UMassAmherst

Team Turn Up

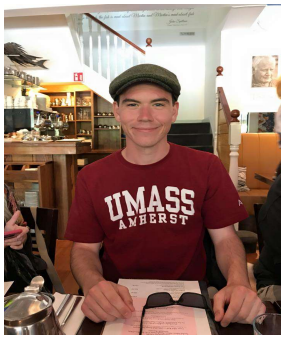
March 4, 2019

Department of Electrical and Computer Engineering

Advisor: Professor Pishro-Nik

Team Members

Nicholas Kafasis
CSE



Harold Healy
EE

Ryan Walsh
CSE

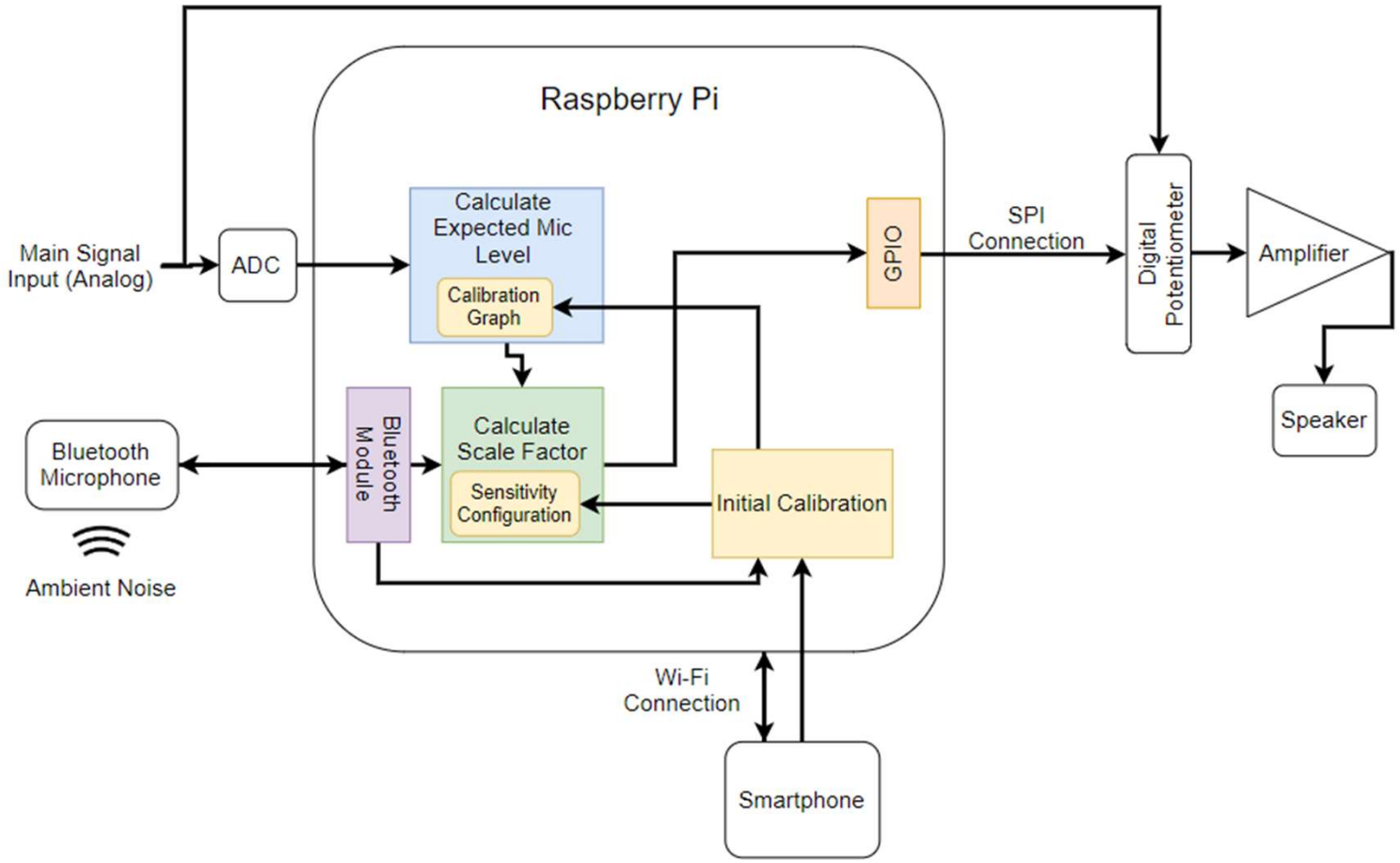
Rahaun Perkins
CSE

Problem Statement

-In environments with dynamic noise levels, frequent volume adjustments for Speakers and TVs are a nuisance.

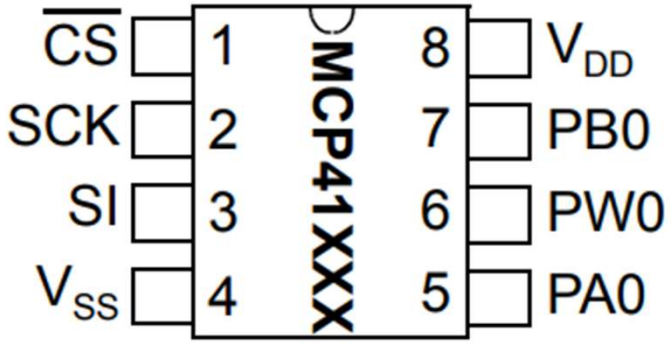
-Our system will be an intermediary device that regulates the volume of audio devices based on levels of ambient noise in a room.

Our Solution: Block Diagram



MCP41010 Digital Potentiometer

- Potentiometer values of 0 Ω to 10 kΩ
- Programming handled by an SPI connection to the Raspberry Pi



LM386n-4 Low Voltage Audio Power Amplifier

- Allows for different gain values based on wiring of circuit
  - Gain range from 20-200
- Circuit wired for gain of 200

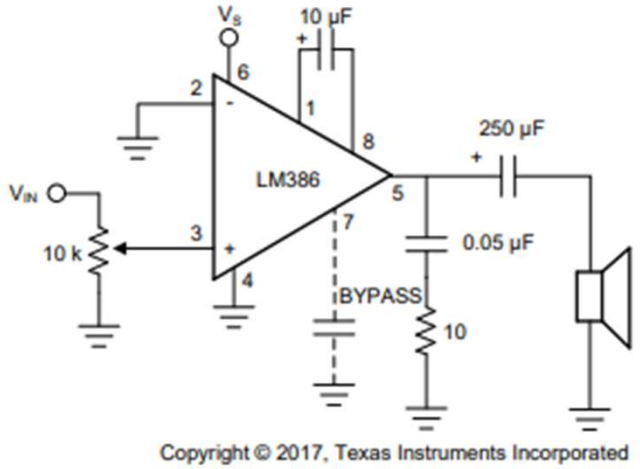
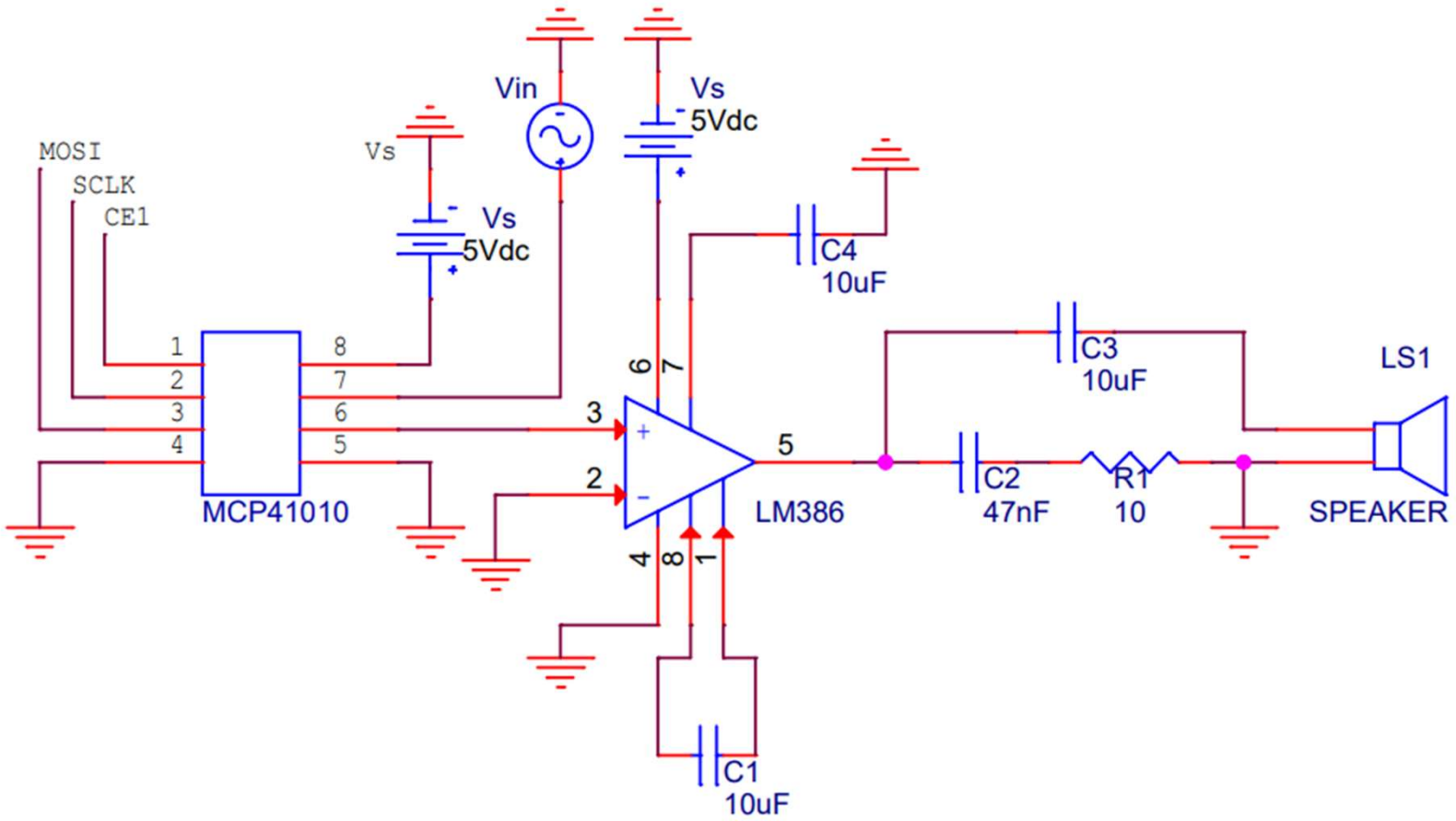


Figure 12. LM386 with Gain = 200

Circuit Diagram



GPIO Integration

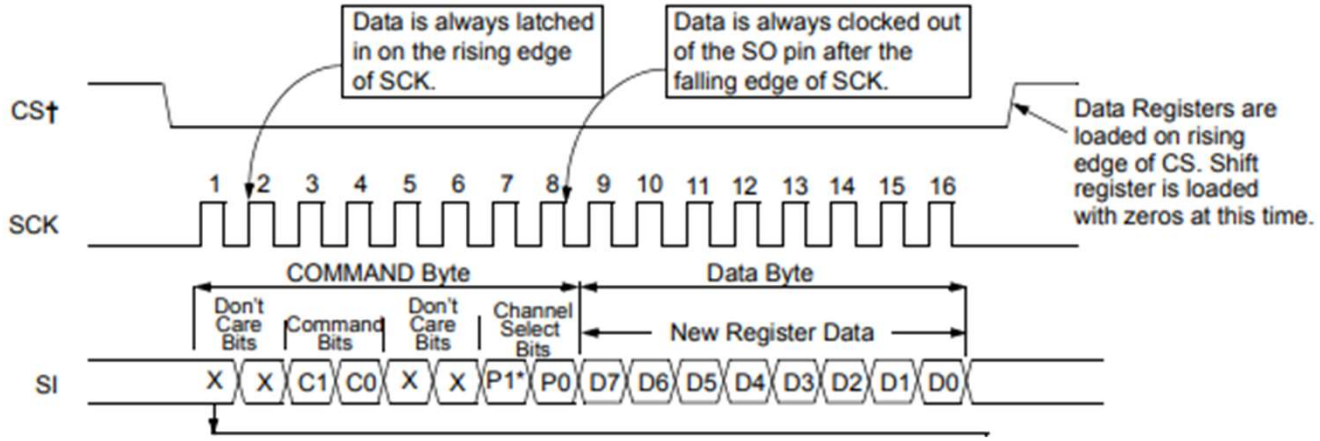
- Raspberry Pi Serial Peripheral Interface (SPI)
  - SCLK - Serial CLocK
  - CE - Chip Enable (often called Chip Select)
  - MOSI - Master Out Slave In
- Sends data to the digital potentiometer
  - 0x00 - 0xFF is equivalent to 0 Ω - 10 kΩ
  - Used to calculate expected scale factor
- Potentiometer used to calculate the input value of the audio amplifier

		Pin no.			
DC Power	3.3V	1	2	5V	DC Power
SDA1, I²C	GPIO 2	3	4	5V	DC Power
SCL1, I²C	GPIO 3	5	6	GND	
GPIO_GCLK	GPIO 4	7	8	GPIO 14	TXD0
	GND	9	10	GPIO 15	RXD0
GPIO_GEN0	GPIO 17	11	12	GPIO 18	GPIO_GEN1
GPIO_GEN2	GPIO 27	13	14	GND	
GPIO_GEN3	GPIO 22	15	16	GPIO 23	GPIO_GEN4
DC Power	3.3V	17	18	GPIO 24	GPIO_GEN5
SPI_MOSI	GPIO 10	19	20	GND	
SPI_MISO	GPIO 9	21	22	GPIO 25	GPIO_GEN6
SPI_CLK	GPIO 11	23	24	GPIO 8	SPI_CE0_N
	GND	25	26	GPIO 7	SPI_CE1_N
I²C ID EEPROM	DNC	27	28	DNC	I²C ID EEPROM
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	
	GND	39	40	GPIO 21	

$$Exp.\,Scale\,Factor = \frac{255 - new\,pot\,value}{255 - base\,pot\,value}$$

GPIO pinout

Command				Channel				Value							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16



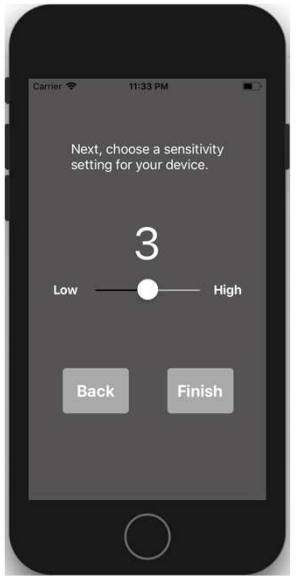
Sensitivity and Threshold

- Sensitivity set by changing size of average mic intensity moving window
  - Range of 1 to 5
  - Sensitivity 1 corresponds to a window size of 4096 samples
    - System can increase from min to max in 22.24s
  - Sensitivity 5 corresponds to a window size of 256 samples
    - System can increase from min to max in 1.39s

- Threshold set at a constant 1.5 (subject to change)
  - We have a range of thresholds available but for simplicity and ease of use, we decided to phase this out of user settings.

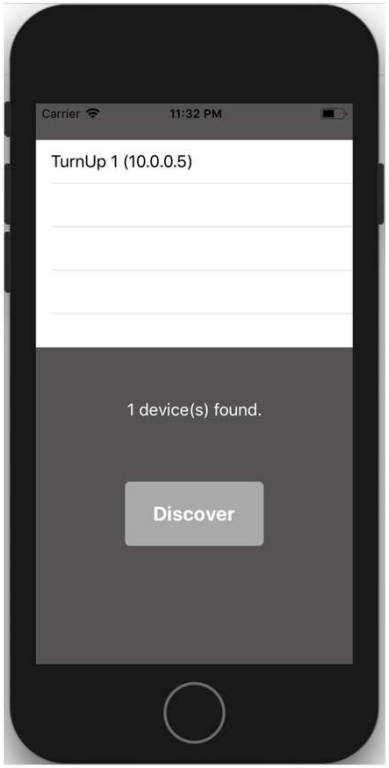
iOS Application: User Interface

- Written in Swift
- iOS app user flow - 5 stages:
  - Welcome Page
  - Device Discovery
  - Calibration
  - User Settings
  - Summary
- Walks user through setting up a device in an effortless fashion



iOS App: iPhone-Device Network Connection

- iPhone discovers and communicates with device via Wi-Fi:
  - No prior knowledge of device IP
- Discovery Process
  - TurnUp device runs UDP server listening for discovery message
  - iPhone sends UDP broadcast discovery message, opens TCP server to listen for response
    - Includes port number that device may respond to
  - TurnUp device responds to iPhone TCP server, opens up TCP server for user connections
    - Response includes device name and a port number that iPhone may connect to
  - iPhone may now connect to this TCP server on the device to send necessary commands
    - E.g. start calibration message, user settings



CDR Deliverables

- ✓ Basic IOS App that will allow user to choose sensitivity and threshold based with sliders
  - Choice of threshold by end user is subject to change upon further review of its necessity
- ✓ System will have the full range of sensitivity(1-10) based on weights of weighted average and size of calculation window
  - Sensitivity levels scaled down to 5, as 10 levels was deemed to precise
- ✓ System will have full range of thresholds
  - System currently allows for the setting of a threshold, but the impact on overall performance is negligible compared to sensitivity

CDR Deliverables

- X Microphone will communicate wirelessly with Pi
  - Currently using a wired connection between the microphone and Raspberry Pi
- ✓ Analog Amplification with Digitally Programmable Gain
  - Amplification being handled by system circuit containing a digitally programmable potentiometer in combination with an analog amplifier.

FPR

- Finish bluetooth integration
- Test on larger speakers
- Integrate PCB
- App UI refinement
- Finish enclosure for Raspberry Pi and PCB

DEMO

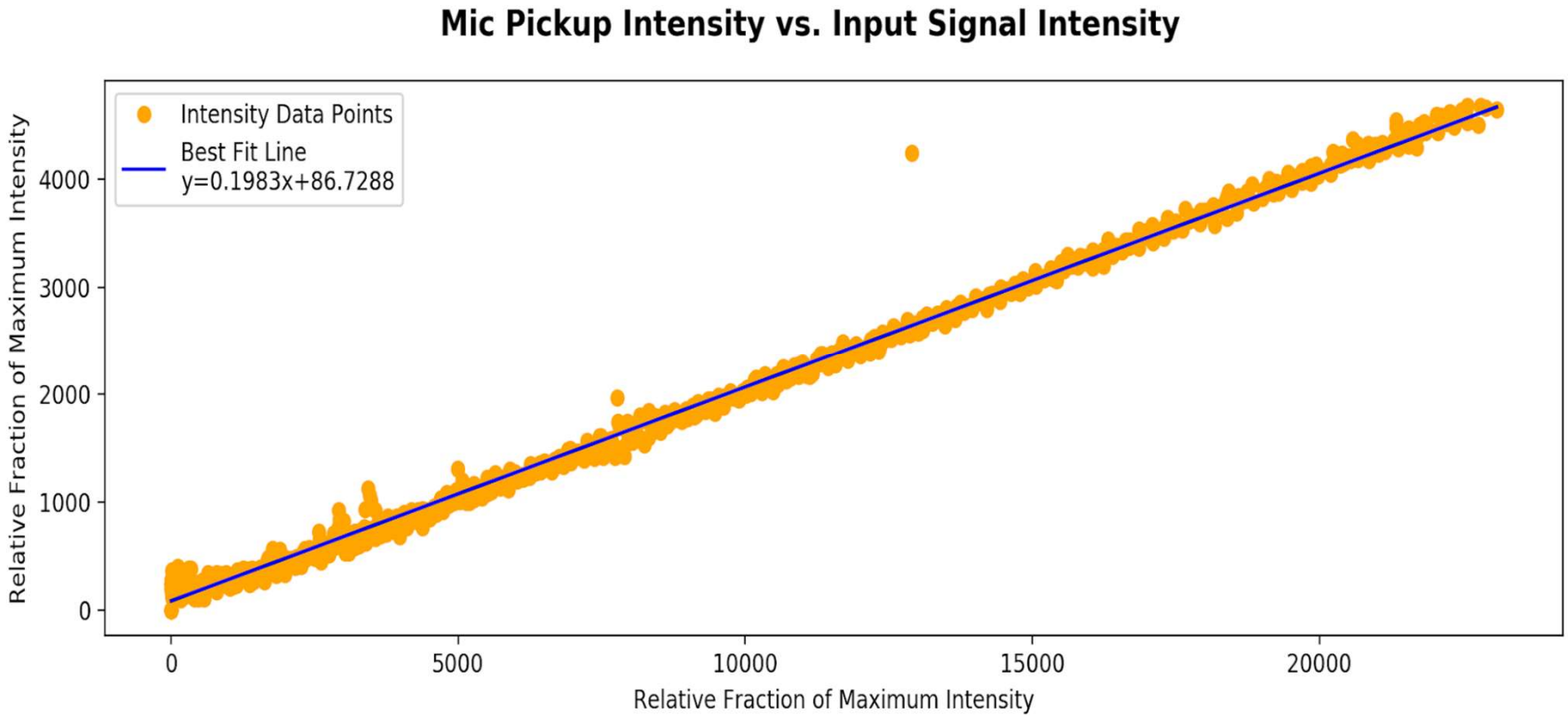
Listen.py

4. Continuously compute the moving averages of mic intensity and expected mic intensity

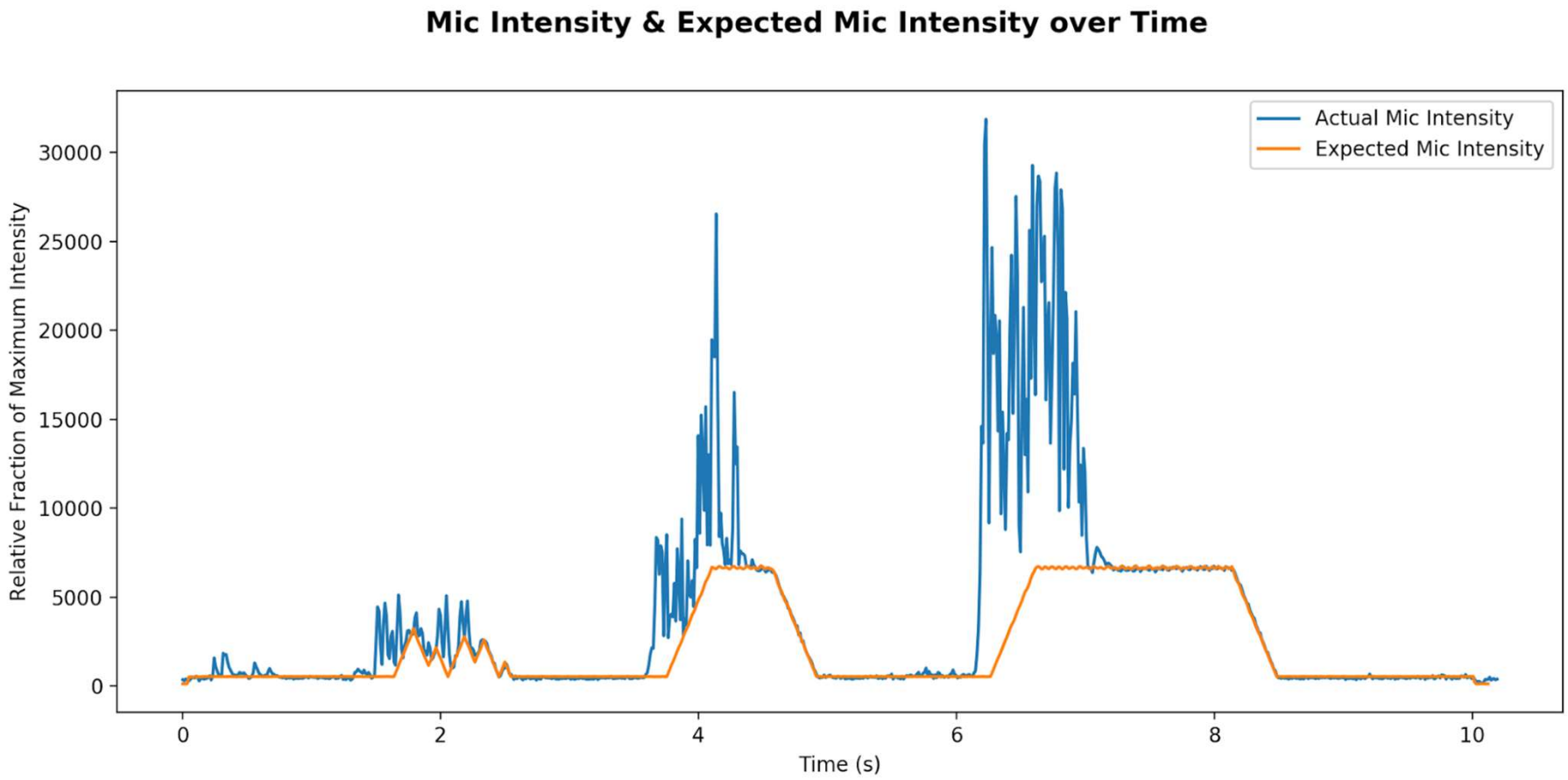
5. Continuously compute ratio of mic intensity average to expected mic intensity average

6. If ratio rises over threshold → increase scale factor;
    Otherwise: If ratio is larger than 1 → decrease scale factor
    Otherwise: keep scale factor at 1

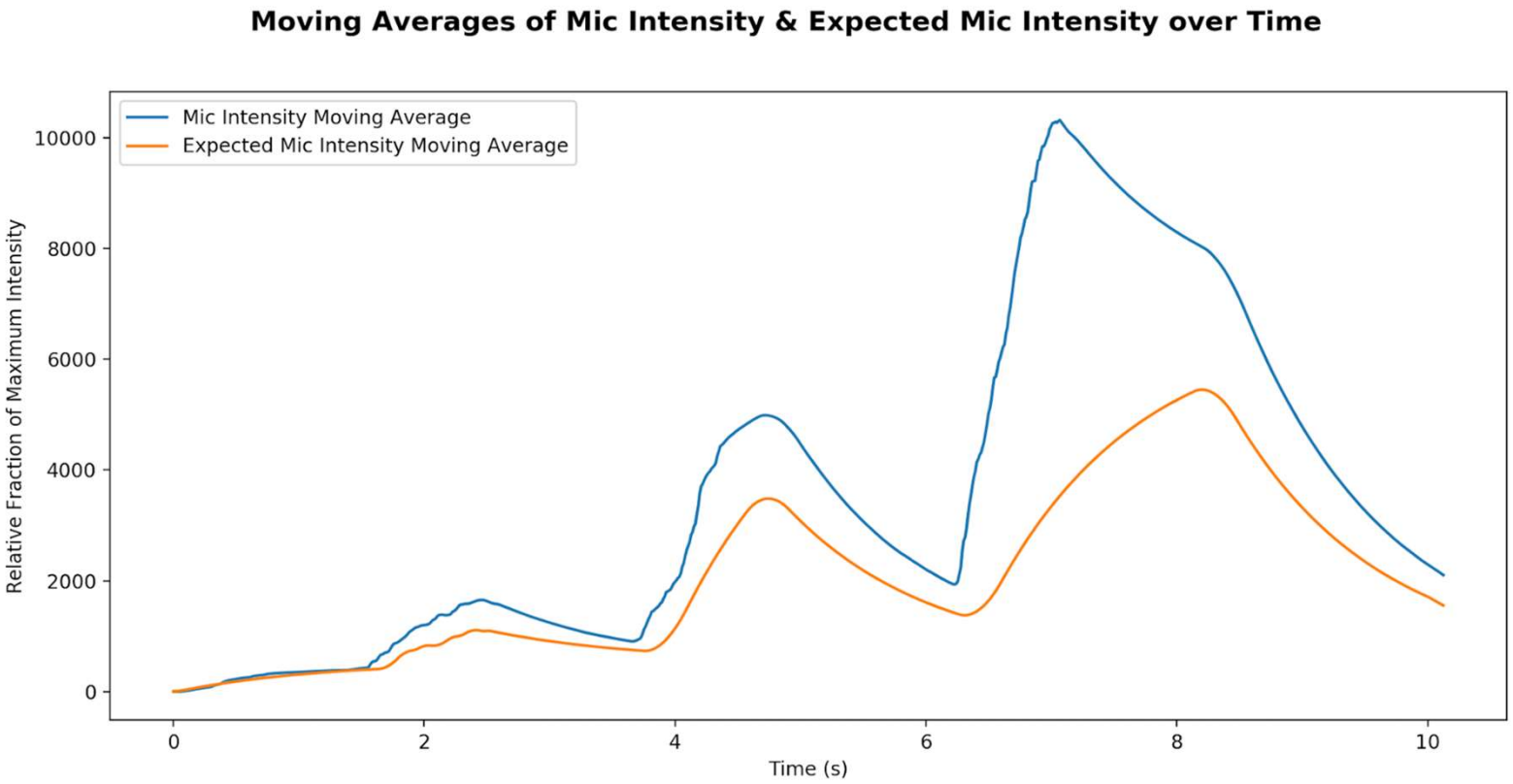
MDR Deliverables: Data Plots - Calibration Graph



MDR Deliverables: Data Plots

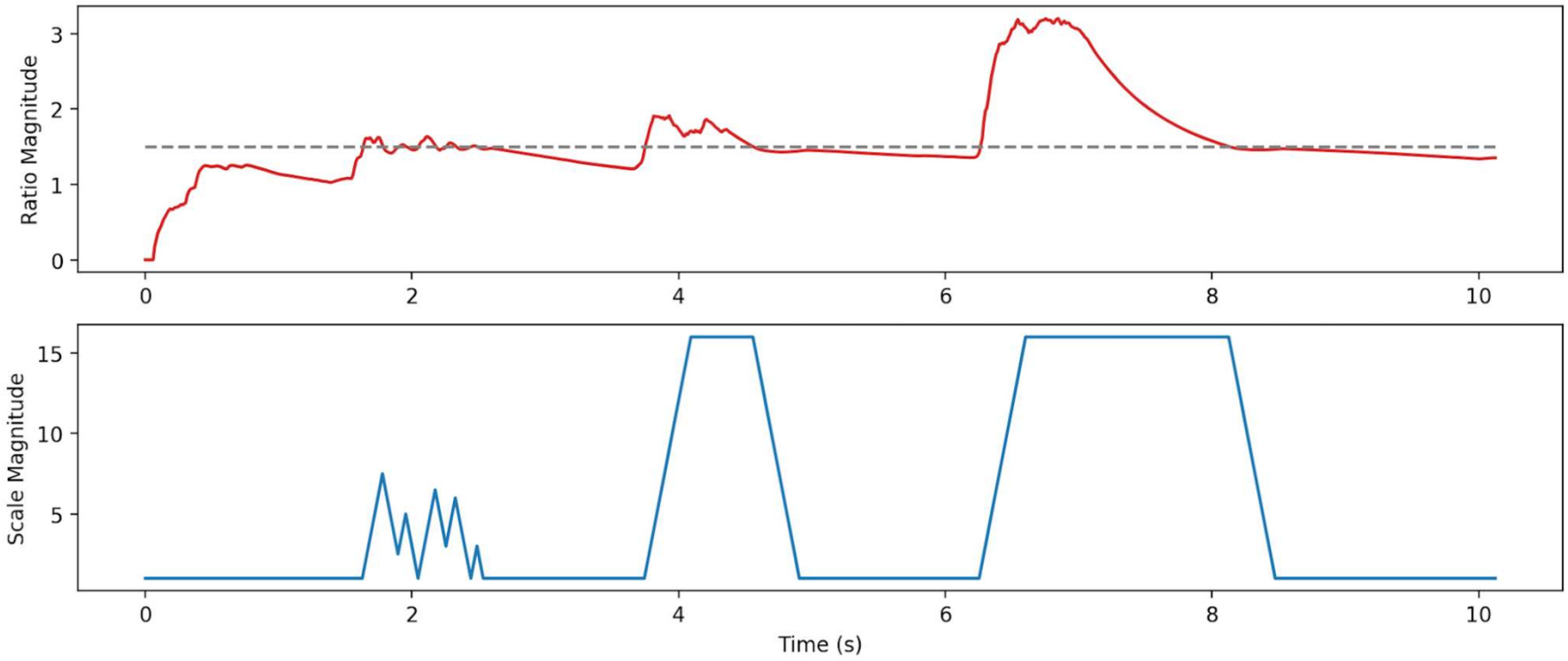


MDR Deliverables: Data Plots



MDR Deliverables: Data Plots

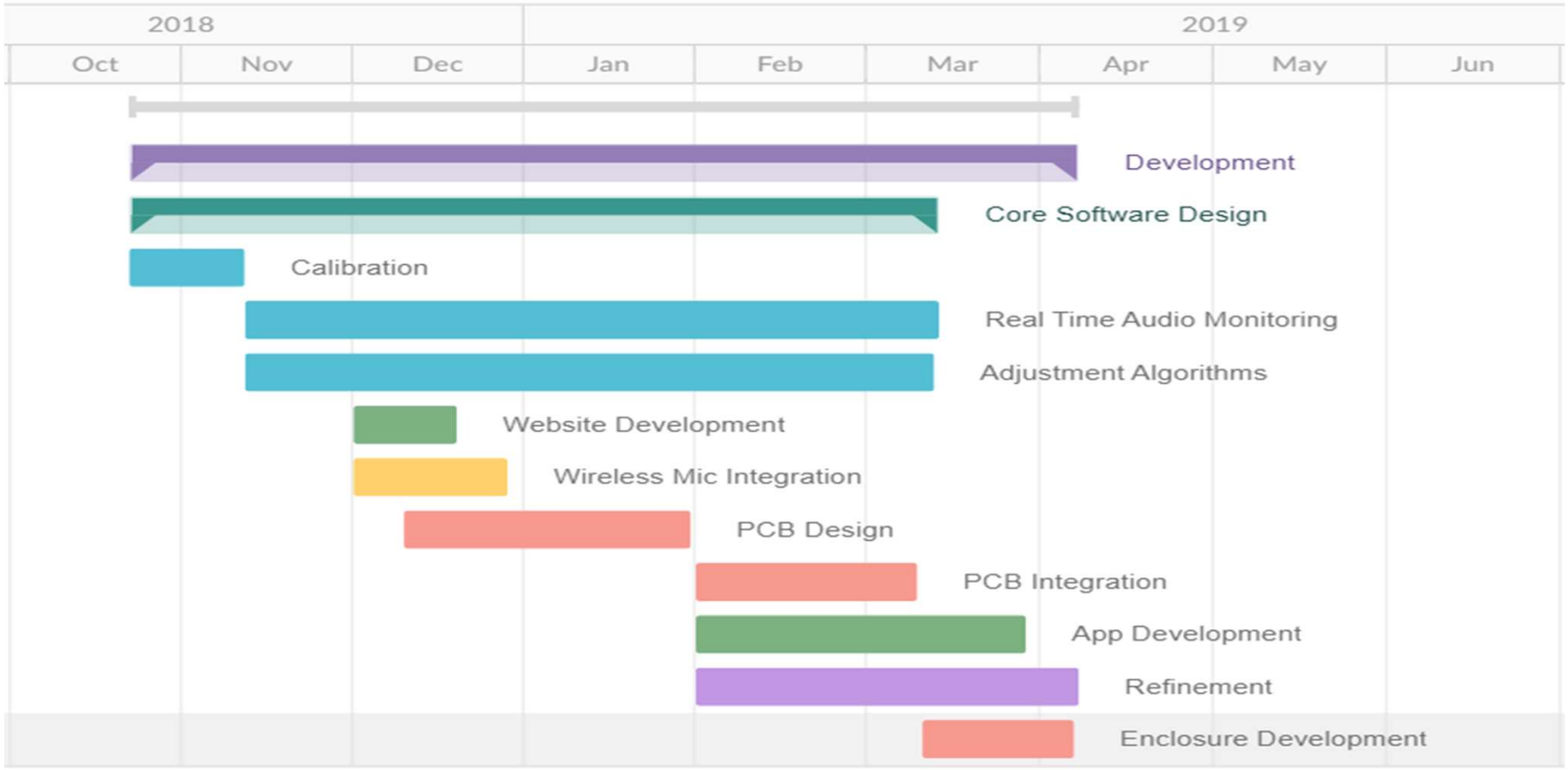
Scale Factor and Avg. Mic Intensity to Avg. Expected Mic Intensity Ratio over Time



Project Breakdown and Responsibilities

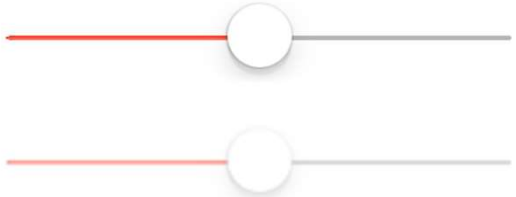
Task name	Start date	End date	Assigned
TurnUp SDP	*10/22/2018*	*04/07/2019*	
Development	**10/22/2018**	**04/07/2019**	
Core Software Design	10/22/2018	03/13/2019	
Calibration	10/22/2018	11/11/2018	R R
Real Time Audio Monitoring	11/12/2018	03/13/2019	N R
Adjustment Algorithms	11/12/2018	03/13/2019	N H
Website Development	12/01/2018	12/19/2018	N Nick Kafasis
Wireless Mic Integration	12/01/2018	12/29/2018	R Rahaun Perkins
PCB Design	12/10/2018	01/31/2019	H Harold Healy
PCB Integration	02/01/2019	03/10/2019	H Harold Healy
App Development	02/01/2019	03/29/2019	R rtwalsh
Refinement	02/01/2019	04/07/2019	R Rahaun Perkins
Enclosure Development	03/11/2019	04/07/2019	H Harold Healy

Gantt Chart


2018
2019
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Development
Core Software Design
Calibration
Real Time Audio Monitoring
Adjustment Algorithms
Website Development
Wireless Mic Integration
PCB Design
PCB Integration
App Development
Refinement
Enclosure Development

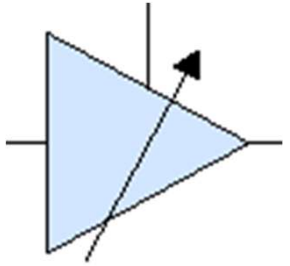
CDR Deliverables

- Basic IOS App that will allow user to choose sensitivity and threshold based with sliders (Ryan)
- System will have the full range of sensitivity(1-10) based on weights of weighted average and size of calculation window (Ryan/Nick)
- System will have full range of thresholds (Expected range of (1.1 - 2) with actual values TBD) (Ryan/Rahaun)



CDR Deliverables

- Microphone will communicate wirelessly with Pi (Rahaun)
- Analog Amplification with Digitally Programmable Gain (Harry)
  - (Instead of Digital Scaling of Samples)



Testing and Validation

Testing: Play a tone through speaker, measure and plot data.

Validation: Graphically show that system increases gain after X seconds of Mic intensity above threshold R

- X is set by sensitivity
  - (expected range is 10 seconds for Min value and close to 0 for Max value)
- R set by threshold setting – (expected range of 1.1-2)
- Consumer satisfaction survey

Challenges Moving Forward

- Integrating the wireless microphone
  - Making sure data is transferred fast enough
- Determining reasonable user preferences through research and polling
- Delivering a clear and concise user interface

Demo

Thank you

Questions?