Analog and Digital Effects Processing Technology (ADEPT)

Group C
Diego Conterno - CpE
Tyler Michaud - EE
Alejandro Porcar - CpE
Dylan Walter - EE

Goals and Objectives

Motivation

- Fusion of audio engineering, music technology, and sound design
- Close the gap between technology and music
- Provide new alternatives for musicians who want to stand out and shape their sound in a unique way
- Have a creative impact on musicians with an interest in electrical and computer engineering

Objectives

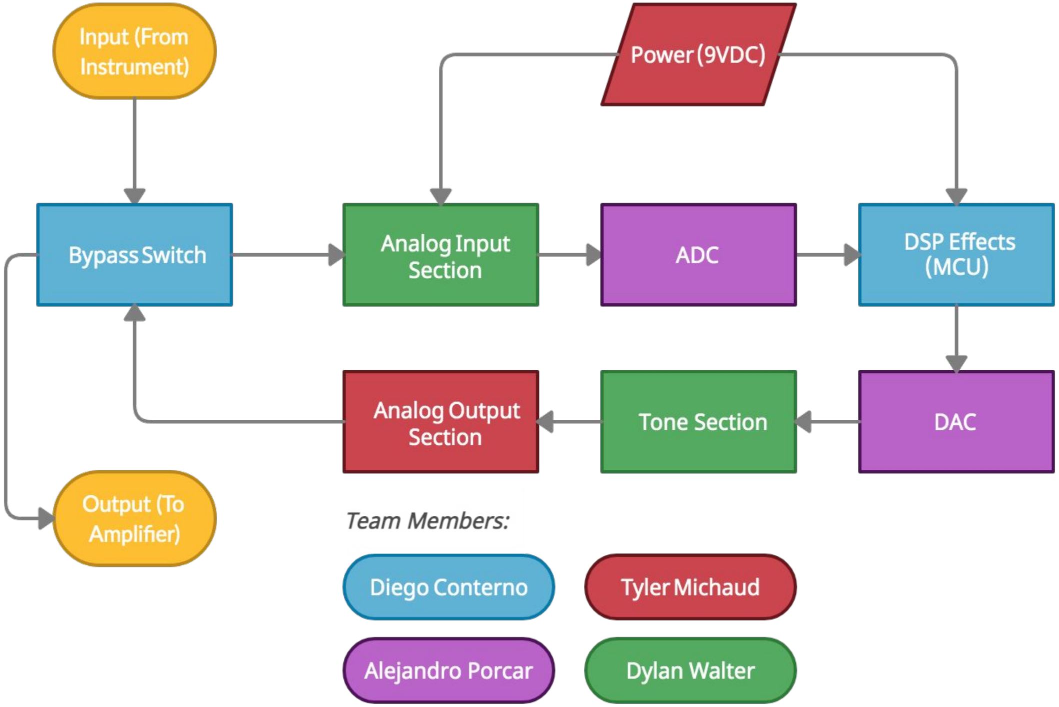
- Affordable
- Easy to Use
- Low Current Draw
- Low Noise Level
- Reliable

Requirements and Specifications

- Analog-to-Digital Conversion
- High Input Impedance
- Low Output Impedance
- Looper Functionality
- Standard ¼” Instrument Cable Compatibility
- 9V Power
- JTAG/SWD Programmability
- LCD Menu and Selection Screen
- Dual Footswitch Functionality

Block Diagram

- Electrical Engineering Team:
    - Tyler Michaud
    - Dylan Walter
- Computer Engineering Team:
    - Diego Conterno
    - Alejandro Porcar



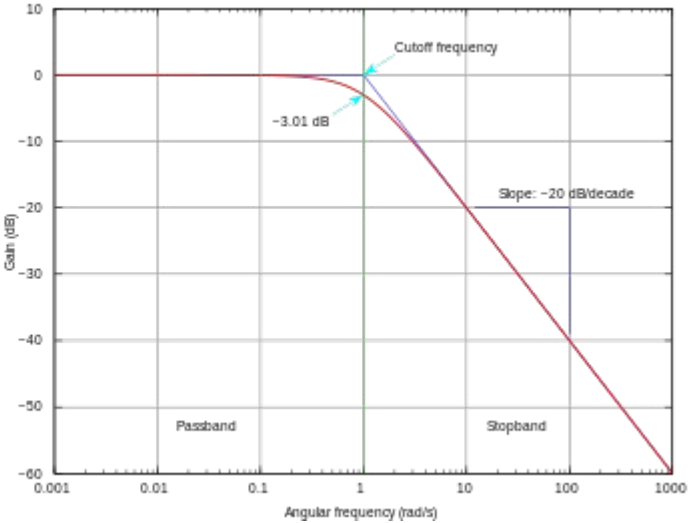
Hardware Design

Footswitches

- 3PDT (3-Pole, Double Throw) mechanical true bypass latching footswitch
  - Used to turn the effect on and off
- SPST (Single Pole, Single Throw) momentary footswitch
  - Used for tap tempo and looper

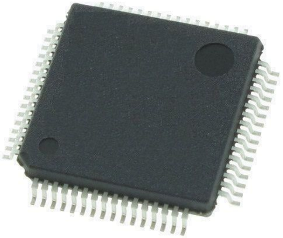
Tone and Volume Controls

- Tone Control
  - Adjusts the cutoff frequency of the instrument signal in order to make the effect sound brighter or darker
  - Logarithmic Potentiometer in series with First Order RC Low Pass Filter
- Volume Control
  - Logarithmic Potentiometer placed at the end of pedal circuit to control overall volume



MCU (STM32F446RC)

- Processing performance
- FPU
- Integrated configurable debug
- Cost



CODEC (PCM 3060)

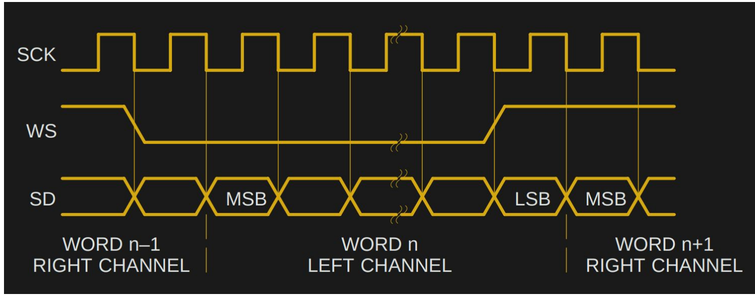
- A CODEC contains both an ADC and DAC

- We chose the PCM 3060 because:

  - 24 bit Stereo Resolution

  -Up to 96kHz Sampling rate (we are using 44.1Khz)

- We will be using I2S Interface to communicate with this chip

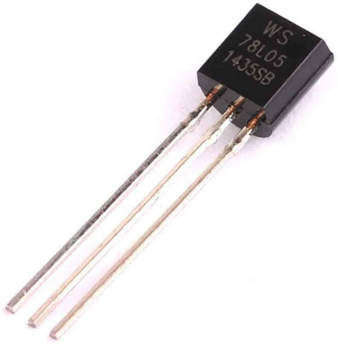


Analog Input and Output Buffers

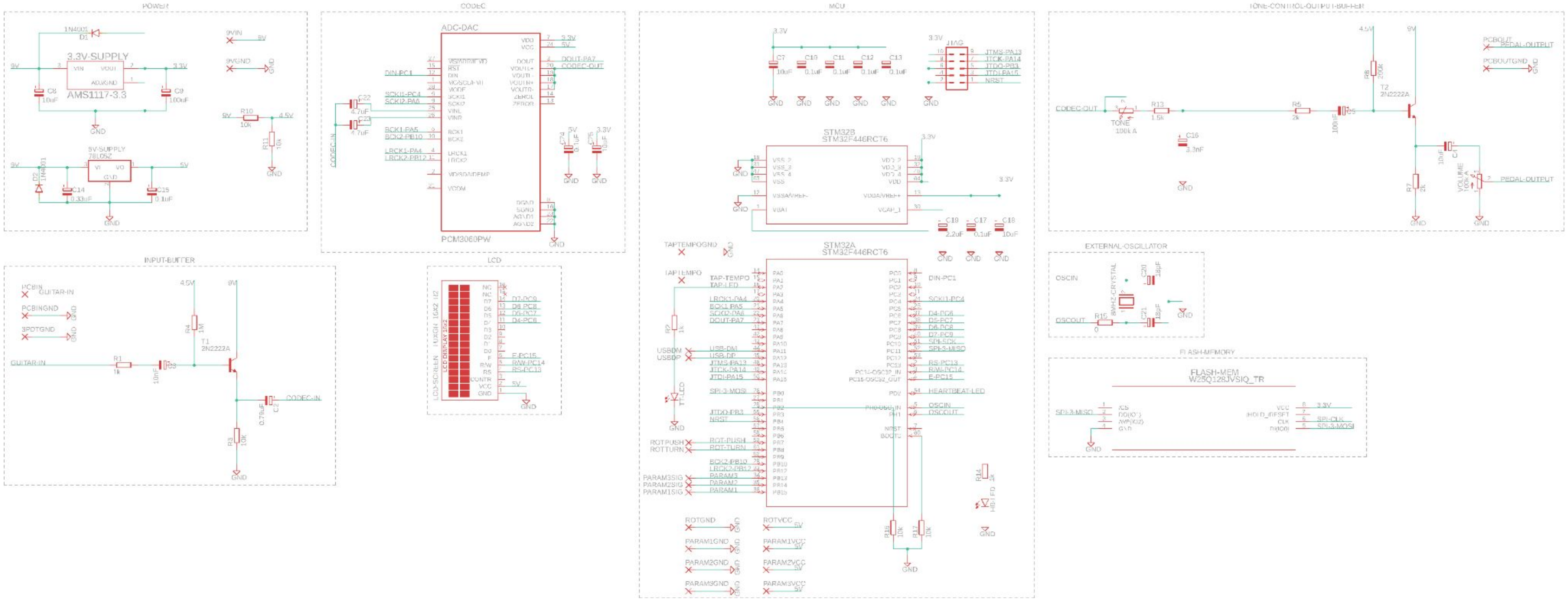
- Active electronic circuit that can provide a change in electrical impedance.
- Maintains signal integrity going in and out of the system.
- Low energy, voltage driven signal.
- High input impedance to low output impedance.
- Emitter follower configuration (standard practice).
- 2N2222A BJT

Power (9VDC)

- 9V to 5V and 9V to 3.3V voltage regulation.
- Linear voltage regulators. For microcontroller, CODEC,and flash memory.
- 78L05Z for 5V and AMS1117 for 3.3V.
- Not power efficient but less noisy.
- 9V and 4.5V(using voltage divider) coupled to input and output buffers.
- Coaxial power jack(2.1mm inside diameter and 5.5mm outside diameter.
- 300mA rating.
- Proper Transient suppression and configuration

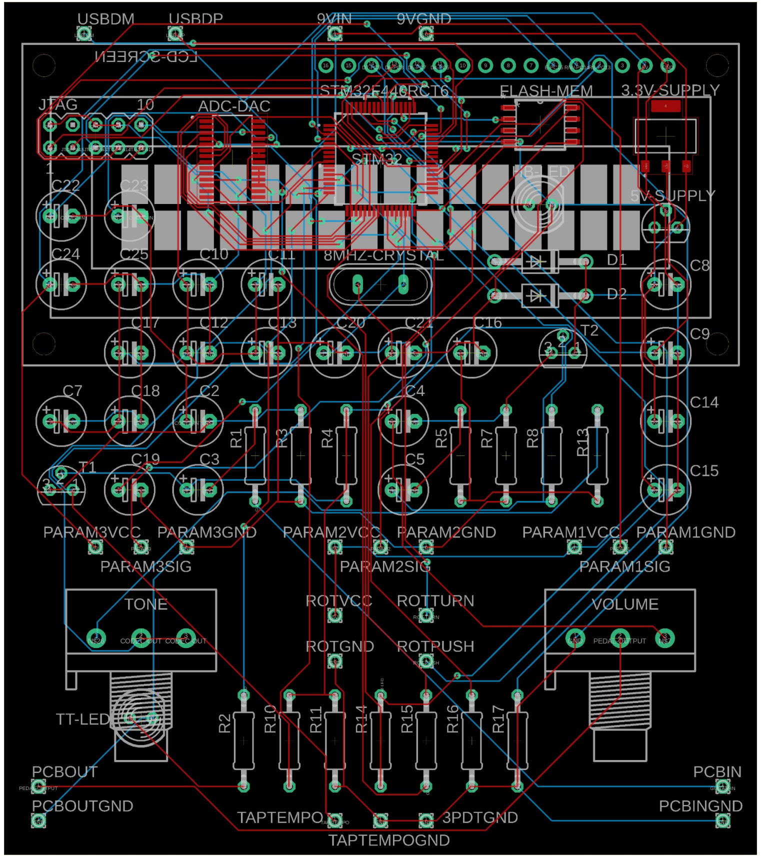


Schematic


POWER
1N4001
D1
3.3V-SUPPLY
AMS1117-3.3
C8
10uF
C9
100uF
5V-SUPPLY
78L05Z
D2
1N4001
C14
0.33uF
C15
0.1uF
9VIN
9VGND
R10
10k
R11
10k
4.5V
9V
3.3V
5V
GND
CODEC
ADC-DAC
PCM3060PW
DIN-PC1
SCKI1-PC4
SCKI2-PA6
BCK1-PA5
BCK2-PB10
LRCK1-PA4
LRCK2-PB12
DOUT-PA7
CODEC-OUT
CODEC-IN
C22
4.7uF
C23
4.7uF
C74
0.1uF
C75
10uF
INPUT-BUFFER
PCBIN
GUITAR-IN
PCBINGND
3POTGND
R1
1k
C3
10nF
R4
1M
T1
2N2222A
C2
0.79uF
R3
10k
CODEC-IN
LCD
LCD-SCREEN
LCD DISPLAY 16x2
D7-PC9
D6-PC8
D5-PC7
D4-PC6
E-PC15
R/W-PC14
RS-PC13
MCU
JTAG
JTMS-PA13
JTCK-PA14
JTDO-PB3
JTDI-PA15
NRST
C7
10uF
C10
C11
C12
C13
0.1uF
STM32B
STM32F446RCT6
STM32A
STM32F446RCT6
C19
2.2uF
C17
0.1uF
C18
10uF
TAPTEMPOGND
TAPTEMPO
TAP-TEMPO
TAP-LED
R2
1k
TT-LED
USBDM
USBDP
USB-DM
USB-DP
SPI-3-MOSI
SPI-SCK
SPI-3-MISO
HEARTBEAT-LED
OSCIN
OSCOUT
ROTPUSH
ROTTURN
ROT-PUSH
ROT-TURN
PARAM3SIG
PARAM2SIG
PARAM1SIG
PARAM3
PARAM2
PARAM1
R14
1k
HB-LED
R16
10k
R17
10k
ROTGND
ROTVCC
PARAM1GND
PARAM1VCC
PARAM2GND
PARAM2VCC
PARAM3GND
PARAM3VCC
TONE-CONTROL-OUTPUT-BUFFER
CODEC-OUT
TONE
100k A
R13
1.5k
C16
3.3nF
R5
2k
C5
100nF
R6
200k
T2
2N2222A
C1
10uF
R7
2k
VOLUME
100k A
PCBOUT
PEDAL-OUTPUT
PCBOUTGND
EXTERNAL-OSCILLATOR
8MHZ-CRYSTAL
C20
18pF
C21
18pF
R15
0
FLASH-MEMORY
FLASH-MEM
W25Q128JVSIQ_TR
SPI-3-MISO
SPI-CLK
SPI-3-MOSI

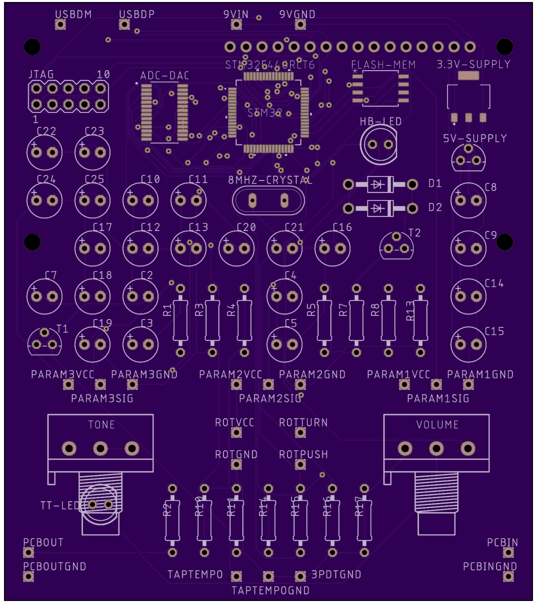
PCB Layout

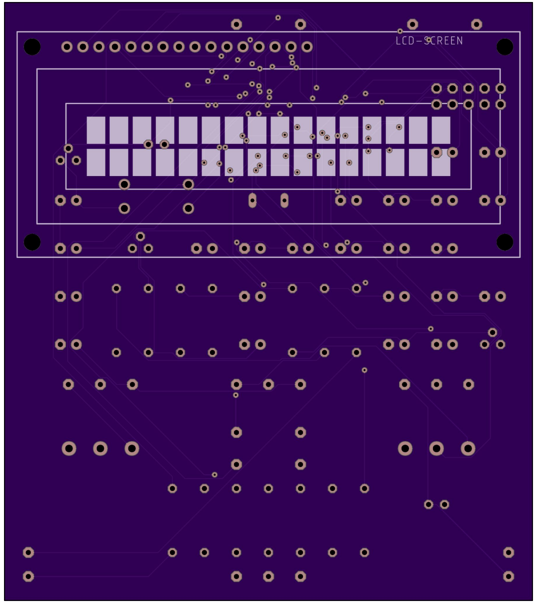
- Layout done using AutoCAD Eagle
- 2 board layers for simplicity
- Default trace widths and spacing used
- PCB Dimensions: 94.92mm x 83.80mm



PCB Layout

- PCB Fabrication done by OSH Park
- Both through-hole (THT) and surface mount (SMT) components integrated in our design



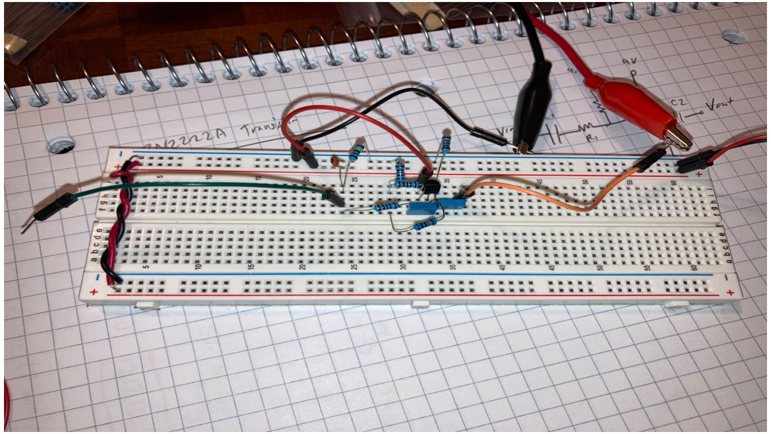


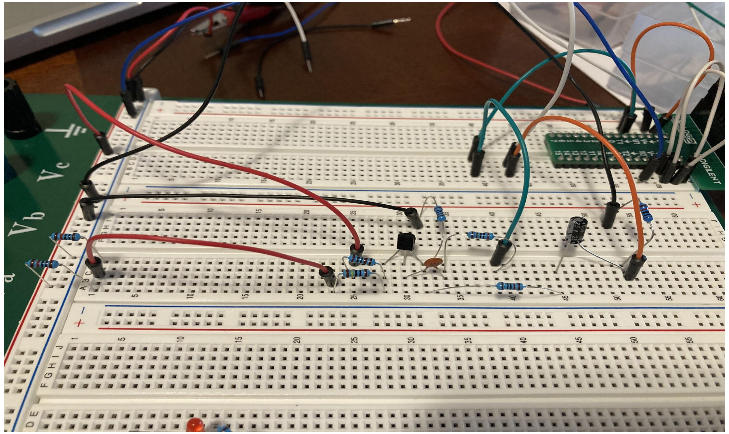
Prototyping and Initial Testing

Power Source Emulation

- External power supply verifies that the correct voltage is powering our device.
- Enables us to observe abnormalities of current consumption.

Input and Output Buffer Prototype



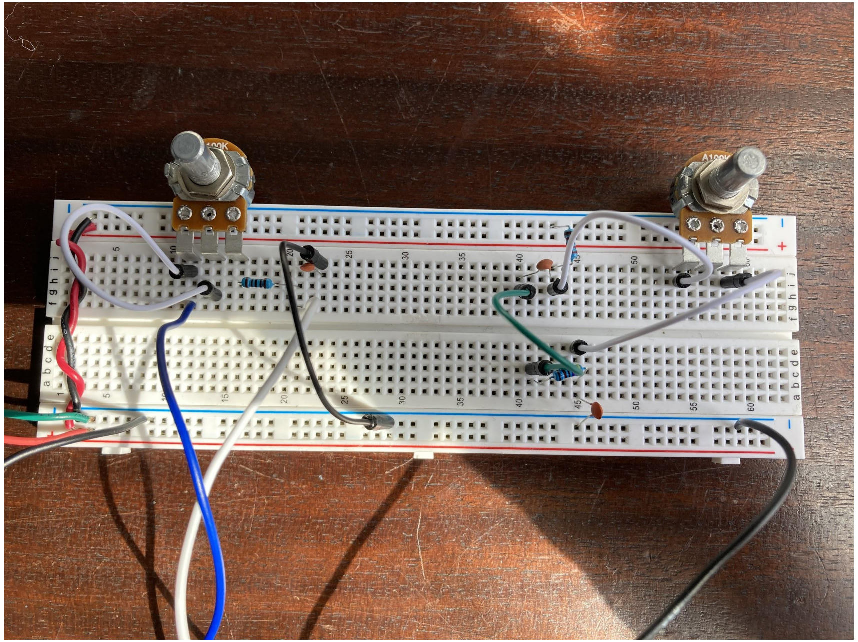


- Measured input/output signal using oscilloscope and function generator.

- All components used are same values and models as simulation.

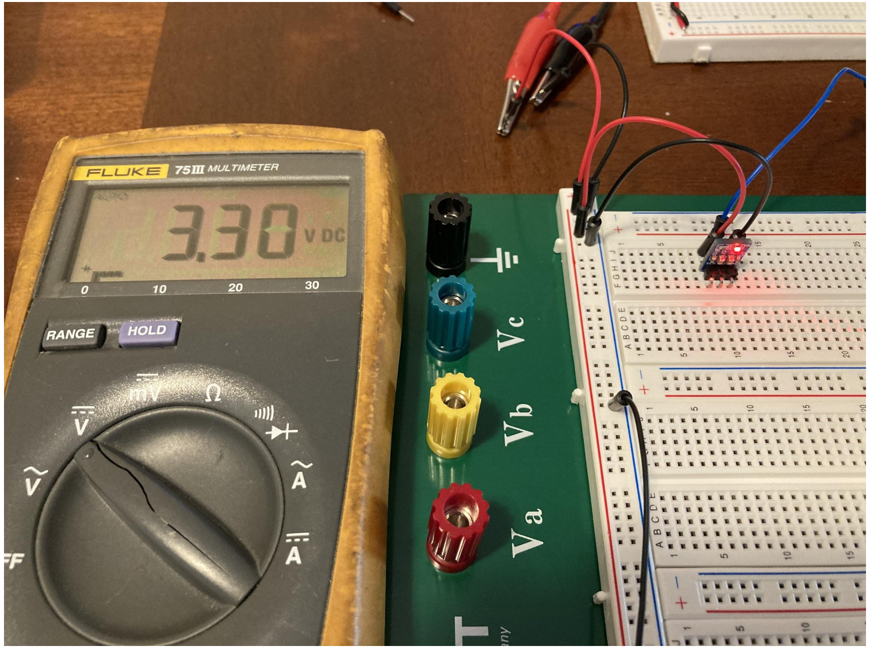
Tone Control Prototype

- Two configurations for comparison:
  - "Bluesbreaker" RC Lowpass Filter
  - "Big Muff" RC Low/Highpass Filter
- Utilizes simple components:
  - Logarithmic potentiometers
  - Passive resistors and capacitors



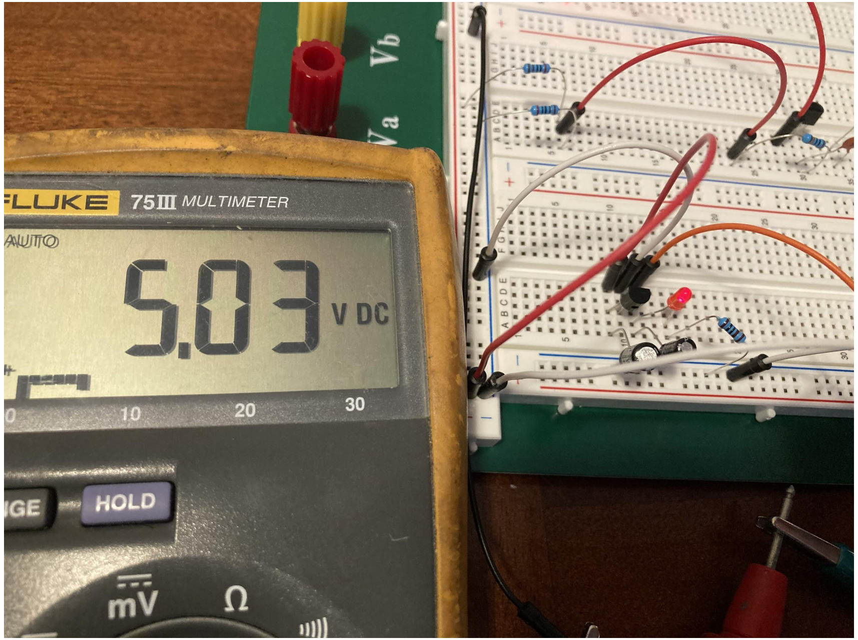
AMS1117 Voltage Regulator

- 9V External Power source
- DMM to confirm output of 3.3V
- Observed for >2 hours for overheating



78L05 Voltage Regulator

- 9V External Power source
- DMM to confirm output of 5V
- Observed for >2 hours for overheating



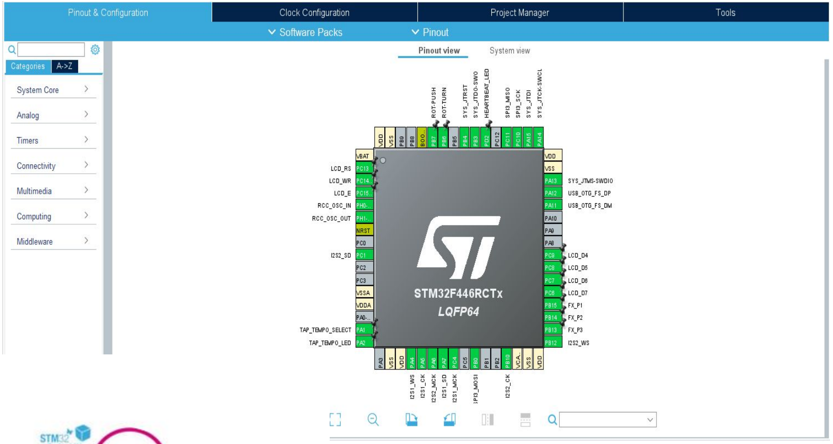
Software Design

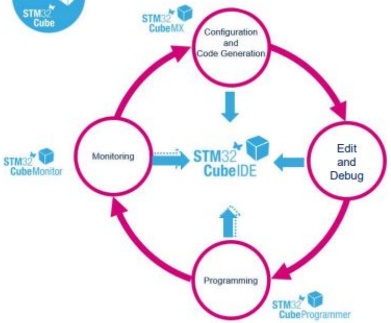
Software Overview

- IDE
- MCU
- CODEC
- User Interface
- DSP
- Dev board
- Libraries

STM32CubeIDE

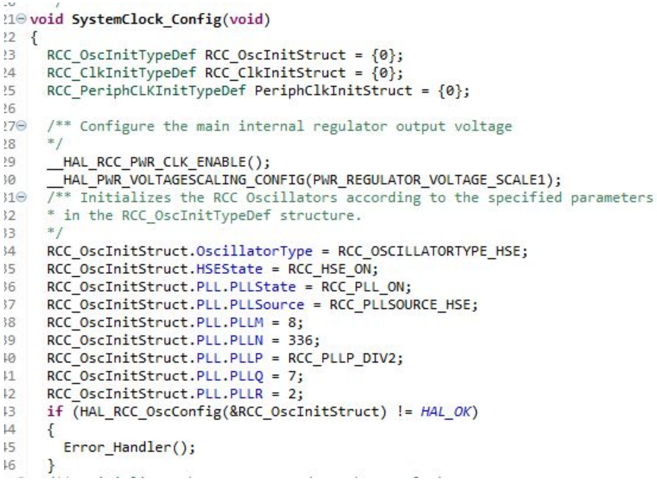
- Purpose:
  - Regularly updated
  - STM32CubeMX integrated into it
  - Pinout & Clock configuration
  - Auto-generated code
- Debugger mode
  - Memory details
- Multi OS support



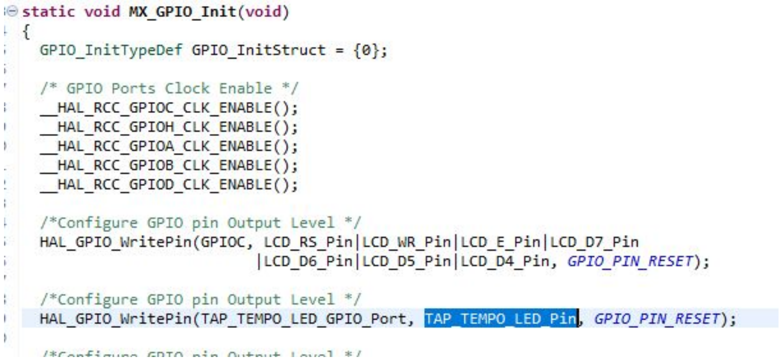


MCU - Overview

- Clock
  - HSE
- I2S
  - CODEC
- SPI
  - External Flash Memory
- GPIO
  - Filter parameters
  - Menu select
  - Tap tempo



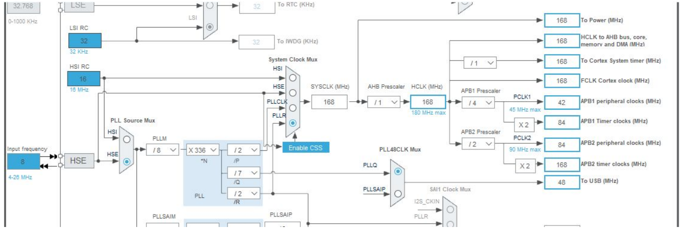
```
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
  RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 8;
  RCC_OscInitStruct.PLL.PLLN = 336;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 7;
  RCC_OscInitStruct.PLL.PLLR = 2;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
```



```
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
  __HAL_RCC_GPIOD_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOC, LCD_RS_Pin|LCD_WR_Pin|LCD_E_Pin|LCD_D7_Pin
                          |LCD_D6_Pin|LCD_D5_Pin|LCD_D4_Pin, GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(TAP_TEMPO_LED_GPIO_Port, TAP_TEMPO_LED_Pin, GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
```

HSE

- Purpose:
  - Prevent phase shift between CODEC and MCU
- Compatibility
  - 4 - 26 MHz
  - g min of 5 mA/V and Gm _crit_max of 1 mA/V.
- Desired Frequency
  - 8 MHz
- Auto clock config (STM32CubeIDE)



External Flash Memory

- Purpose:
  - Extra storage for audio
- Record audio at 44.1 kHz
  - 16-bit sample rate (MCU/Flash mem)
- Example: 1 second track recording
  - 1 sec * 44,100 samples/sec * 16 bit/sample = 705, 600 bits (88.2 kB of memory required)
- Common recording length
  - 1 - 5 minutes
  - ~10 MB per minute
- 200+ MB external memory

STM32 Nucleo Development Board

- Solid environment for testing and debugging
- Alternative when prototype board undergoes changes
- Same processor
- Cheap alternative for developing from home
- Easier learning curve of IDE and processor programming

Effects

1. Bitcrusher: decreases sample rate and bit depth
2. Chorus/Vibrato: time-based doubling effect that sinusoidally alters the pitch of the signal using delay lines
3. Compressor: alters the dynamics of the signal such that quiet sounds are increased and loud sounds are suppressed
4. Distortion: increases gain to the point of saturation
5. Delay: repeats signal impulse a certain number of times [feedback] at a specified rate [delay time]
6. Filter/Autowah: utilizes an envelope filter in tandem with a cutoff/resonance filter that will sweep through the filter frequencies upon the input of an impulse from the instrument

Effects (cont.)

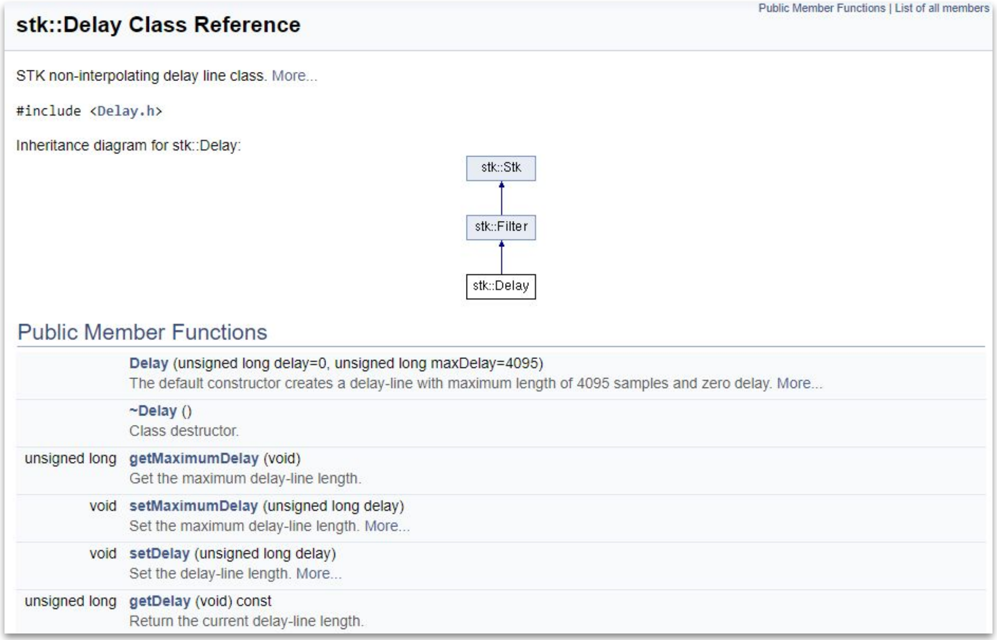
7. Flanger: similar to chorus, but with shorter delay lines
8. Looper: allows the user to record, play, overdub, stop/pause, erase musical loops
9. Phaser: sweeps through frequency spectrum at a specified rate and depth
10. Pitch Shifter/Harmonizer: alters the original pitch of the signal, with the ability to mix with dry signal for harmony
11. Reverb: similar to delay but more spatial/atmospheric, and with more delay lines
12. Tremolo: rapid increase and decrease in volume at a specified rate and depth

The Synthesis ToolKit Library (STK)

- To reproduce these effects in Code, it is necessary to perform a series of operations to the digitized sound bits
- The STK Library is an audio signal processing library written in C++
- It was designed to facilitate rapid development of audio synthesis software
- Contains several classes and functions to produce different sound effects

Effects Coding Implementation using STK Library

- Using Delay Implementation as Example
- Contains functions to manipulate delay parameters (like setting delay length)
- Distortion can be obtained by combining different functions



Public Member Functions | List of all members

stk::Delay Class Reference

STK non-interpolating delay line class. More...

```
#include <Delay.h>
```

Inheritance diagram for stk::Delay:

Public Member Functions

	Delay (unsigned long delay=0, unsigned long maxDelay=4095) The default constructor creates a delay-line with maximum length of 4095 samples and zero delay. More...
	~Delay () Class destructor.
unsigned long	**getMaximumDelay** (void) Get the maximum delay-line length.
void	**setMaximumDelay** (unsigned long delay) Set the maximum delay-line length. More...
void	**setDelay** (unsigned long delay) Set the delay-line length. More...
unsigned long	**getDelay** (void) const Return the current delay-line length.

User Interface

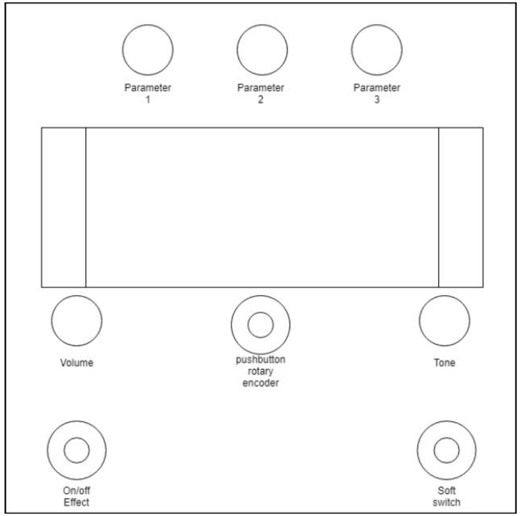
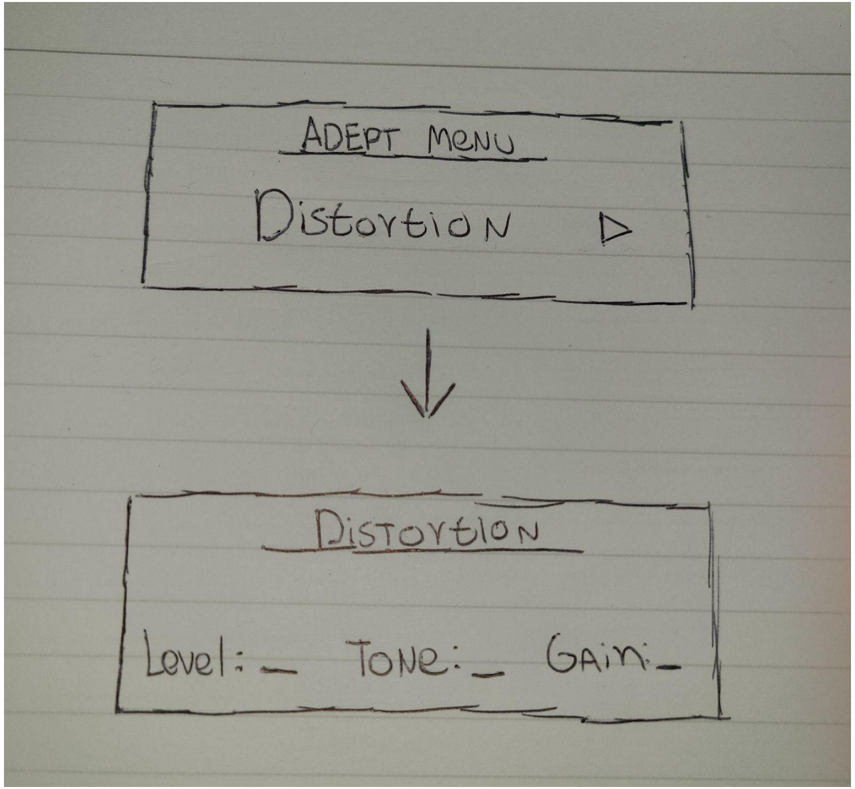


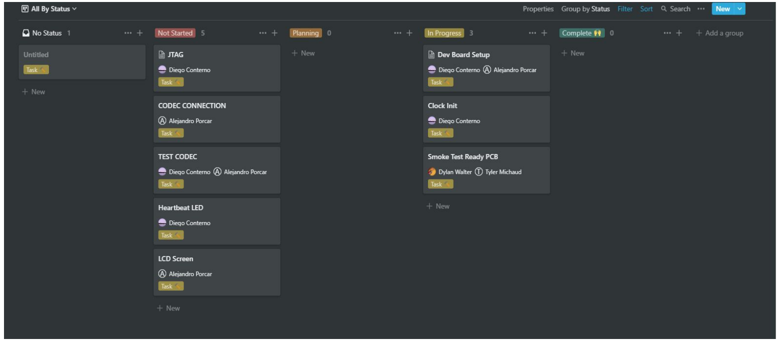
Figure 52: Prototype Diagram



Administrative Content

Project Planning Tools

- Notion
  - Project management tool
  - Easy record keeper
  - Notifications
  - Lightweight
- Google Drive
  - Storage
  - Share docs
- Discord



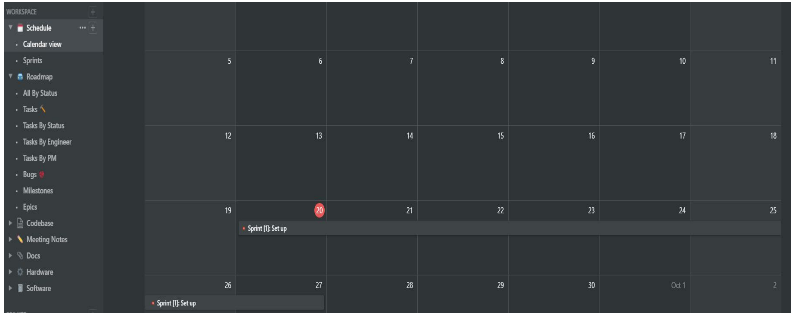
No Status 1
- Untitled — Task

Not Started 5
- JTAG — Diego Conterno — Task
- CODEC CONNECTION — Alejandro Porcar — Task
- TEST CODEC — Diego Conterno, Alejandro Porcar — Task
- Heartbeat LED — Diego Conterno — Task
- LCD Screen — Alejandro Porcar — Task

Planning 0

In Progress 3
- Dev Board Setup — Diego Conterno, Alejandro Porcar — Task
- Clock Init — Diego Conterno — Task
- Smoke Test Ready PCB — Dylan Walter, Tyler Michaud — Task

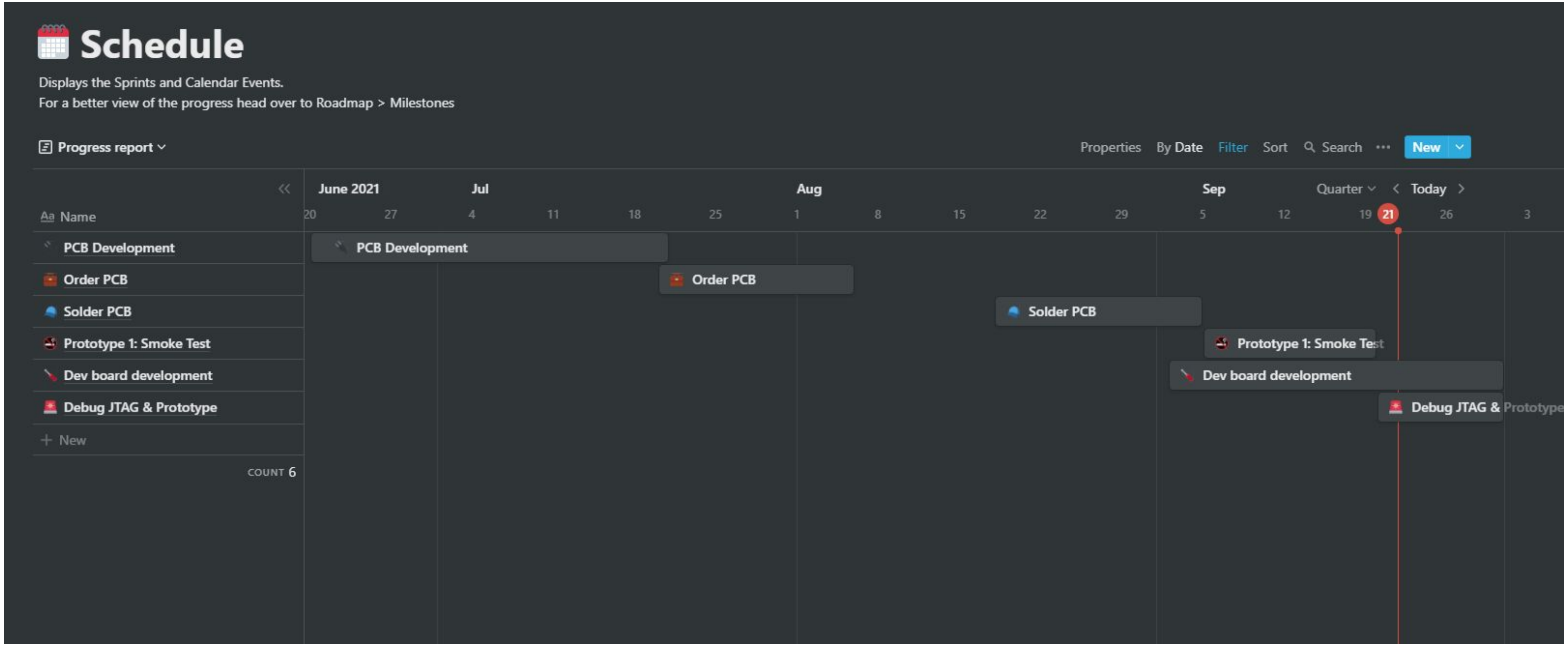
Complete 🙌 0



Sprint [1]: Set up

Sprint [1]: Set up

Progress Chart



Schedule

Displays the Sprints and Calendar Events.
For a better view of the progress head over to Roadmap > Milestones

Progress report

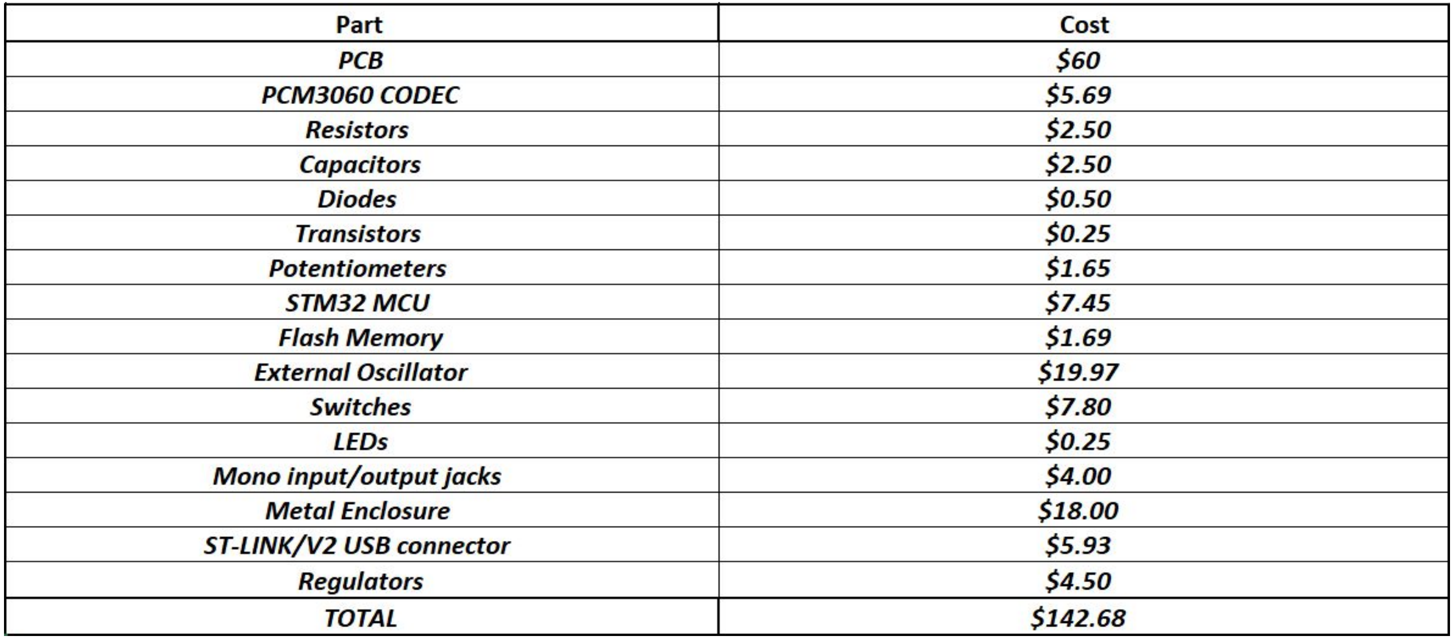
Name

- PCB Development
- Order PCB
- Solder PCB
- Prototype 1: Smoke Test
- Dev board development
- Debug JTAG & Prototype

+ New

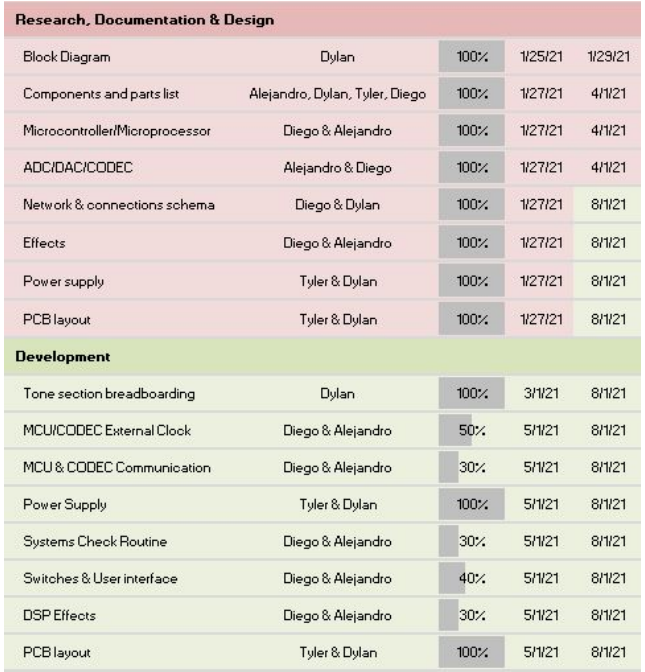
COUNT 6

Cost of Materials



Part	Cost
PCB	*$60*
PCM3060 CODEC	*$5.69*
Resistors	*$2.50*
Capacitors	*$2.50*
Diodes	*$0.50*
Transistors	*$0.25*
Potentiometers	*$1.65*
STM32 MCU	*$7.45*
Flash Memory	*$1.69*
External Oscillator	*$19.97*
Switches	*$7.80*
LEDs	*$0.25*
Mono input/output jacks	*$4.00*
Metal Enclosure	*$18.00*
ST-LINK/V2 USB connector	*$5.93*
Regulators	*$4.50*
TOTAL	*$142.68*

Progress percentage



Research, Documentation & Design				
Block Diagram	Dylan	100%	1/25/21	1/29/21
Components and parts list	Alejandro, Dylan, Tyler, Diego	100%	1/27/21	4/1/21
Microcontroller/Microprocessor	Diego & Alejandro	100%	1/27/21	4/1/21
ADC/DAC/CODEC	Alejandro & Diego	100%	1/27/21	4/1/21
Network & connections schema	Diego & Dylan	100%	1/27/21	8/1/21
Effects	Diego & Alejandro	100%	1/27/21	8/1/21
Power supply	Tyler & Dylan	100%	1/27/21	8/1/21
PCB layout	Tyler & Dylan	100%	1/27/21	8/1/21
Development				
Tone section breadboarding	Dylan	100%	3/1/21	8/1/21
MCU/CODEC External Clock	Diego & Alejandro	50%	5/1/21	8/1/21
MCU & CODEC Communication	Diego & Alejandro	30%	5/1/21	8/1/21
Power Supply	Tyler & Dylan	100%	5/1/21	8/1/21
Systems Check Routine	Diego & Alejandro	30%	5/1/21	8/1/21
Switches & User interface	Diego & Alejandro	40%	5/1/21	8/1/21
DSP Effects	Diego & Alejandro	30%	5/1/21	8/1/21
PCB layout	Tyler & Dylan	100%	5/1/21	8/1/21

Current Issues and Future Solutions

- JTAG connection
  - Currently working on troubleshooting pinout issues along with ST-link configuration.
  - Solution: Record of bugs, forums, update prototype board. Later PCB revisions may fix this issue.
  - Parallel development: Dev board, breakout printed circuit board.
- LCD display configuration
  - The LCD backlight and contrast pins were not connected in the initial design.
  - Solution: Solder the connection on the PCB using magwire for now. Later PCB revisions will fix this issue.
- Rotary encoder
  - Only able to turn in one direction due to GPIO pin not being mapped properly
  - Solution: Solder the connection on the PCB using magwire for now. Later PCB revisions will fix this issue.
- Software development
  - Need to start assembling libraries, creating LCD menus, mapping parameter knobs, and programming effects algorithms

Thank you!

Questions?