UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Team 9: EpiCap

ECE 445

Shiru Shong, Casey Bryniarski, Qihang Zhao

12/3/2021

Introduction

Problem

- Expensive hospital bills for epilepsy patients
  - $2.5 billion annually
- Epilepsy patients: lower annual income and higher unemployment
- Sleep deprived

Introduction

Current Outpatient Solution

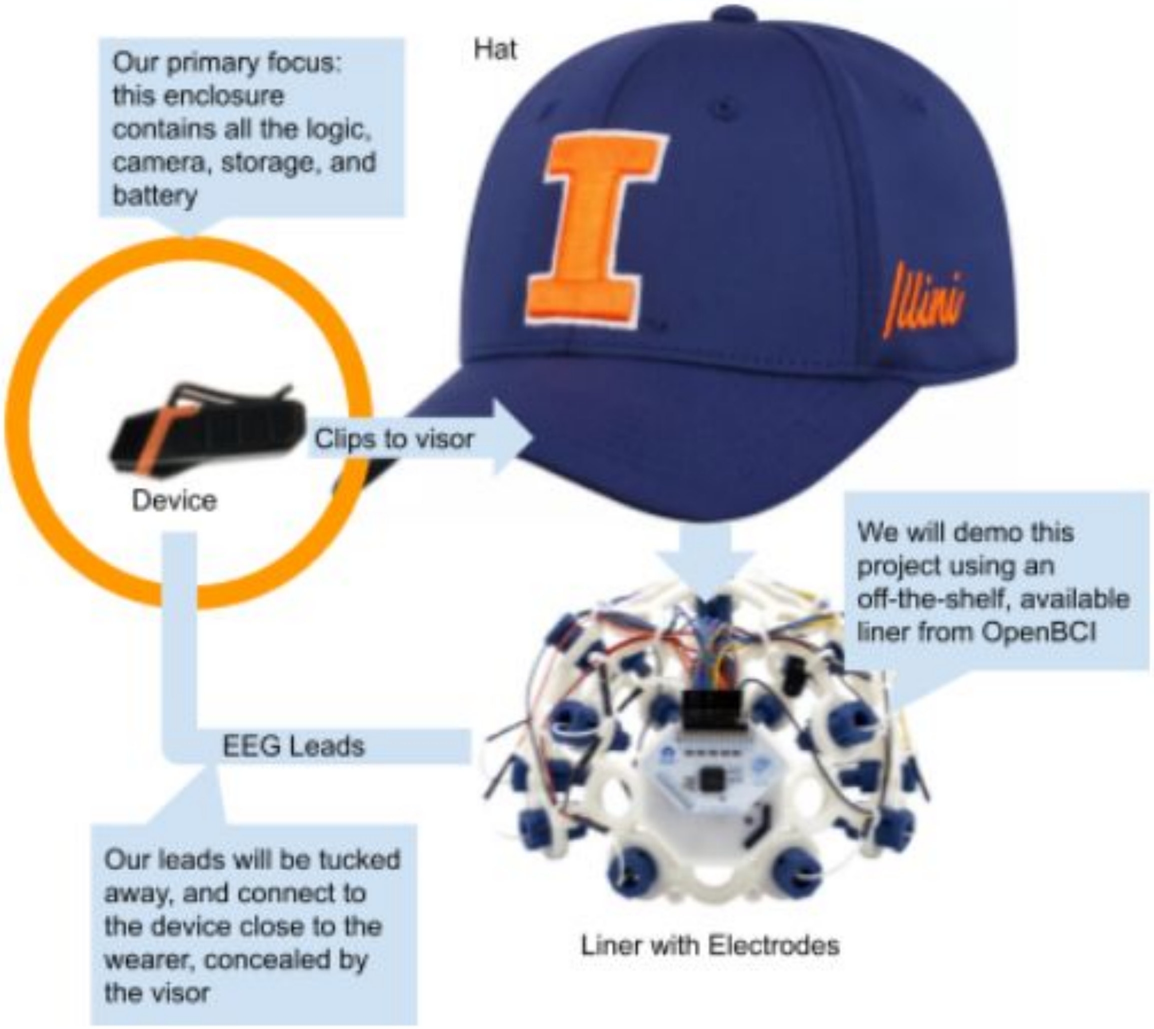
- Bulky ambulatory equipments
  - Surrender important responsibilities
  - Affect social activities

- No video recording

Introduction

Solution

- Continue daily activities
- Not bulky
- Has video recording functions



Our primary focus: this enclosure contains all the logic, camera, storage, and battery

Hat

Device

Clips to visor

EEG Leads

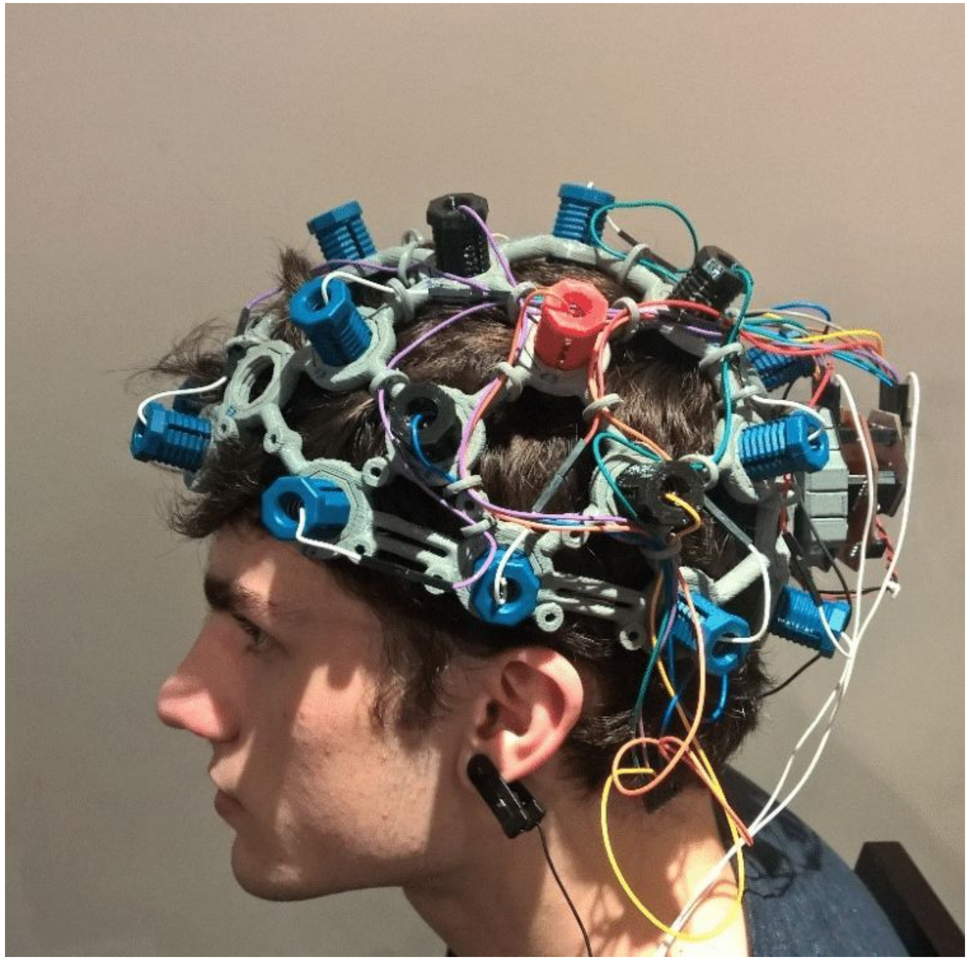
We will demo this project using an off-the-shelf, available liner from OpenBCI

Our leads will be tucked away, and connect to the device close to the wearer, concealed by the visor

Liner with Electrodes

Introduction

Solution



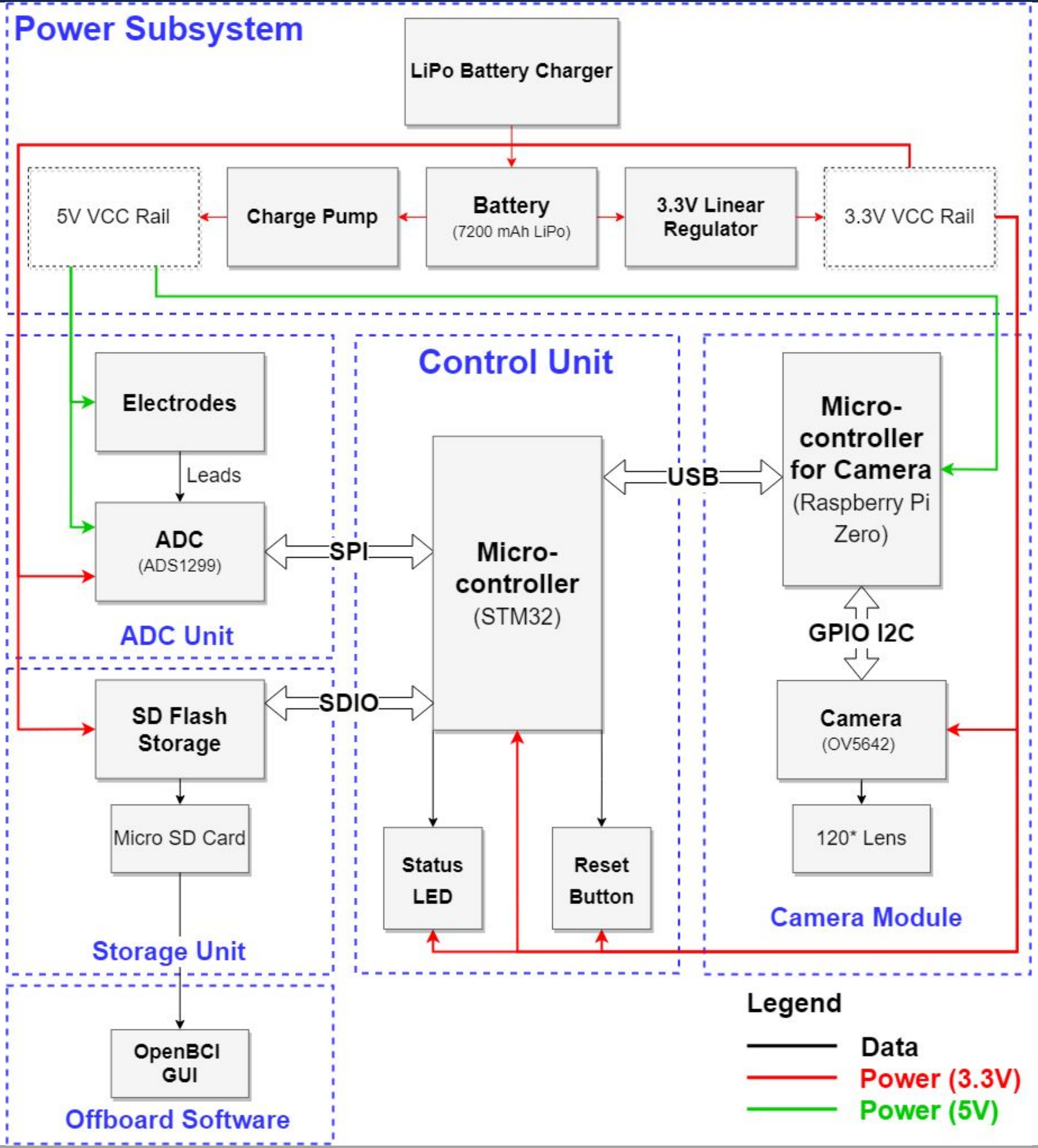
Objectives

High Level Requirements

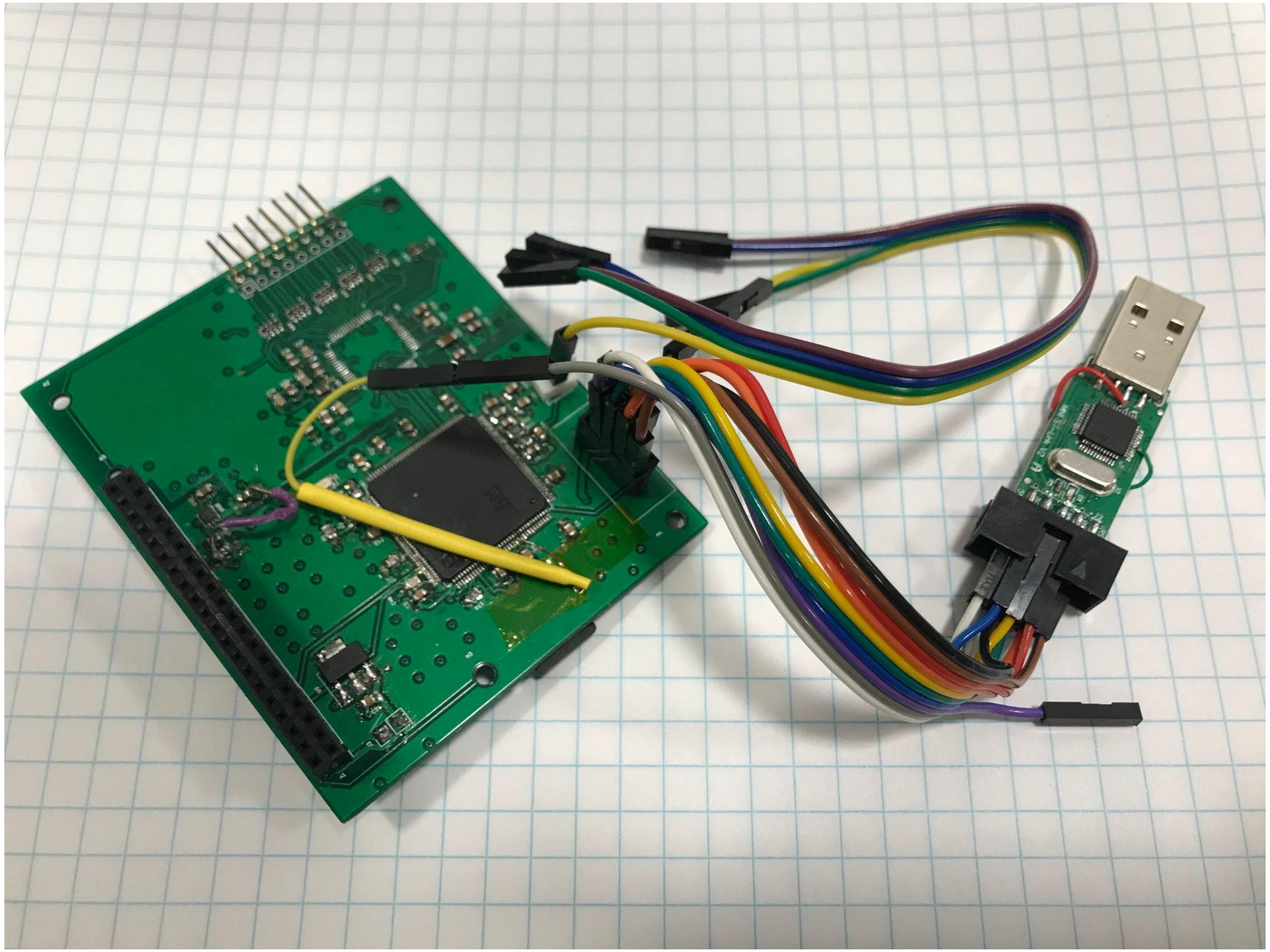
- The EEG cap must be discreet and all the main devices components must be within the cap and cap visor (enclosure volume = 72 mm x 36 mm x 25 mm).
- Record EEG data at 240 +/- 5% Hz sampling rate for at least 24 hours and be able to store EEG data– electrical activity of the brain during a seizure on 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.

Design

Block Diagram #1

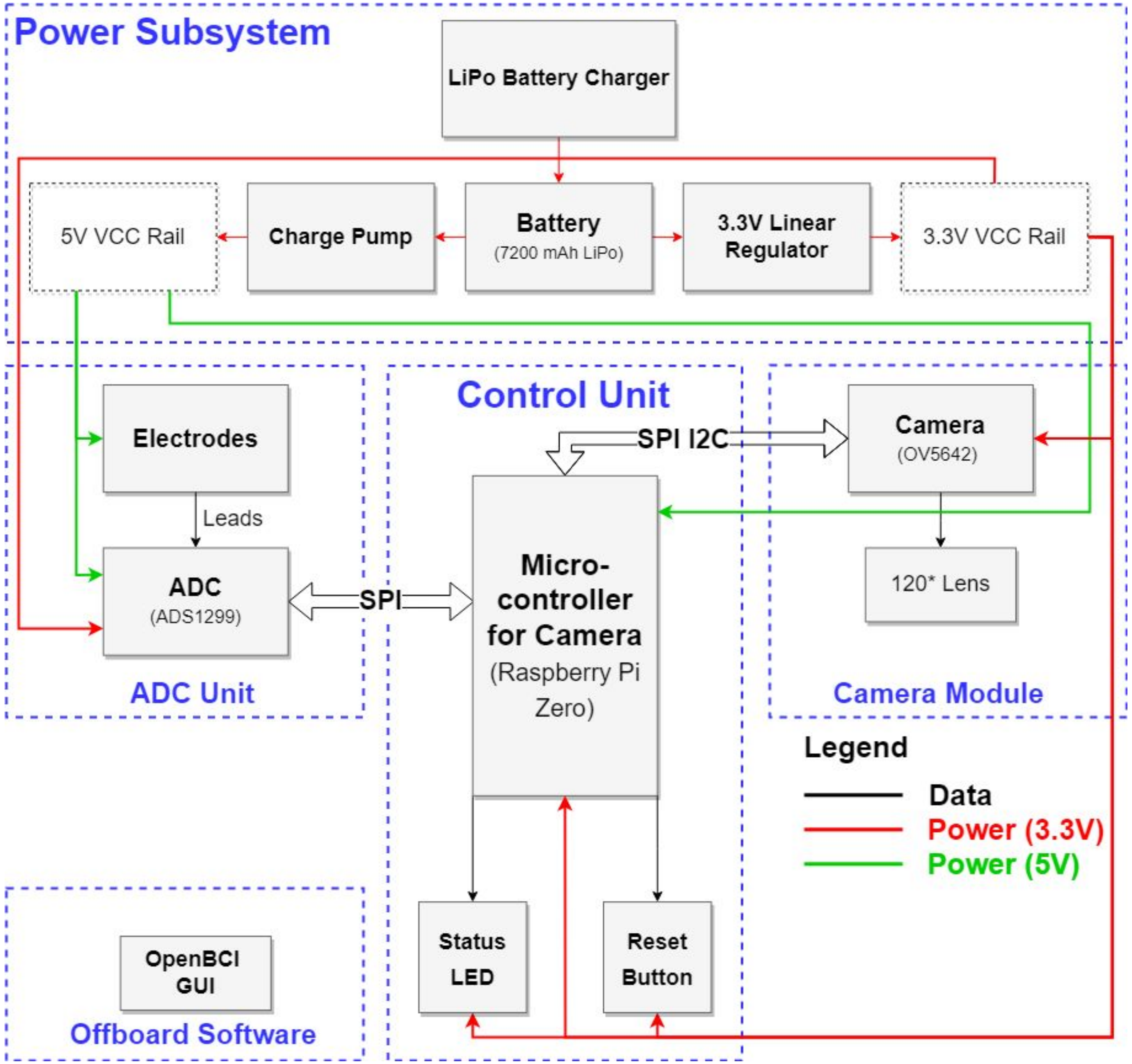


Design

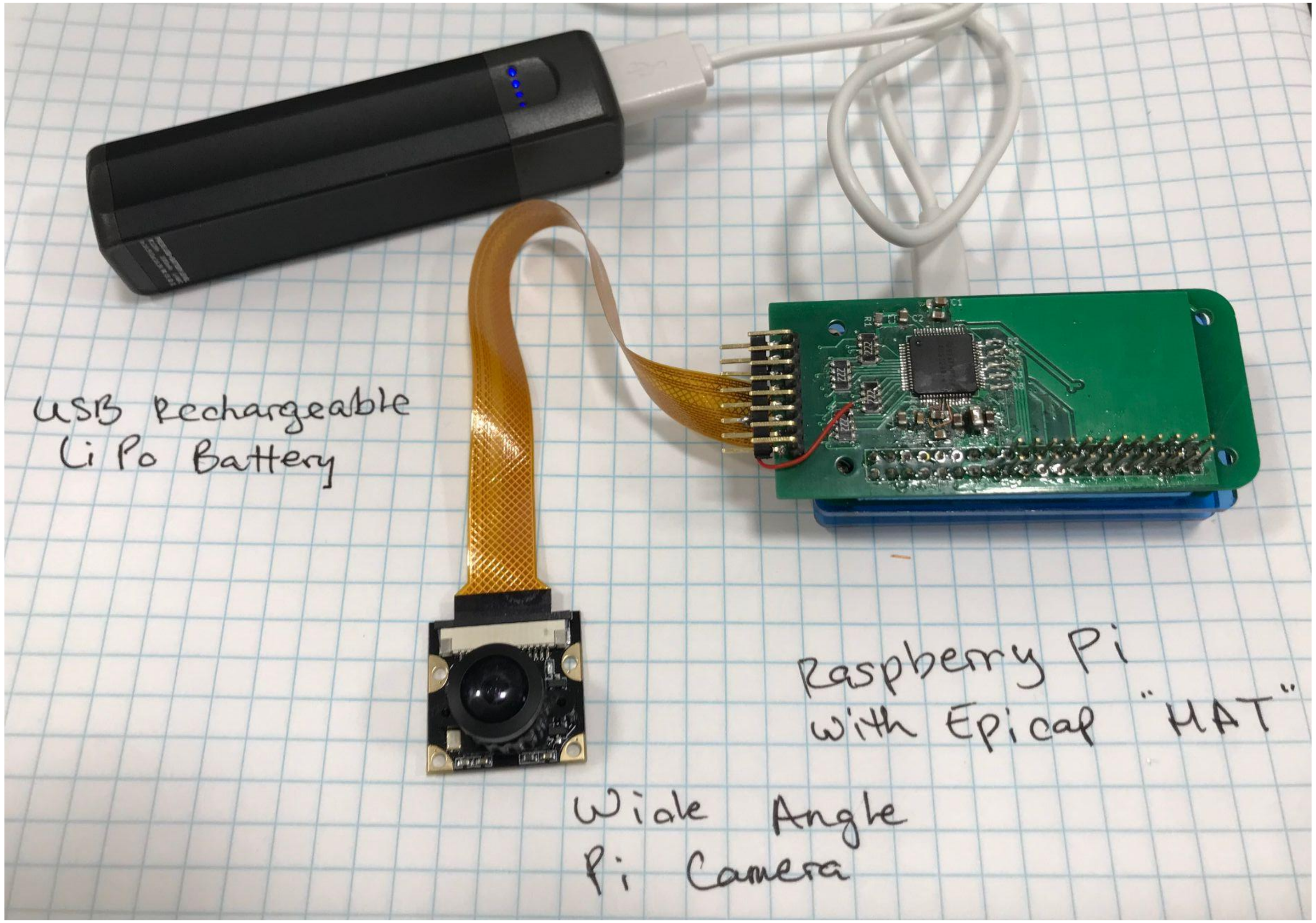


Design

Block Diagram #2



Design


USB Rechargeable
Li Po Battery
Raspberry Pi
with Epicap "HAT"
Wide Angle
Pi Camera

Power Subsystem

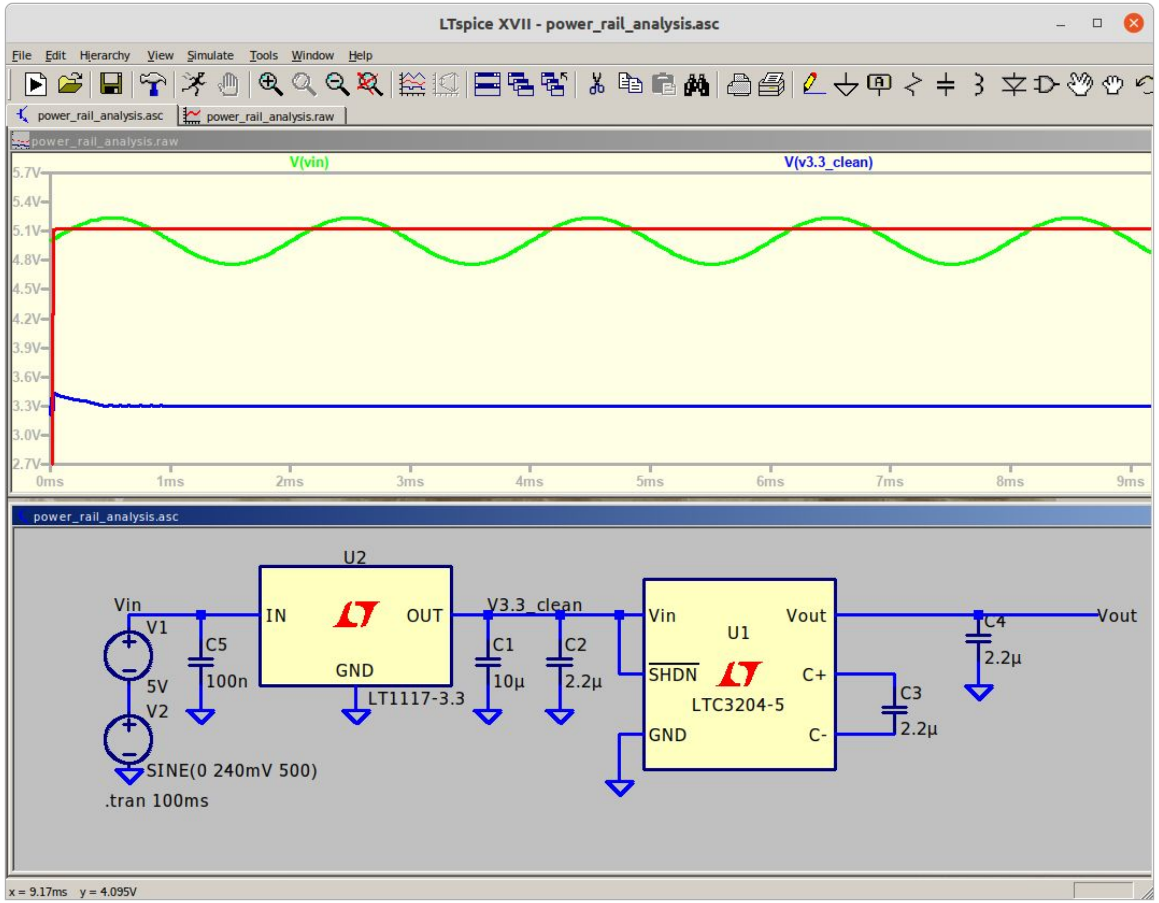
We require...

- Battery power
  - We can't have patients confined to a wall outlet.
- 3.3V and 5V lines
  - 3.3V logic, 5V bias for our sensor

Power Subsystem

Simulation

- Simulated with measured ~240mV ripple of USB portable battery output
- Output noise of 5V rail (critical)
    - sub -uV result

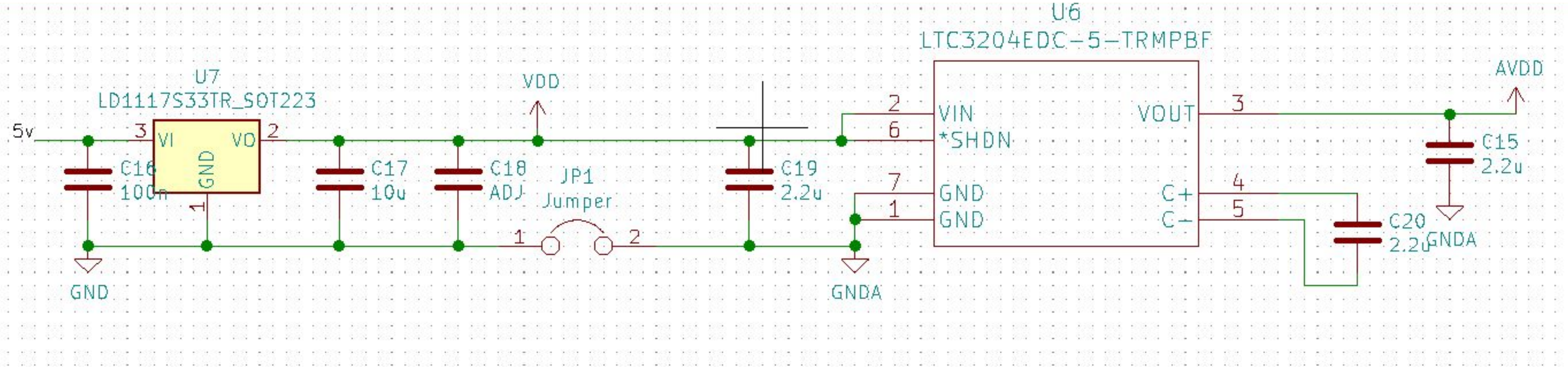


SINE(0 240mV 500)

.tran 100ms

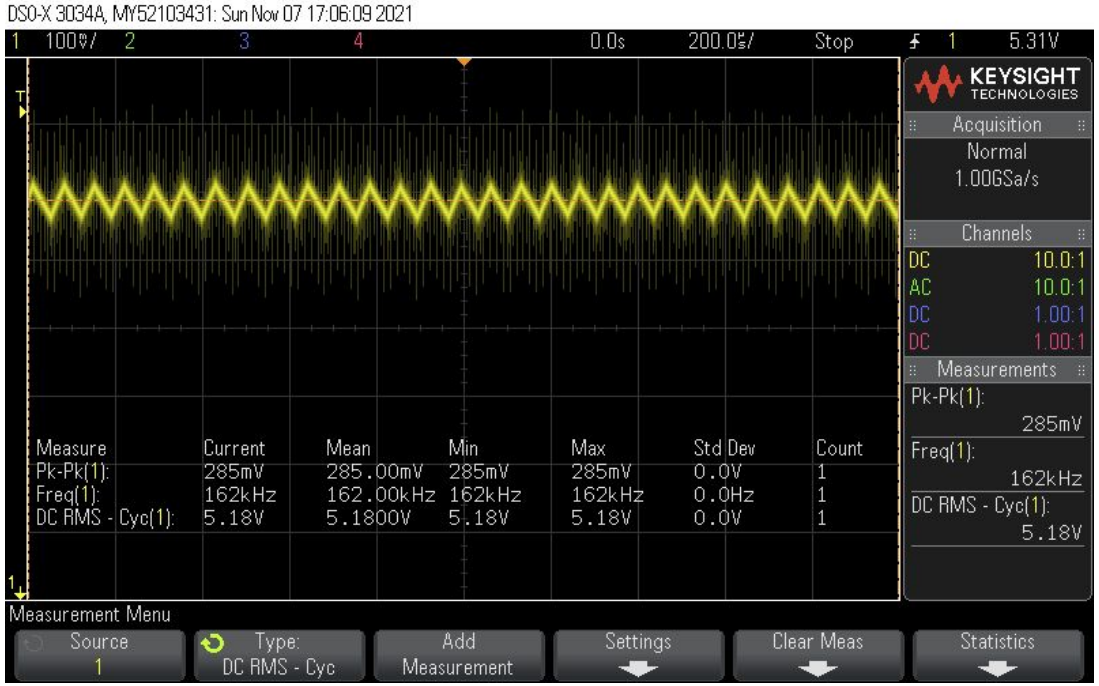
x = 9.17ms y = 4.095V

Implementation



Power Subsystem

Measurement



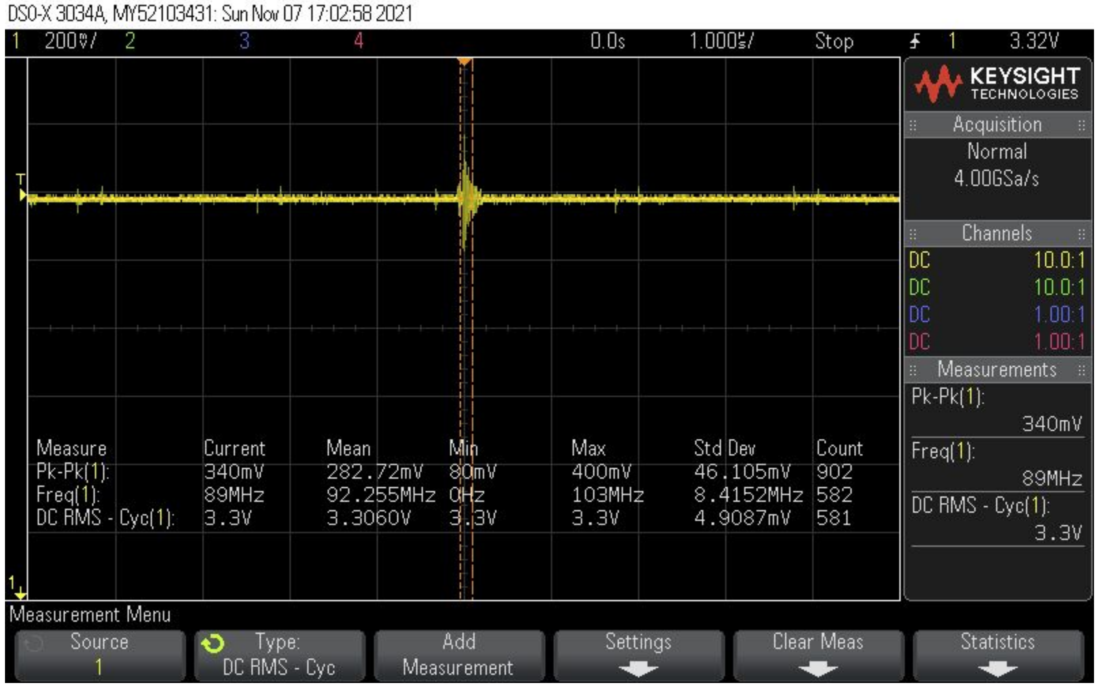
```
DSO-X 3034A, MY52103431: Sun Nov 07 17:06:09 2021
1  100mV/  2  3  4    0.0s  200.0µs/  Stop  1  5.31V

Acquisition
Normal
1.00GSa/s

Channels
DC  10.0:1
AC  10.0:1
DC  1.00:1
DC  1.00:1

Measurements
Pk-Pk(1):
           285mV
Freq(1):
          162kHz
DC RMS - Cyc(1):
           5.18V

Measure          Current   Mean        Min      Max      Std Dev   Count
Pk-Pk(1):        285mV     285.00mV    285mV    285mV    0.0V      1
Freq(1):         162kHz    162.00kHz   162kHz   162kHz   0.0Hz     1
DC RMS - Cyc(1): 5.18V     5.1800V     5.18V    5.18V    0.0V      1
```



```
DSO-X 3034A, MY52103431: Sun Nov 07 17:02:58 2021
1  200mV/  2  3  4    0.0s  1.000µs/  Stop  1  3.32V

Acquisition
Normal
4.00GSa/s

Channels
DC  10.0:1
DC  10.0:1
DC  1.00:1
DC  1.00:1

Measurements
Pk-Pk(1):
           340mV
Freq(1):
           89MHz
DC RMS - Cyc(1):
            3.3V

Measure          Current   Mean        Min      Max      Std Dev     Count
Pk-Pk(1):        340mV     282.72mV    80mV     400mV    46.105mV    902
Freq(1):         89MHz     92.255MHz   0Hz      103MHz   8.4152MHz   582
DC RMS - Cyc(1): 3.3V      3.3060V     3.3V     3.3V     4.9087mV    581
```

Power Subsystem

Summary

- Able to achieve goal of sub 2% voltage ripple for our power rails
  - Simulation no substitute for physical testing
- Able to reduce voltage ripple from USB battery by order of 4
- Able to utilize Raspberry Pi 5V rail and USB battery capability

Sensor Subsystem

Quick preface on EEGs

- Ear clips get bias voltages for scalp electrodes
- Scalp electrodes compared against these references
- Send signal over SPI to ADC
  - Begin streaming data to our logic subsystem

OpenBCI. *Ultracortex Mark VI Headset*. 2018, OpenBCI, LLC, New York, NY.

Sensor Subsystem

Introduction

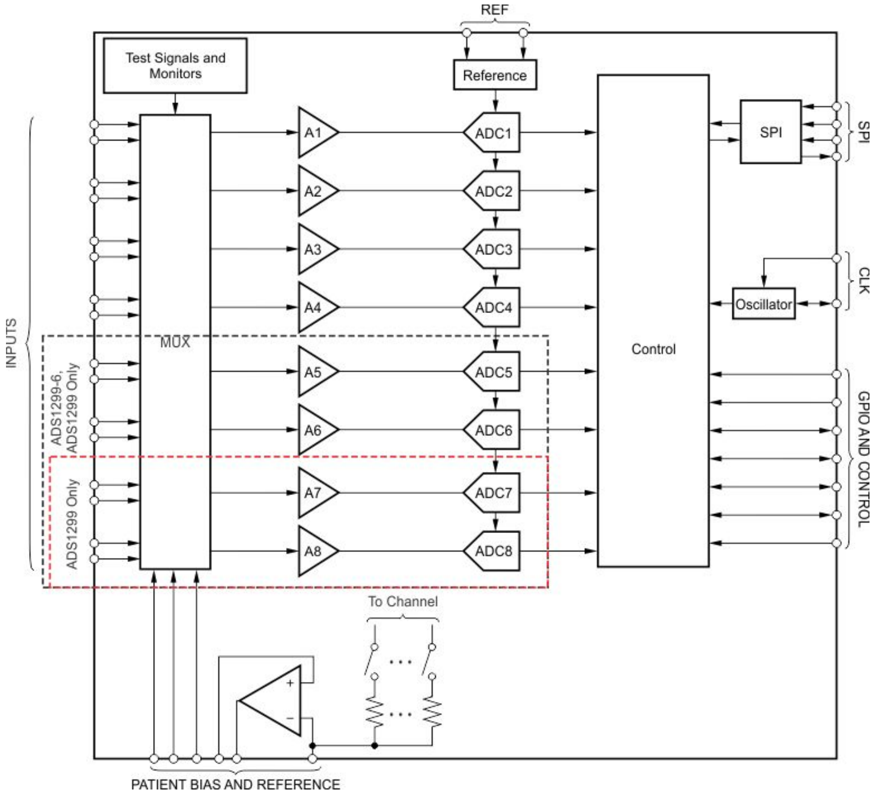
- ADC - analog to digital converter
  - Must be able to discriminate ~uV and sample ~240 Hz
  - Control, data over SPI
- Trade offs
  - More sensitivity - more susceptible to ESD
    - Noise from other parts in our system
  - More leads vs. space constraints

Texas Instruments. *TQFP64 package*. 2016, Texas Instruments, Inc., Dallas, TX.

Sensor Subsystem

Signal Conditioning

- High speed communication can create noise elsewhere
  - Split ground planes between 3v3 and 5v
- TVS diodes
  - Protects from static electricity
- Size/Mechanical considerations
  - Use resistor and capacitor arrays whenever possible

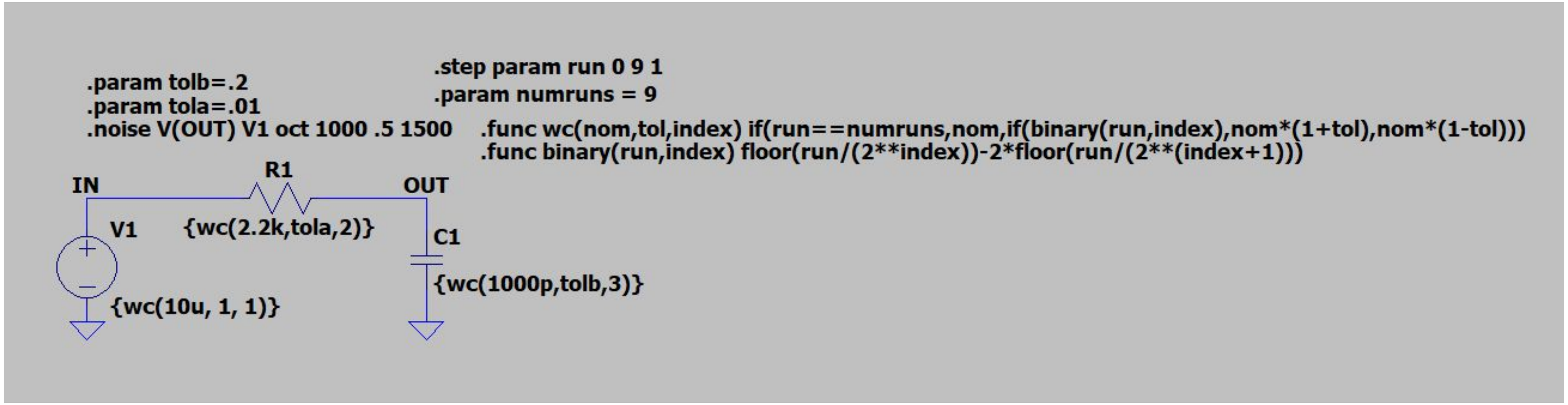


Texas Instruments.ADS1299 Block kDiagram. 2018, Texas Instruments, Inc., Dallas, TX.

Sensor Subsystem

Simulation - Isolation

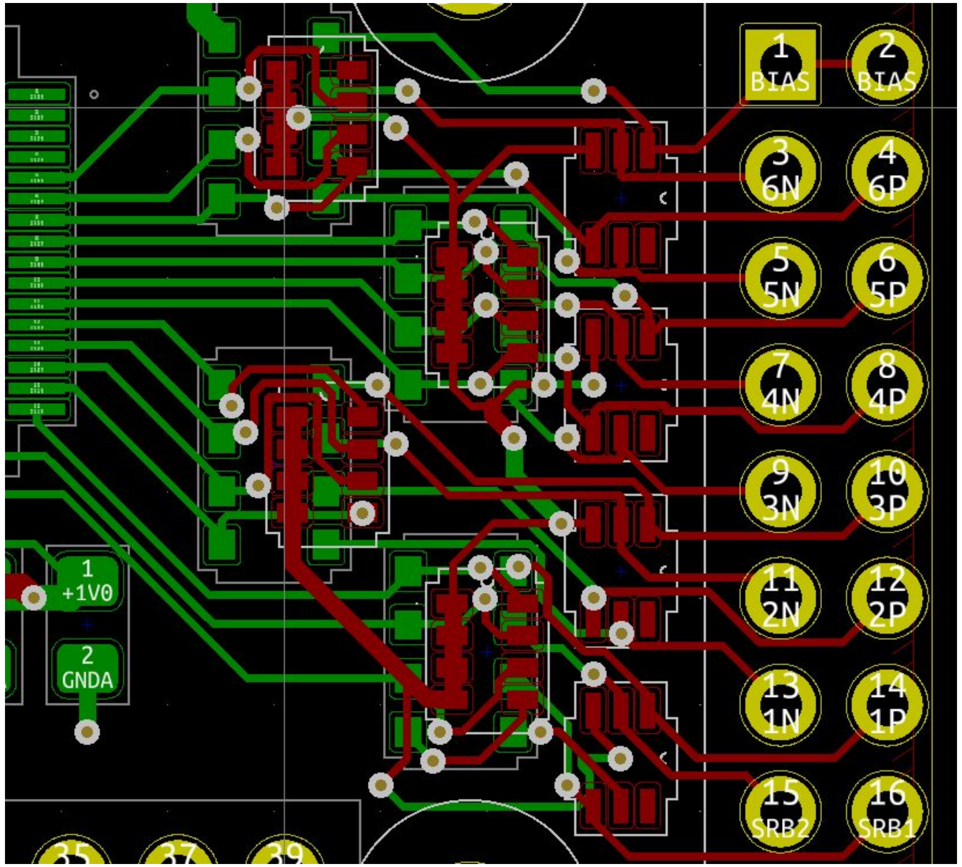
- RC network creates low-pass filter
- Result - signal attenuated beginning at 240Hz

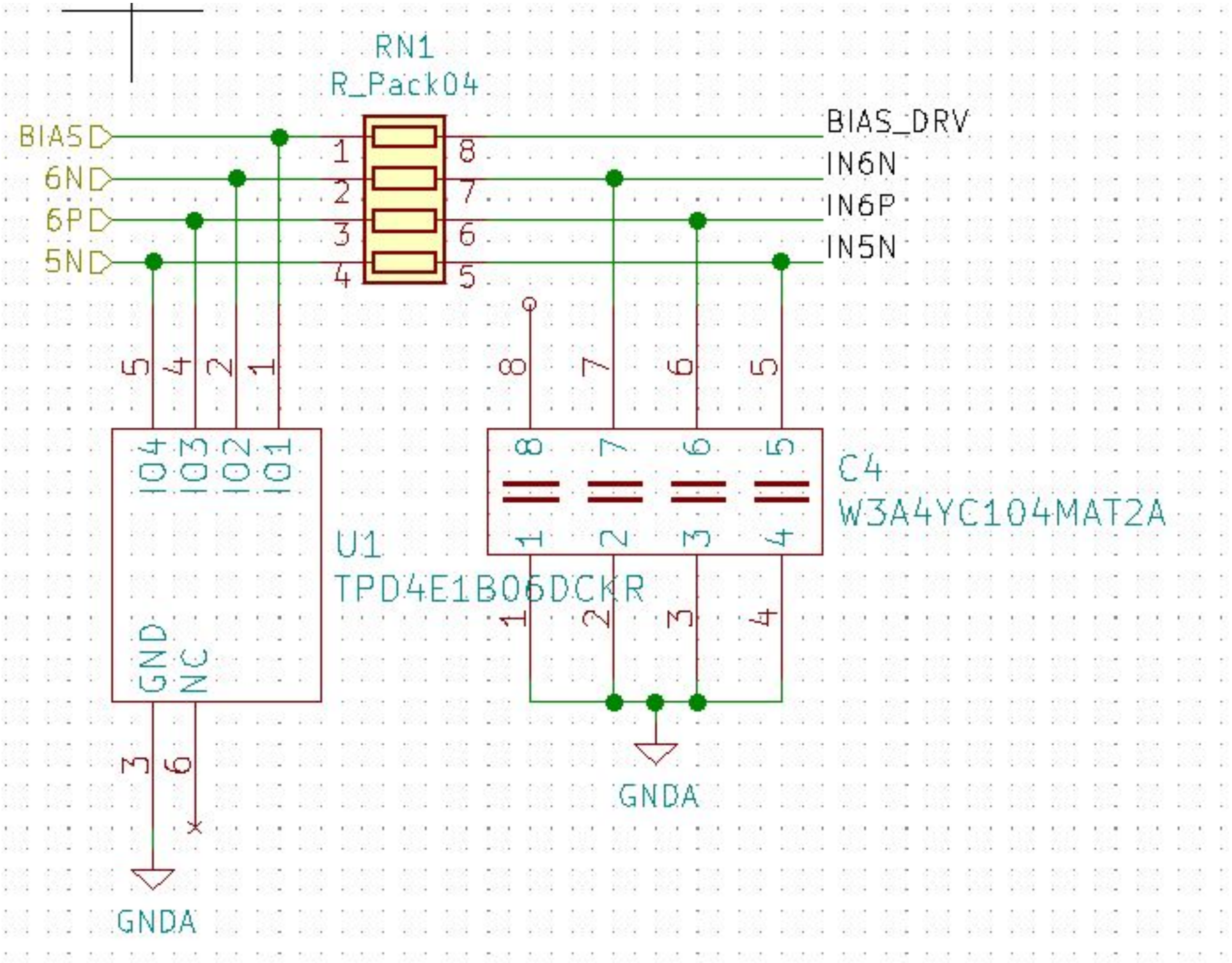


```
.param tolb=.2
.param tola=.01
.noise V(OUT) V1 oct 1000 .5 1500
.step param run 0 9 1
.param numruns = 9
.func wc(nom,tol,index) if(run==numruns,nom,if(binary(run,index),nom*(1+tol),nom*(1-tol)))
.func binary(run,index) floor(run/(2**index))-2*floor(run/(2**(index+1)))
```

Sensor Subsystem

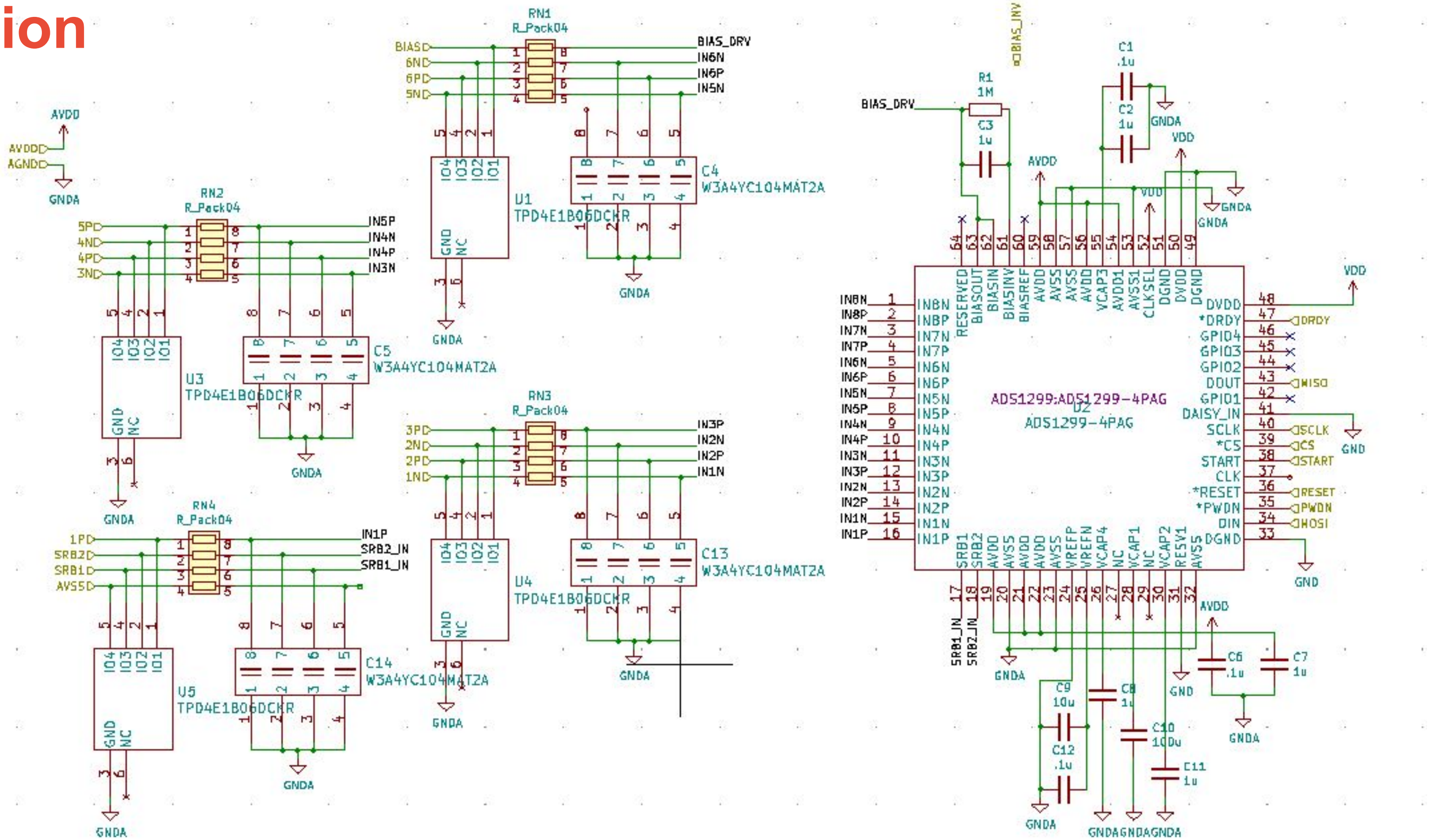
Implementation





Sensor Subsystem

Implementation



Sensor Subsystem

ADC Results

- Biasing capacitors have trouble remaining charged on new board
- Successfully able to communicate between Pi and ADC chip (logic side)
- Firmware pin assignments too hard to recover from

Control Subsystem

The STM32

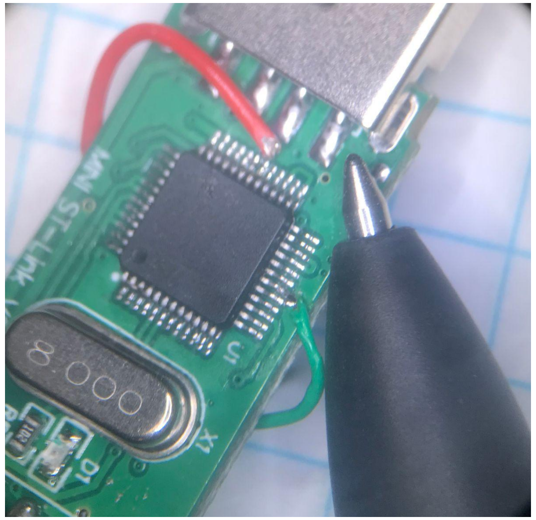
- Common microcontroller in use
  - Popular in automotive, appliance, high-reliability applications
- Configurable pins
  - Flexibility
- Adequate computing power
  - Must have enough memory for ADC + video task
    - Proved hard to source specific product line
- Proprietary "DCMI" camera interface

Mouser Electronics. *LQFP144 Package*.
2014, Mouser Electronics, Inc., Mansfield, TX

Control Subsystem

STM32 Results

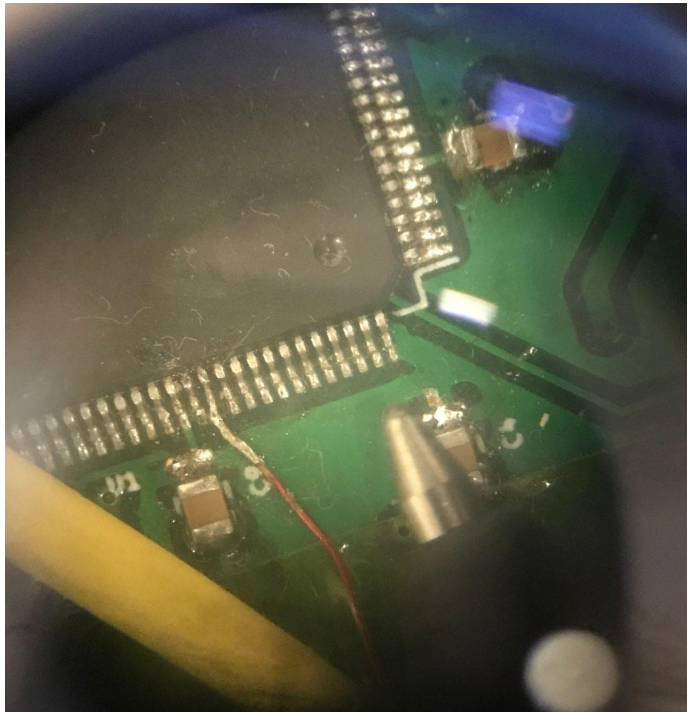
- Chose SWDIO to debug
  - Was able to flash firmware and trace through our program as it executed on chip
- Chose to omit external crystal oscillator
  - Rely on internal RC clock
- Prioritized cleanliness of traces when assigning pins
  - Readily configurable but tricky to modify in IDE



Jumps made on STLink 2.1 Programmer for nRST and SWO

Control Subsystem

STM32 Results



Our SWO pin to STLink

Control Subsystem

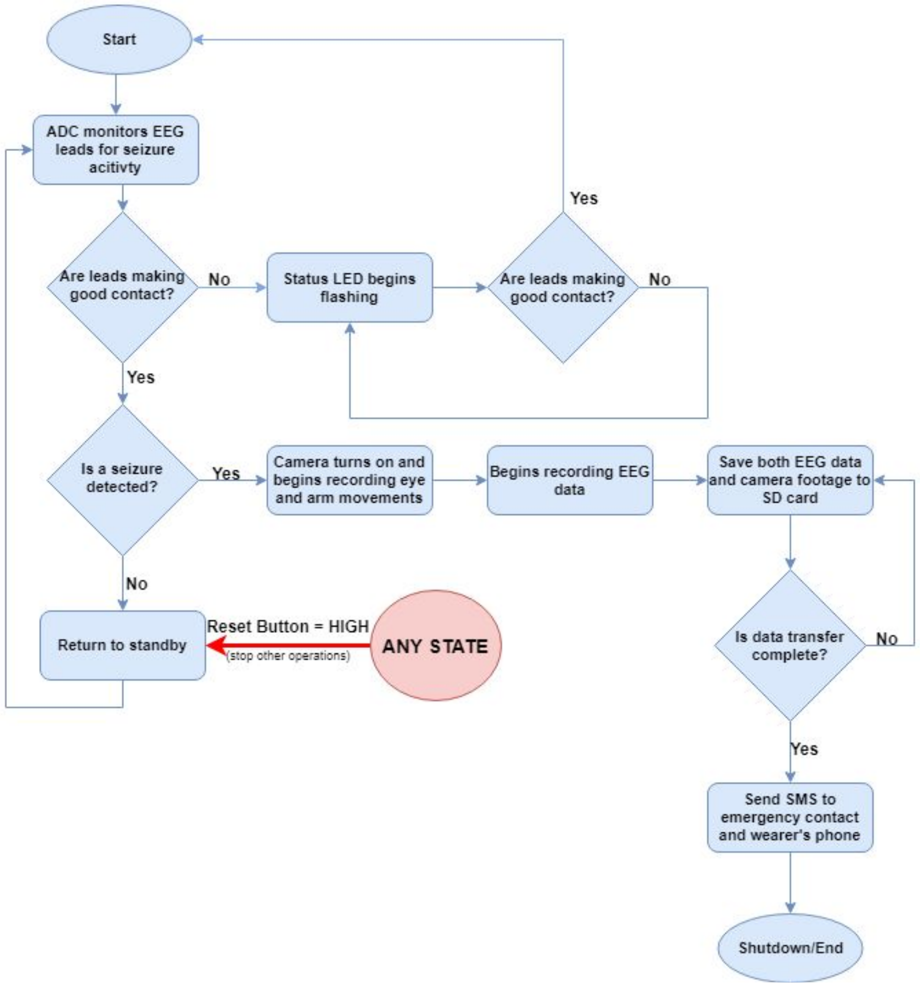
Control Subsystem 2: The Raspberry Pi

- Already present in design, and much more powerful than STM32
- Essentially a desktop PC
- Has fewer GPIO pins, but isn't a hassle to solder

Software Algorithm

- ADC receives and sends the EEG data to STM32 board or Raspberry Pi
- STM32 calculates their average, and compares it with the average of the previous set of data.
- If current average is larger than a threshold, STM32 will send a signal to the camera to ask it to start the recording.

Microcontroller Firmware Flowchart



Start

ADC monitors EEG leads for seizure acitivty

Are leads making good contact?

No

Status LED begins flashing

Are leads making good contact?

Yes

No

Yes

Is a seizure detected?

Yes

Camera turns on and begins recording eye and arm movements

Begins recording EEG data

Save both EEG data and camera footage to SD card

No

Return to standby

Reset Button = HIGH

(stop other operations)

ANY STATE

Is data transfer complete?

No

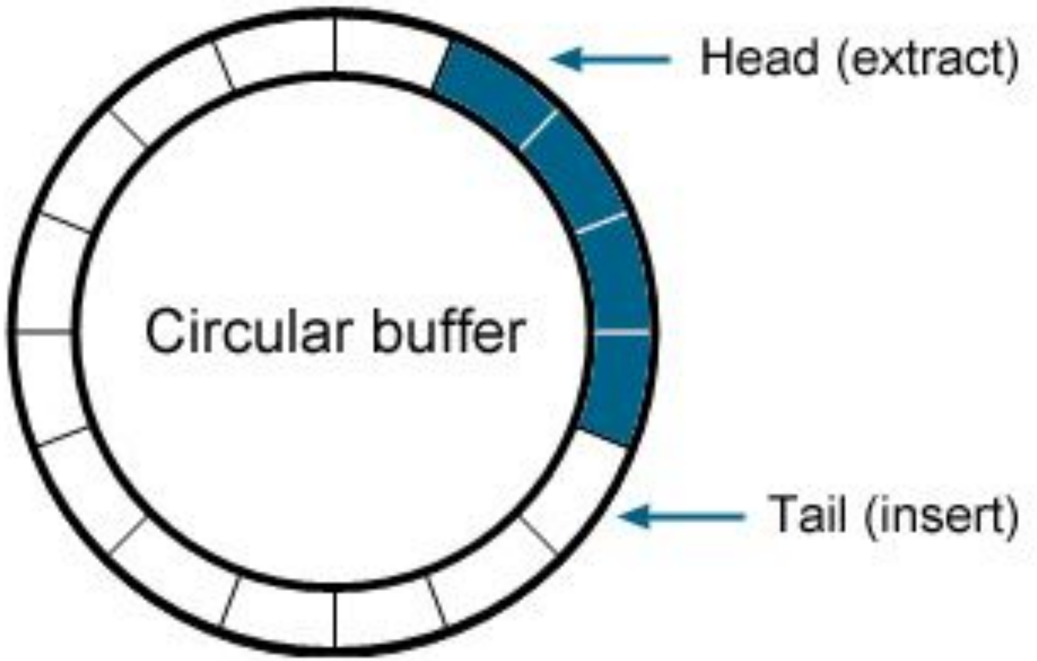
Yes

Send SMS to emergency contact and wearer's phone

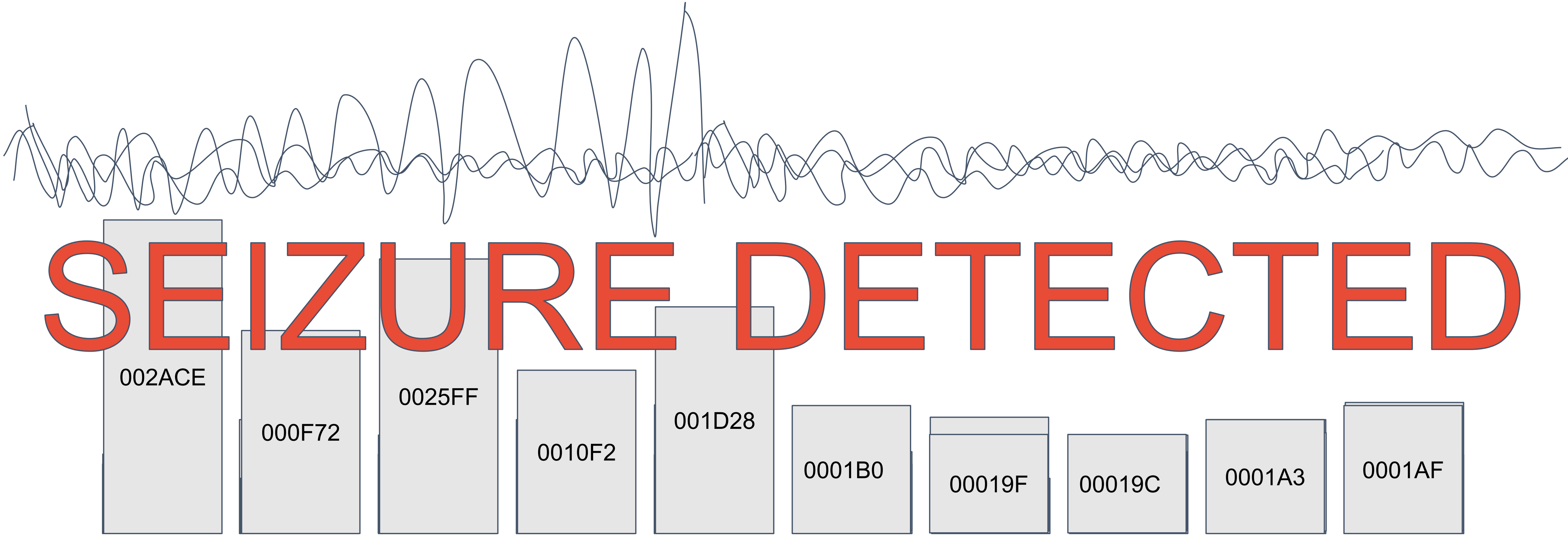
Shutdown/End

Ring buffer

- A single, fixed-size buffer, which connects end to end, achieves buffering the data streaming without storing any unused data.



Software Algorithm (animation)



SEIZURE DETECTED

average(buf[0:4]) ⇒ average(buf[5:9])

STM32 Software based on C

STM32 board failed to communicate with the ADC

- Our STM32 does not have an external crystal, HSE, to set up the system clock, so we always meet the trace no synchronization problem.
    - Relying on internal HSI clock can cause temperature-dependent jitter
        - May create communication issues with high-speed SPI and JTAG debug lanes

- STM32 has invalid SPI communication because of the wrong pin assignment (PA4 for CS and any other Pin for DRDY)

Raspberry Pi Software based on Python

Ringbuffer

- A single, fixed-size buffer, which connects end to end, achieves buffering the data streaming without storing any unused data.

SPI communication and system call

- We directly make a system call to the camera part in the ADC file, to avoid any unexpected exception or bugs.

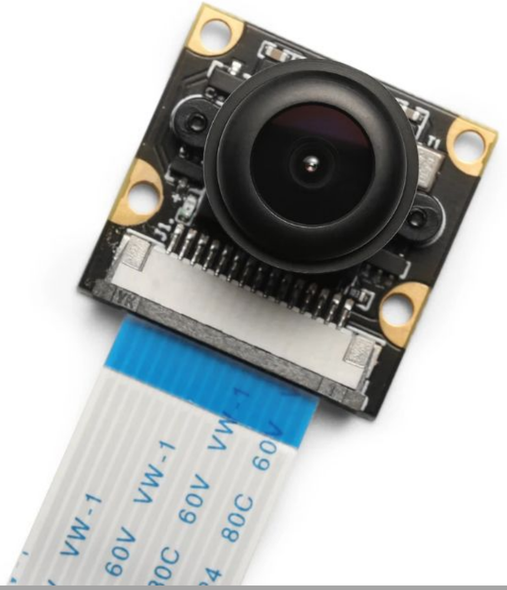
Shell script

- We set up a shell script to let the whole project can run automatically when our Pi OS boots up.

Camera Module

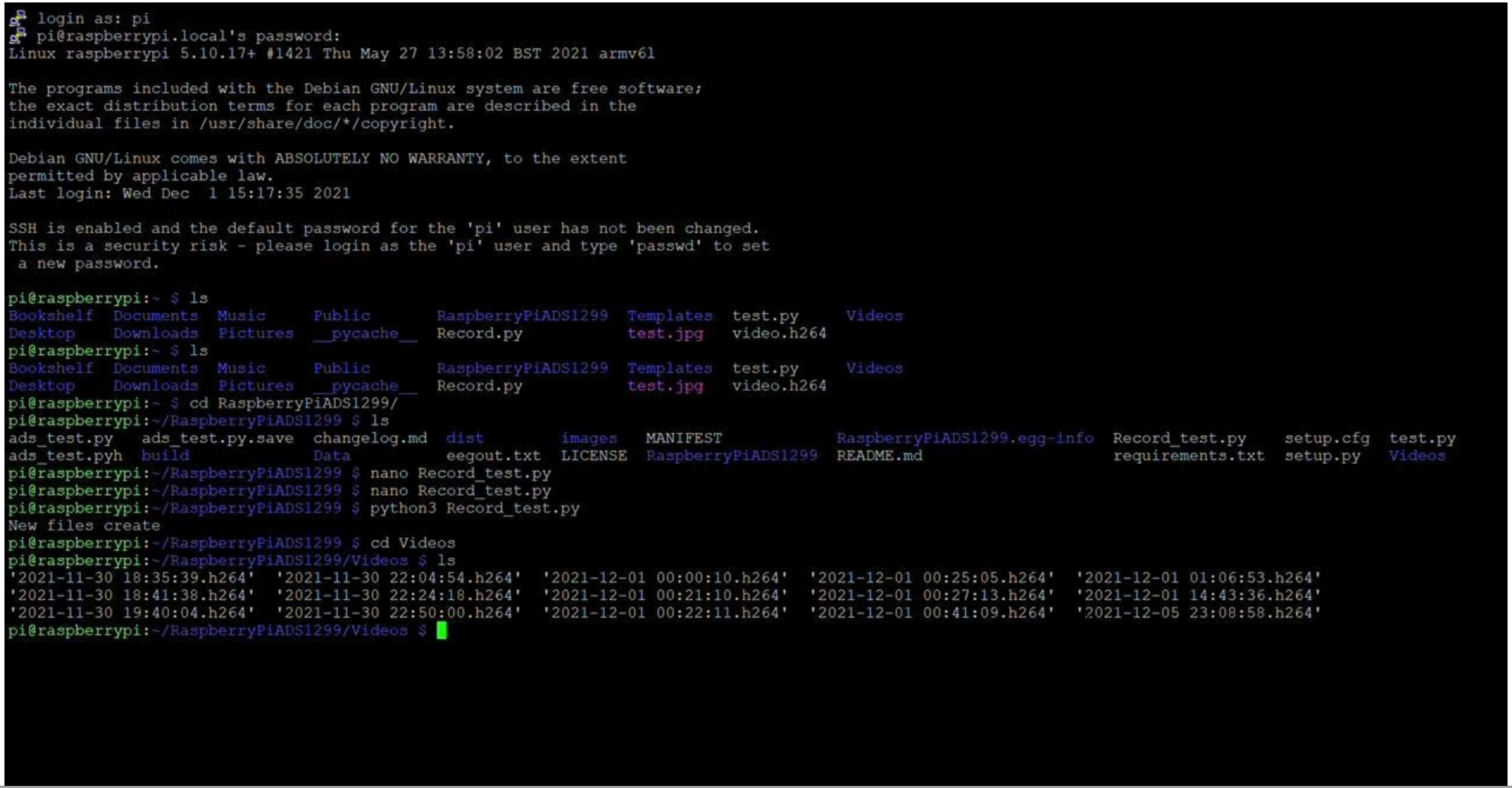
We require...

- Wide-angle camera that can record patient's eye and arm movement to shoulder height (minimum 240p → **480p**)
    - Wide Angle FOV160° 5-Megapixel
    - Located at cap visor
- Start recording when detects a seizure
- Save file onto an SD card for physician use later



Camera Module

Implementation



```
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 5.10.17+ #1421 Thu May 27 13:58:02 BST 2021 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec  1 15:17:35 2021

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $ ls
Bookshelf  Documents  Music     Public       RaspberryPiADS1299  Templates  test.py     Videos
Desktop    Downloads  Pictures  __pycache__  Record.py           test.jpg   video.h264
pi@raspberrypi:~ $ ls
Bookshelf  Documents  Music     Public       RaspberryPiADS1299  Templates  test.py     Videos
Desktop    Downloads  Pictures  __pycache__  Record.py           test.jpg   video.h264
pi@raspberrypi:~ $ cd RaspberryPiADS1299/
pi@raspberrypi:~/RaspberryPiADS1299 $ ls
ads_test.py   ads_test.py.save  changelog.md  dist        images   MANIFEST            RaspberryPiADS1299.egg-info   Record_test.py    setup.cfg  test.py
ads_test.pyh  build             Data          eegout.txt  LICENSE  RaspberryPiADS1299  README.md                     requirements.txt  setup.py   Videos
pi@raspberrypi:~/RaspberryPiADS1299 $ nano Record_test.py
pi@raspberrypi:~/RaspberryPiADS1299 $ nano Record_test.py
pi@raspberrypi:~/RaspberryPiADS1299 $ python3 Record_test.py
New files create
pi@raspberrypi:~/RaspberryPiADS1299 $ cd Videos
pi@raspberrypi:~/RaspberryPiADS1299/Videos $ ls
'2021-11-30 18:35:39.h264'  '2021-11-30 22:04:54.h264'  '2021-12-01 00:00:10.h264'  '2021-12-01 00:25:05.h264'  '2021-12-01 01:06:53.h264'
'2021-11-30 18:41:38.h264'  '2021-11-30 22:24:18.h264'  '2021-12-01 00:21:10.h264'  '2021-12-01 00:27:13.h264'  '2021-12-01 14:43:36.h264'
'2021-11-30 19:40:04.h264'  '2021-11-30 22:50:00.h264'  '2021-12-01 00:22:11.h264'  '2021-12-01 00:41:09.h264'  '2021-12-05 23:08:58.h264'
pi@raspberrypi:~/RaspberryPiADS1299/Videos $
```

Camera Module

Results

Software: OpenBCI GUI

Visualization of EEG Data

- Software tool for visualizing, recording, and streaming EEG data
- Data: live-time, played back, saved to your computer in .txt format
- Widgets
  - Time Series
  - FFT Plot
  - Head Plot

Software: OpenBCI GUI

Visualization of EEG Data

Conclusion

Steps to iterate upon

- Work more with Raspberry Pi
  - $5 per Pi vs. $10 per for STM32 in 3000+ bulk order
  - Raspberry Pi HAT standard
- Redesign board
  - Solve biasing circuit issues
- Software implementation
  - Validate using formal methods