Pegasus-21

Cumulative Design Review

Senior Design Project

Spring 2016

Our Team

Istvan Kreisz
EE

Keith Miller
CSE

Trevor Berry
EE

Zlatan Aksamija
Advisor

Pegasus-21

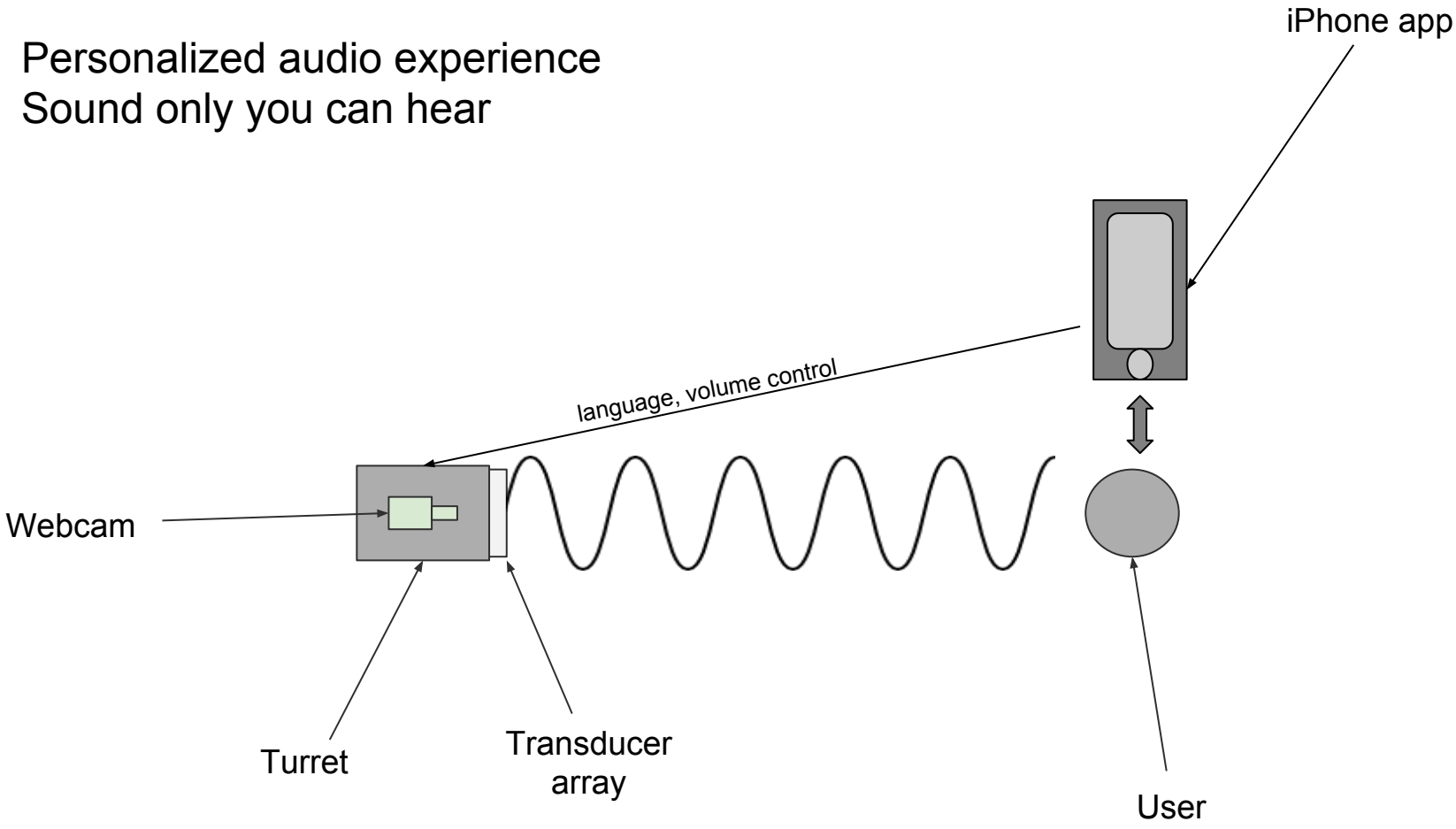
A turret-mounted speaker capable of transmitting audio in a narrow beam that tracks the listener.

System Overview

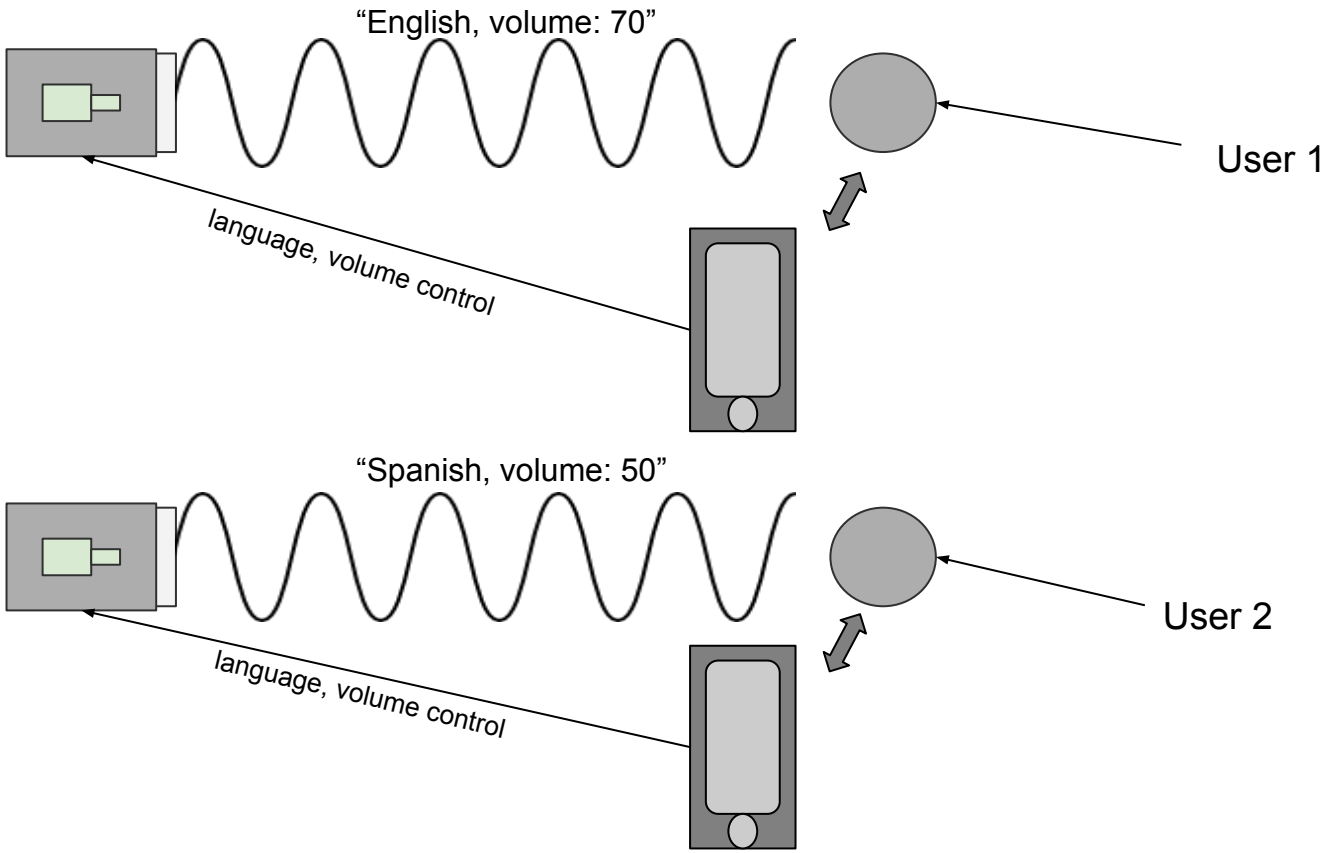
- Users take a short video clip of their faces using our iPhone app
- This video is used to train our neural network to recognize a face
- Once trained, the turret mounted webcam searches for and tracks the user, focusing the audio beam in their direction
- Users use the iPhone app to select the language and control the volume of the audio

Why?

- Personalized audio experience
- Sound only you can hear



Multi User System



Some Background: Acoustic Directionality

-Directionality of sound correlates with the size of speaker compared to wavelength. The formula for far field directivity of a flat circular piston in an infinite baffle is

$$p(\theta) = \frac{p_0 J_1(k_a \sin\theta)}{k_a \sin\theta}$$

where...

$$k_a = \frac{2\pi a}{\lambda}$$

p_0 = pressure on axis

a = piston radius

J1 = Bessel Function

θ = Angle off axis

-Piezoelectric transducers approximate the ideal acoustic piston

-Directionality is achieved by maximizing the piston radius and minimizing the wavelength

("Loudspeaker" n.d.)

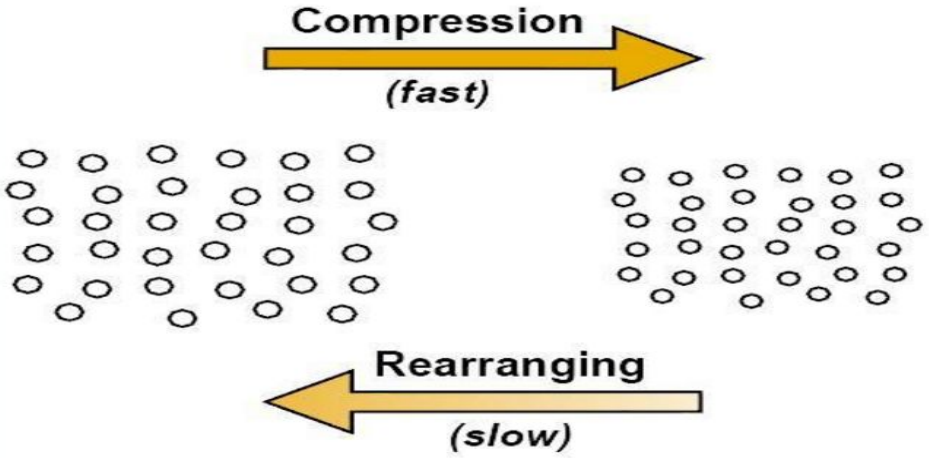
Sound From Ultrasound

Technology to create directional, highly focused soundbeams:

-An array of transducers emits ultrasonic carrier waves in the direction of target

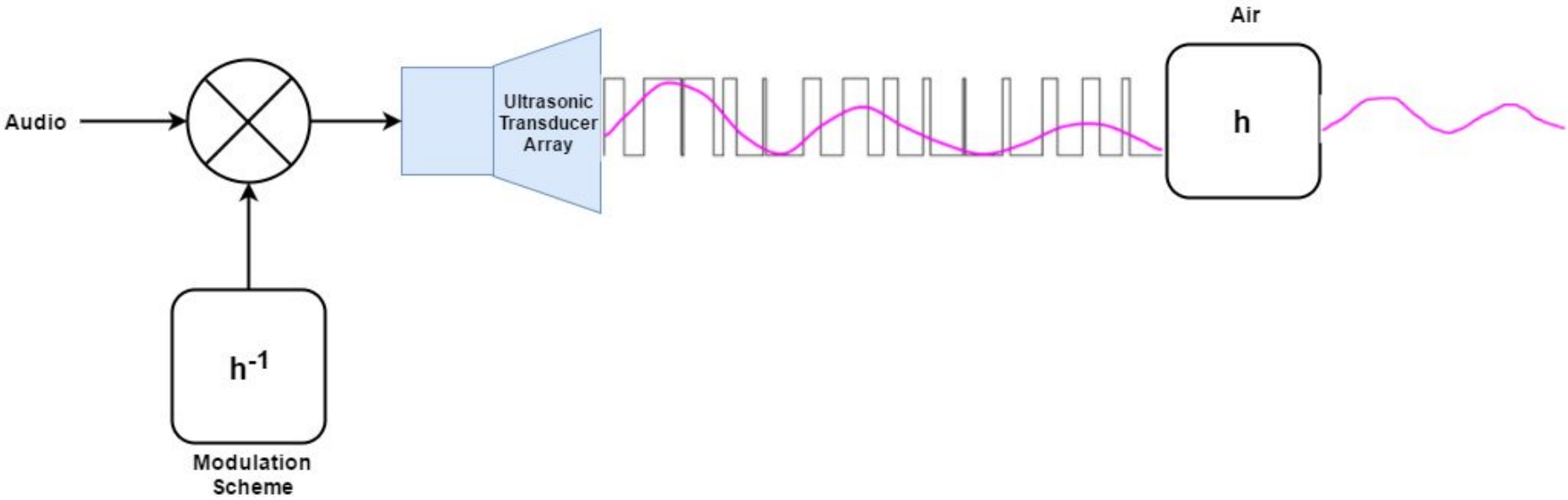
-Air, as a nonlinear medium, acts as a demodulator (no need for receiver)

-When the modulated wave reaches a person the modulated sound becomes audible



(Nema, “Ultrasonic Directive Speaker”)

Sound From Ultrasound



http://www.soundlazer.com/what-is-a-parametric-speaker/

CDR Deliverables

Keith:

- Fully working software pipeline for tracking, interfaced with the turret

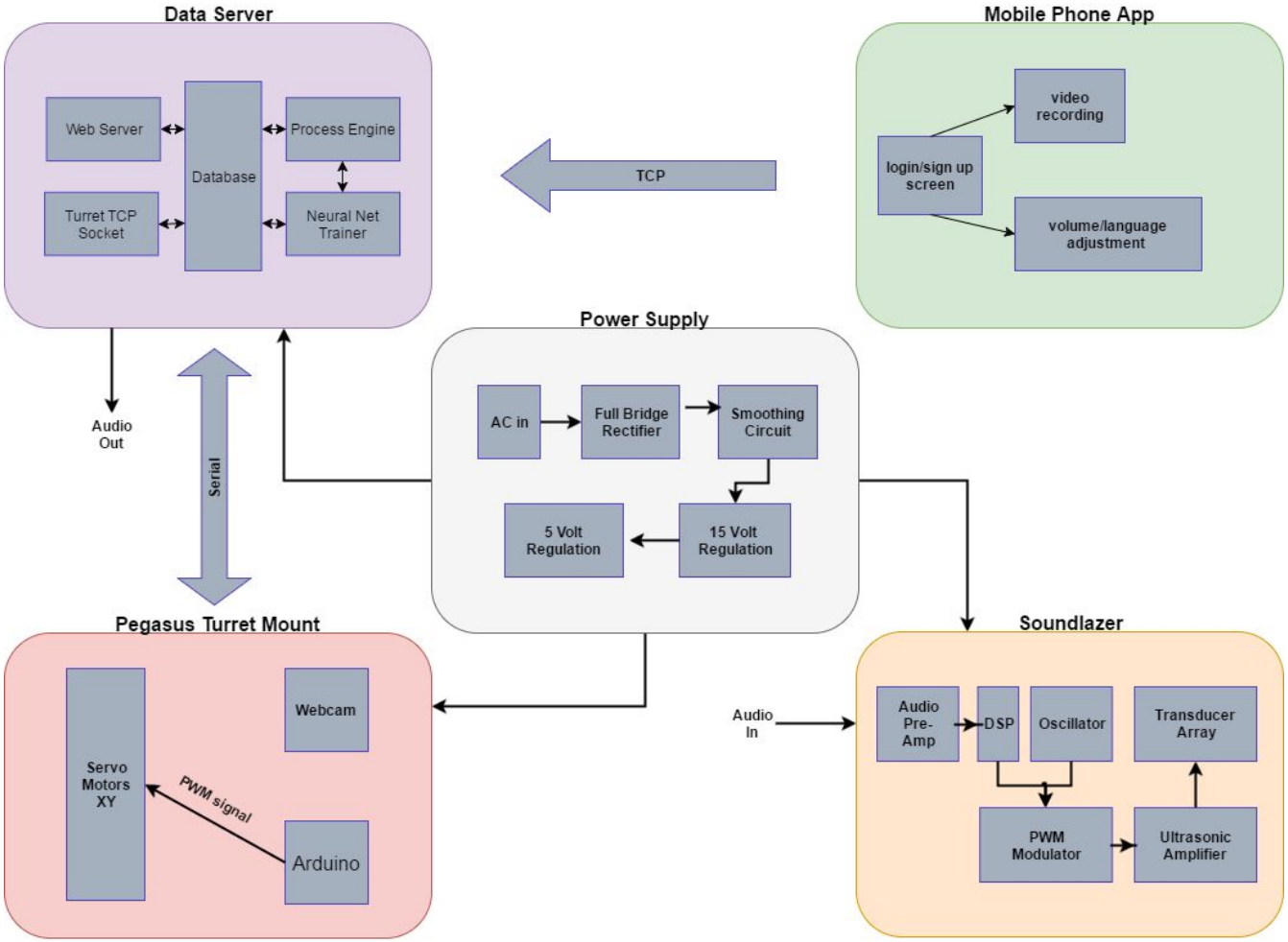
Trevor:

- Build universal power supply
- 3D print acoustic horn array

Istvan:

- Build iPhone application skeleton

System Block Diagram



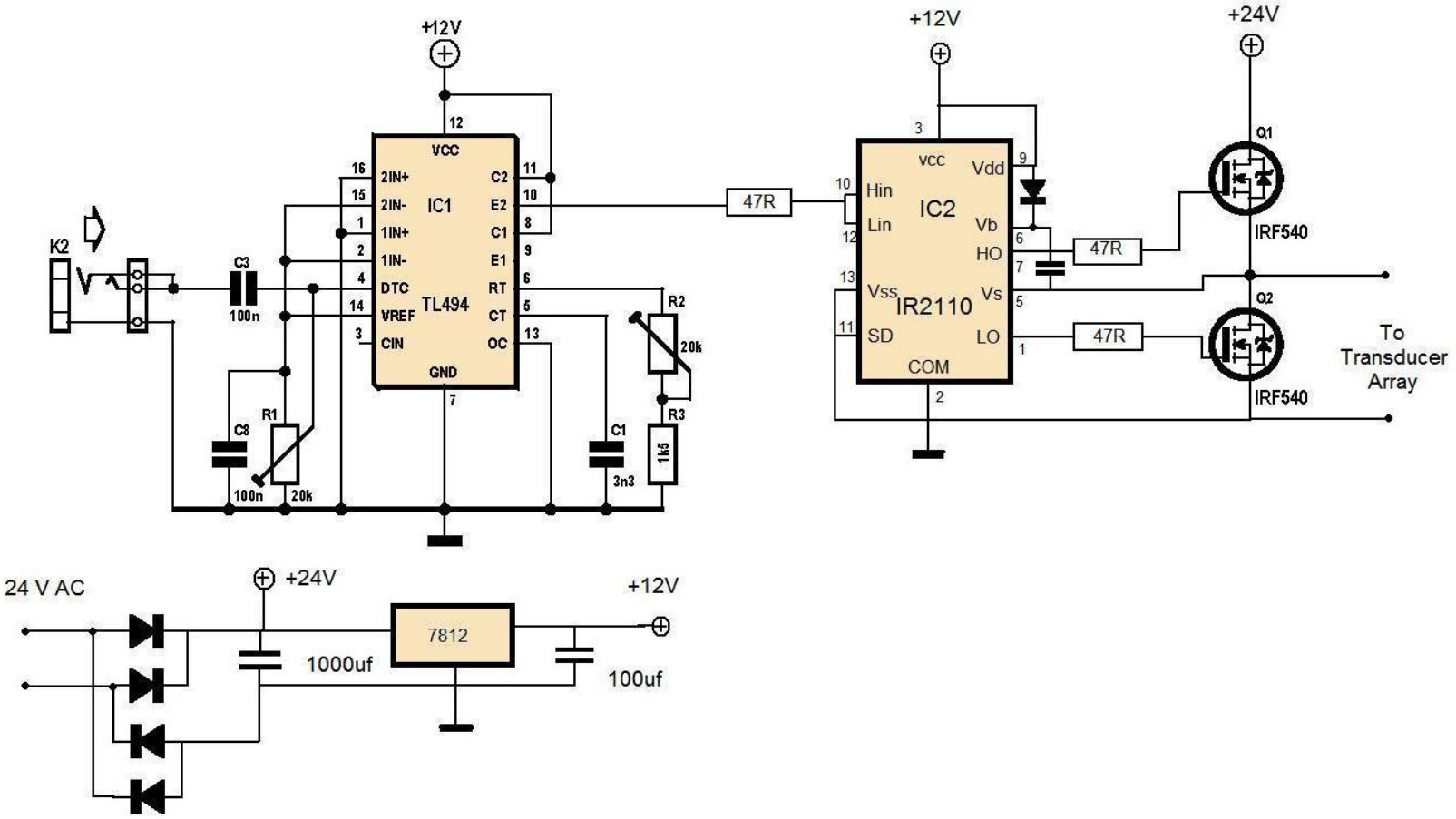
Soundlazer - Overview

- Soundlazer emits directional audio signal
- Mounted onto Pegasus Turret
- Gets audio input from data server
- Audio is controlled through iPhone App

Soundlazer - Specs

- 12VDC power input
- 120dB ultrasonic output
- Usable range indoors: 30.0 feet
- Beam size of around 3 feet
- 40 GHz Carrier Wave
- Pulse Width Modulation
- Digital Signal Processing
- 5 Watt Carrier Wave

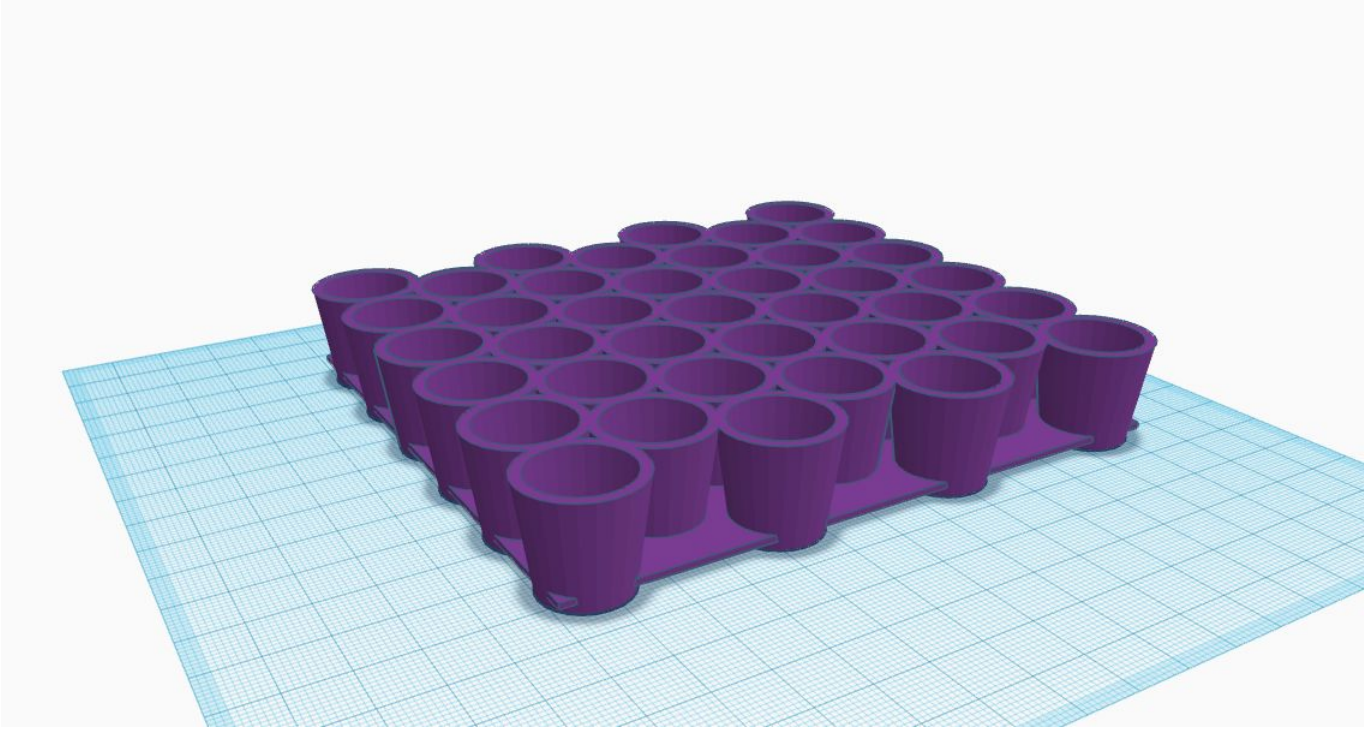
Soundlazer - Ultrasonic Directional Speaker Circuit



Soundlazer - Improved Directionality

- Custom 3d design of an acoustic horn array.

-Acts as an acoustic transformer, increasing speaker diameter and directionality



Soundlazer - Measured Performance

SPL at 7 Feet		
	Indoors	Outdoors
On Axis	65dB	63dB
5ft	52dB	48dB

SPL at 30 Feet		
	Indoors	Outdoors
On Axis		55dB
5 ft.		47dB

Soundlazer - What's left to be done?

- Redesign acoustic horn array
- Add microphone integration to soundlazer
- Edit DSP algorithms if time permits

Data Server - Overview

- Receives user information from the iPhone application
- Communicates over serial with the turret to run the facial recognition and determine the appropriate servo positions
- Multiprocessing enables us to run the iPhone server process separate from the turret process
- Utilizes processor affinity
- Trains/stores neural network

Data Server - Specs

MacBook Docker

```
[root@9155f23a7f10:~/openface# time ./demos/cl
/usr/local/lib/python2.7/dist-packages/sklear
  "in 0.17 and will be removed in 0.19", Depr

=== images/examples/carell.jpg ===
Predict SteveCarell with 0.97 confidence.

=== images/examples/adams.jpg ===
Predict AmyAdams with 0.81 confidence.

=== images/examples/lennon-1.jpg ===
Predict SteveCarell with 0.50 confidence.

=== images/examples/lennon-2.jpg ===
Predict DavidBoreanaz with 0.43 confidence.

real    0m8.744s
user    0m7.230s
sys     0m3.870s
[root@9155f23a7f10:~/openface#
[root@9155f23a7f10:~/openface#
root@9155f23a7f10:~/openface#
```

Nvidia TX1

```
[ubuntu@tegra-ubuntu:~/openface$ time ./demos/classif
/usr/local/lib/python2.7/dist-packages/sklearn/lda.p
  "in 0.17 and will be removed in 0.19", Deprecation

=== images/examples/carell.jpg ===
Predict SteveCarell with 0.99 confidence.

=== images/examples/adams.jpg ===
Predict AmyAdams with 0.64 confidence.

=== images/examples/lennon-1.jpg ===
Predict SteveCarell with 0.52 confidence.

=== images/examples/lennon-2.jpg ===
Predict DavidBoreanaz with 0.53 confidence.

real    1m4.067s
user    1m2.980s
sys     0m0.950s
```

Data Server - NVIDIA Jetson TX1

- GPU: NVIDIA Maxwell GPU with 256 CUDA cores (1 TFLOP)
- CPU: Quad-Core Cortex-A57 64-bit ARM Processor
- DRAM: 4 GB DDR4
- Storage: 16GB fast eMMC 4.51
- Wi-Fi

Data Server - Preexisting Technologies

- OpenCV: computer vision and media manipulation library (Python)
- OpenFace: facial recognition project
- Torch: scientific computing Framework (Lua)
- SciKit-Learn: machine Learning library (Python)
- dlib: highly optimized machine learning
- FFMPEG: converts video formats
- Basic socket and serial libraries: communication to other subsystems

Data Server - What's left to be done?

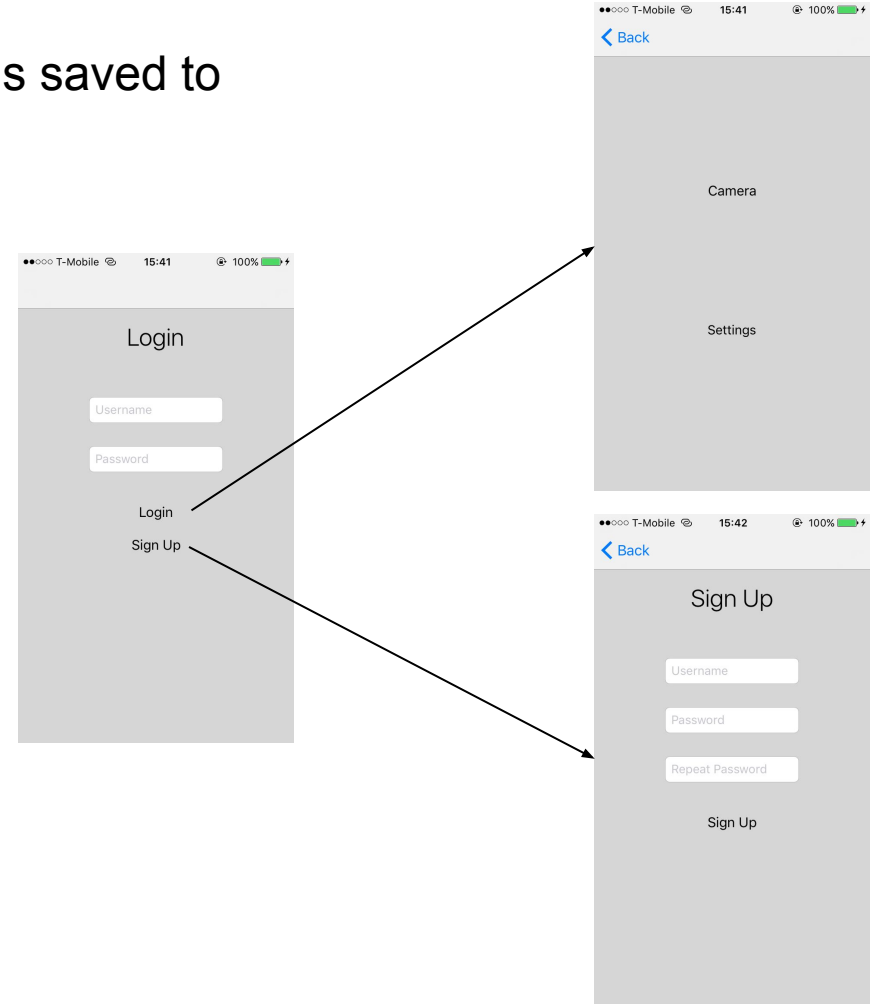
- Various optimizations
  - Overclocking
  - Rebuilding crucial libraries with CUDA
  - C++ rewrite
  - Removing Image Display and reduce logging
- PID tuning
  - Compensate for potential low frame rates
  - Smooth the tracking
- Generic Classifier

iPhone App - Overview

- Provides a user interface to interact with the system
- Has two main functionalities:
  - Allows the user to train the neural network by sending a video to the server
  - Allows the user to control the volume and language of the audio

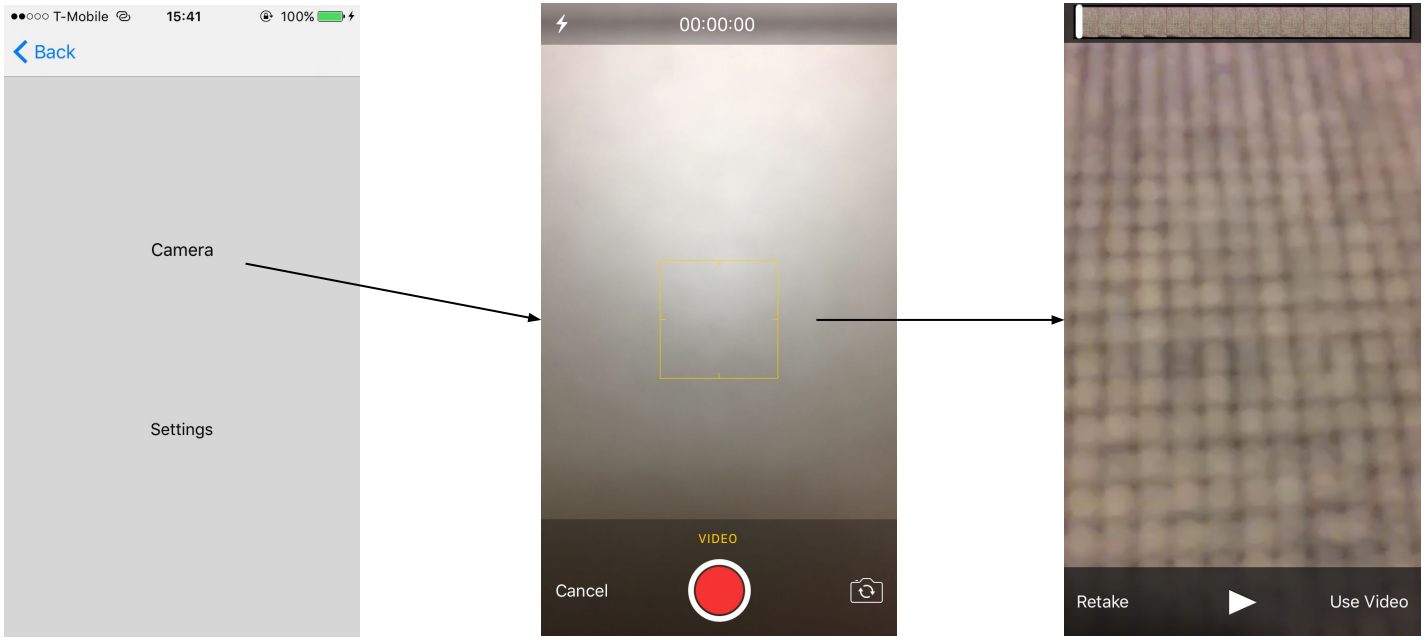
iPhone App - Login/Sign Up

- Once signed up, user information is saved to Core Data
- Login button navigates to Main Menu, automatically establishes connection to data server
- Navigating back to the Login screen closes TCP connection



iPhone App - Neural Network Training

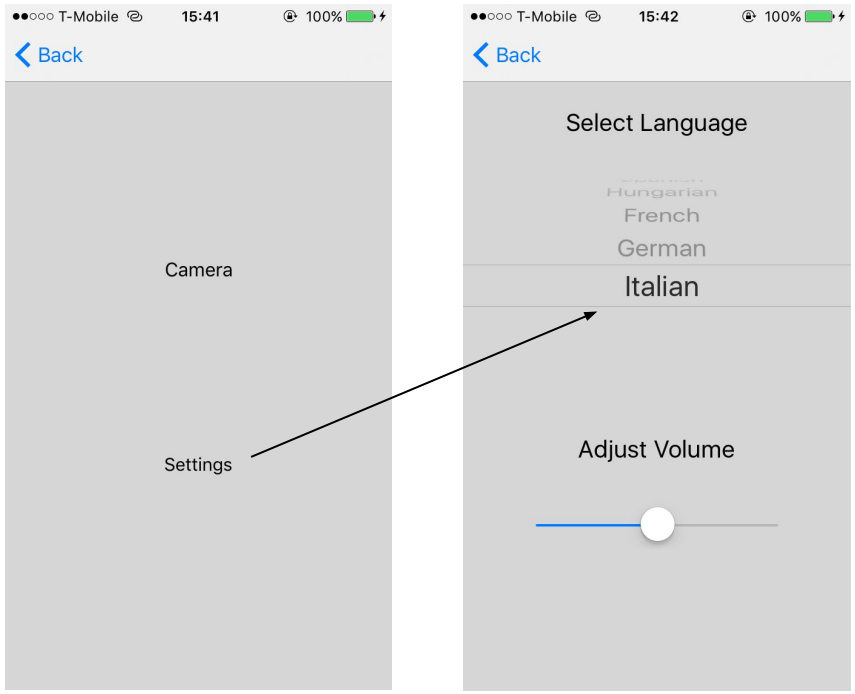
- "Camera" button brings up the camera
- Allows the user to take a short video
- "Use Video" button sends the video to the server to train the neural network



iPhone App - Settings

Allows the user to:

- select the language
- control volume of the audio



iPhone App - Specs

- Development: Xcode 8.2, Swift 3
- TCP connection with data server
- Username, language and volume settings are sent as simple strings
- Video file is sent to the server in .mov format
- Multithreaded: connecting to server runs off the main thread to avoid freezing up the UI

iPhone App - Specs

- Used frameworks:
  - “CoreData”: framework to permanently save user details (name, password, language, volume)
  - “SwiftSocket”: CocoaPod to establish TCP communication with the server and send data to it
  - “MobileCoreServices” framework for video recording

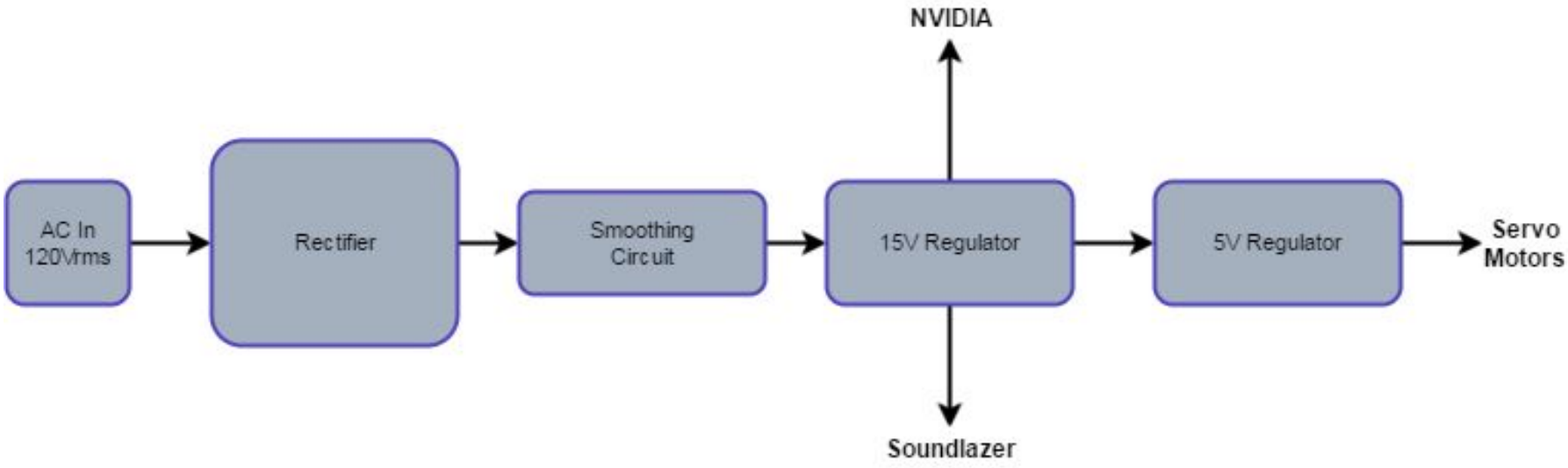
iPhone App - What's left to be done?

- More multithreading
- More error handling
- UI upgrade

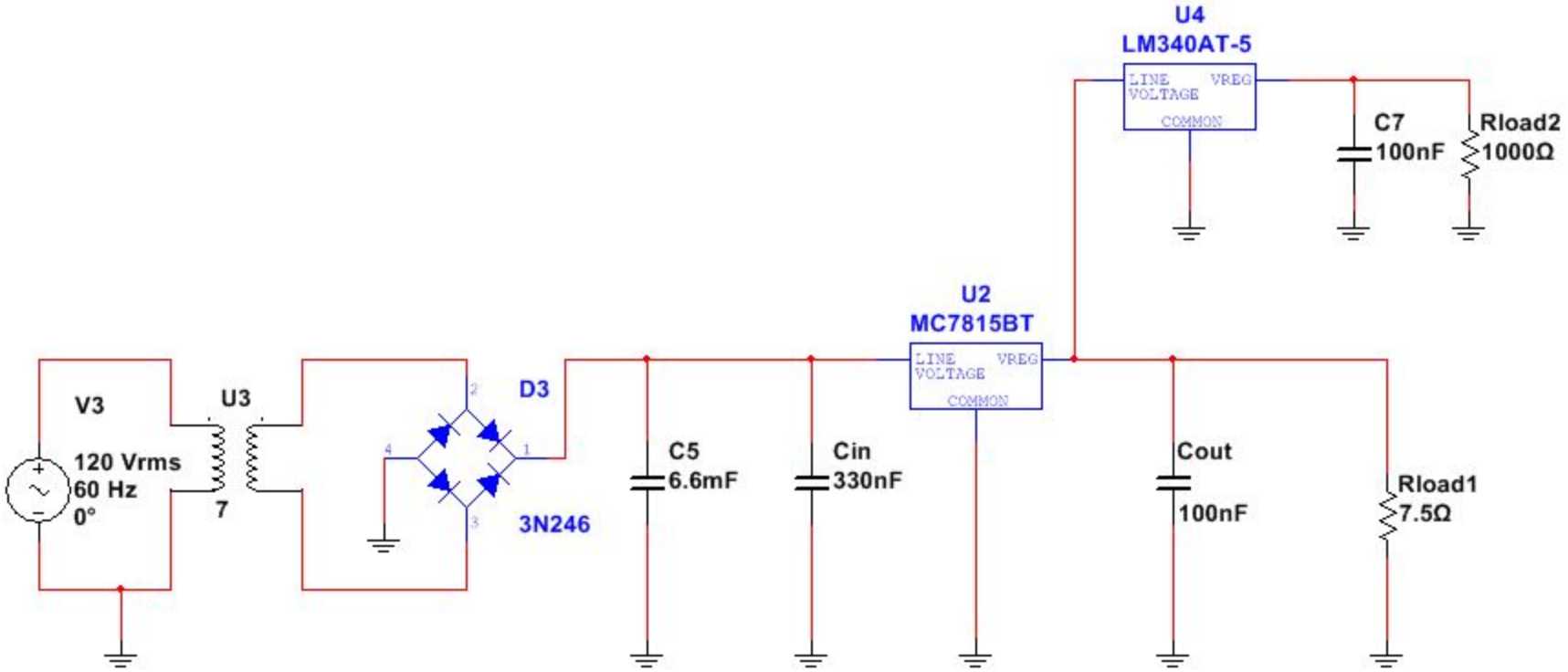
Universal Power Supply - Overview

- Linear Regulating Power Supply
- Capable of supplying 30 Watts
- Two voltage levels
- Low noise and low ripple

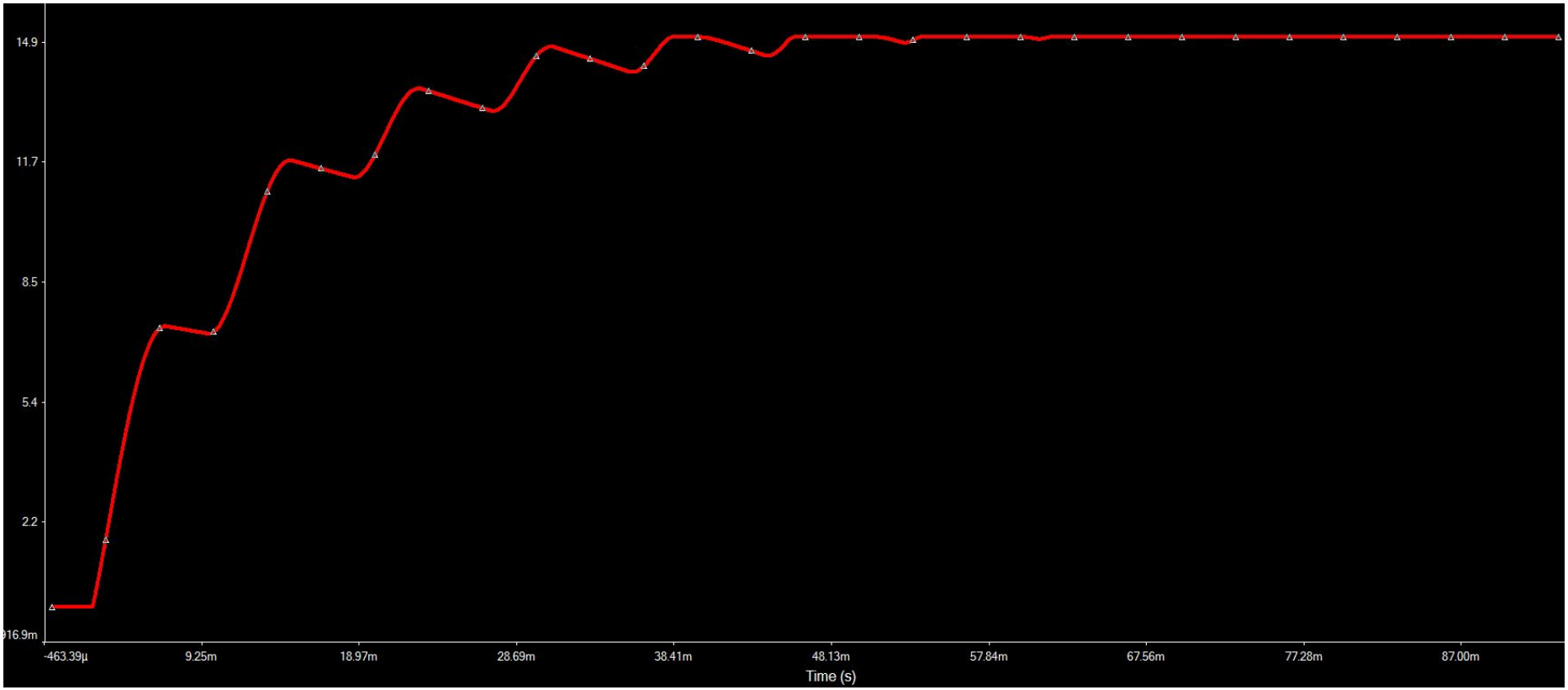
Universal Power Supply - Block Diagram



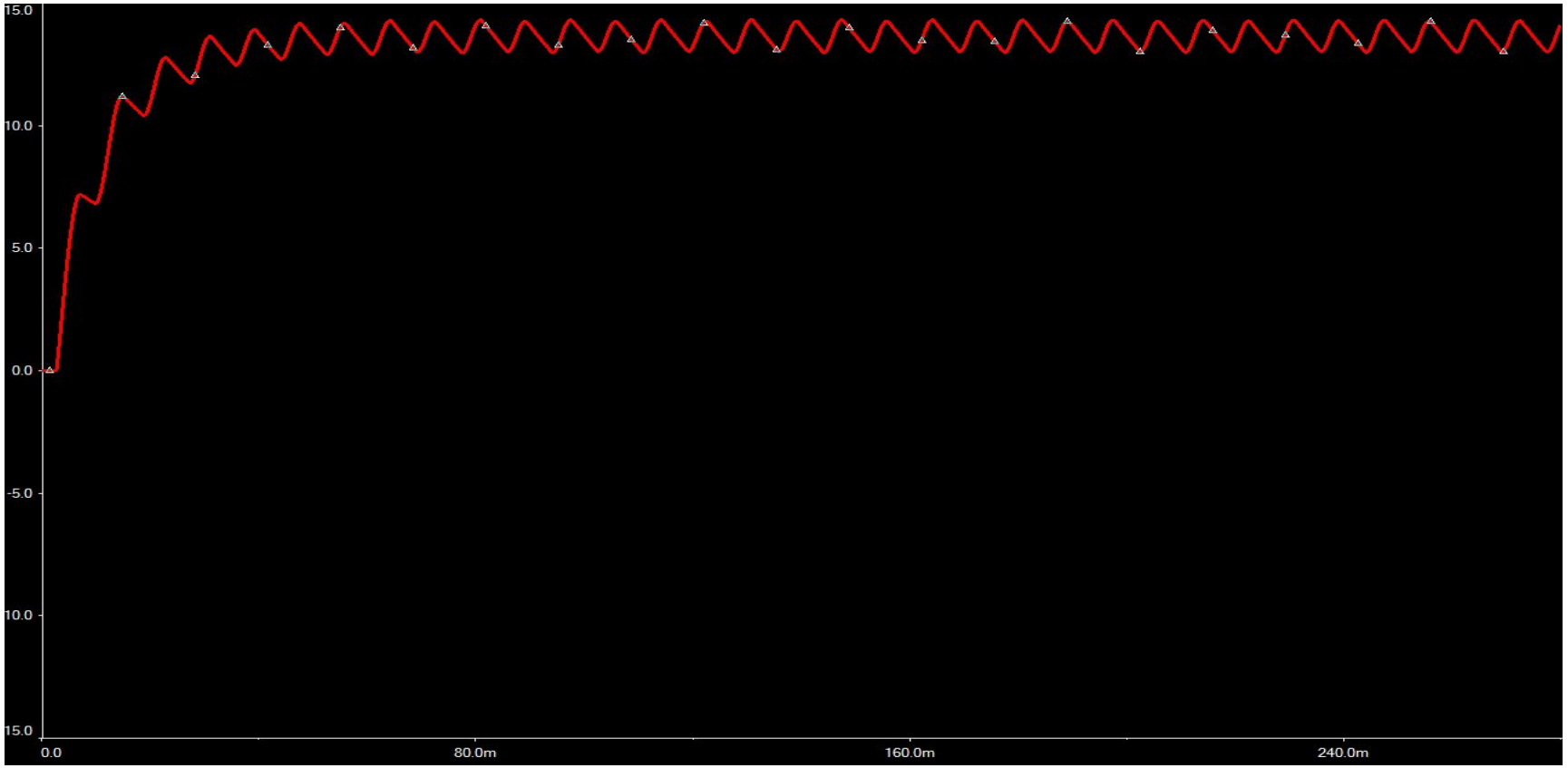
Universal Power Supply - Circuit Diagram



Voltage Ripple at 1 Ampere



Voltage Ripple at 2 Amperes (Max Current)

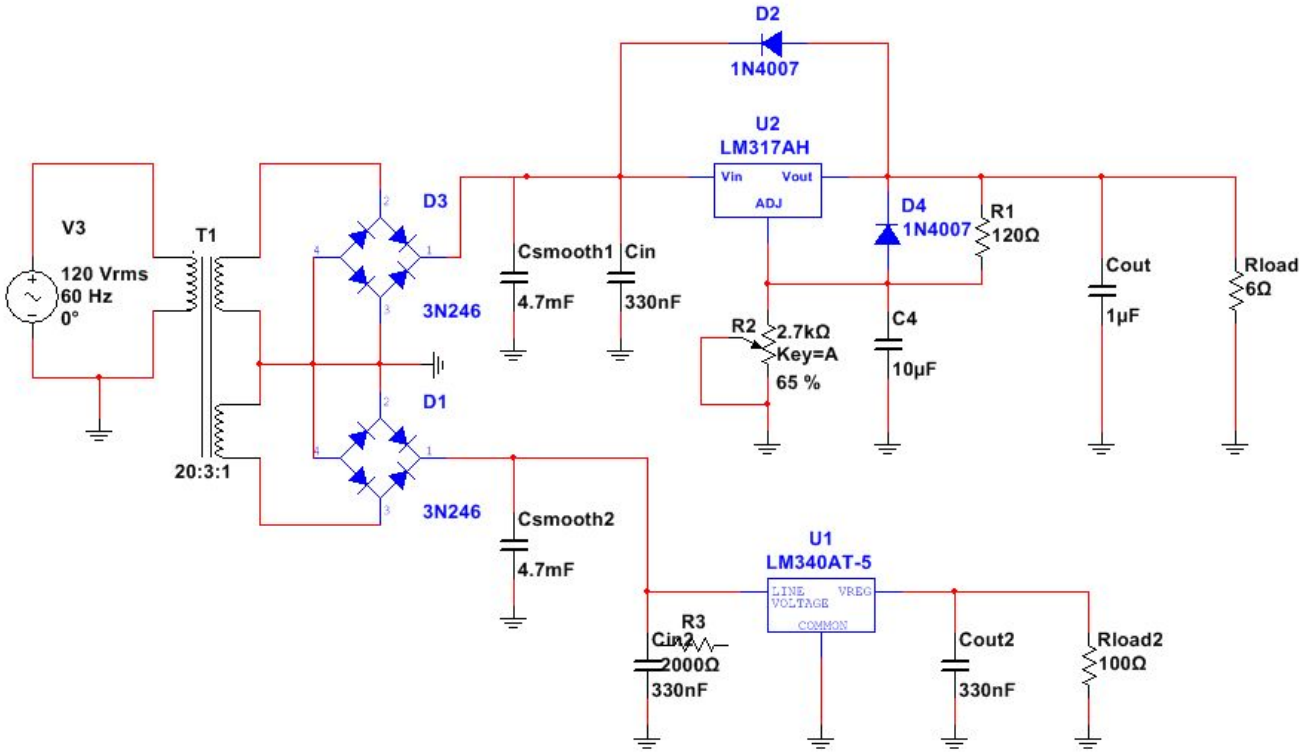


Vripple = 1.2 Vpk-pk @ 120Hz

Universal Power Supply - Present Issues

- Transformer primary resistance is 64 ohms, creates a large voltage drop as current increases thereby limiting the transformable AC voltage
- Series voltage regulator connection increases the current passing through LM350 and voltage drop across the LM340 resulting in devices overheating

Universal Power Supply - Circuit Redesign



Pegasus Turret - Overview

- Controls direction of the audio beam
- Controls direction of the camera
- Locks and tracks a specific user

Pegasus Turret - Functionality

- Webcam sends live video to server
- Server sends back updated X & Y position to Arduino
- Arduino turns servos to the Updated position
- Arduino and Soundlazer are powered from Power Supply

Pegasus Turret - Specs

Servo Motors

- Input Voltage: 6V
- Operating Speed: 0.15sec/60deg
- Stall Torque: 3.95 kg-cm
- Controlled by PWM input signal

Webcam

- 3.0 Megapixel
- 1280 x 720 pixels
- RightLight Technology

Arduino Nano

- Input Voltage: 7-12V
- Logic Level: 5V
- Clock Speed: 16MHz
- PWM Capable

Pegasus Turret - Specs

- Serial connection between Arduino and TX1
- Serial connection between Webcam and TX1
- Rotation angles:
  - In X direction: 0 - 180
  - In Y direction: 0 - 130
- PWM signal from arduino controls the servos
- Used library: “Servo.h”: Allows easy communication between the Arduino and the servos

Pegasus Turret - What's left to be done?

- Design and 3D print a turret chassis
- Design and 3D print new soundlazer mount

FDR Deliverables

Istvan:

- Upgrade UI, increased error handling and multithreading

Trevor:

- Rebuild power supply with new design using a PCB
- Design and print Turret chassis

Keith:

- Software optimizations
- PID Tuning
- Generic Classifier

Prototype Demo

Questions?...


UNIVERSITY OF MASSACHUSETTS
AMHERST 1863
ENSE PETIT PLACIDAM
SUB LIBERTATE QUIETEM