Group 20: Funetics

MAUREEN FLINTZ, MEYCHELE CHESLEY, EJ ORTIZ, & DANIEL FALCONER

Project Description

Hands on phonetic learning device

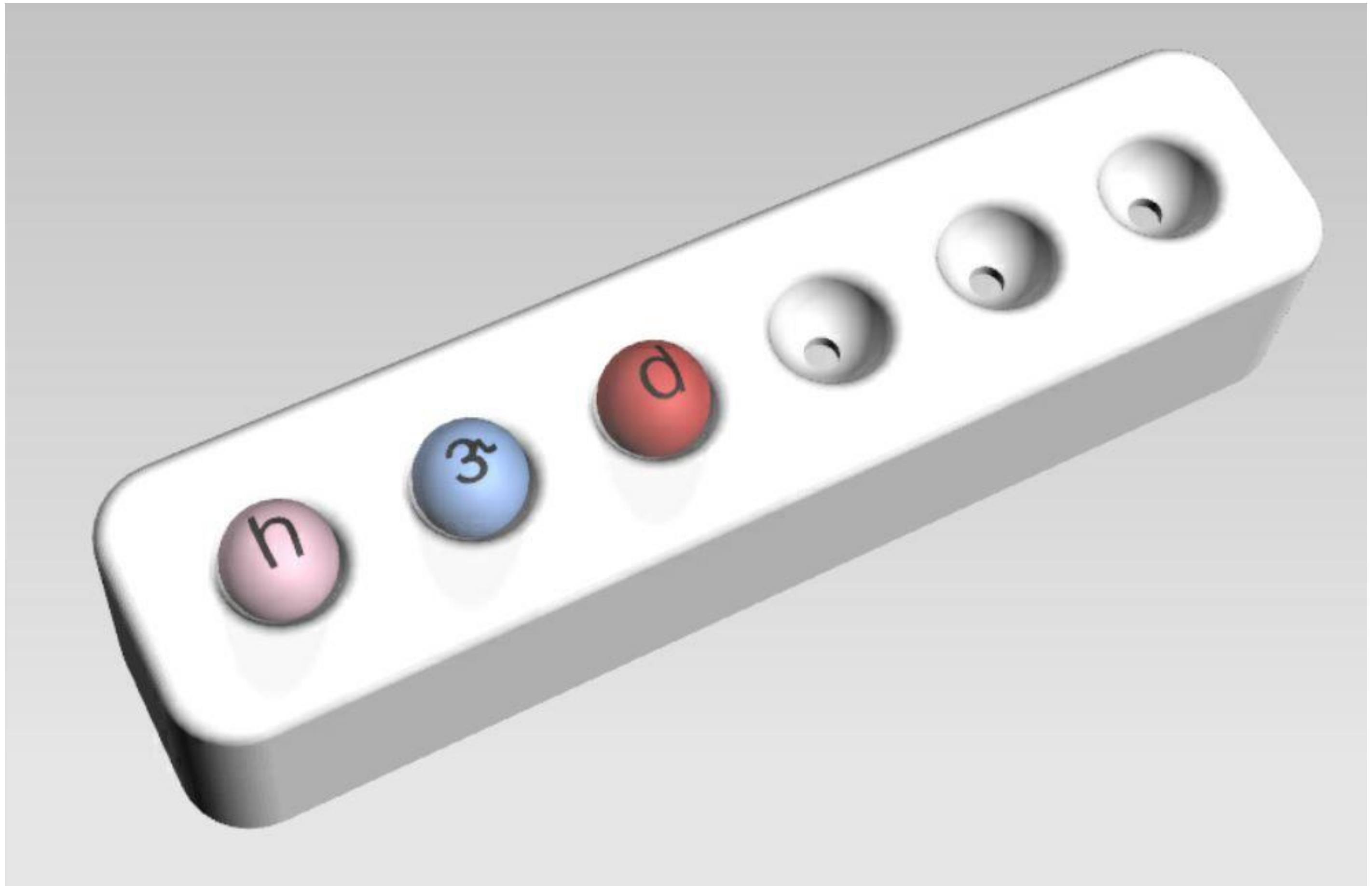
Alternative and Augmentative Communication (AAC) encompasses the different communication methods used to supplement or replace speech

International Phonetic Alphabet (IPA) contains 40 different symbols representing each phonetic sound in the English language.

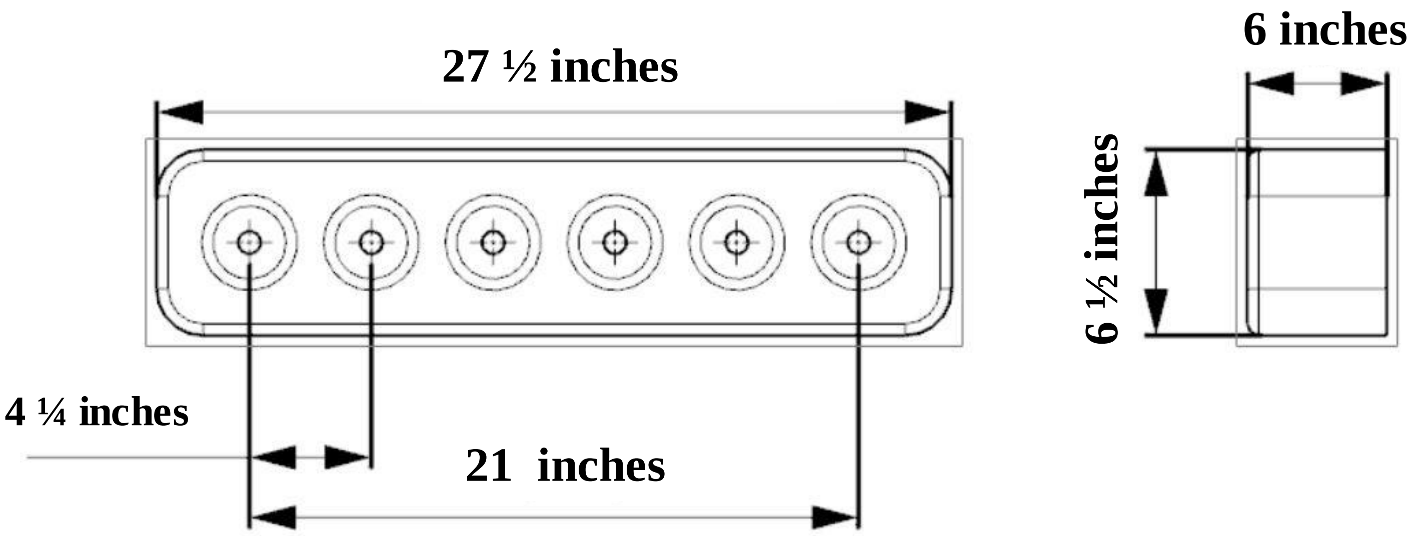
Limited to American English language

heard

h ɝ d



Housing



- Wood material
- Plastic 70 mm spheres changed to 2 1/8 inch wide cards
- Back hinged to access internal electronics

/r/ Phoneme

Keyword	**IPA Transcription**
Ran	**r æ n**
Heard	**h ɝ d**
Her	**h ɝ**
Manner	**m æ n ɚ**
Deer - dear	**d ɪ r**
Ram	**r æ m**
Rook	**r ʊ k**
Sir	**sɝ**
Were	**w ɝ**
Rack	**r æ k**
Work	**w ɝ k**
Hinder	**h ɪ n d ɚ**
Winner	**w ɪ n ɚ**

/s/ and /l/ Phonemes

Keyword	**IPA Transcription**
Hand	**h æ n d**
Man	**m æ n**
Woman	**w ʊ m ə n, w ɪ m ə n**
would – wood	**w ʊ d**
Week	**w i k**
Wind	**w ɪ n d**
Hook	**h ʊ k**
Hood	**H ʊ d**

Other Assorted Sounds

Keyword	**IPA Transcription**
Kiss	**k ɪ s**
Look	**l ʊ k**
see - sea	**s i**
Sand	**s æ n d**
Sack	**s æ k**
Miss	**m ɪ s**

Motivation

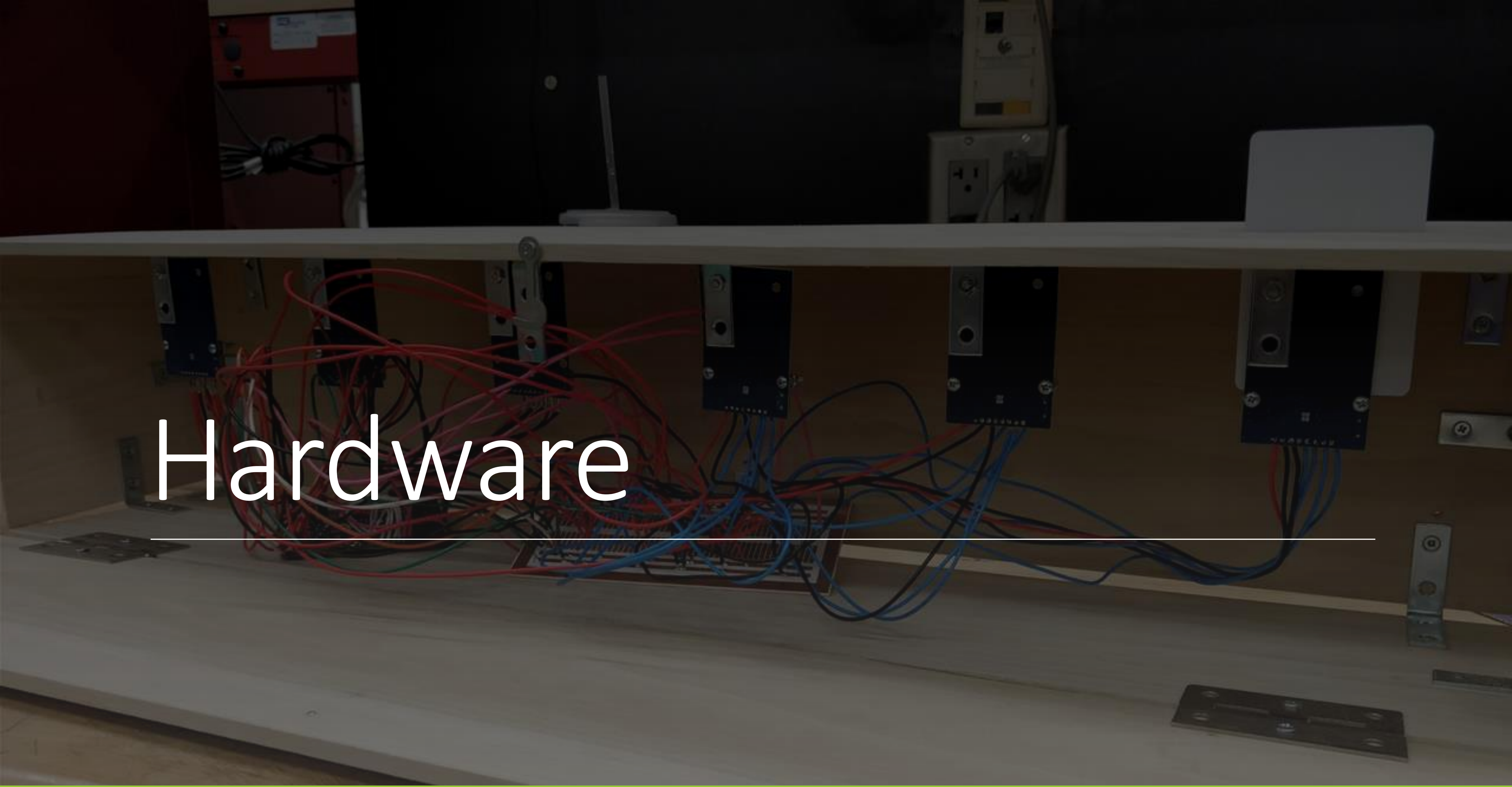
- Young learners
- “Non verbal” communication
- Articulation disorders
- English as a second language ESL
- Fun and interactive learning experience for everyone

Specifications and Requirements

- Audio at 50-60 dBA
- 6-8 ohm impedance
- Signal-to-Noise Ratio (SNR): 70 – 100 dB

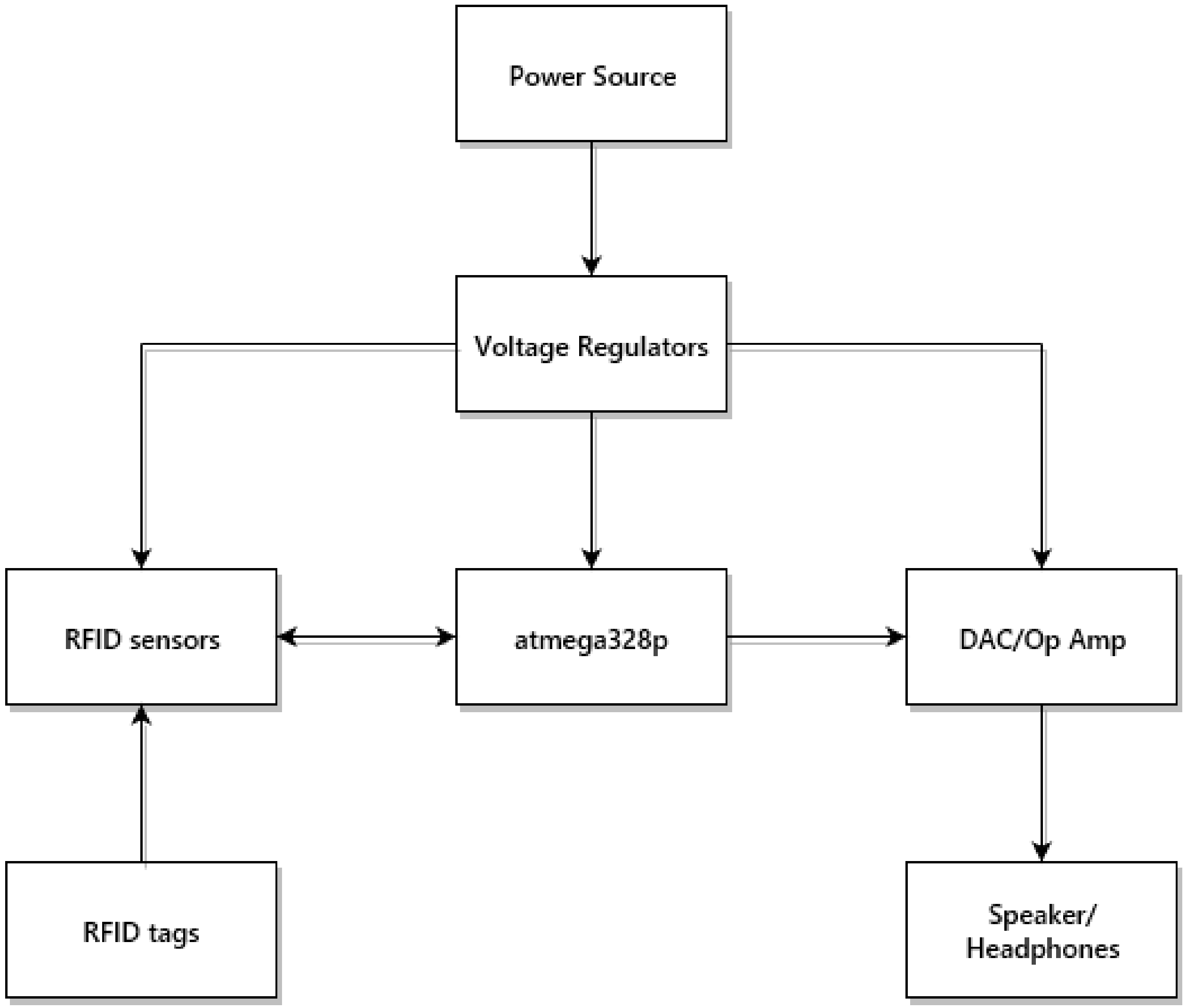
Goals

- Runs on a rechargeable battery
- Correct pronunciation of individual phenomes and words
- Use colored LEDs synchronized with sound outputs
- Housing durability
- Works with variety of card placements



Hardware

Hardware Block Diagram


Power Source

Voltage Regulators

RFID sensors

atmega328p

DAC/Op Amp

RFID tags

Speaker/
Headphones

Microcontroller Choices

MSP430G2

- Clock Speed – 16MHz
- RAM – 512 byes
- Storage – 16 KB
- Digital I/O pins – 16
- $0

ATmega2560
(Arduino Mega)

- Clock Speed – 16MHz
- RAM – 8 KB
- Storage – 256 KB
- Digital I/O pins – 54
- $38.50 (~$12)

ATmega328
(Arduino Uno)

- Clock Speed – 16MHz
- Ram – 2 KB
- Storage – 32 KB
- Digital I/O pins – 14
- Analog I/O pins – 6
- $22.00 (~$2)

Wireless Communications

	RFID	**NFC**	**Bluetooth**
Range	20 ft	10 cm	10 m
Frequencies	120kHz – 150kHZ, 13.56MHz, 433MHz, 902 – 928MHz, 2450MHz – 5800MHz	13.56MHz	2.4GHz
Frequency Standards	Unregulated low frequency, ISM bands including ISO/IEC and FeliCa	ISO/IEC 14443, ISO/IEC 18092, FeliCa	ISM band
Passive Tag Implementations	Yes	Yes	No

MFRC522

Voltage	3.3V
Current	6.5mA
Interface Support	SPI, I2C, UART
Frequency	13.56MHz
Range	5 - 10 cm
Standards Support	ISO/IEC 14443 A/MIFARE, NTAG
Average Cost	$5/unit (low volume)

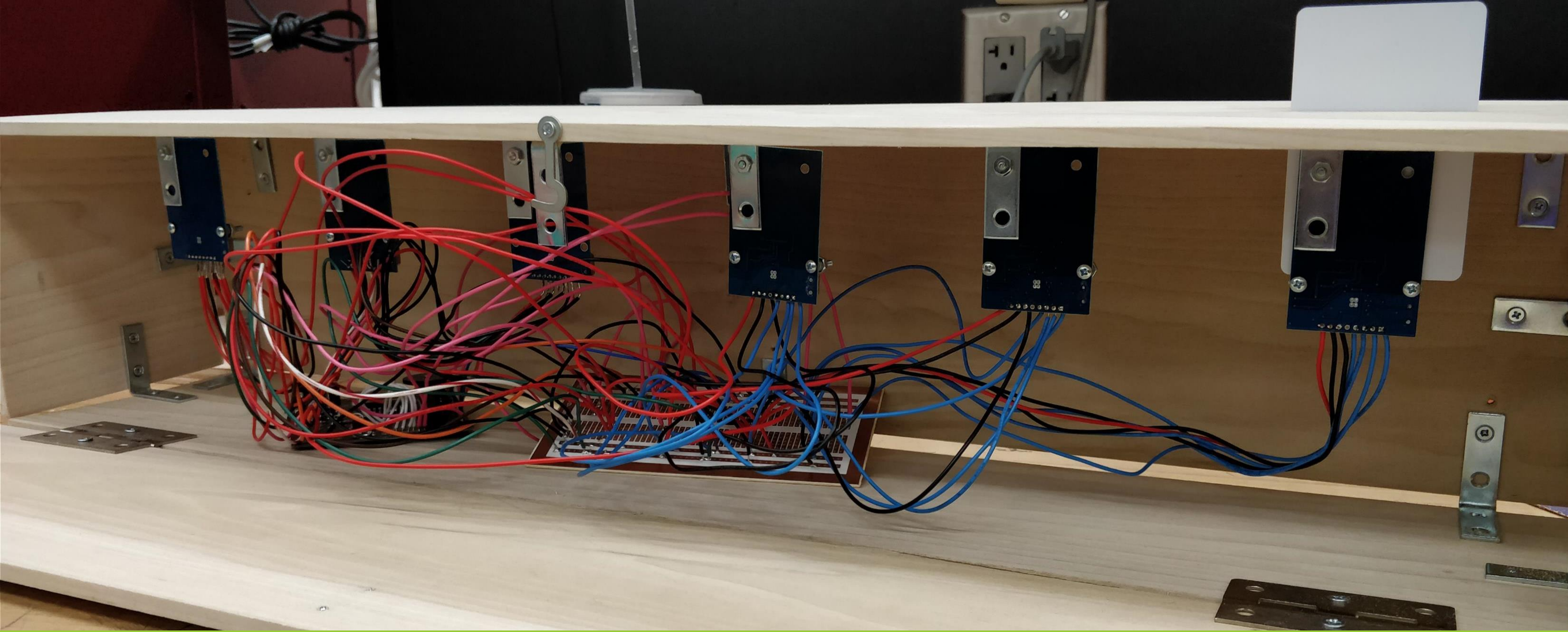
DAC and Op Amp

DAC – MPC4921

- 12 BITS
- SPI
- USES LESS PINS THAN AN R2R LADDER

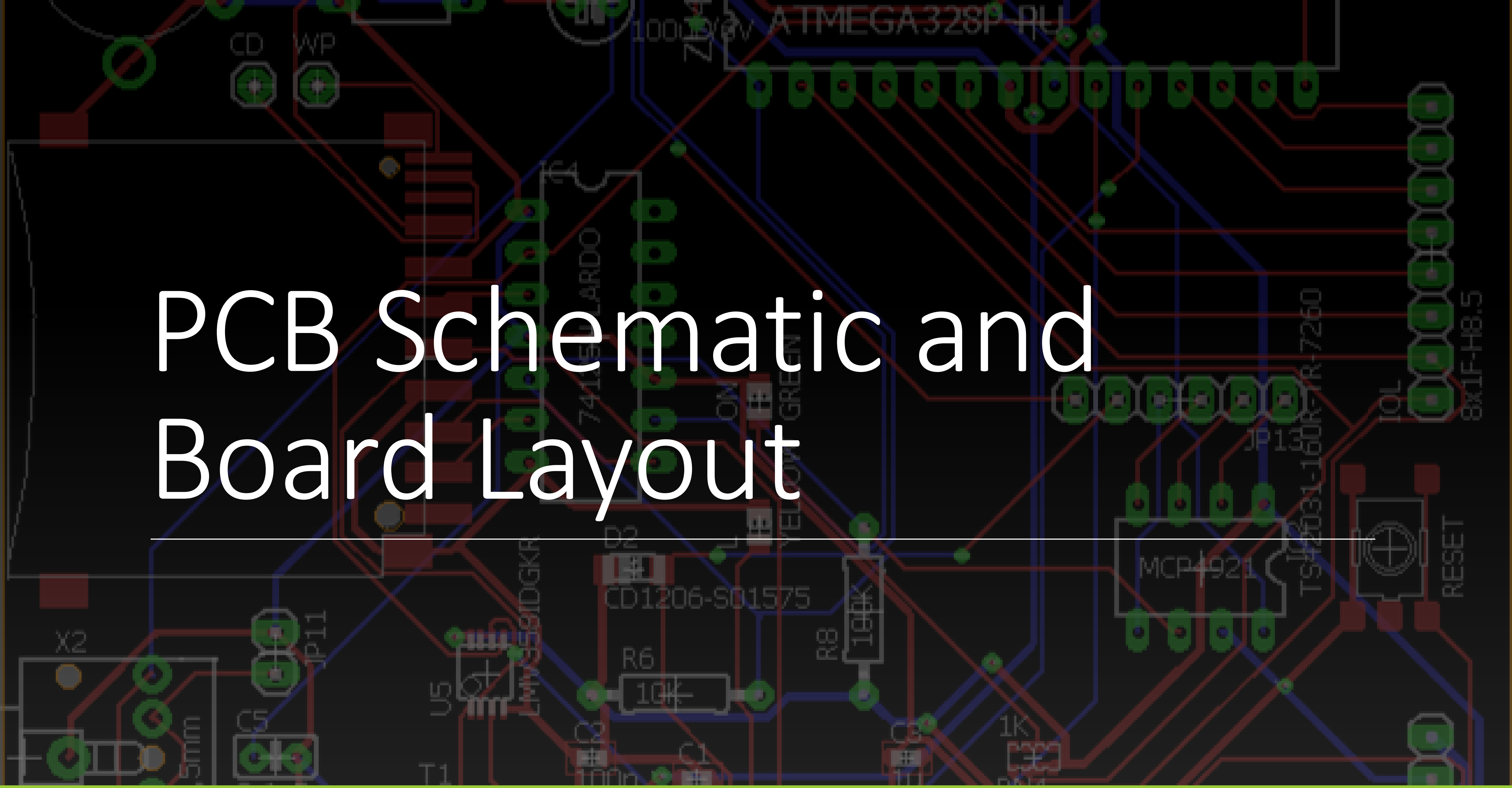
OP AMP – TL072

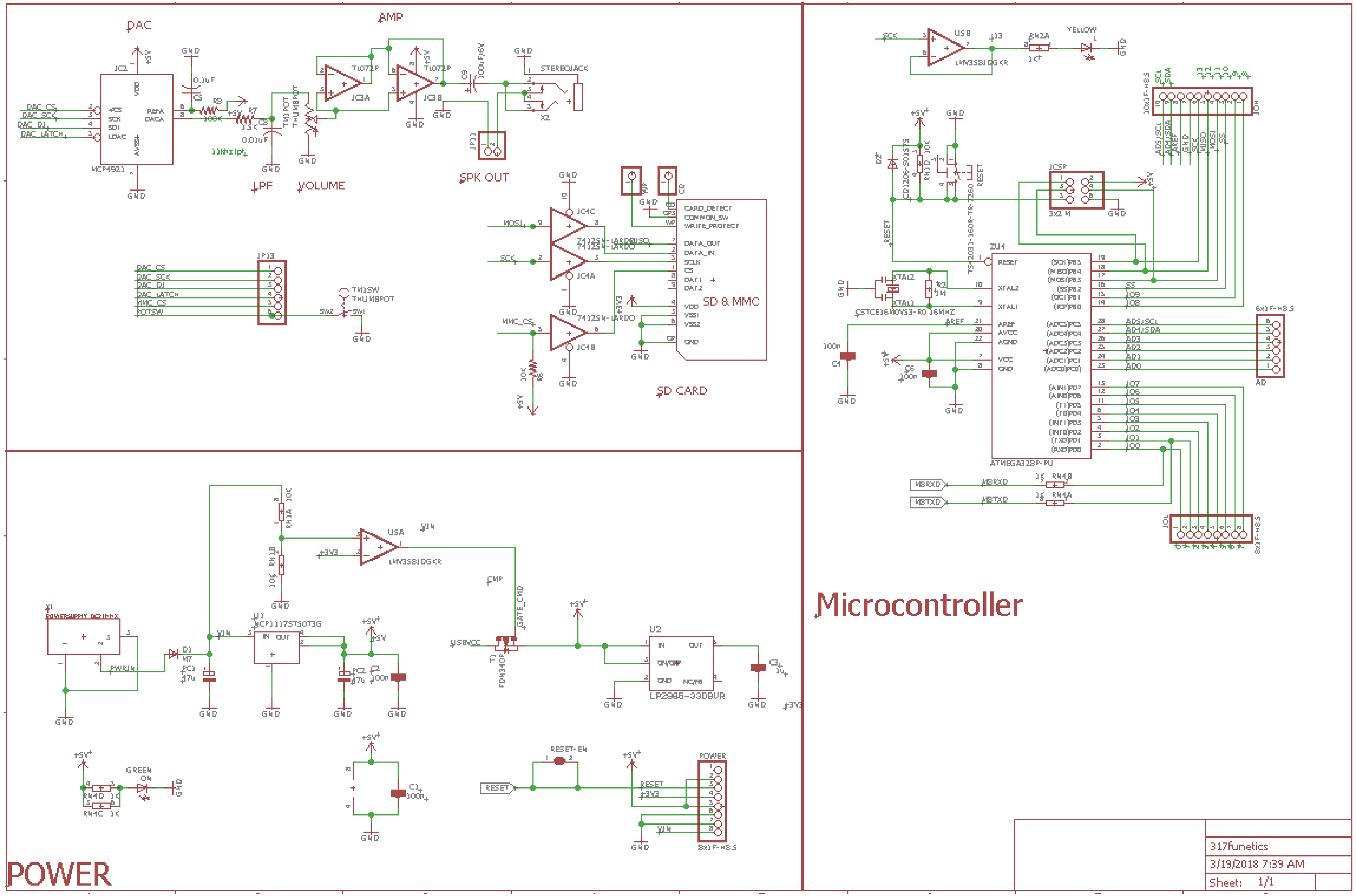
- LOW HARMONIC DISTORTION
- LOW NOISE

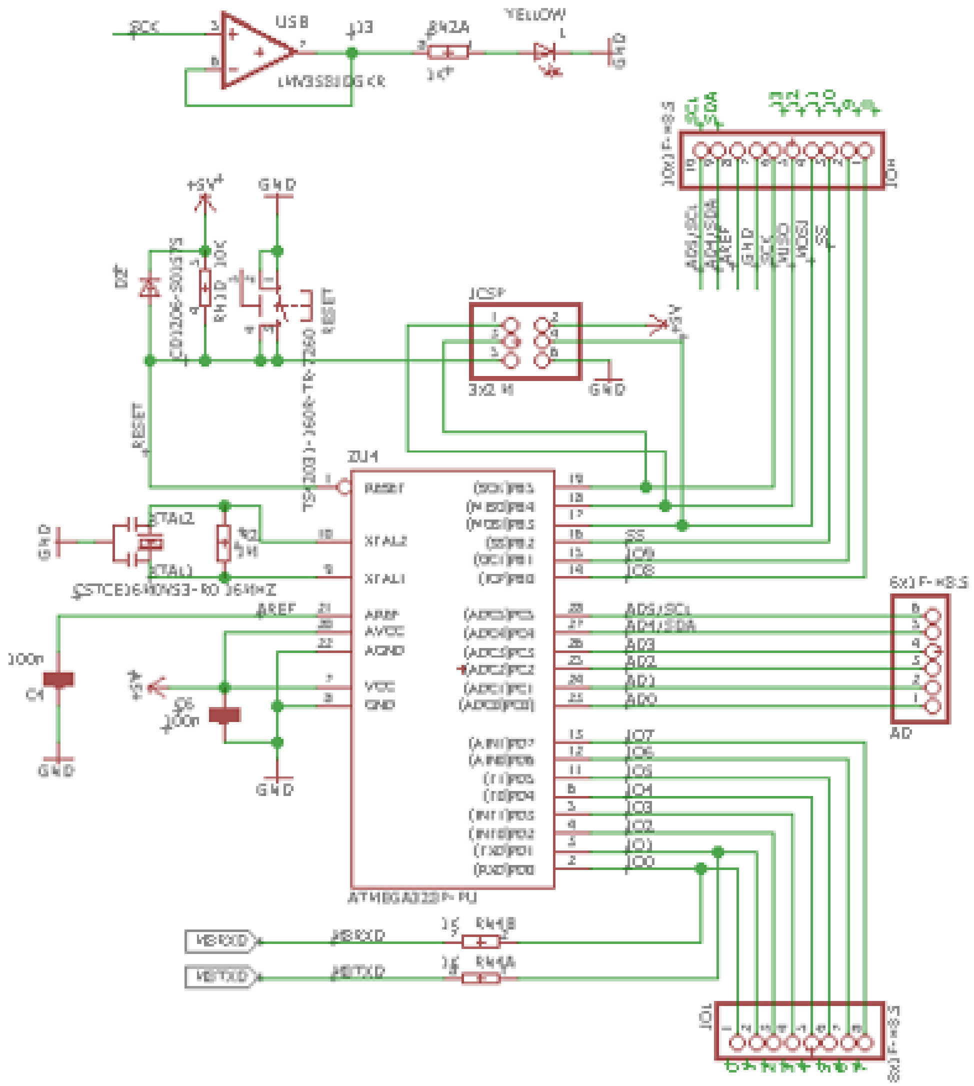


Six 522s, Final Testing

PCB Schematic and Board Layout

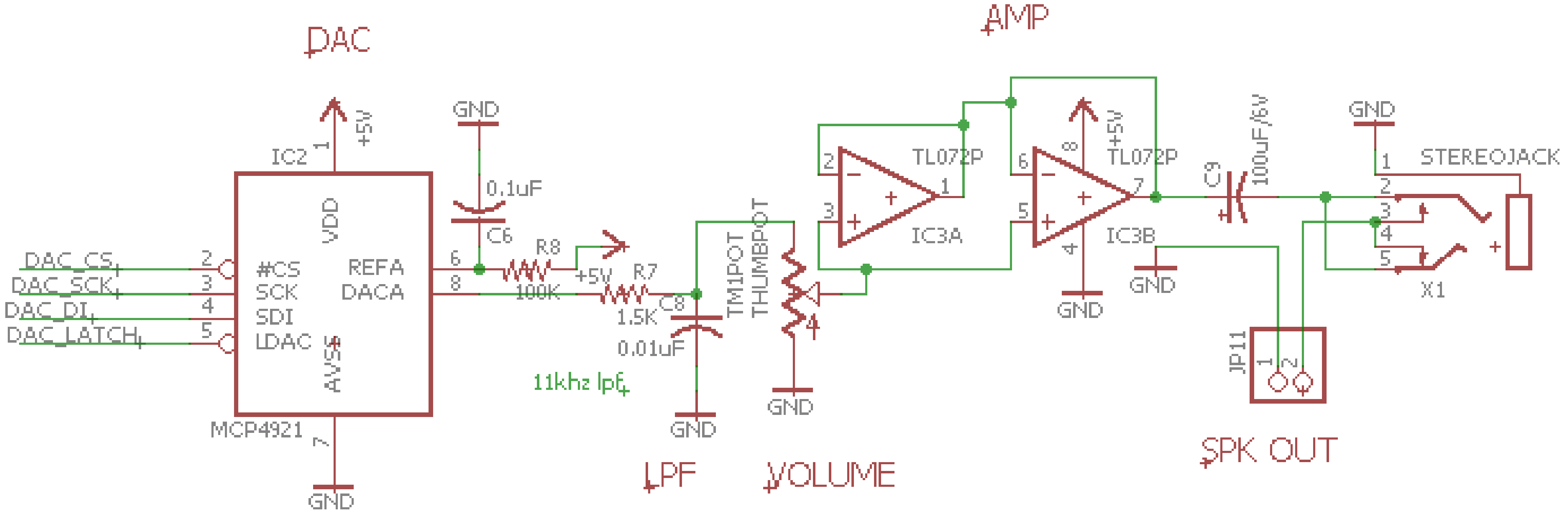



DAC
AMP
LPF
VOLUME
SPK OUT
STEREOJACK
MCP4921
TL072P
SD & MMC
SD CARD
74125N
JP13
THUMBPOT
POWER
LM358IDGKR
MCP1117ST50T3G
LP2985-33DBVR
FDN340P
RESET-EN
GREEN
ON
YELLOW
ICSP
ATMEGA328P-PU
CSTCE16M0V53-R0 16MHZ
MBRXD
MBTXD
Microcontroller
317funetics
3/19/2018 7:39 AM
Sheet: 1/1



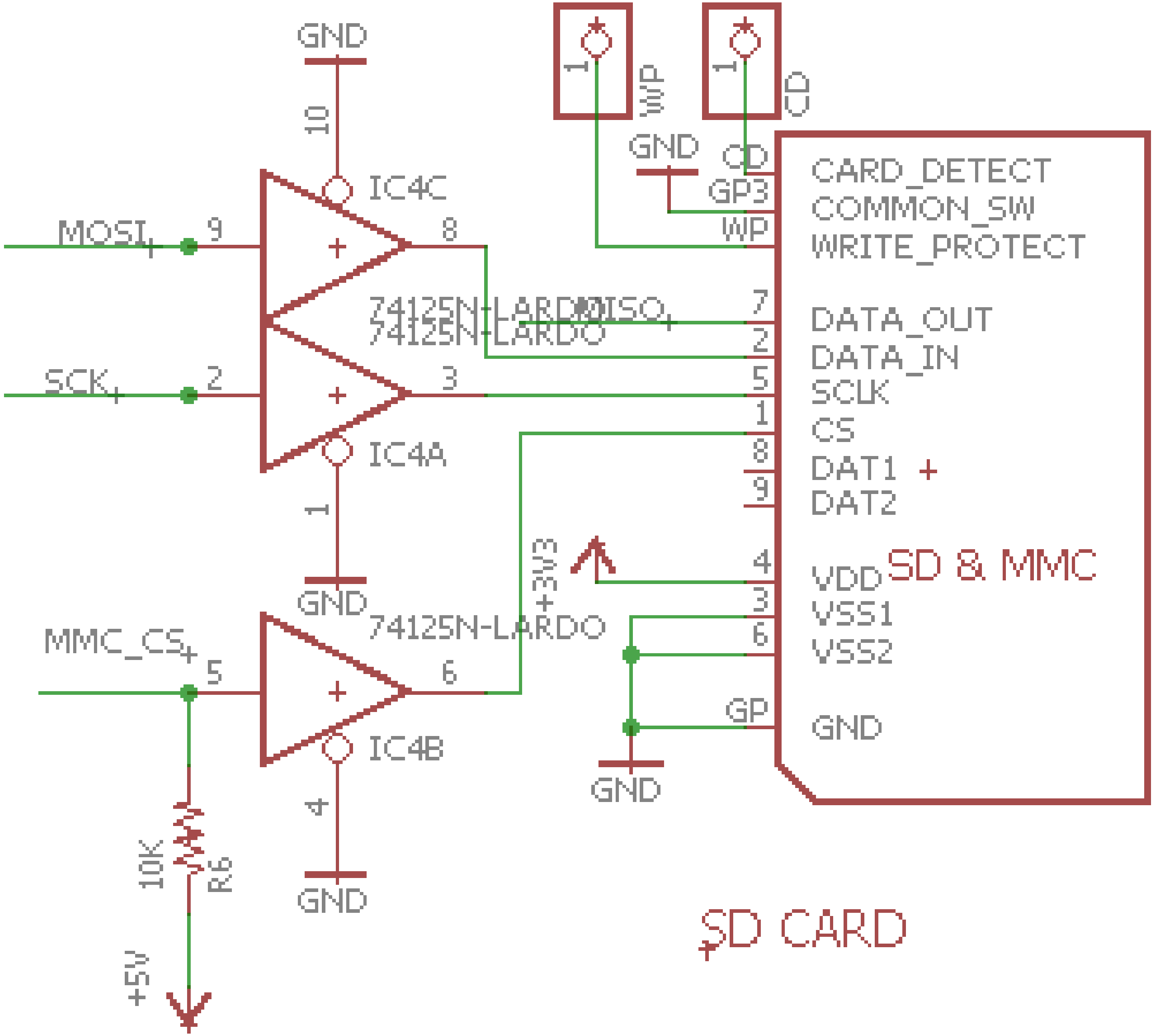
Microcontroller

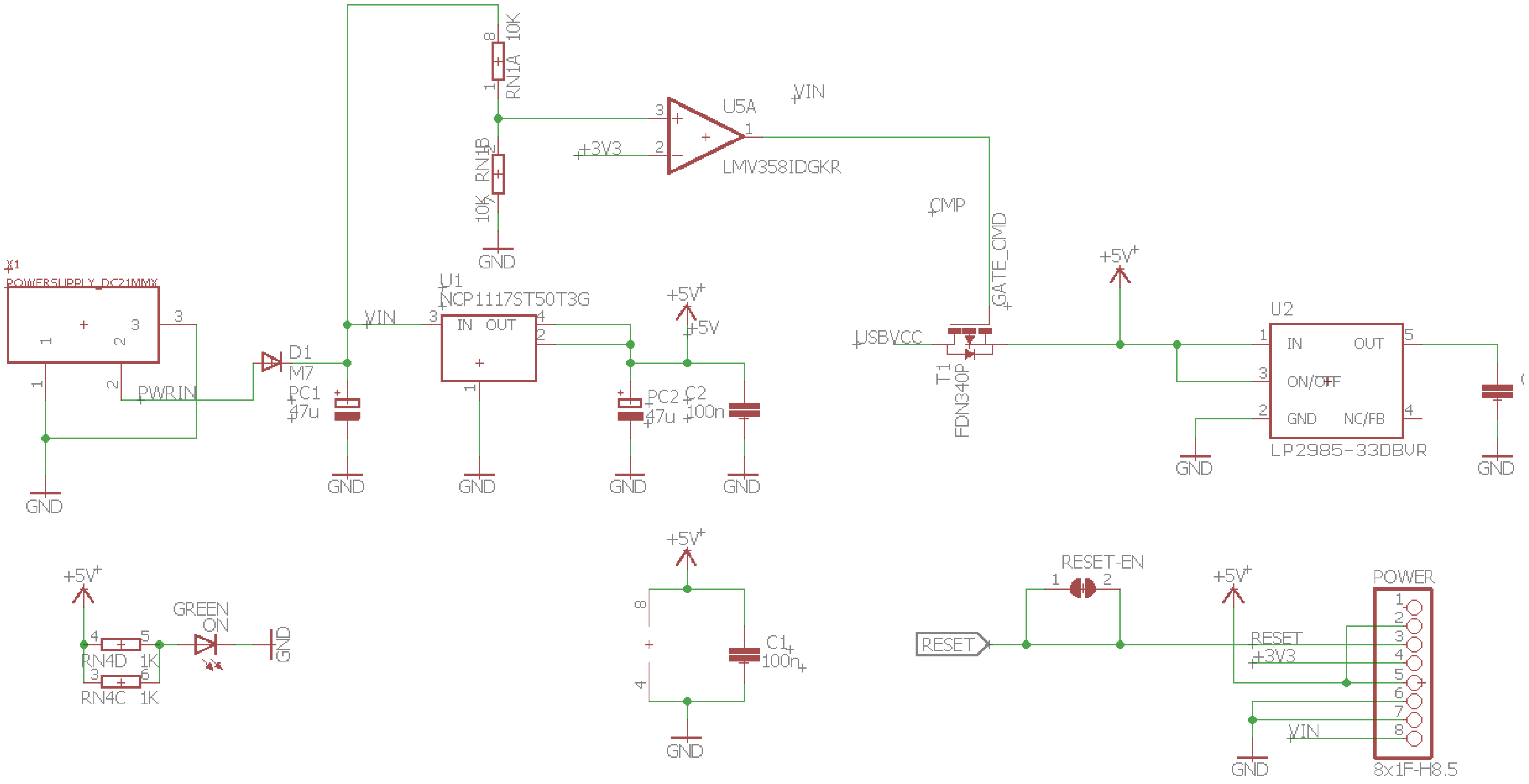
Microcontroller



DAC and Op Amp

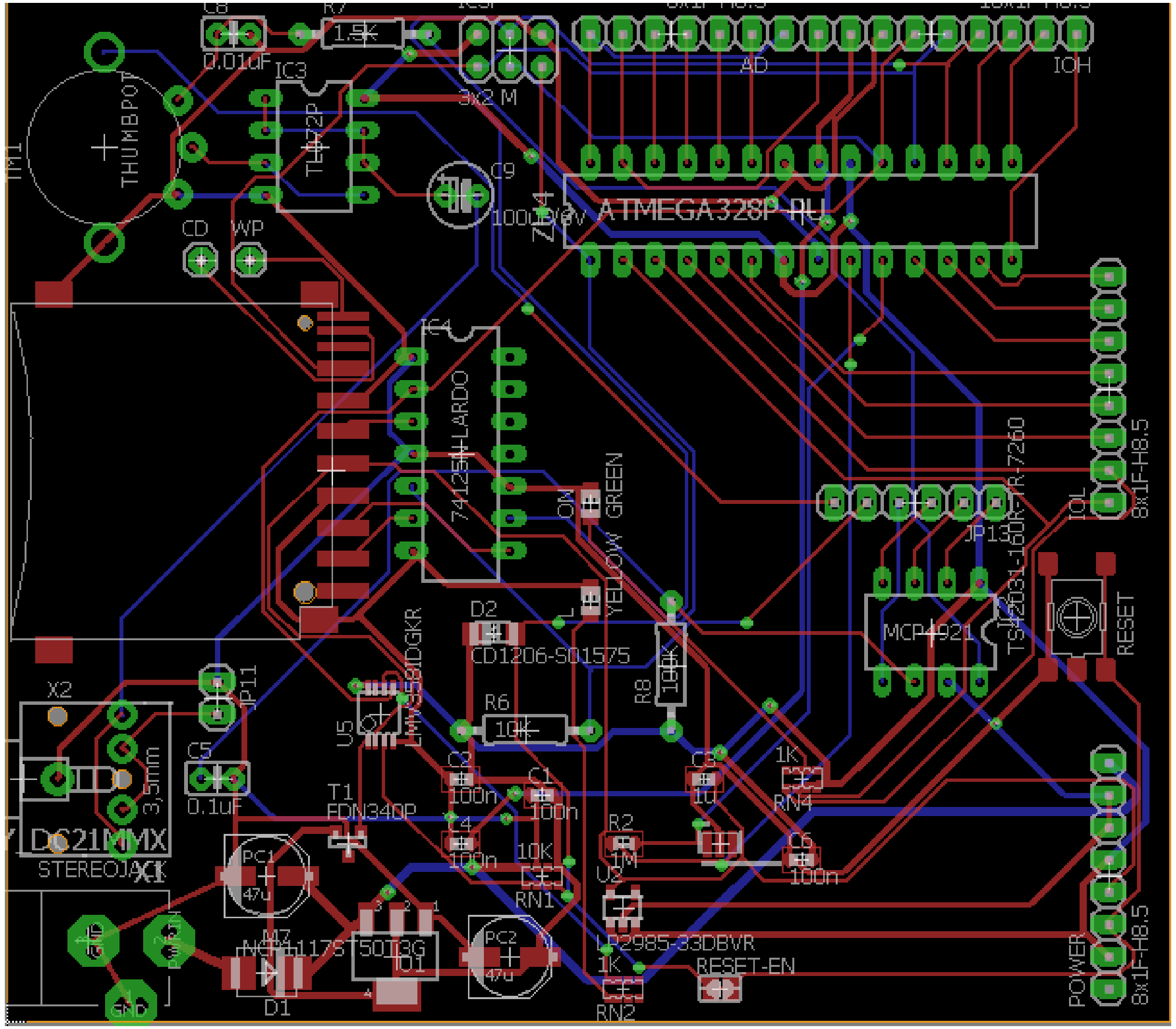
SD Card Interface



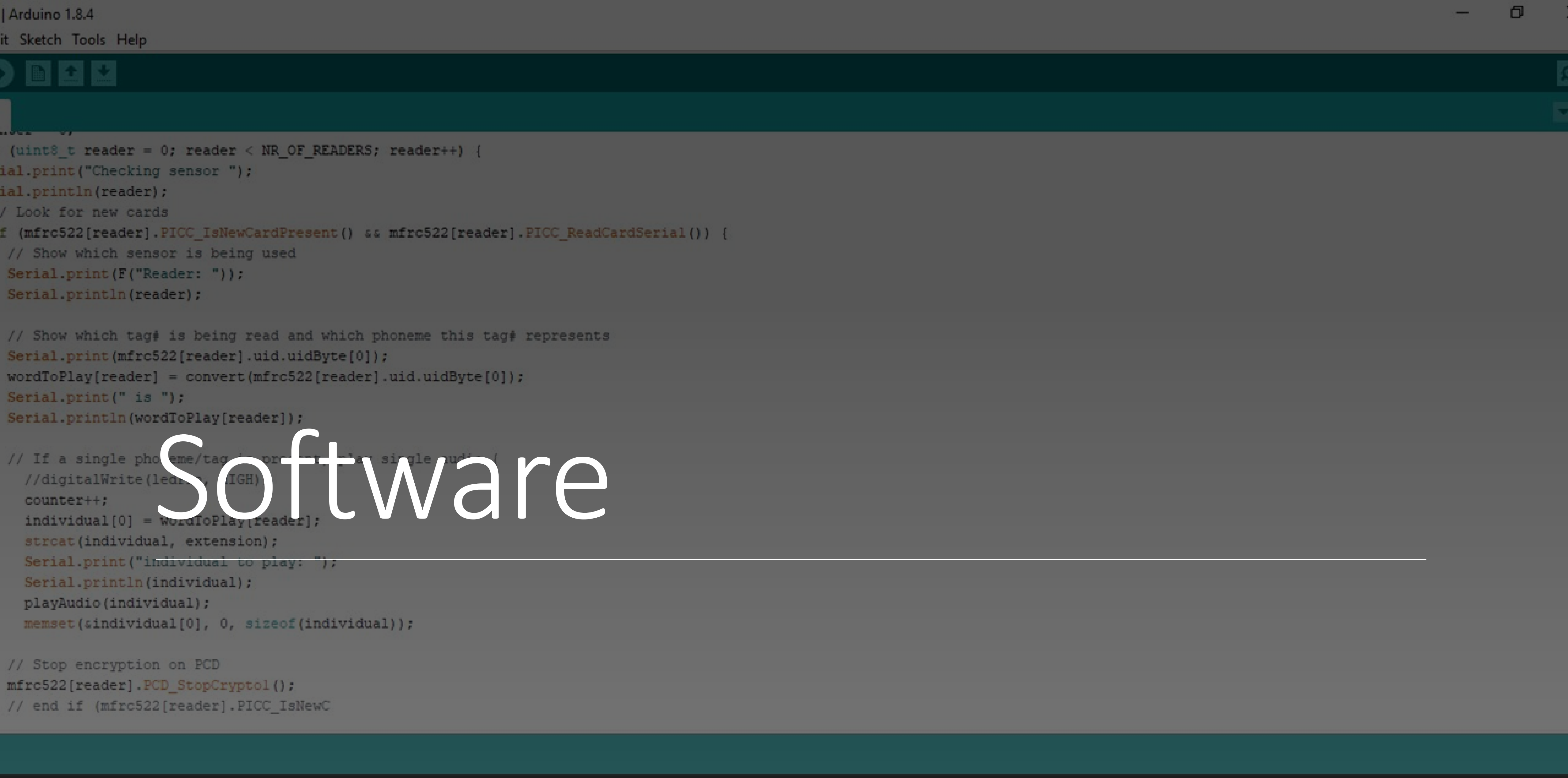


Power Supply

PCB Board (Routed)



Software



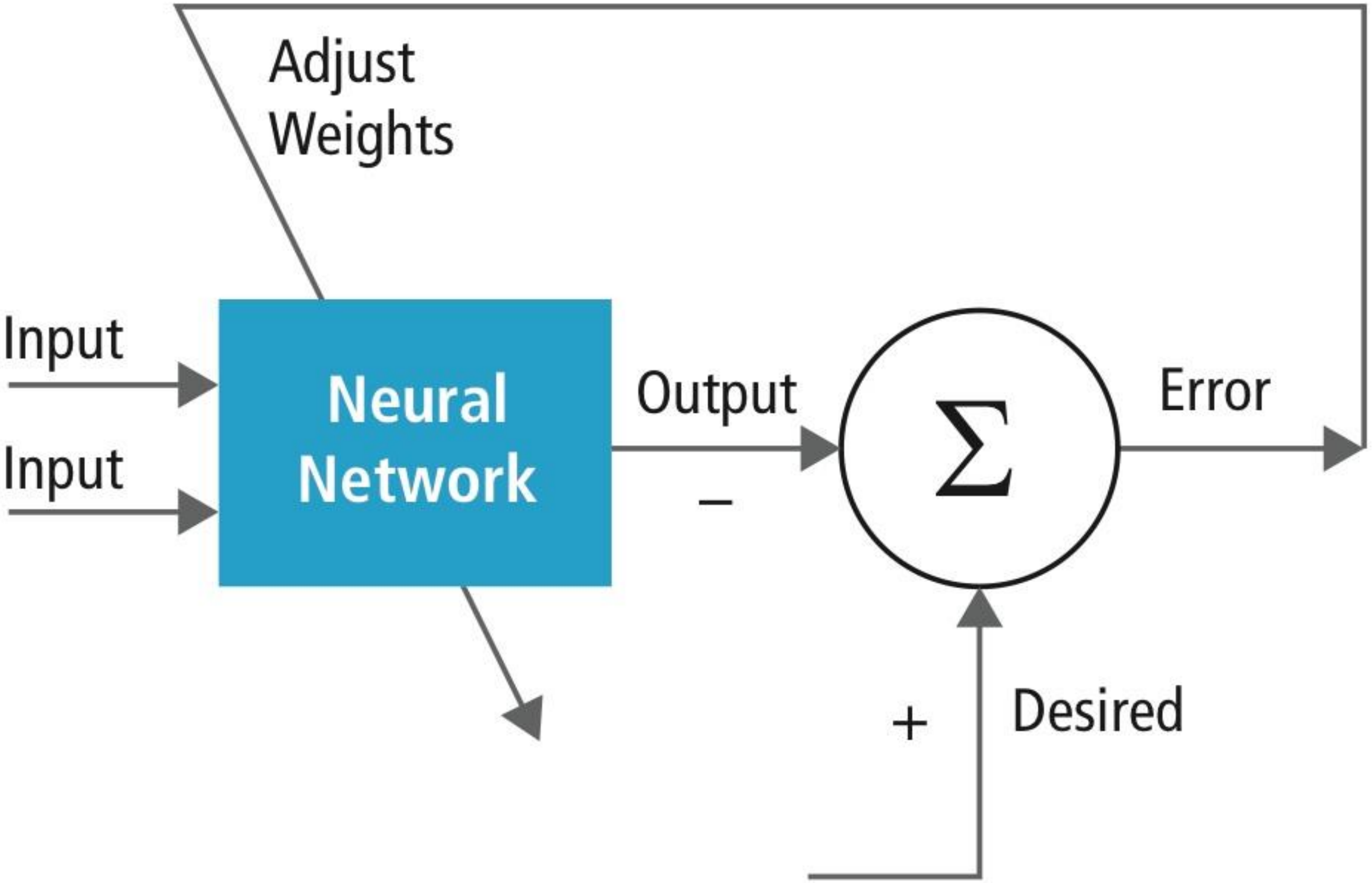
```
 (uint8_t reader = 0; reader < NR_OF_READERS; reader++) {
ial.print("Checking sensor ");
ial.println(reader);
/ Look for new cards
f (mfrc522[reader].PICC_IsNewCardPresent() && mfrc522[reader].PICC_ReadCardSerial()) {
 // Show which sensor is being used
 Serial.print(F("Reader: "));
 Serial.println(reader);

 // Show which tag# is being read and which phoneme this tag# represents
 Serial.print(mfrc522[reader].uid.uidByte[0]);
 wordToPlay[reader] = convert(mfrc522[reader].uid.uidByte[0]);
 Serial.print(" is ");
 Serial.println(wordToPlay[reader]);

 // If a single phoneme/tag is present, play single audio {
   //digitalWrite(ledPin, HIGH)
   counter++;
   individual[0] = wordToPlay[reader];
   strcat(individual, extension);
   Serial.print("individual to play: ");
   Serial.println(individual);
   playAudio(individual);
   memset(&individual[0], 0, sizeof(individual));

 // Stop encryption on PCD
 mfrc522[reader].PCD_StopCrypto1();
 // end if (mfrc522[reader].PICC_IsNewC
```

Creating the Audio

Originally, we decided to use all computerized voices. But, due to restraints, we opted to have full words produced in computer voices and phonemes in human voice.

This was ideal and practical for full words. Our computer voice can easily be used to generate any word whereas recording every word would be tremendously time consuming.

In contrast, phonemes are the most important part to enunciate. Teaching a computer such precise pronunciation properly was not only difficult but also lacking in data and samples. The time invested to just record a human voice for such a small set of sounds where we know it would sound the way it should was the better choice.



Computer Voice

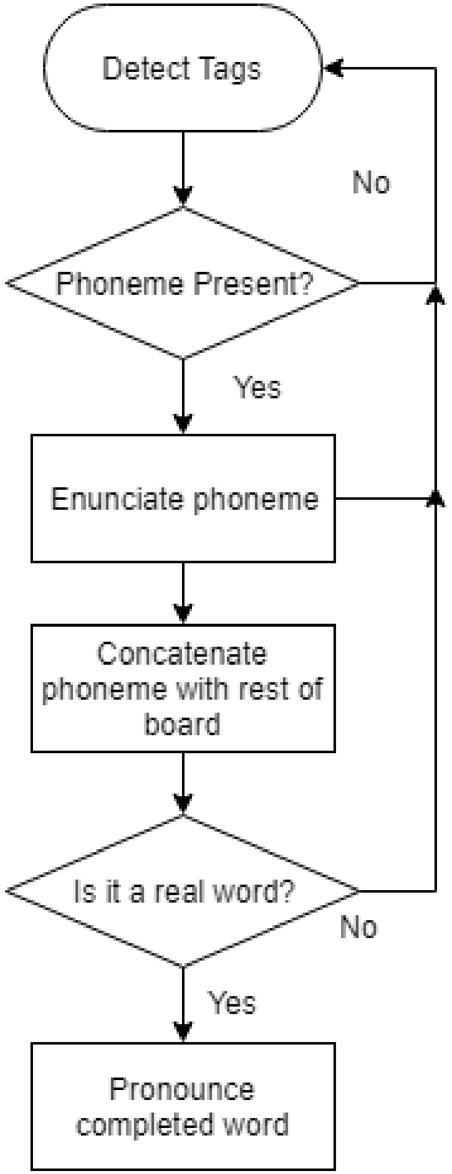
Finalizing the Audio

Once the audio files were created, it was necessary to configure them to match our hardware.

Having all the components connected to our board caused the audio to have a slow down. To compensate, we lowered the speed of the audio to match that of the board so it could have a normal playback speed again.

This results in our audio to play with configurations of 8 bit 11kHz

Software Flowchart

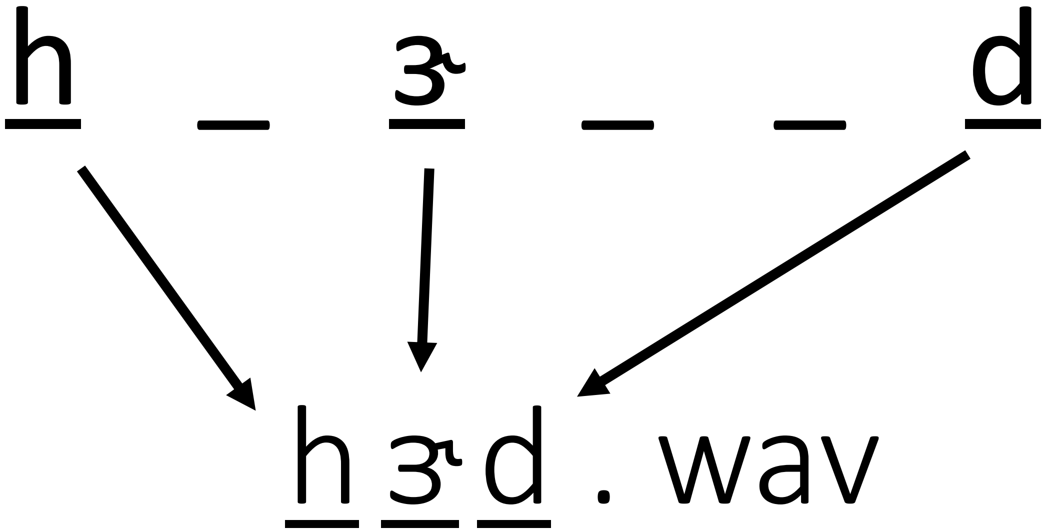


_ _ _ _ _ _

h _ _ _ _ _

h _ _ _ _ d

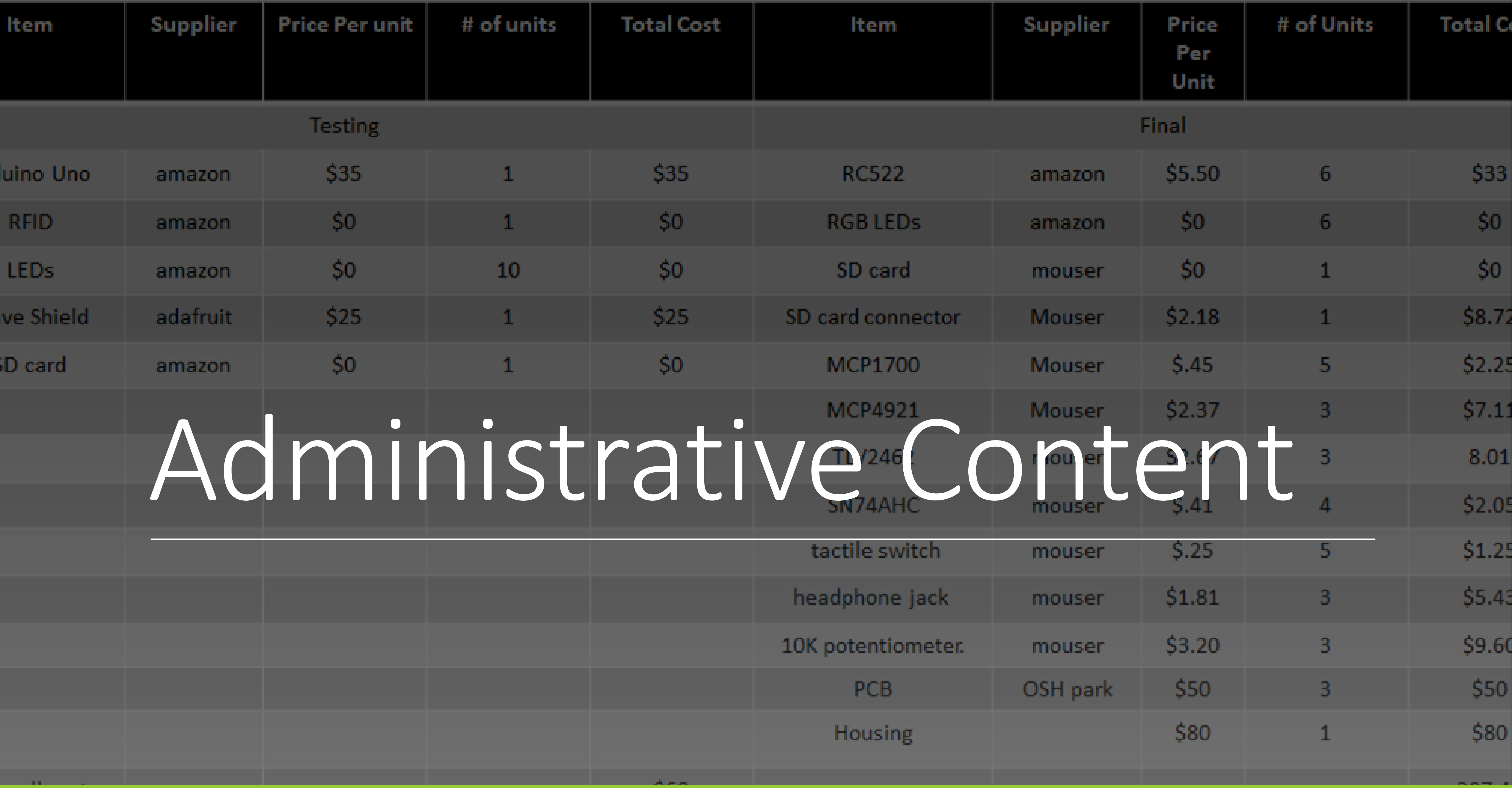
h _ ɝ _ _ d



h _ ɝ _ _ d

hɝd.wav

Administrative Content



Item	Supplier	Price Per unit	# of units	Total Cost	Item	Supplier	Price Per Unit	# of Units	Total C
		Testing					Final		
luino Uno	amazon	$35	1	$35	RC522	amazon	$5.50	6	$33
RFID	amazon	$0	1	$0	RGB LEDs	amazon	$0	6	$0
LEDs	amazon	$0	10	$0	SD card	mouser	$0	1	$0
ve Shield	adafruit	$25	1	$25	SD card connector	Mouser	$2.18	1	$8.7
D card	amazon	$0	1	$0	MCP1700	Mouser	$.45	5	$2.2
					MCP4921	Mouser	$2.37	3	$7.1
					TLV2462	mouser	$2.67	3	8.01
					SN74AHC	mouser	$.41	4	$2.0
					tactile switch	mouser	$.25	5	$1.2
					headphone jack	mouser	$1.81	3	$5.4
					10K potentiometer.	mouser	$3.20	3	$9.6
					PCB	OSH park	$50	3	$50
					Housing		$80	1	$80

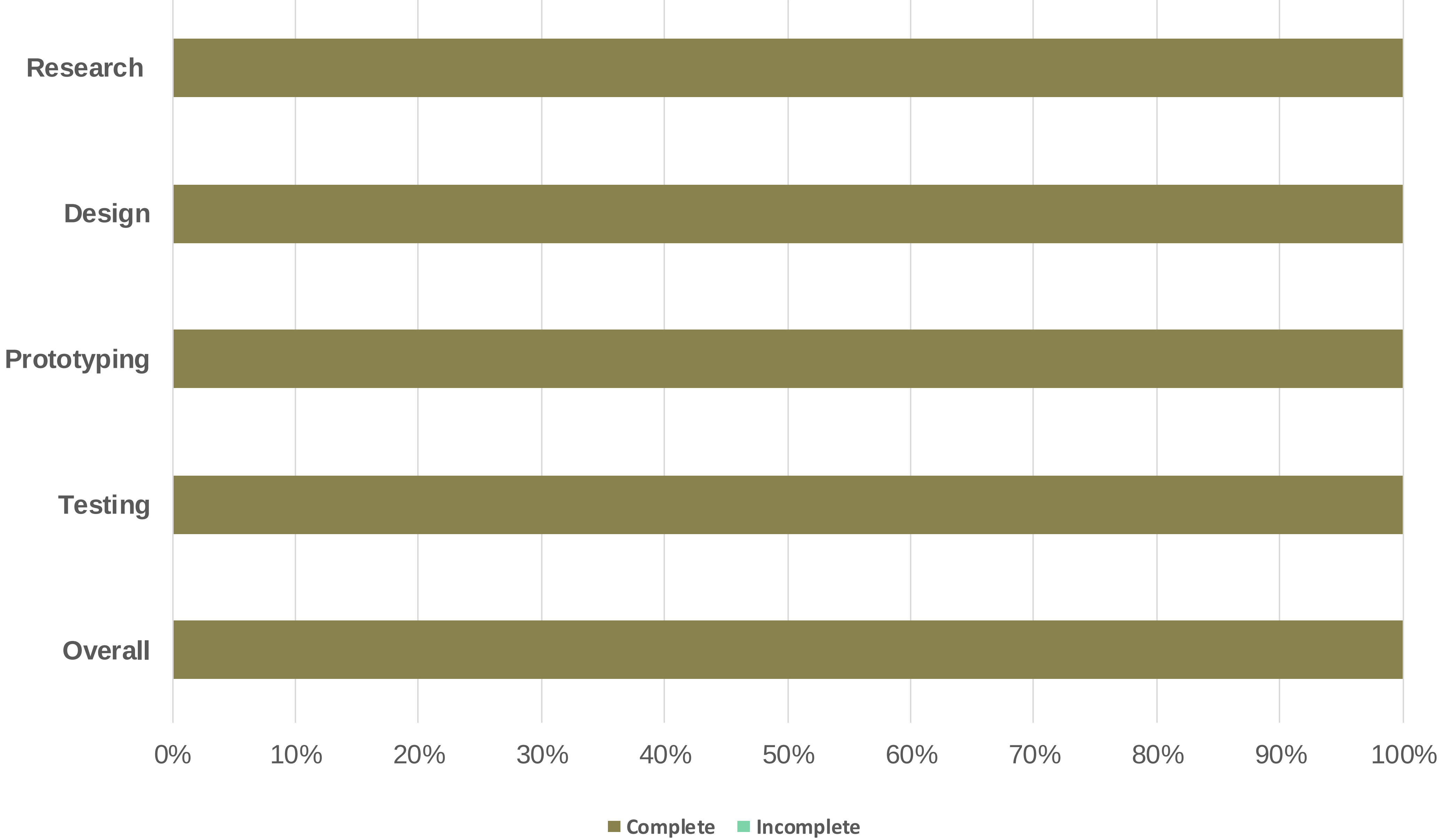
Work Distribution

	Maureen	**Daniel**	**Meychele**	**EJ**
Hardware: PCB	Primary	Secondary		
Hardware: Electronics	Secondary	Primary		
Hardware: Housing	Secondary		Primary	
Software: RFID & Audio			Secondary	Primary

Budget

Item	Supplier	Price Per unit	# of units	Total Cost	Item	Supplier	Price Per Unit	# of Units	Total Cost
Testing					Final				
Arduino Uno	Amazon	$35	1	$35	PCB	OSH park	$150	3	$150
RFID	Amazon	$0	1	$0	RC522	Amazon	$0	6	$0
LEDs	Amazon	$0	10	$0	SD card	Walmart	$20	1	$20
Wave Shield	adafruit	$25	1	$25	SD card connector	Mouser	$2.18	1	$8.72
SD card	Amazon	$0	1	$0	NCP1117	Mouser	$.45	5	$2.25
					MCP4921	Mouser	$2.37	3	$7.11
					TLV2462	mouser	$2.67	3	8.01
					SN74AHC	mouser	$.41	4	$2.05
					tactile switch	mouser	$.25	5	$1.25
					headphone jack	mouser	$1.81	3	$5.43
					10K potentiometer.	mouser	$3.20	3	$9.60
					Additional hardware	hardware store	$100		$100
					Housing		$80	1	$80
Overall cost				$60					$394.42

Overall Progress



Beyond Senior Design

Able to respond to any combination of phonetics and construct each word string and audio file as a new entity regardless if it is a real word or not.

Implement other languages

Can be branched into similar designs outside of just human speech

Issues

- Inconsistencies with RFID tags/cards being activated by sensor and sending data
- Extremely uncooperative wiring

Demo and Q&A time!