Voice-Controlled Motor-Driven Pottery Wheel

Critical Design Report

Sebastian Ngo, Christa O'Reilly, Austin Volpe

Client: Dr. Joanna Thomas

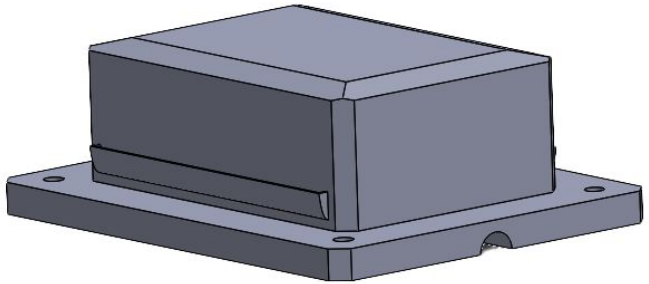
Manager: Dr. Stephen Hill

Introduction

- Client: Dr. Joanna Thomas
- Problem: Lack of accessibility of standard foot-pedal operated motor-driven pottery wheel
- Task: Modify pottery wheel to be voice-controlled

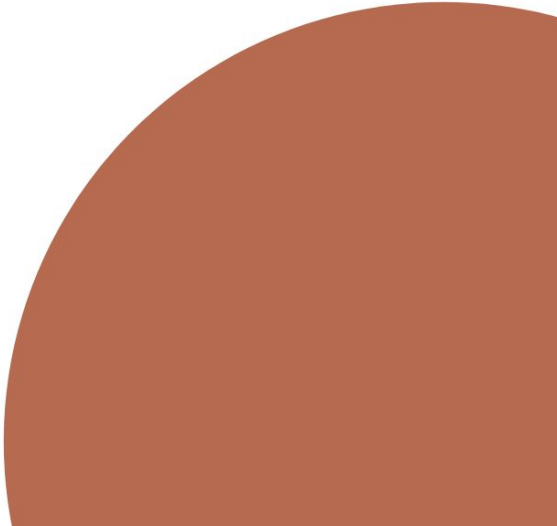
PDR Summary

- Goal: Remove potentiometer and wire motor to microprocessor
- Given several specifications by client
- Analyzed multiple options:
  - Microprocessor - Arduino Nano 33 BLE Sense
  - Housing - 3D Printed PLA
- Initial Expected Cost: $214 per unit



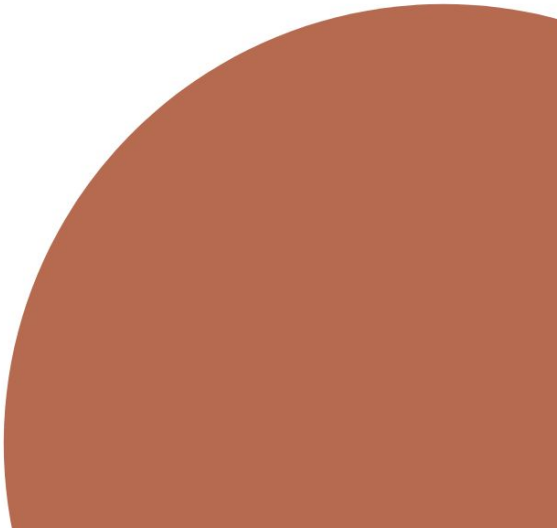
Specifications

- Voice - controlled
- 10 distinct speeds
- Respond to a set of commands
- Programmable name
- Respond to a voice of 50dB within 3 feet
- LED indicator light



Criteria

- Specifications given to us; however we added some more
- Safe to use
- Time of construction
- Cost to build
- Reliability
- Coding ease



Roles and Responsibility

- Sebastian Ngo
  - Computer Engineer
  - Coding and software
- Christa O’Reilly
  - Biomedical Engineer
  - Hardware and team management
- Austin Volpe
  - Mechanical Engineer
  - Physical modifications and housing design

Design Revision

- Used a Google Dataset that provided an abundant amount of voice samples from random people (1500 for each voice command)
- Curate/Mix samples of background samples to simulate users saying a command in a noisy environment

Design Revision

- Original Design: Implementing a trigger word to allow the microcontroller to listen for commands and have the pottery wheel change up to 10 different speeds.

- Revision: Microcontroller does not have a trigger word, and the wheel can only go up to 4 speeds.

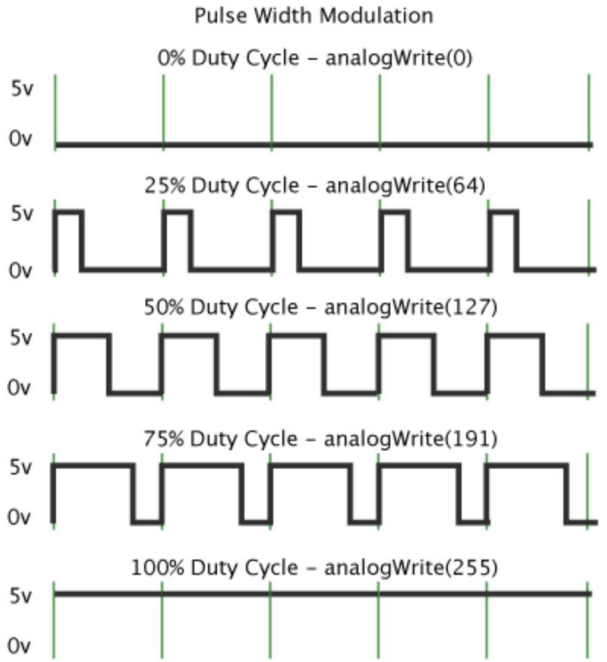
Design Revision

- Imported the neural network model into the arduino IDE and coded the interaction between the microcontroller and pottery wheel

Design Revision

- Used analogWrite() to specify rotation speed

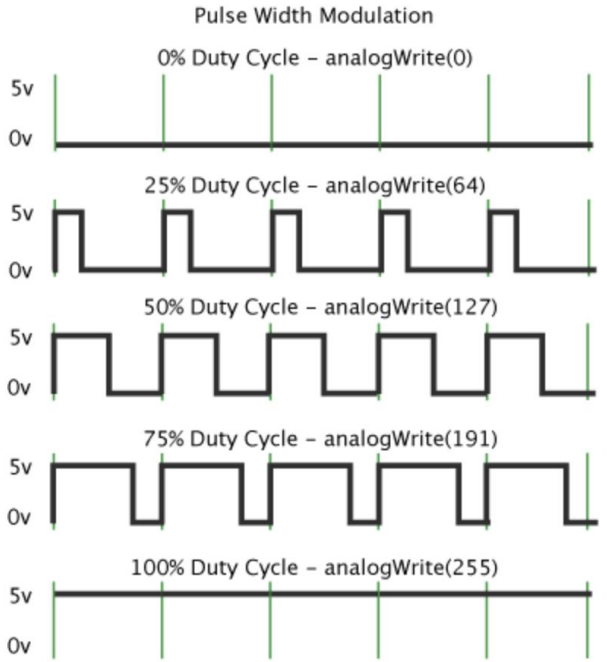
- PWM only works on pins 3,5,6,9,10, therefore we used pin 6 as our output



Design Revision

- Pottery Wheel was staggering in between analogWrite(0-255)

- Solution: Increase the frequency of the pins



Design Revision

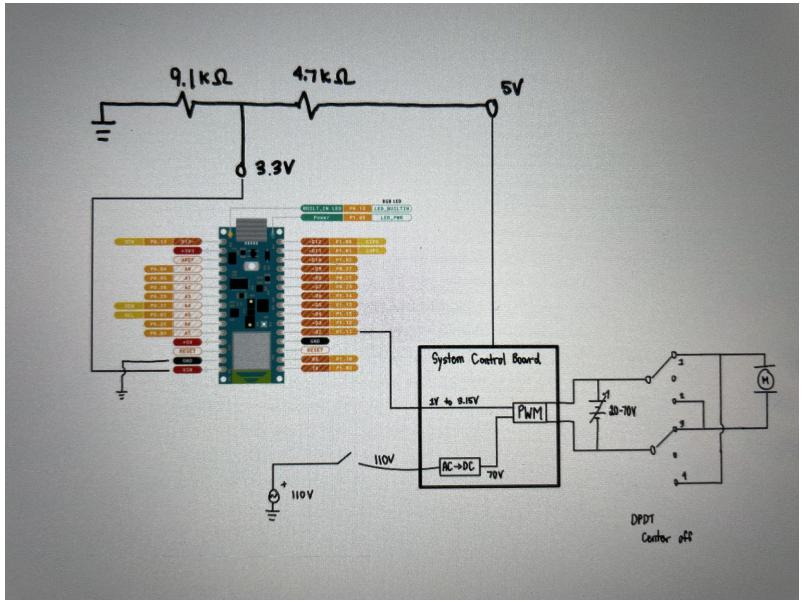
- Removed analogWrite() and used a library called mbed.os which allows us to control the frequency of the designated pin

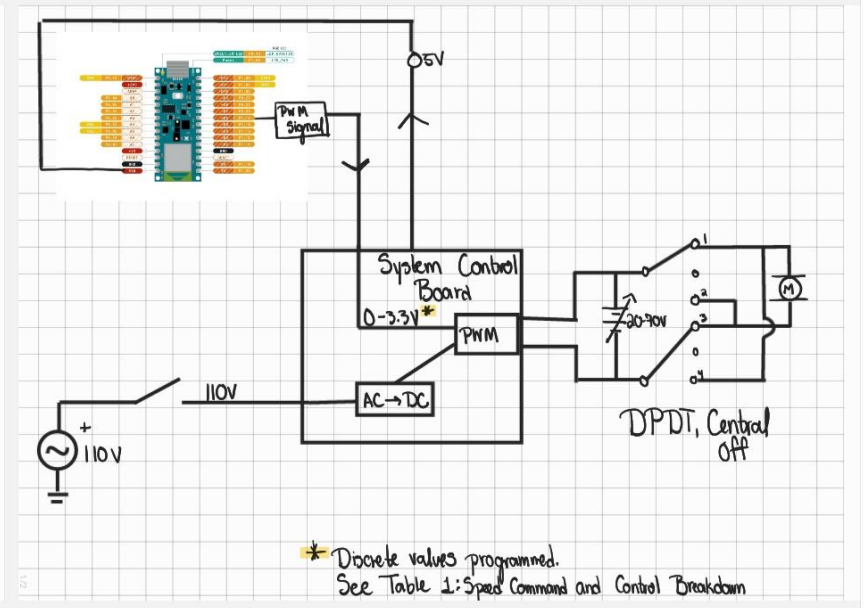
- Increased pin 6 from 500Hz to 25kHz.

```
#include "mbed.h"

#define PWM_PIN_COUNT 1

mbed::PwmOut pwmPin[PWM_PIN_COUNT]
{

  ( digitalPinToPinName( 6 ) )

};

pwmPin[0].period( 0.00004 ); // 25kHz

  pwmPin[0].write( 0.4 );
```

Design Revision

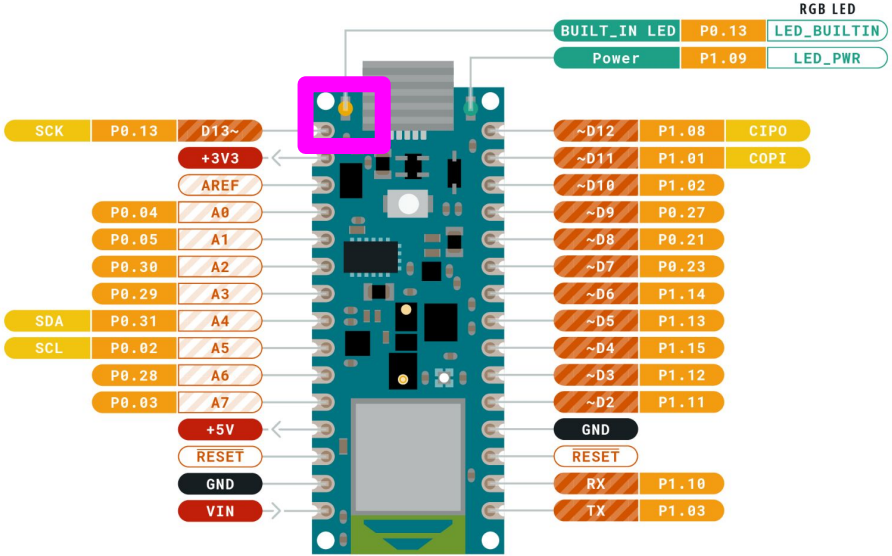
- Final Revision: getting rid of the voltage divider





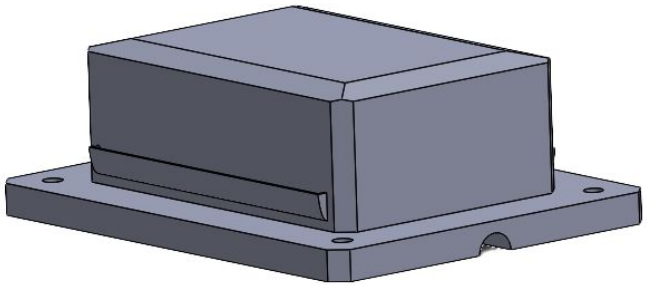
Design Revision

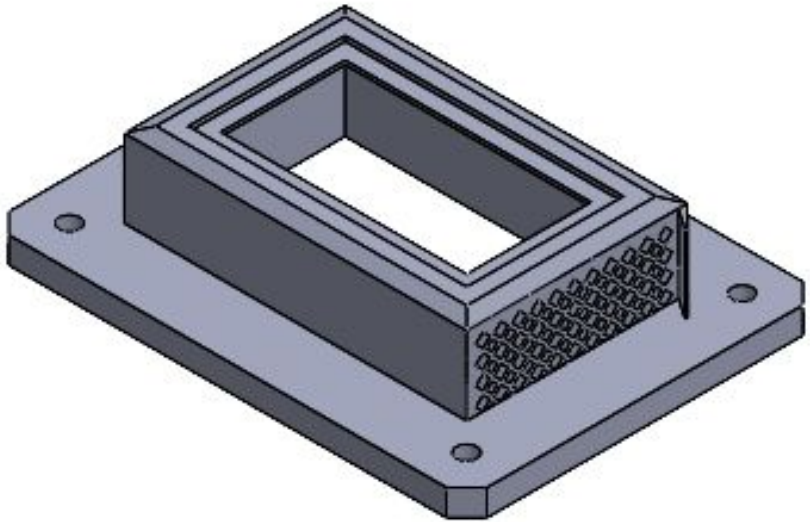
- Revision: Using built-in LED



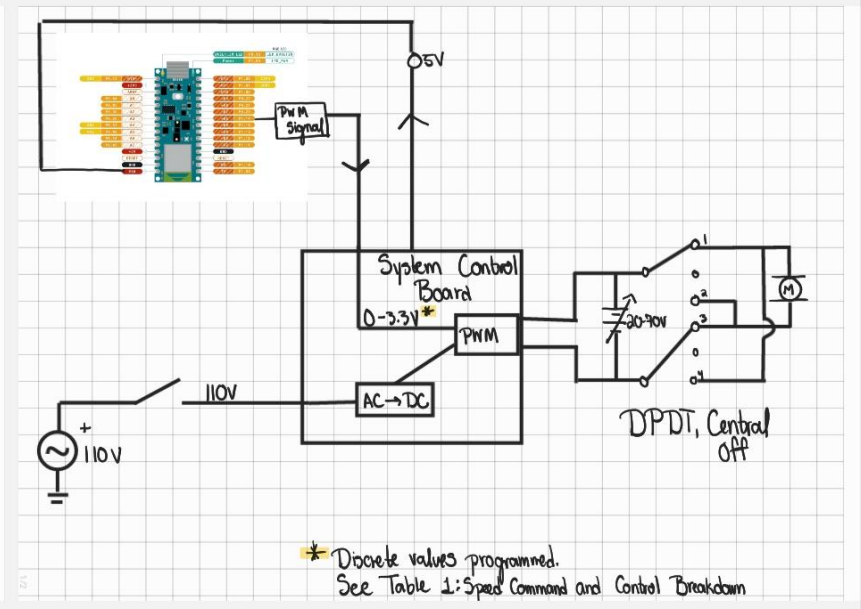
Design Revision - Electrical Housing

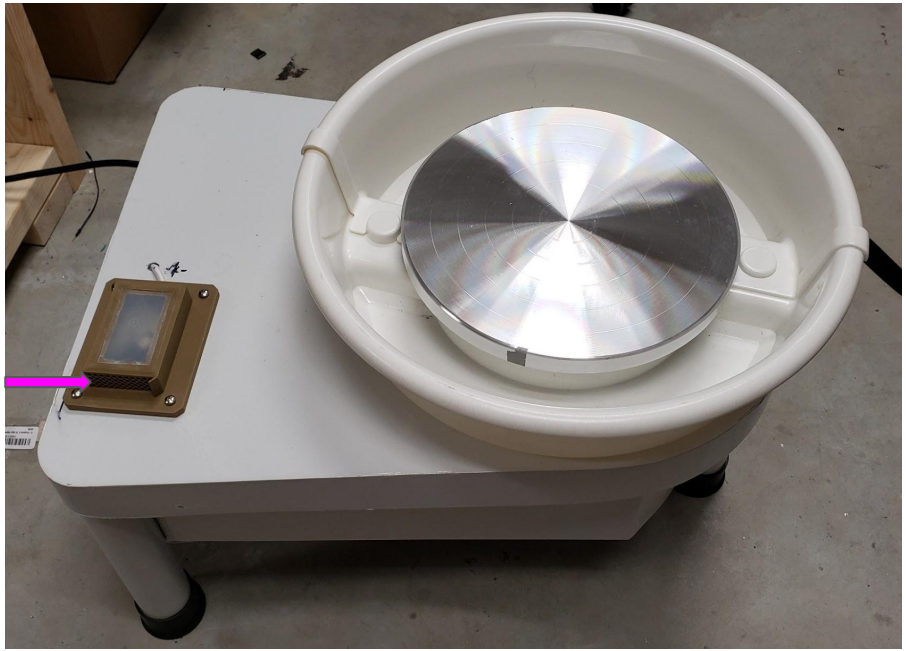
- Original box dimensions (left): 100mm x 70mm x 35mm
- New box dimensions (right): 84.5mm x 54.5mm x 20.75mm
- Translucent LDPE top
- One side open with lattice structure





Final Design





Final Design

https://drive.google.com/file/d/1vBZUzALuUZbJJ7moRylLQpB6ywibruYV/view?usp=sharing

https://drive.google.com/file/d/1kOR277iv4HxQKSw67mto6k9YN1AAgqDK/view?usp=sharing

Test Results

- It Worked!
  - Smooth transitions
  - No stopping or stuttering during the spinning
  - RPM=73.7(volts)-72.7

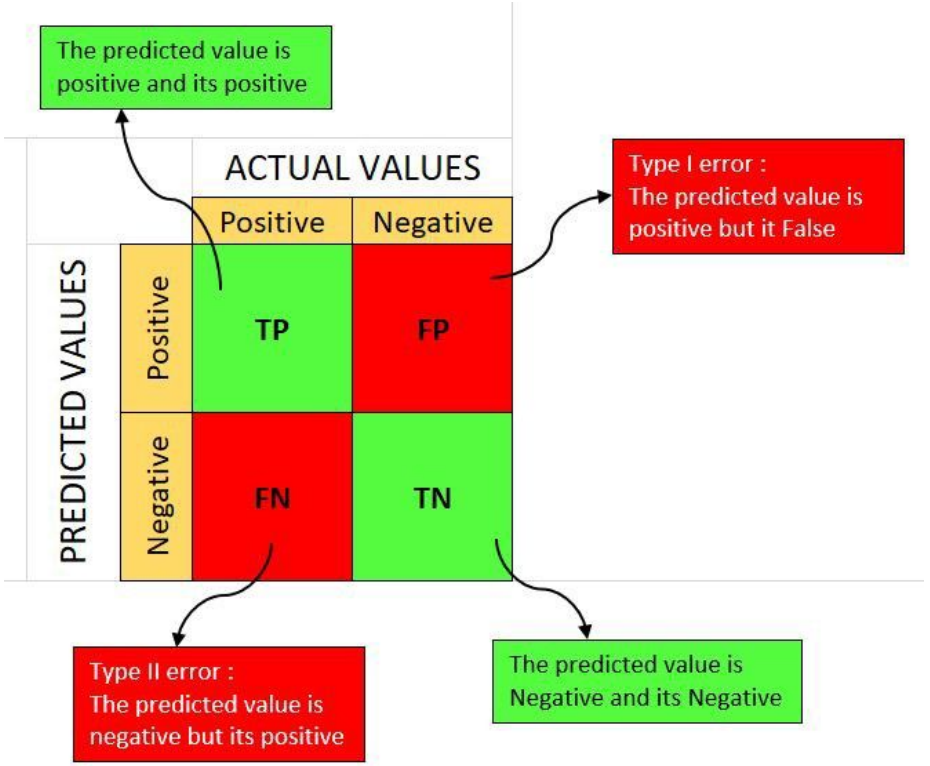
Speed Command	Duty Cycle Value	Voltage	Rotation per Minutes	LED Color
Zero	0	0 Volts	0	Black/Off
One	0.4	1.30 Volts	23.11	Red
Two	0.6	1.96 Volts	71.75	Green
Three	0.8	2.62 Volts	120.4	Blue
Four	1.0	3.30 Volts	170.5	Yellow

Test Results

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



Test Results

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

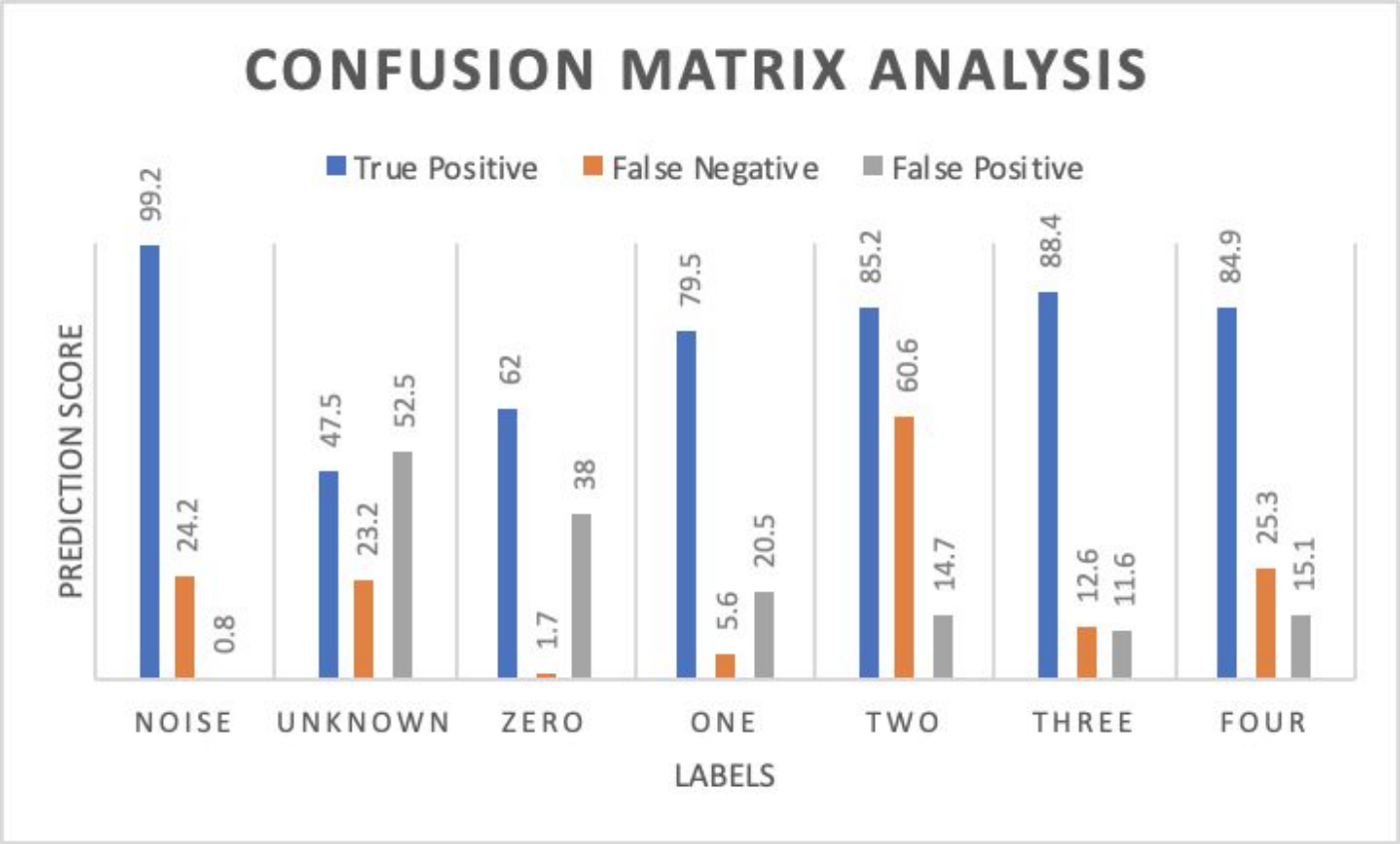
ACCURACY
78.5%

LOSS
0.69

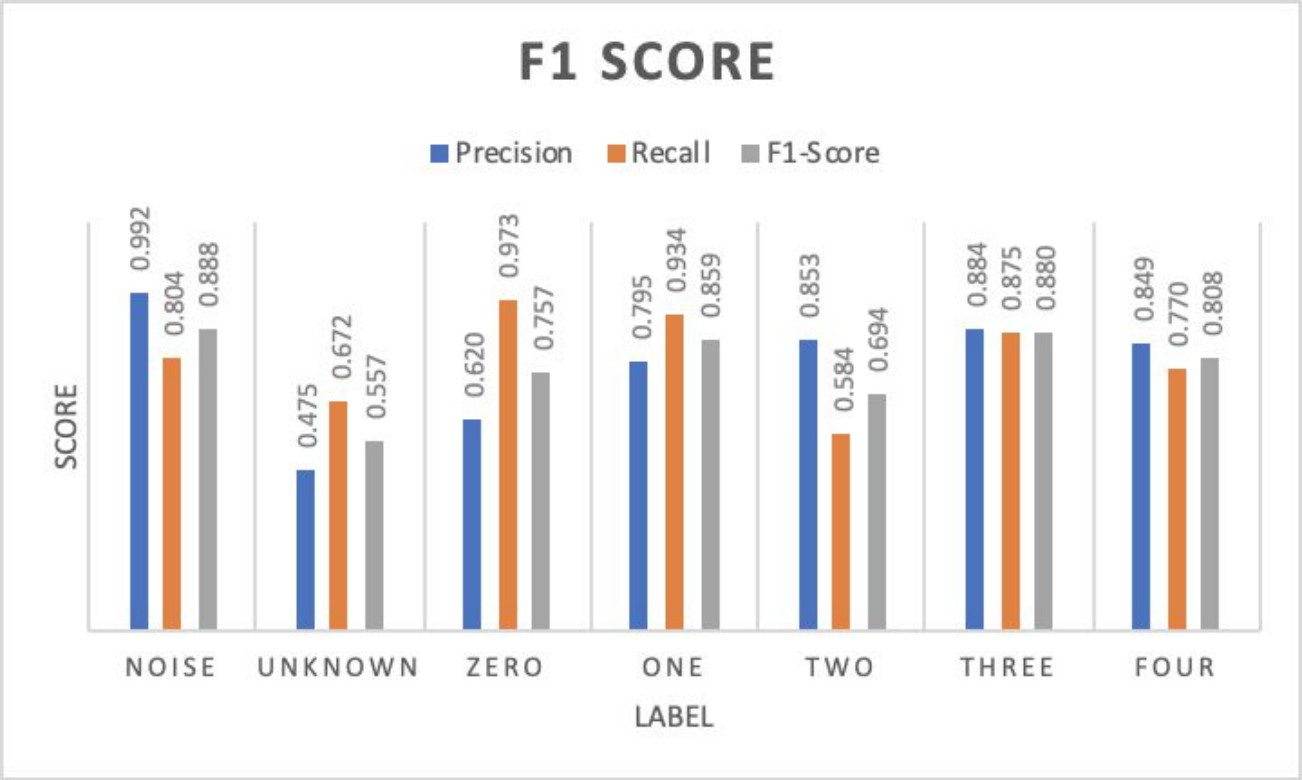
Confusion matrix (validation set)

	_NOISE	_UNKNOWN	FOUR	ONE	THREE	TWO	ZERO
_NOISE	**99.2%**	0.4%	0%	0%	0%	0.4%	0%
_UNKNOWN	8.6%	47.5%	9.5%	4.5%	8.6%	20.4%	0.9%
FOUR	4.2%	3.8%	**84.9%**	1.1%	0%	6.0%	0%
ONE	3.2%	8.2%	7.7%	**79.5%**	0%	1.4%	0%
THREE	1.6%	0.8%	0.8%	0%	**88.4%**	8.0%	0.4%
TWO	4.3%	2.2%	6.1%	0%	1.7%	**85.2%**	0.4%
ZERO	2.3%	7.8%	1.2%	0%	2.3%	24.4%	**62.0%**
F1 SCORE	0.89	0.55	0.82	0.86	0.88	0.68	0.76

Test Results



Test Results

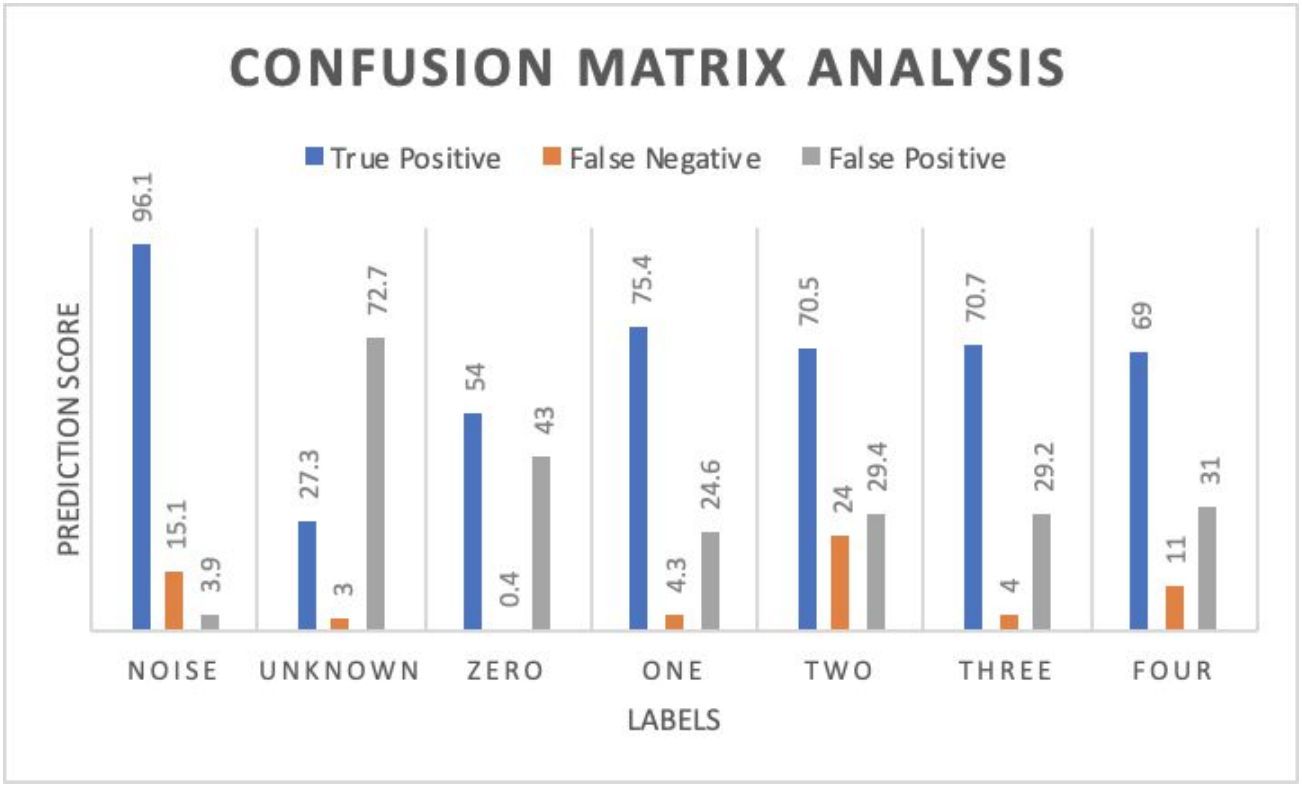


Test Results

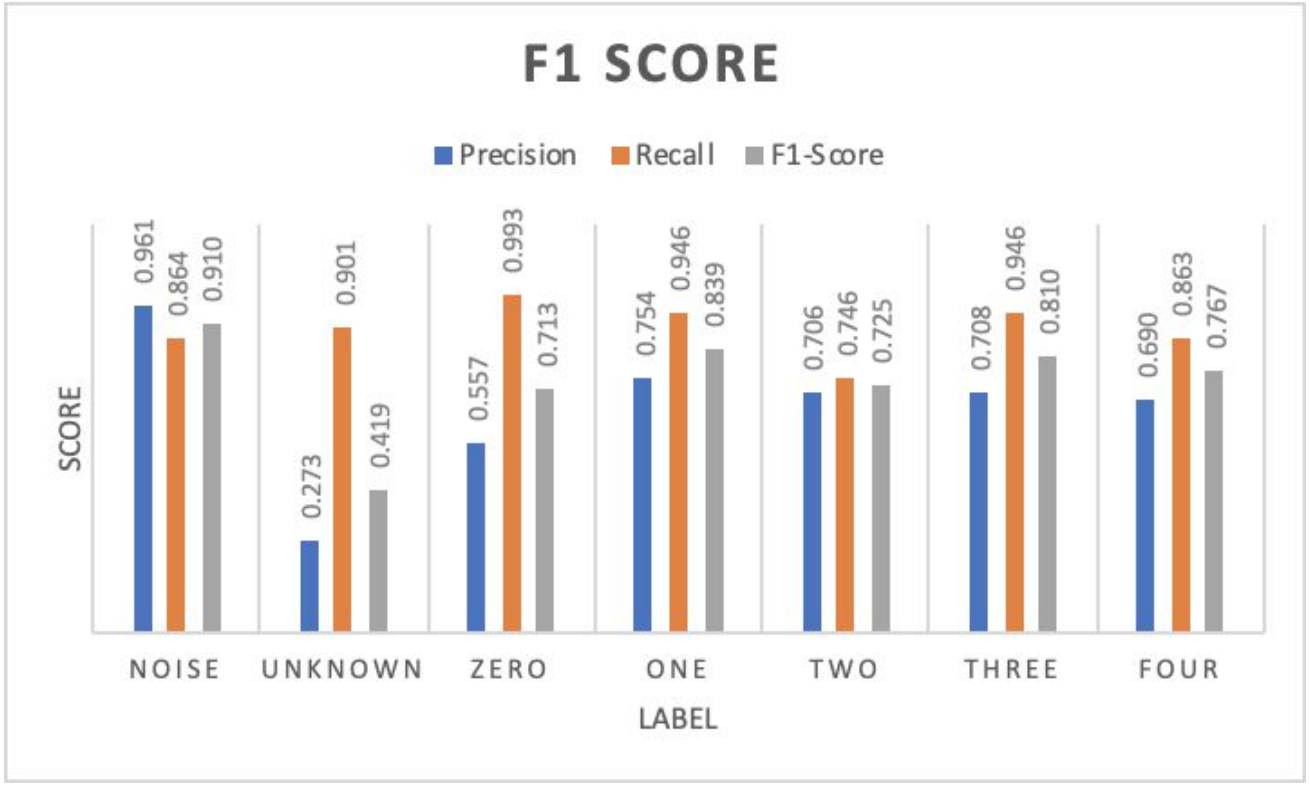
ACCURACY
66.62%

	_NOISE	_UNKNOWN	FOUR	ONE	THREE	TWO	ZERO	UNCERTAIN
_NOISE	96.1%	0%	0%	0%	0%	0%	0%	3.9%
_UNKNOWN	5.5%	27.3%	7.6%	4%	2.9%	1.8%	0.4%	50.5%
FOUR	0.7%	1.0%	69.0%	0.3%	0%	2.4%	0%	26.6%
ONE	3.0%	0.6%	2.1%	75.4%	0%	0%	0%	18.9%
THREE	2.0%	0%	0.3%	0%	70.7%	6.4%	0%	20.5%
TWO	2.1%	0.3%	1.0%	0%	0.7%	70.5%	0%	25.3%
ZERO	1.8%	1.1%	0%	0%	0.4%	13.4%	54.0%	29.3%
F1 SCORE	0.91	0.42	0.77	0.84	0.81	0.73	0.70	

Test Results



Test Results



Cost Breakdown

- Total Cost: 299 dollars.

Item	Total Cost	Individual Cost
Arduino Nano 33	$22.50	N/A
Arduino Nano 33 (post inflation)	$38.85	N/A
9.1K Ohm Resistor (15 pack)	$9.77	$0.65
4.7K Ohm Resistors (10 pack)	$5.76	$0.58
10K Ohm Resistors (100 pack)	$5.00	$0.05
OPA4196	$5.00	N/A
LCD Screen	$12.00	N/A
External Microphone	$9.00	N/A
Pottery Wheel	$163.99	N/A
Bolts (need two packs)	$3.02	$1.51
Washers	$1.00	N/A
Nuts (need two packs)	$1.88	$0.94
PLA	$22.99	N/A

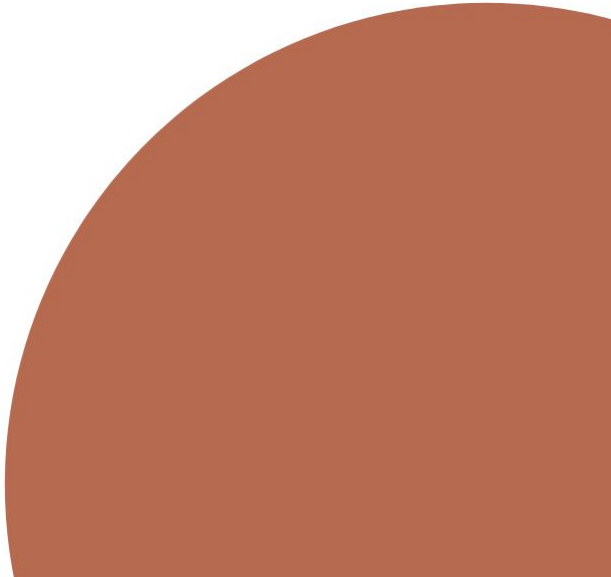
Cost Breakdown

- Project Cost: $231 dollars

Item	Total Cost	Individual Cost
Arduino Nano 33	$38.50	N/A
Pottery Wheel	$163.99	N/A
Bolts (need two packs)	$3.02	$1.51
Washers	$1.00	N/A
Nuts (need two packs)	$1.88	$0.94
PLA	$22.99	N/A

Recommendations

- Different pottery wheel
  - More voltage and power
- Better Microcontroller
  - More neural network training
- Increase budget



Project Summary

- Modified a motor-driven pottery wheel to be voice-controlled
    - Responds to “Zero,” “One,” “Two,” “Three,” “Four”
- Used an Arduino Nano and made housing from 3D printed PLA/LDPE
- Unit Cost: $231
- Final product works well, but we have a few recommendations before proceeding

Questions

Contact Information
Sebastian Ngo:SebastianSy.Vuchi.Ngo@live.mercer.edu
Christa O'Reilly: christa.oreilly@live.mercer.edu
Austin Volpe: austin.m.volpe@live.mercer.edu