27℃.

2.2 Control Unit

The control subsystem, supplied by the power subsystem, communicates and controls other subsystems by managing the SPIs for the camera, flash storage, and the ADC (analog to digital) converter. The control subsystem, which consists of the STM32 microcontroller, will also communicate with the bus to allow EEG data coming from the ADC and camera footage to be saved on the on-chip flash storage and Raspberry Pi Zero's SD card simultaneously.

2.2.1 Microcontroller

Our microcontroller must internally be able to have enough memory overhead to save our software, implement a ring buffer or FIFO, and use that ring buffer and/or FIFO to cache data streaming from our EEG and our camera and save it to SDIO. For the purposes of EEG data, we are collecting 8 leads, 16 bit resolution at 240 Hz. 8 * 16 bits * 240 Hz = ~3.84 kBps. We predict that for a 240p 60fps video, we require 240 x 320 x (8 bit depth) x 3 colors, at 60fps our throughput is 230KB/frame, roughly 13.82 MB/sec. In order to prevent losing frames while our microcontroller does another task, we implement a video FIFO buffer, creating a stack of partial frames, potentially reducing the total amount of memory we have in use. Using DMA, we can move these frames off the buffer and into the SDIO/SD card flash storage present on the board. Due to present shortages, we could not acquire a microcontroller that had adequate memory overhead to accomplish the video task. We instead offload the video task temporarily to a Raspberry Pi Zero W and use it as an external peripheral to our MCU. Our STM32F205CT7 that we were able to procure allows us to implement the ADC portion adequately and communicate with the Raspberry Pi Zero W's I2C, SPI, USART lines. Our microcontroller also exposes plenty of internal ADC and GPIO pins so we may monitor our battery and various other parts of the system, our voltage rails, buttons and indicator LEDs.

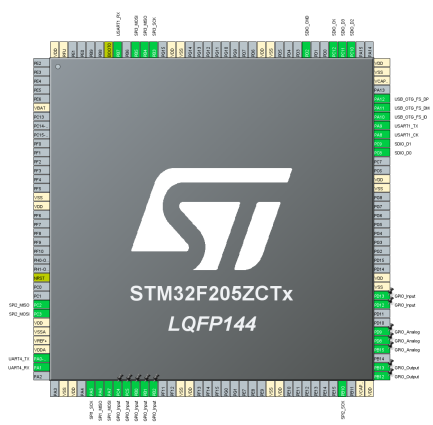


Figure 6. Our Microcontroller Pinout

Table 4. Requirements and Verification Table for Microcontroller

Requirements	**Verification**
1. Microcontroller must successfully boot under battery power	1. a. Connect 3.3V DC power supply to battery terminals present on board b. Using our programmer header, flash MCU firmware to board that constantly loops a "hello world" statement to our UART serial console. Determine that our print statement is being sent to serial console as intended. c. Disconnect our DC source and attach our charged battery to the MCU. Verify that our intended message and the one we receive is identical to what we had received earlier.
2. Microcontroller must be able to poll and double buffer ADC data	2. a. Connect 3.3V DC power supply to battery terminals present on board b. Flash firmware that captures our ADC output and stores it into a ring buffer, then sends voltage amounts to our serial interface c. Disconnect power supply, and use a voltage divider to create a 33uV constant source. Monitor the output of the constant source with a bench multimeter. d. Attach our battery to our board. Once communication is established, attach our constant source to the ADC pin being monitored. Make minute adjustments and verify that our serial data being sent matches what our bench multimeter is reading from our test supply, within 5% margin of error.

<table>
<tr><td>3. Microcontroller must be able to write to SD FAT32 filesystem</td><td>3.
a. Format a FAT microSD card with a text file containing a few lines of text.
b. Insert the microSD card into the carriage on the board. Connect 3.3V DC power supply to battery terminals present on board.
c. Flash firmware that overwrites our initial text file with the printf() ADC data we previously sent over console, while also displaying the data on console
d. Perform a similar test to our previous verification, and disconnect the supplies and battery and completely power down the board. Remove the SD card and verify that the data on the SD card matches the data in our serial console.</td></tr>
<tr><td>4. Microcontroller must be able to send a "begin recording" alert to our camera peripheral</td><td>4.
a. Create a microSD card formatted with a Raspbian distribution. Write firmware for our MCU that will loop over and send data to the UART serial terminal pins we share with the Raspberry Pi
b. Connect 3.3V DC power supply to battery terminals present on board, and flash our MCU with our test firmware.
c. Connect leads to expose Raspberry Pi's serial terminal and establish a connection with our test PC.
d. Verify Raspberry Pi has successfully booted via our serial terminal. Using a script, verify that the other serial tty present on the Pi is receiving our dummy MCU data being sent.</td></tr>
<tr><td>5. Microcontroller must be able to raise an alarm depending on ADC voltage detecting a seizure-like activity or a bad contact</td><td>5.
a. Flash firmware to the device that can interpret all of our ADC leads' data and send printf() statements to console</td></tr>
</table>

	regarding our seizure criteria or a bad contact b. Connect 3.3V DC power supply to battery terminals present on board, and flash our MCU with our test firmware. c. Send test data that determine edge cases of our criteria down our EEG leads. Determine that our intended detection behavior is present in our serial output.

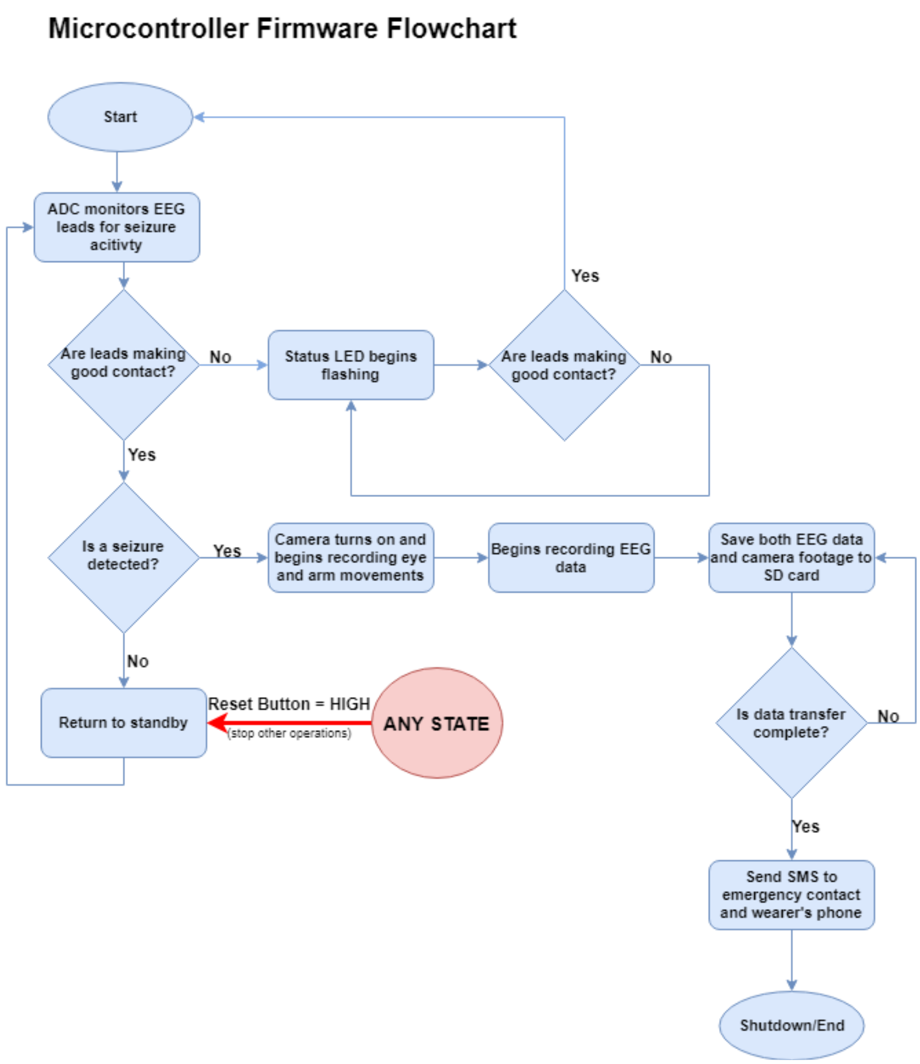


Figure 7. Microcontroller Firmware Flowchart

2.2.2 Status LEDs

Status LEDs, directly connected to the GPIO pins of the microcontroller, would be used to signal various states of the device. We will use one LED to alert wearers that not all the electrodes have valid contacts with the wearer's head. Another LED will represent what mode the device is in (standby vs recording/seizure detected). Our final LED will use an ADC pin from our microcontroller to monitor the battery voltage, and turn on when the battery is low. The microcontroller firmware will interpret these conditions using the ADC pins directly on the MCU or the data lines coming in from our sensitive EEG-grade ADC.

Table 5. Requirements and Verification Table for Electrodes Status LEDs

Requirements	**Verification**
1. Must be visible during bright sunlight (>500 lumens).	1. Use a light meter (also known as lux meter) that uses the light sensor to measure the illuminance of the LED. Verify that the LED emits > 500 lumens in a dark and light environment.
2. Must be able to withstand a maximum of 40 +/-5% mA drive current.	2. a. Use a constant-current test circuit (figure 3) and connect the output of the voltage regulator to the VDD node. b. Adjust the Rs (potentiometer) value to deliver 20 mA to the load, measured by the multimeter. c. Record the voltage drop across the LED at this drive current. d. Adjust the Rs (potentiometer) value to deliver 40 mA to the load, measured by the multimeter. e. Verify that the diode is still in working condition with a similar diode voltage drop compared to the voltage drop at 20 mA drive current (+/- 5%).

2.2.3 Reset Button

The reset button, when pressed, will allow the patient to force the microcontroller's state machine to go standby mode no matter what state the microcontroller is in. This reset button would stop all ongoing operations of other devices and go back to ADC monitoring EEG data.

Table 6. Requirements and Verification Table for Reset Button

Requirements	**Verification**
1. Must be easy to be pressed by the wearer of the EpiCap and actually resets the whole microcontroller state.	1. Button can be pressed without much force. When pressed, verify that the EpiCap has stopped all operations and has resumed to standby mode.

2.3 ADC (Analog to Digital) Unit

The ADC subsystem, which consists of the ADS1299 (8 channel), would convert the analog signal received from the electrode leads into quantifiable data such as the EEG data. As the electric signal is detected from the electrode leads, the received voltage level would be digitized into EEG data. The EEG data would then be read and processed by the microcontroller, which would be stored on the on-chip storage during a seizure.

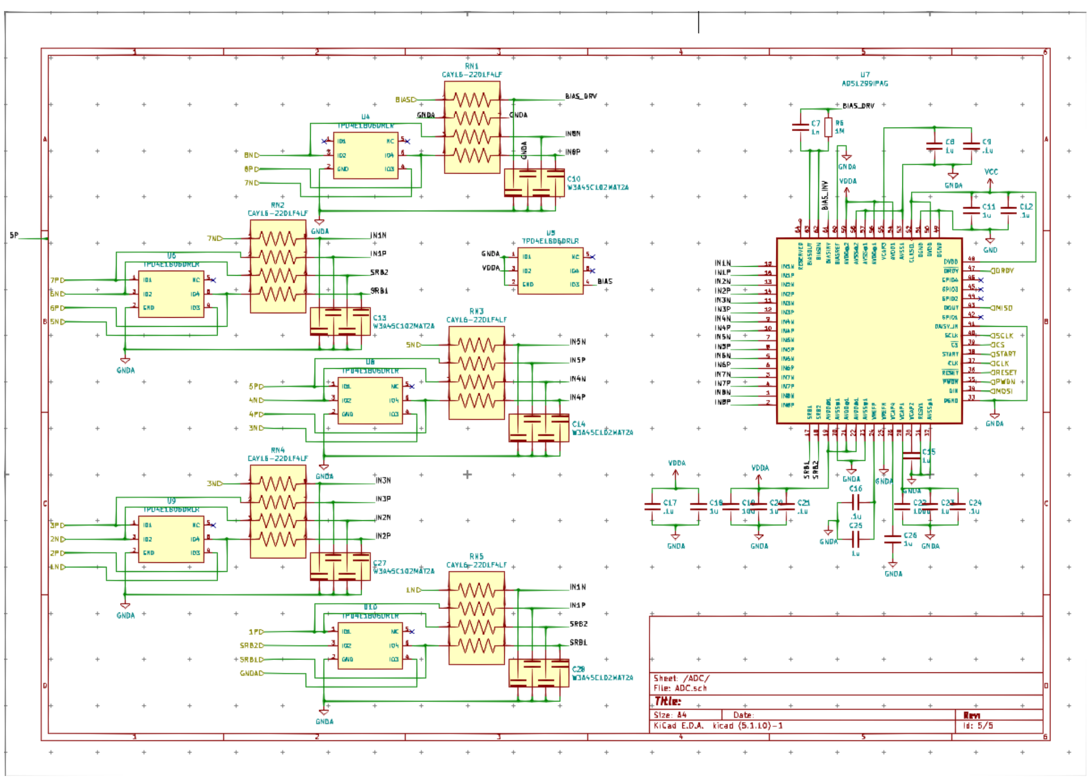


Figure 8. Schematic of ADC Unit

2.3.1 ADC (ADS1299)

The ADC converts the analog signal collected from the electrode leads into digitized EEG traces. The EEG data will be polled to determine if a seizure is occuring. When a seizure is detected, this EEG data is stored on our onboard SD storage.

Table 7. Requirements and Verification Table for ADC

Requirements	**Verification**
1. Must be sampling at a rate of 240 +/-5% Hz.	1. a. Flash firmware similar to our earlier MCU verification, that sends data from our ADC down a serial UART port. Connect the device to our 3.3V battery and determine that our device has booted and is talking on our serial console. b. Hook a lead pin from the ADC up to a function generator, with a voltage divider to step down the output from the generator. c. Sweep a very low voltage sine wave (uV scale after the divider) near 120Hz, with a span of 20 Hz in either direction and in discrete steps, each one at least 10 seconds. d. View the sweep data collected from the ADS recording, either through serial from the MCU or the SD Card. Determine the folding frequency and the sampling rate.
2. Must not pick up any external signals from other peripherals on the board.	2. a. Perform an EEG test with leads, but no head, in the hat, for a long duration, in a reasonably EM-free area. b. While recording data, poll every other peripheral in sequence, for a minute or so each, making sure there's another minute of dead time in between devices running.

	c. View the data. Identify and determine any concerning, spurious, emissions.
3. Provide reasonably low noise (<μV).	3. a. Provide a low-power analog signal to the ADC in the range of μV. b. Take oscilloscope measurements to measure the output voltage ripple signal is less than 1 μV.

2.3.2 Electrodes

Electrodes, directly attached to the cap and connected to the ADC, would be collecting electrical signals generated from brain activity. The electrical signals would then be viewed as voltage signals measured by the ADC.

Table 8. Requirements and Verification Table for Electrodes

Requirements	**Verification**
1. Must remain in contact with the scalp in the event of a seizure	1. a. Wear the EpiCap with all the electrodes in contact with the scalp. Confirm that all the electrodes do detect electrical signals initially, using a continuity check on a multimeter. b. Move around and re-enact big physical movements such as falling, jumping, walking, etc. c. Confirm that all the electrodes are still in contact with the scalp and collect electrical signals from the brain accurately.

2. Must be collecting accurate electrical signals (can have +/- 5% error).	2. a. Provide a low-power analog signal, generated from the function generator to the one electrode lead of the original OpenBCI Ultracortex Mark IV Cap. b. Use the OpenBCI GUI software to record the EEG data received by the Cyton 8-channel board. c. Repeat steps A and B with our own EpiCap board. Ensure that the EEG data collected from our board stays within the 5% error of the data collected from the OpenBCI Cap. d. Repeat steps A, B, and C with the 7 other electrodes used for the 8 channel electrode input.

2.4 Camera Module

The camera module, which consists of an OV5642 wide-angle camera, would be directly connected to the microcontroller through SPI and DCMI. The camera module would receive a signal from the microcontroller of when to be turned on and start recording a video. The camera module would then send the processed camera footage back to the microcontroller to be sent to the storage unit for local SD card storage. The camera module is expected to draw a maximum 200mA, depending on clock frequency and image transfer rate.

2.4.1 Wide-Angle Camera (OV5642)

The OV5642 camera will be set to have a maximum image transfer rate of 60 fps and VGA resolution (640x480) *or a resolution of 240p*. The OV5642 requires an analog input power of 3.3V and will have an input clock frequency of 24 +/- 5% MHz (as this is the DCMI frequency of our microcontroller). The camera will collect all the camera data while the microcontroller processes the data in order to store it onto the on-board storage.

Table 9. Requirements and Verification Table for OV5642 Camera

Requirements	**Verification**
1. Must have a minimum 60 fps and 240p resolution.	1. Write a device script to record and save a short video from the camera, saving it to

	onboard SD. Determine the fidelity of the video using software playback in VLC.
2. Be able to track both eye movements and arm movements to shoulder height from the camera footage (verify that it is mounted at a reasonable angle on cap visor).	2. a. Mount the camera on the cap visor at the desired angle. b. Use the microcontroller to turn on the camera and collect footage in order to process and save the final camera footage onto the SD card. c. Verify using the saved camera footage that both eye movements and arm moments to shoulder height can be observed from the camera footage. d. Repeat steps A, B, and C if the verification fails.
3. Must draw maximum drive current of 300 mA when recording footage at 60 fps and 240p.	3. a. Connect the appropriate nodes to VDD in order to turn on the camera. b. Start recording with the camera at 60 fps and 240p. c. Use a multimeter to measure the drive current on all input and output pins. d. Ensure that the drive current must be less than 300 mA.
4. Camera size is less than 40x40 mm and weight less than 30g.	4. a. Measure the length and width of the camera chip to ensure that it is less than 40x40 mm. b. Weigh the camera chip to ensure that it is less than 30g.

2.4.2 Microcontroller for Camera (Raspberry Pi Zero)

The microcontroller for the camera, which is chosen to be Raspberry Pi Zero, has a built-in camera interface that would allow the STM32 to communicate with the camera. As the Raspberry Pi Zero includes an on-board SD card, the camera footage would be stored locally on the Pi's SD card. *(Please note that the reason we have chosen to use the Raspberry Pi Zero as a camera microcontroller and storage for camera footage is that the STM32 that we have ordered* *does not contain enough RAM for video capture. The STM32 that would be powerful enough to capture video is currently out of stock due to the chip storage.)* The Raspberry Pi Zero would be directly connected to the STM32 through USB and connected to the camera with the GPIO IC2 lane. The maximum current draw of the Raspberry Pi Zero should be 300 mA.

Table 10. Requirements and Verification Table for Camera Microcontroller

Requirements	**Verification**
1. Maximum current draw of Raspberry Pi Zero when receiving data from the camera must be less than 300 mA.	1. a. Attach the input GPIO leads to the multimeter b. Load the firmware that allows the Raspberry Pi Zero to receive data from the camera module. c. Ensure that the current through the load during data transfer must be less than 300 mA when powered by the 5V input.

2.5 Storage Unit

The storage unit, directly connected to the microcontroller through the SDIO interface bus, includes an SD card. This storage unit would be the onboard storage used to store EEG data during seizure events. The EEG data saved on the SD card can be viewed and analyzed using the OpenBCI GUI software.

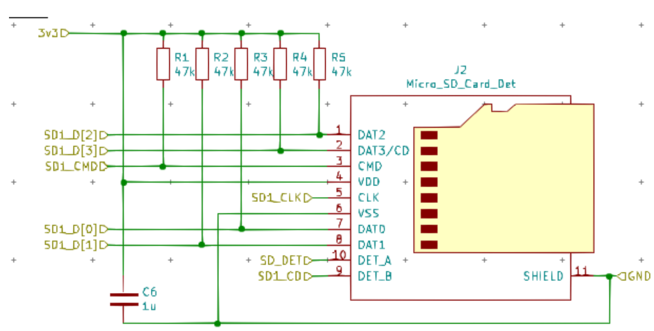


Figure 9. Schematic of Storage Unit

2.5.1 Micro SD Flash Card

The Micro SD flash card, connected to the board by the Hirose Card Chassis, would be directly written into by the microcontroller. The micro SD card would be saving the EEG data during a seizure. When writing to the SD card, the current draw can go up to 100 mA. The SD card would require a 3.3V power input. The SD card would have a FAT32 filesystem as we will need more than 2GB of storage.

Table 11. Requirements and Verification Table for Micro SD Flash Card

Requirements	**Verification**
1. The data transferred to the micro SD card must be valid and ensure that none of the radiators can corrupt the data once saved.	1. a. Write a simple test code to the microcontroller that allows the microcontroller to write a dummy .txt file to the micro SD card. b. Verify that the file stored on the micro SD card is the exact same as what was written in the .txt file when viewed on another device.
2. Require a FAT32 filesystem format.	2. a. Verify from the component's datasheet that the SD card is FAT32 filesystem formatted. b. If not, we would need to format the card before use following the datasheet's instructions.

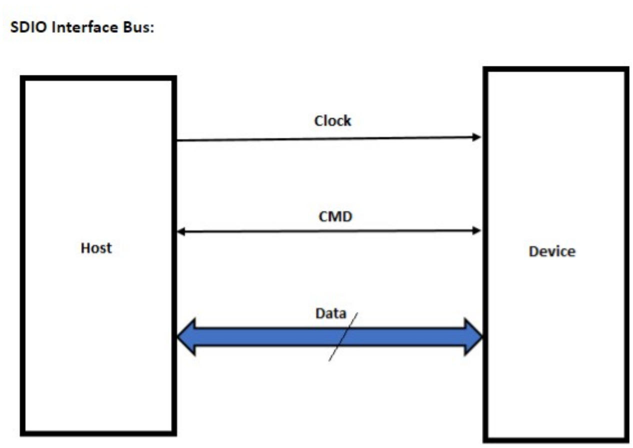


Figure 10. SDIO Interface Between Microcontroller and Micro SD Card

3. Costs

Table 12. Bill of Materials

Bill of Materials		
Component	**Qty**	**Price**
Raspberry Pi Zero W	1	$5
B006603 ArduCam Pi Zero OV5642 Camera Module	1	$17.99
STM32F205ZCTx MCU	1	$13.73
LD1117S33TR_SOT223 Linear Regulator	1	$0.55000
TPS60151DRV Charge Pump ->5V	1	$1.94
TPD4E1B06DRLR TVS Isolation Diode	6	$0.65
ADS1299IPAG 8 Channel ADC	1	$70.88000
DM3BT-DSF-PEJS microSD Card Carriage	1	$2.93
SDSDQM-032G-B35 microSDHC card, 32G	2	$7.50
Single-cell LiPo battery, 5000 mAh	1	$8.50
1568-PRT-15217-ND LiPo Battery Charger	1	$10.50
Total	**XX**	**$140.17**

3.1 Labor

We predict each team member will have roughly 80 hours of work to complete for the duration of the semester, about 12 hours per week. We will charge $50/hour for each member's contribution. Additionally, we require an additional $125/hour for our company's upkeep. Our total, then, is 80 * 50 * 3 + 125 * 80 = $22000, for our labor costs.

4. Schedule

Table 13. Project Schedule and Task Allocation

Week	**Shiru Shong**	**Casey Bryniarski**	**Qihang Zhao**
9/27	Complete Design Document. (8 hours)	Complete Schematic for EpiCap. (8 hours)	Complete Design Document.
10/4	Complete Design Review. Get the PCB board approved and order first around. Buy all the necessary components for the project. Test OpenBCI EpiCap.	Complete Design Review. Finalize & Review PCB. Get the PCB board approved and order the first round. Test OpenBCI EpiCap. (5 hours)	Complete Design Review. Finalize & Review PCB. Buy all the necessary components for the project. Test with OpenBCI GUI.
10/11	Complete unit-testing for each component. Complete PCB Board by soldering and mounting components. Start testing on-board hardware.	Complete unit-testing for each component. Complete PCB Board by soldering and mounting components. Start testing on-board hardware. (8 hours)	Work on the firmware of the microcontroller with other subsystems such as wireless module, camera module, ADC unit, and storage unit.
10/18	Test on-board hardware system, identify and fix issues. Start Version 2 PCB design.	Test on-board hardware system, identify and fix issues. Start Version 2 PCB design. (16 hours)	Finish the firmware code for the microcontroller.
10/25	Finish on-board hardware testing. Execute the whole EpiCap testing, including firmware. Order Version 2 PCB board design.	Finish on-board hardware testing. Execute the whole EpiCap testing, including firmware. Order Version 2 PCB board design. (16 hours)	Test firmware code with the entire hardware component. Start Version 2 firmware coding.
11/1	Complete the Version 2 PCB board and perform the final debugging process.	Complete the Version 2 PCB board and perform the final debugging process. (8 hours)	Finish Version 2 firmware coding and continue the firmware debugging process.
11/8	Perform final complete system testing. Prepare for Mock Demo and demonstration.	Perform final complete system testing. Prepare for Mock Demo and demonstration. (8 hours)	Perform final complete system testing. Prepare for Mock Demo and demonstration.

11/15	Mock demo.	Mock demo.	Mock demo.
11/22	Fall break.	Fall break.	Fall break.
11/29	Demonstration. Work on presentation. Start the final paper.	Demonstration. Work on presentation. Start the final paper.	Demonstration. Work on presentation. Start the final paper.
12/6	Final presentation and final paper.	Final presentation and final paper.	Final presentation and final paper.

5. Tolerance Analysis

One important consideration is that we must ensure that dry electrodes can collect EEG data properly in the event of a seizure.

In order to illustrate this, we can compare the normal EEG signal to the seizure EEG signal. The following data were collected from seven different recordings. Non-seizure data was recorded 5 minutes before seizure data. The seizure segment is a 5s sliding window with a 2.5s interval. All of them show variable patterns.

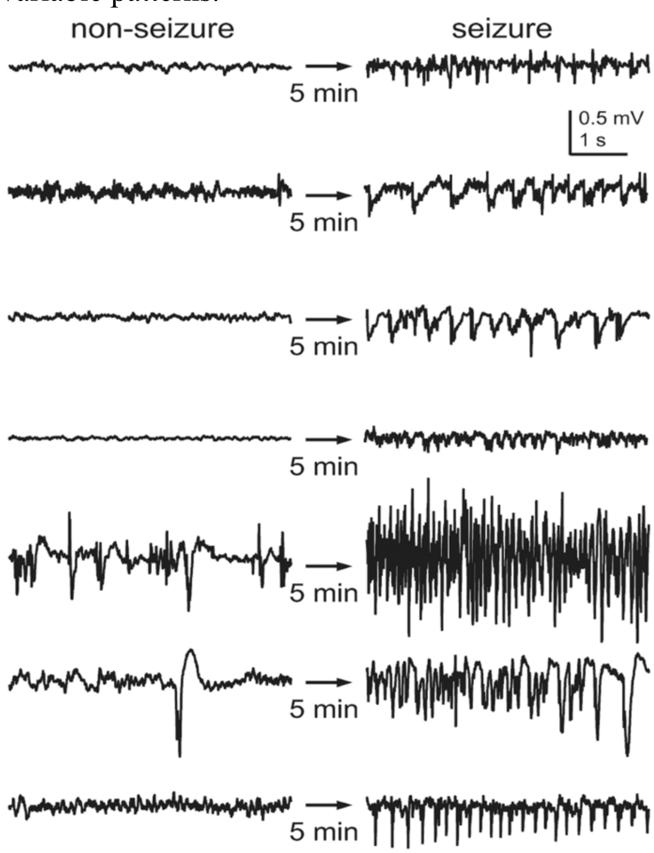


Figure 11. EEG change during 5 minutes from non-seizure to seizure [5]

As we can see clearly from the figure above, the seizure signal changes more severely than the non-seizure signal does. Since we need a criterion to determine the seizure, we chose to regard the current EEG as the seizure signal by considering the signal which has a 1.2 times higher mean absolute amplitude in frequency compared with the nearby EEG data [5]. If there was no large amplitude change in frequency, the data was not regarded as a seizure signal.

Following is the actual logic:

1. Calculate the mean
2. Calculate how far away each data point is from the mean using positive distances.
3. Add those deviations together
4. Divide the sum by the number of data points

Therefore, the equation for this mean absolute deviation is

$$MAD = \frac{\sum|xi - \bar{x}|}{n}$$

For example, we can set a 5s time interval to calculate its MAD. We should evenly extract 100 points from this set. After we get the frequency of each point, we can calculate the mean of the sum of these frequencies.

$$Mean(\bar{x}) = \frac{\sum x}{n}$$

The next step is to calculate the distance between each data point and the calculated mean $\bar{x}$.

$$\text{Distance} = |xi - \bar{x}|$$

The third step is to add these deviations together.

$$\text{Sum of distance} = \sum|xi - \bar{x}|$$

The fourth step is to divide the sum by the number of data points, and there should be 100 data points.

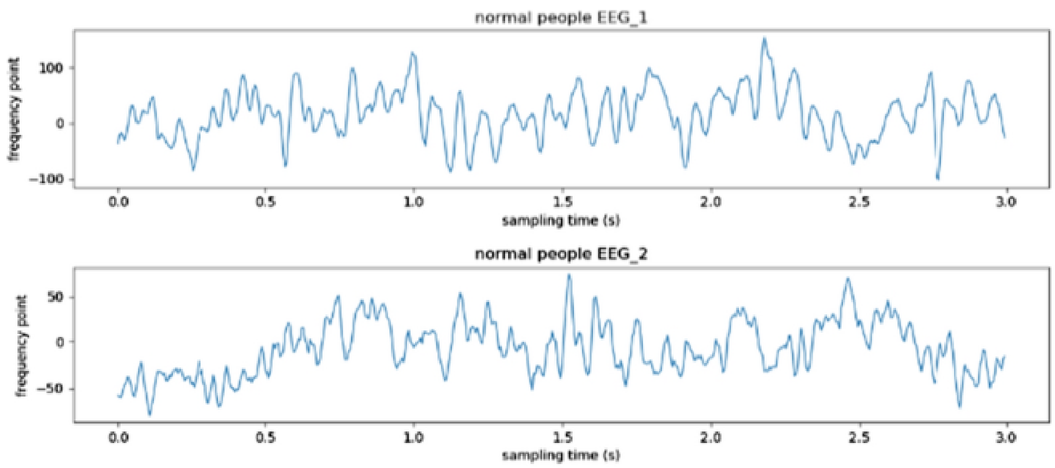
$$\text{Mean absolute deviation} = \frac{\sum|xi - \bar{x}|}{n}$$

The last step is to compare this Mean absolute deviation with the normal EEG data, which is the patient's daily normal EEG data. If the mean absolute deviation is 1.2 times higher than the normal EEG value, it means that the patient is in the state of seizure.

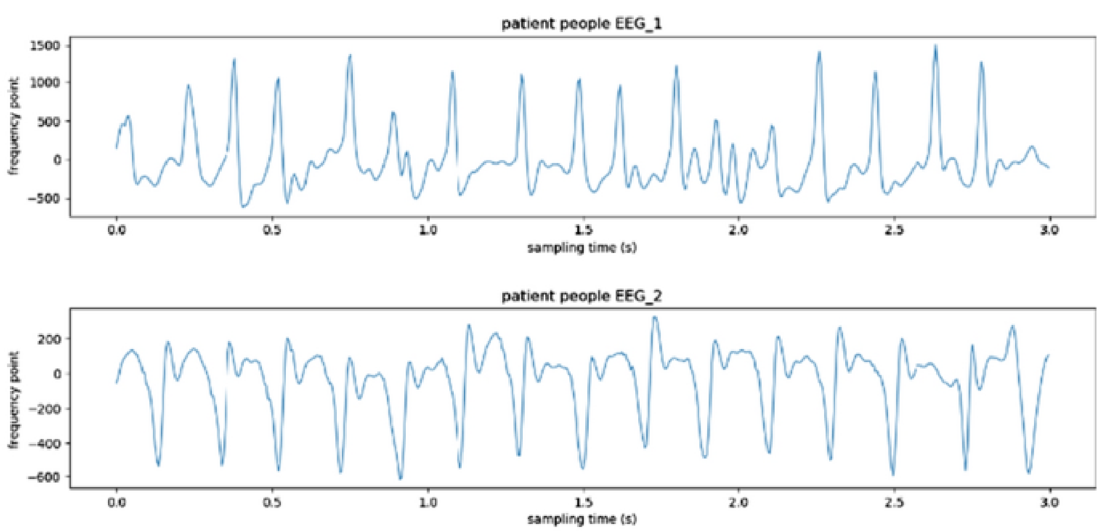
- If $MAD \simeq normal\ EEG\ value$:
  The patient is not in a state of seizure.
- If MAD > 1.2*(normal EEG value):
  The patient is in a state of seizure.

We will use our STM32 microcontroller to set a normal EEG value as a criterion according to the daily normal EEG data of the patient, and then set an upper bound of the daily normal EEG data, which should be less than 1.2 times of the normal value. The algorithm of our STM32 microcontroller includes such work. Once our microcontroller receives the voltage from the ADC unit correctly and it can properly decide whether the patient is in the state of seizure or not according to our algorithm of STM32, the microcontroller can send the result signal to the console.

In order to demonstrate more clearly why we use the so-called normal EEG data instead of a specific frequency range, below is the illustration plot of the frequency of the normal EEG and seizure EEG. Note that the EEG frequency is varied from one person to another person, which is why we want to get the normal EEG data of each patient. For example, the first person's normal EEG range is from -100 Hz to 100 Hz, while the second person's normal EEG range is from -50 Hz to 50 Hz. As we can see it is also common for large different scales of seizure for different patients, which is from -500 Hz to 1500 Hz for the first patient, and -600 Hz to 200 Hz for the second patient. Therefore, it is better for us to first collect the daily normal EEG data of the patient. After we set this value as an upper bound of normal EEG value, if the calculated MAD is 1.2 times higher than this upper bound, then we regard this patient as in the state of seizure.



(a)



(b)

Figure 8. EEG data from healthy persons and patients [6]

One of the other mission critical areas is the connection between our electrodes and the ADC. We design an RC circuit to perform a low-pass filter on all of the signals coming from our head. Our resistors are 2.2k with about 1% tolerance, while our capacitors are X7R with a 1000pF +- 20% tolerance. We drew up the RC isolation circuit in LTSpice:

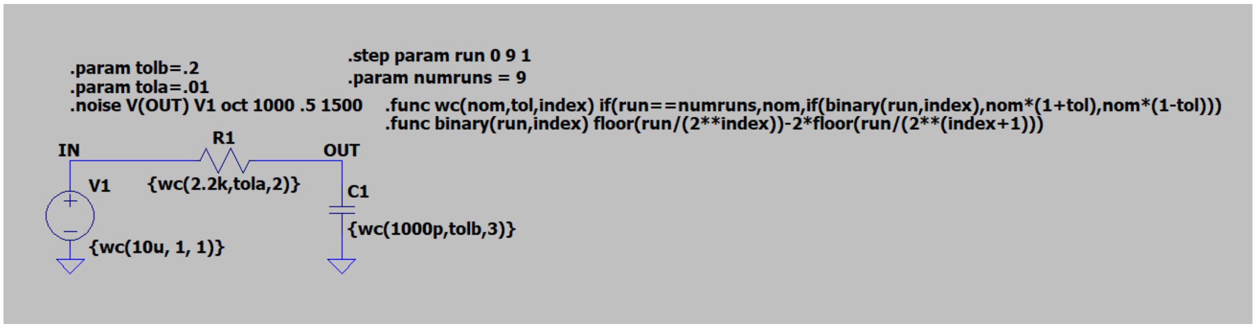


Figure 12. Our LTSpice tolerance analysis model for our EEG leads.

This circuit performed a noise analysis to determine if at all extents of our tolerances we still don't see any premature attenuation. Our resultant graph looks like this:

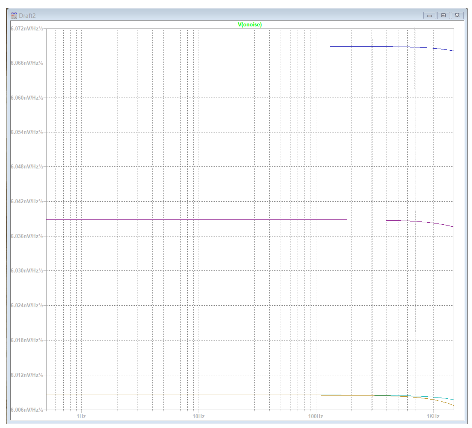


Figure 13. Our model output. Note the attenuation bends at 240Hz.

Through our noise analysis, we have demonstrated that even in our worst cases of part tolerance we will never clamp signals before our critical 240Hz mark. When the RC constant is at a minimum (and the components are at their minimum extent, attenuation still doesn't begin before 240Hz. We can say, in confidence, that our components will meet our design specifications for the ADC, provided we are supplied with components that have the tolerances we selected.

7. Ethics and Safety

There could be several concerns regarding our project. The cap confronts several risks and vulnerabilities as a result of the use of rechargeable lithium-ion batteries, electrodes, cables, chips, flash storage, camera, and the possibility of a patient's unexpected fall.

First, there are electrodes that remain in contact with the scalp in the event of a seizure. Even though this is a special hat that collects medical EEG information, we still need to make sure patients do not feel any difference or discomfort wearing the hat compared to a regular hat. As stated in the subsystem requirements, the electrodes must be readily adapted to different ball caps and different hat sizes. We must also take precautions that our device does not create more danger for a patient in the event of a fall. In order to solve these issues, we will adjust the gap between the cap and the patient's head by using one of the most common ways of fixing an ordinary cap: an elastic strap, which can better keep the cap on the patient's head steadily and capture important EEG information in the scenario that the patient falls. We will also line up our wires and chips so that they are distributed around the edge of the hat so that the patient will not be injured by these parts if they fall.

The second concern is that EpiCap is a wearable gear, and patients will need to wear it for long periods to get complete EEG data. If any materials are mixed with chemicals that are harmful to the human body, it is potentially dangerous for the patient and may lead to an erroneous diagnosis from doctors. In order to solve this problem, we need to ensure that both the battery and board present no hazard to the patient - especially in the event of high heat, moisture, and any sort of mechanical shock where according to the IEEE Code of ethics I.1: we must "hold paramount the safety, health, and welfare of the public…" [7].

The third concern is that the EpiCap will collect highly confidential EEG data of the patient. We must ensure that patient information can only be accessed by the doctor and is kept confidential to visitors, as according to the IEEE Code of ethics I.1 we must "hold paramount the safety, health, and welfare of the public…"[7]. In order to solve this issue, we can design a password system for doctors and encrypt all patient information. For example, each doctor will have separate accounts and passwords. When the doctor receives data from the GSM chip, they will give their patient an identification number. As a result, only the corresponding physician has the patient's EEG data, which cannot be viewed by the outside viewer.

The fourth concern is that the EpiCap will not have the danger of electrocution from the electrodes. We must ensure that no patients or doctors get hurt according to the IEEE Code of ethics I.1 we must "hold paramount the safety, health, and welfare of the public…"[7]. In order to solve this issue, we can attach a layer of insulation material to the inside of the hat, to avoid direct contact between any electrodes to the skin of patients.

We will rigorously test our design to ensure that the final product is safe for patients and doctors alike. We intend to comply with the IEEE code and the corresponding safety or regulator standards such as OSHA or FCC. Additionally, we will seek and accept any improvements regarding our project and due to the nature of our design, we will need to work with the medical school, so that we will appropriately credit others' efforts, according to IEEE code of ethics I.5: "to seek, accept, and offer honest criticism of technical work…" [7].

8. References

[1] S.-Y. Chen, N. Wu, L. Boulanger, and P. Sacco, “Antiepileptic drug treatment patterns and economic burden of commercially-insured patients with refractory epilepsy with partial onset seizures in the United States,” *Journal of Medical Economics*, vol. 16, no. 2, pp. 240–248, 2012. [Accessed September 25, 2021].

[2] Healthcare Cost and Utilization Project, 2014 National Data. [Online]. Available: hcupnet.ahrq.gov/. [Accessed September 25, 2021].

[3] J. A. Cramer, Z. J. Wang, E. Chang, A. Powers, R. Copher, D. Cherepanov, and M. S. Broder, “Healthcare utilization and costs in adults with stable and uncontrolled epilepsy,” *Epilepsy Behav.*, vol. 31, pp. 356–362, 2014. [Accessed September 25, 2021].

[4] Kobau R, Zahran H, Thurman DJ, et al. Epilepsy surveillance among adults—19 states, Behavioral Risk Factor Surveillance System, 2005. *MMWR Surveill Summ.* 2008;57(6):1-20. [Accessed September 25, 2021].

[5] K.-O. Cho and H.-J. Jang, “Comparison of different input modalities and network structures for deep learning-based seizure detection,” *Nature News*, 10-Jan-2020. [Online]. Available: https://www.nature.com/articles/s41598-019-56958-y. [Accessed: 28-Sep-2021].

[6] I. Aliyu and C. G. Lim, “Selection of optimal wavelet features for epileptic EEG Signal Classification with LSTM,” *Neural Computing and Applications*, 09-Jan-2021. [Online]. Available: https://link.springer.com/article/10.1007/s00521-020-05666-0. [Accessed: 03-Oct-2021].

[7] IEEE, “IEEE code of ethics,”. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 28-Sep-2021].