Real-Time Sound Visualization

Team 43
Lin Le, Qian Chen, Xinyue Yu

Introduction

- A hepler for music lover.

- A device that recognizes music notes.

- Real time display.

Outline

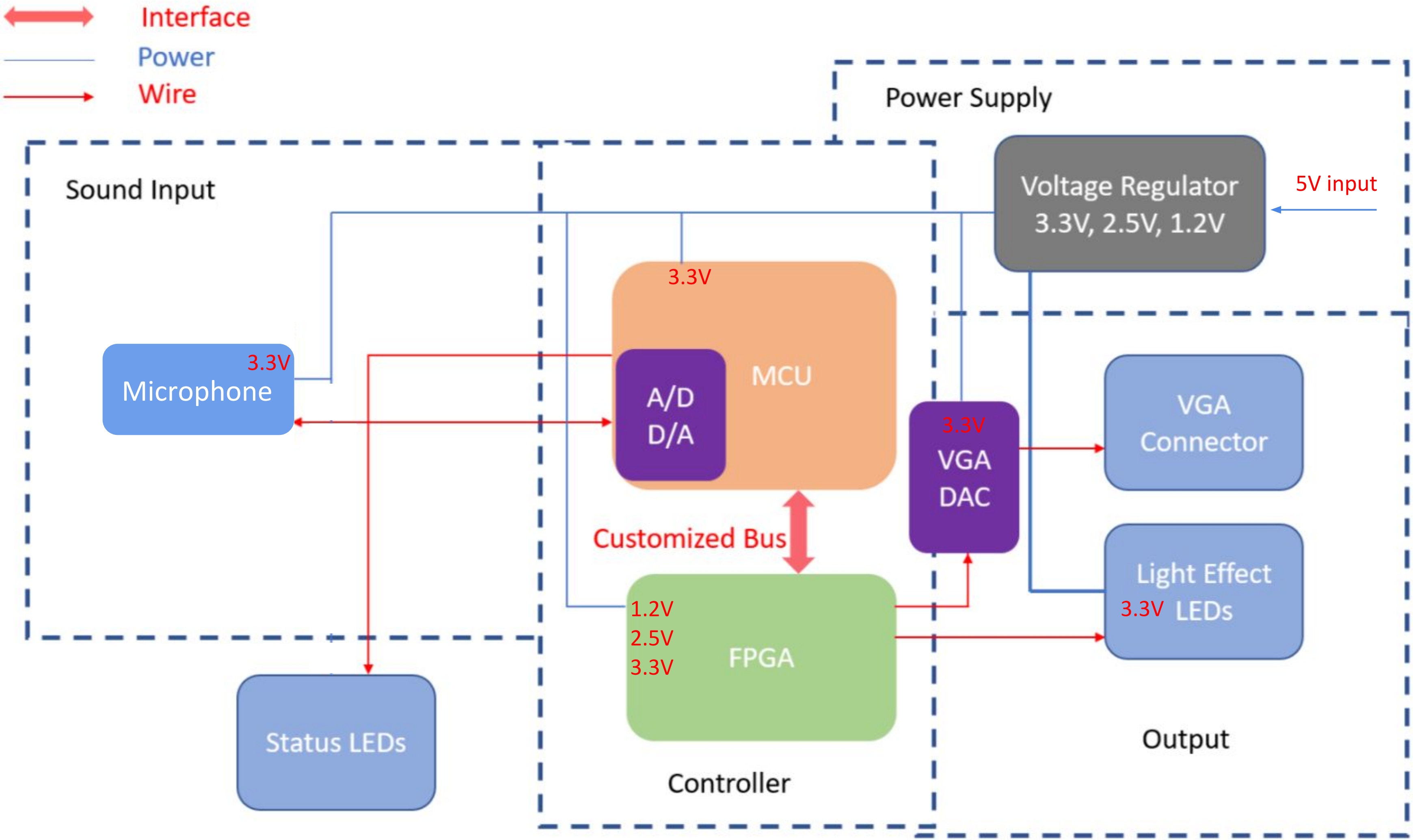
- Objective
- Review of original design, requirements, and verifications
- Functionalities & Tests
- Successes and Failure; Challenges
- Future work

Objective

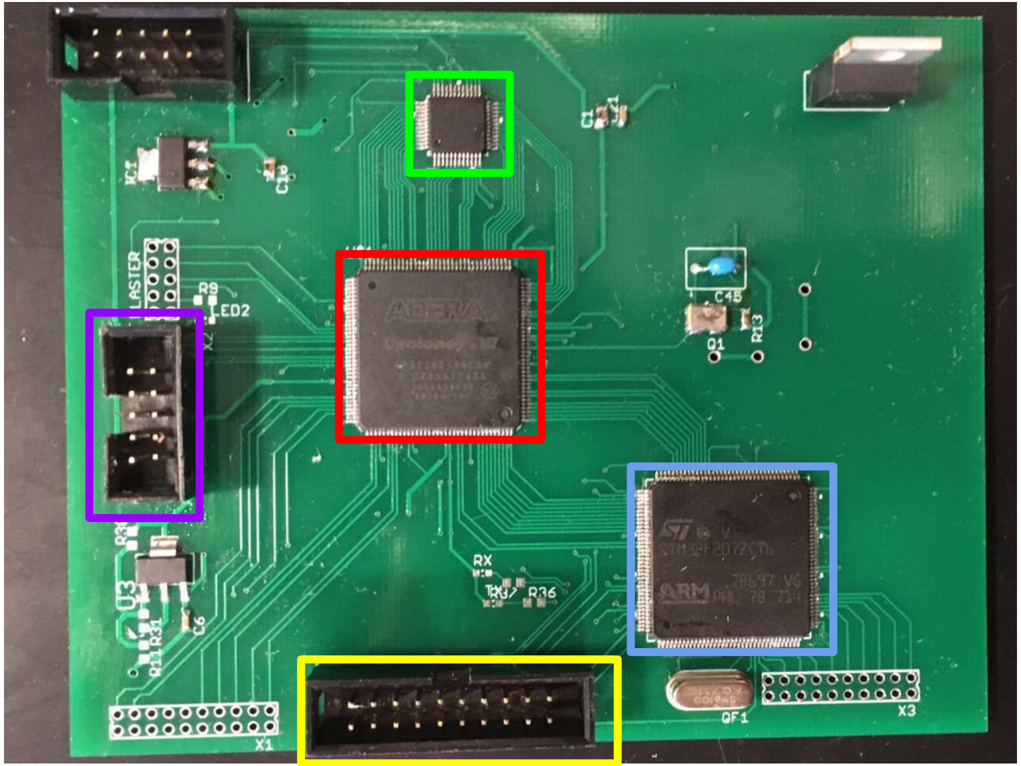
- Capture music signal.
- Detect the correct pitch.
- Display on screen (VGA port).

System Design

- Perform Fast Fourier Transform(FFT) on the sound data in MCU
- Displayed Music transcript on the VGA display using FPGA
- Mounted all the component on a PCB (printed circuit board).

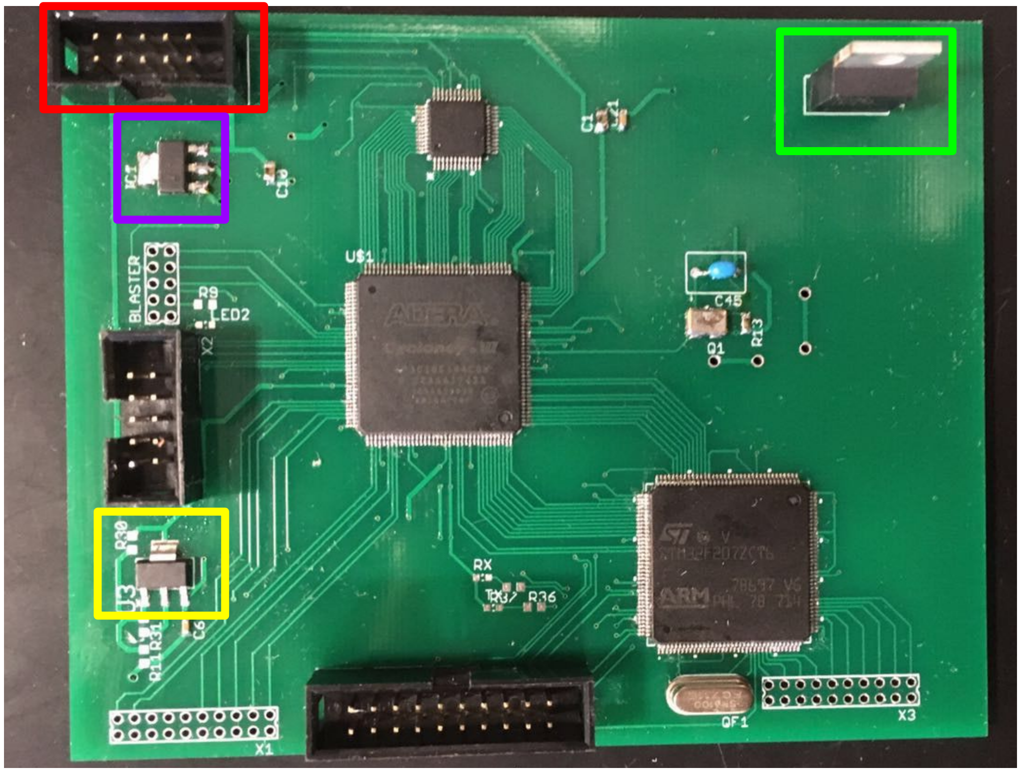

Interface
Power
Wire
Power Supply
Voltage Regulator
3.3V, 2.5V, 1.2V
5V input
Sound Input
Microphone
3.3V
3.3V
A/D
D/A
MCU
Customized Bus
3.3V
VGA
DAC
VGA
Connector
Light Effect
3.3V LEDs
1.2V
2.5V
3.3V
FPGA
Status LEDs
Controller
Output

PCB design



- STM32F2 MCU
- Cyclone III FPGA
- VGA DAC
- ST-Link
- J-tag

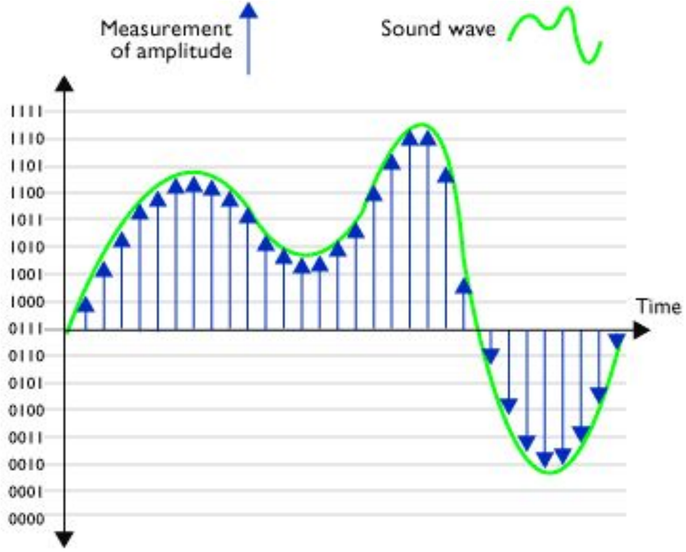
PCB design



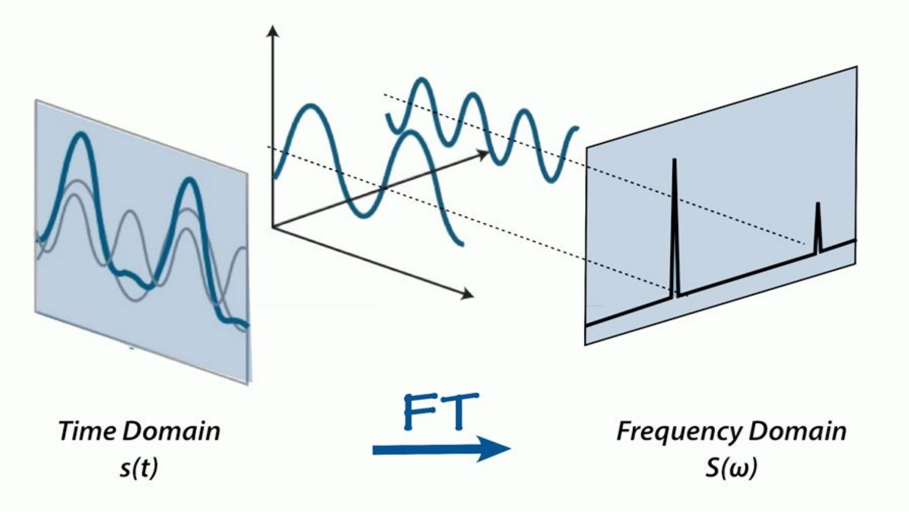
- 3.3V voltage regulator
- 2.5V voltage regulator
- 1.25V voltage regulator
- 10 Pin header
  - 5V input
  - VGA output
  - 3.3v, 2.5v, 1.25v test points

MCU and pitch detection

- Analog to digital conversion
- Fourier Transform

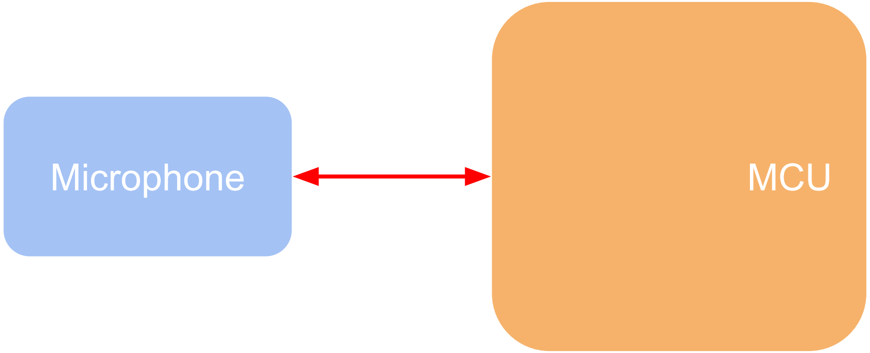


Each measurement is assigned a number (byte) according to its amplitude. The end result is a file comprising a string of bytes, eg ...
1001 1110 0001 1010 0111 0100 1111 1101 etc



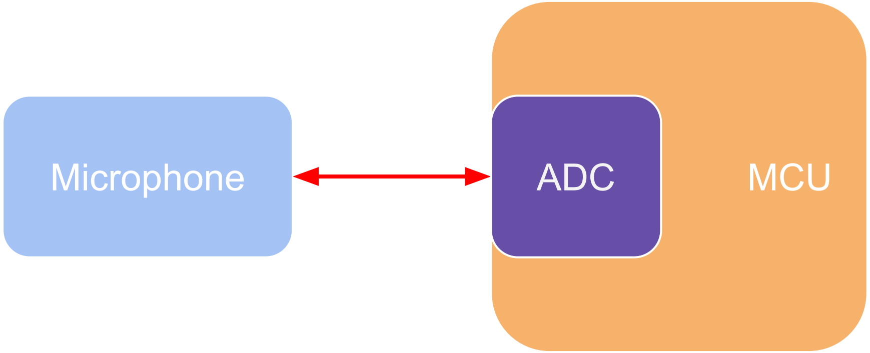
MCU and pitch detection

- 12 bits resolution A/D conversion
- Sample frequency at 22.1kHz
- 4096 points FFT



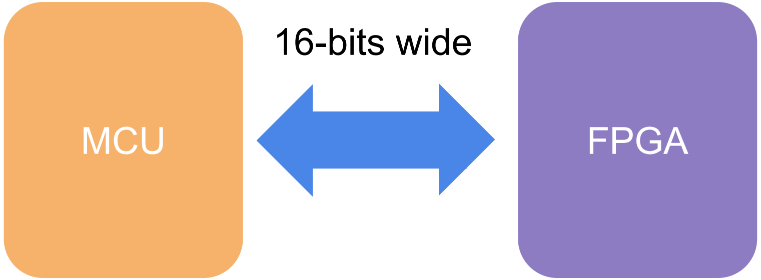
MCU and pitch detection

- Interruption after every sample
- Find the strongest peak in frequency domain
- Write the output into data bus



MCU: Outputting result

- MCU connect to FPGA through a 16-bits customized data bus
- The high 8 bits for frequency
- The low 8 bits for intensity

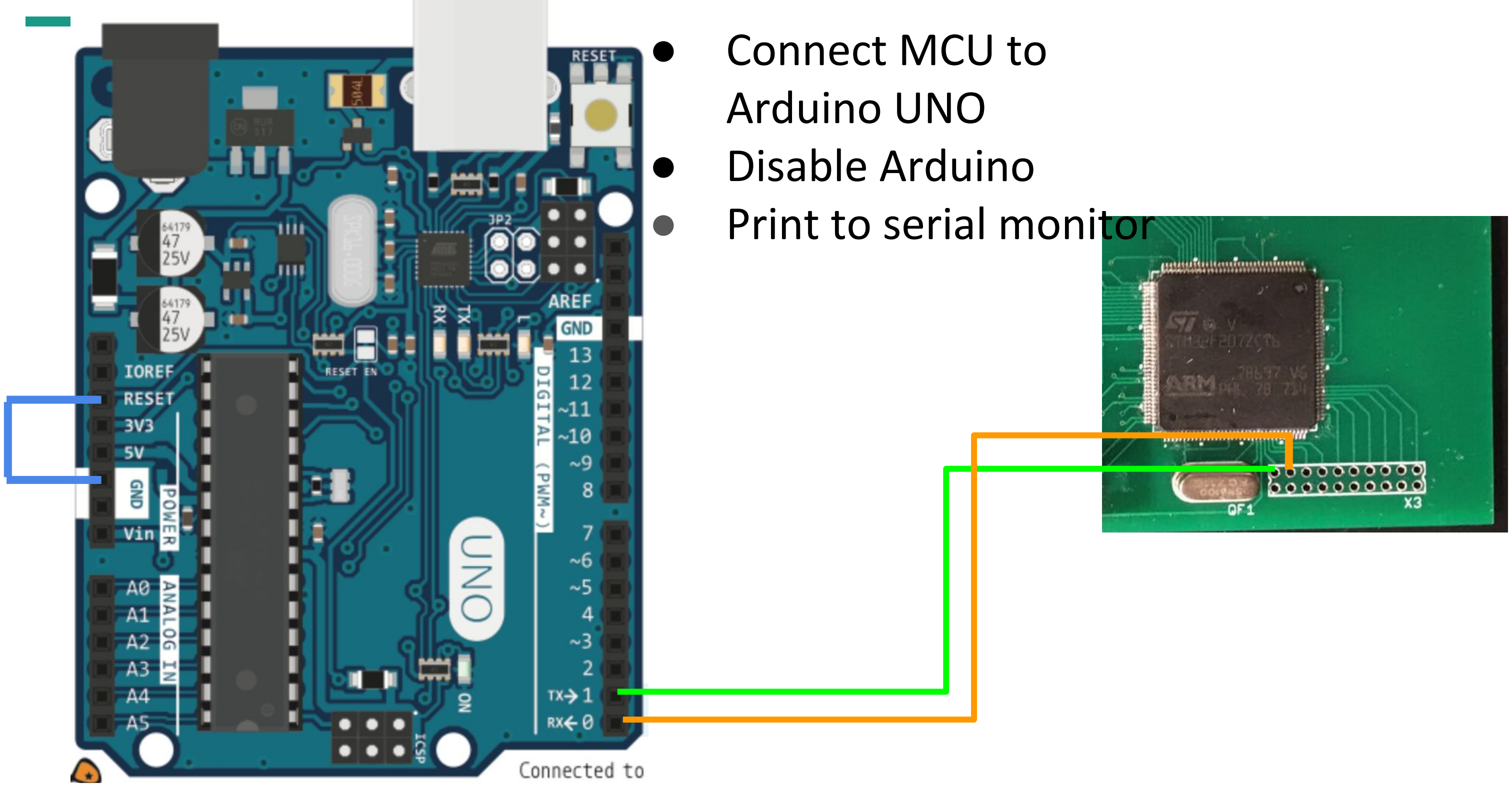


Other Test: MCU printing

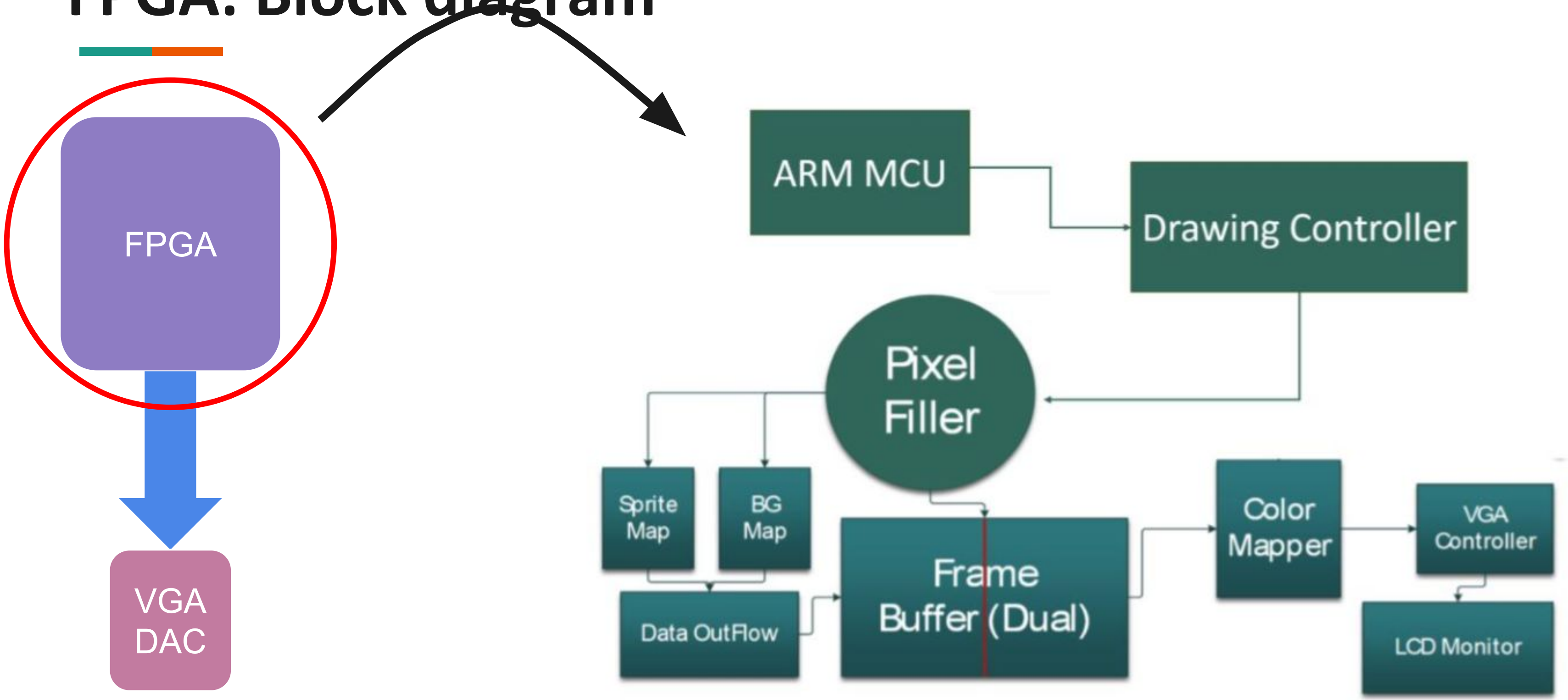
- How to print from MCU?
  - MCU support UART (universal asynchronous receiver-transmitter)
  - Redirect the printf() function
- Need usb to UART bridge
  - Arduino as a UART bridge

Other Test:

- Connect MCU to Arduino UNO
- Disable Arduino
- Print to serial monitor



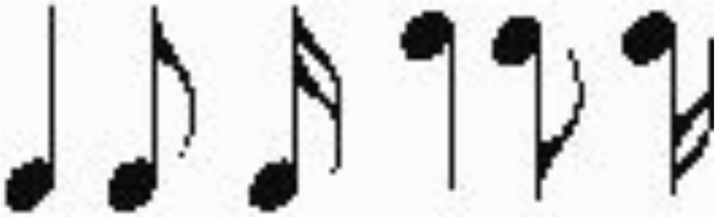
FPGA: Block diagram

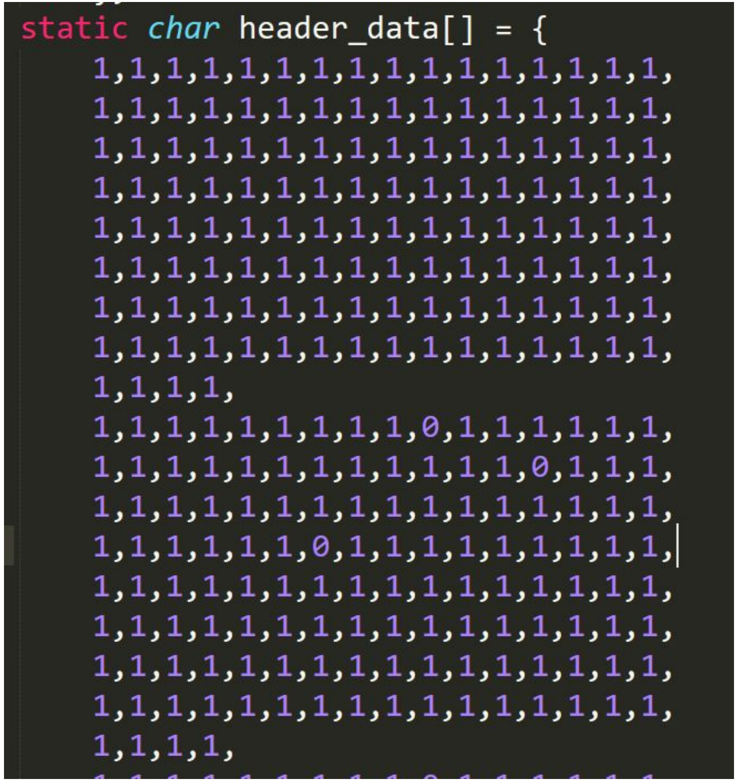


FPGA: Drawing logic

- Get background color indices
- Find music note on sprite map
- Write into frame buffer.

FPGA: Sprite Map



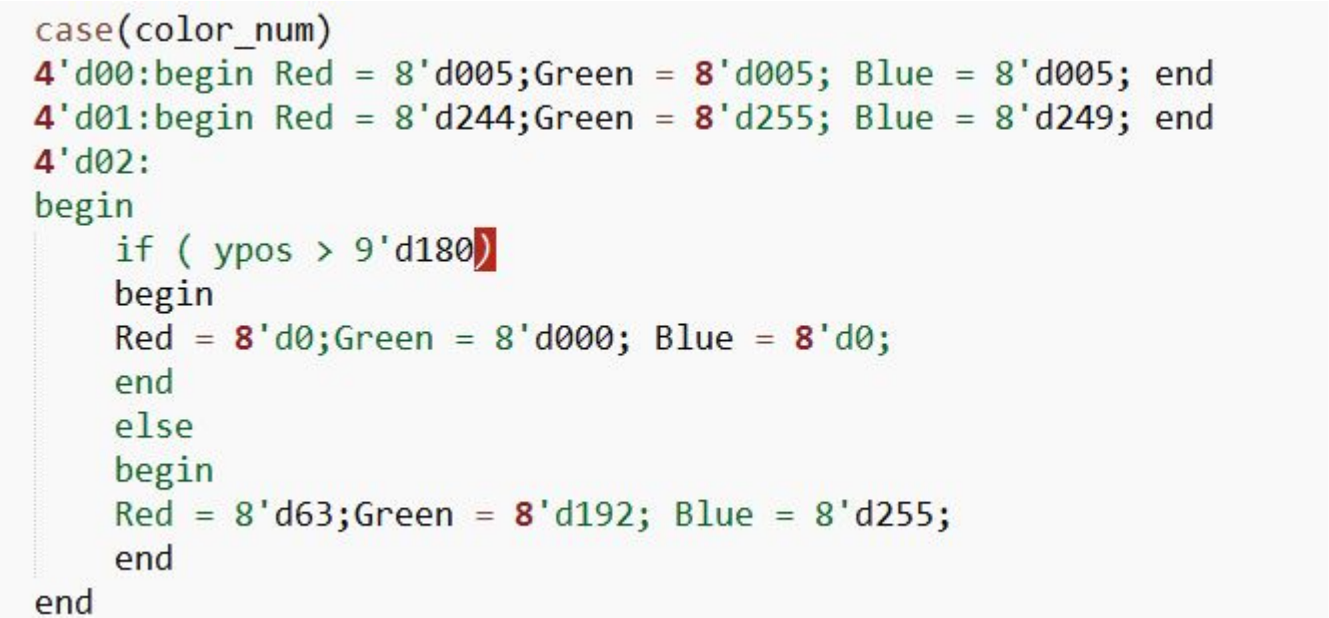


```
static char header_data[] = {
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,
	1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
	1,1,1,1,
```

FPGA: Background

FPGA: Color Mapper

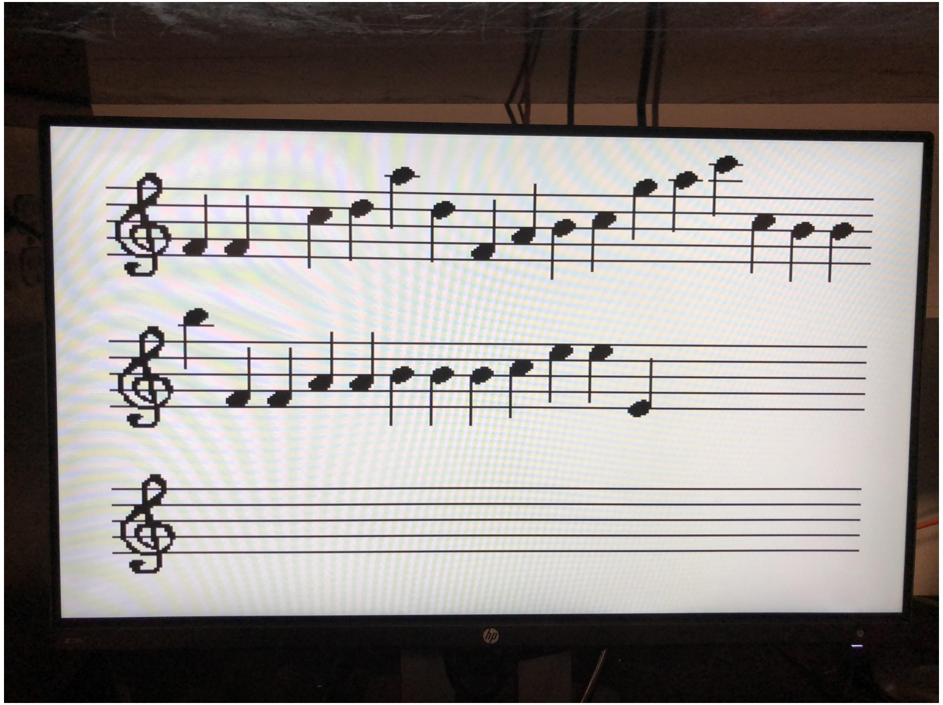
- Black
- White
- Transparent



```
case(color_num)
4'd00:begin Red = 8'd005;Green = 8'd005; Blue = 8'd005; end
4'd01:begin Red = 8'd244;Green = 8'd255; Blue = 8'd249; end
4'd02:
begin
    if ( ypos > 9'd180)
    begin
    Red = 8'd0;Green = 8'd000; Blue = 8'd0;
    end
    else
    begin
    Red = 8'd63;Green = 8'd192; Blue = 8'd255;
    end
end
```

FPGA: Output

- Display on screen



FPGA: Output

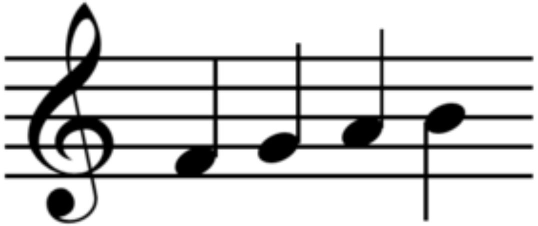
- Volume unit indicator

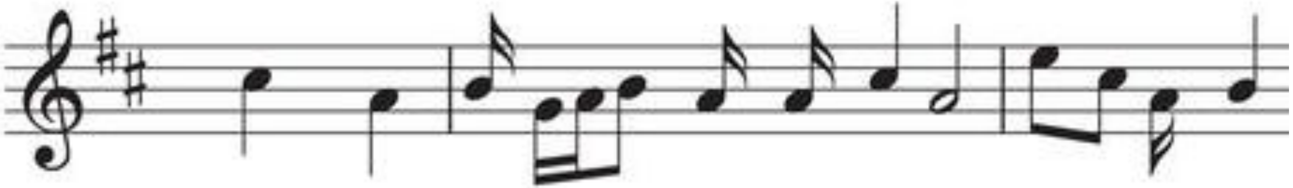
FPGA: Display Test

- We used the switch to mimic the input notes.
    - Out of range note
    - Change lines

Software Future Work

- Now : quarter note
- Future: Variables of rhythm

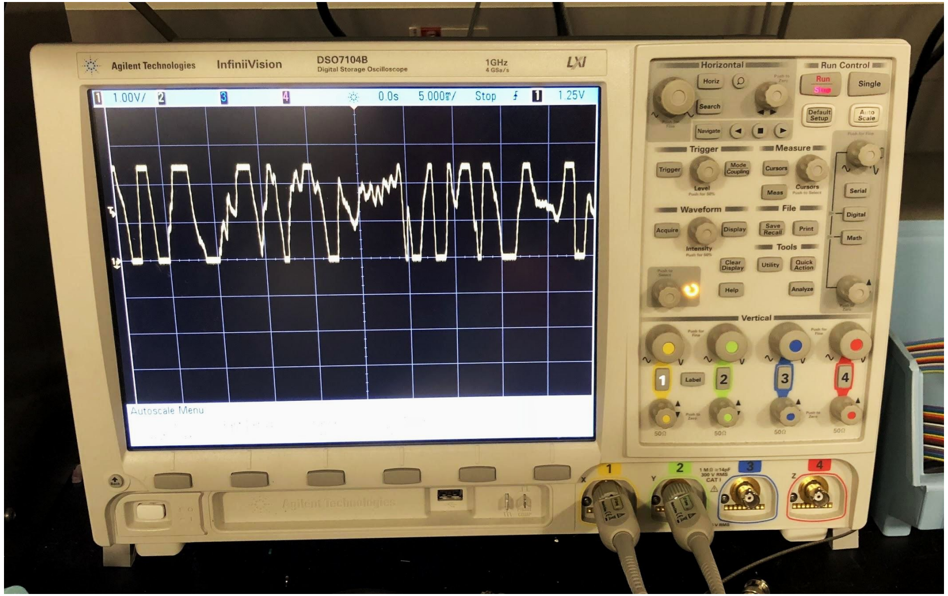




Requirements And Verification

- Microphone
- VGA
- MCU
- FPGA
- Power Supply
- Other

R&V: Microphone



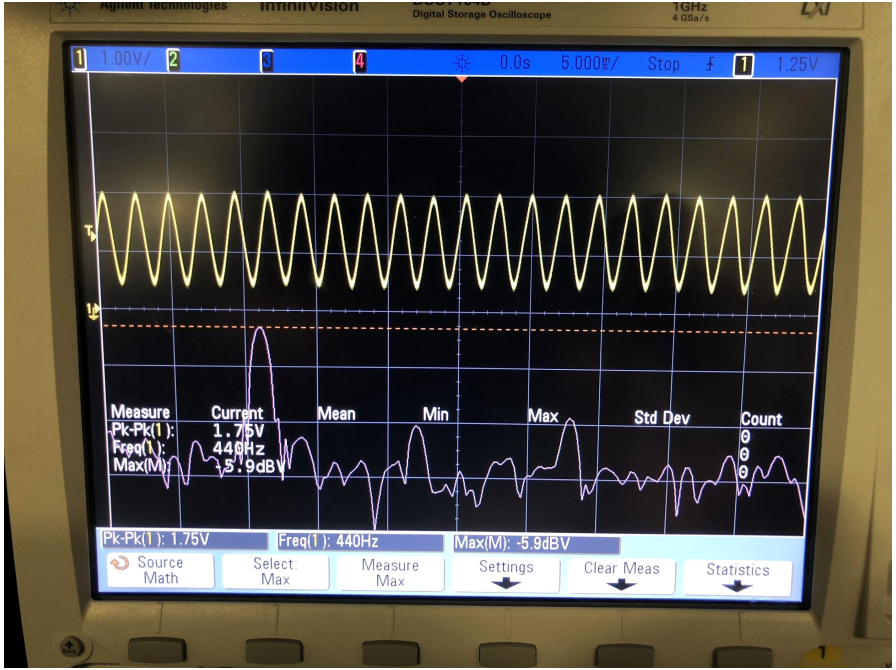
Requirement

- □ Microphone module outputs voltage between 0 to 3.3v.

Verification

- □ Read the output voltage through oscilloscope

R&V: Microphone



```
Measure      Current    Mean   Min   Max   Std Dev   Count
Pk-Pk(1):    1.75V                                   0
Freq(1):     440Hz                                   0
Max(M):      -5.9dBV                                 0
```

```
Pk-Pk(1): 1.75V    Freq(1): 440Hz    Max(M): -5.9dBV
```

Requirement

❏ The output of microphone must have a signal to noise ratio of at least 10 dB.

Verification

❏ Connect the microphone to MCU through the Audio jack to read data from oscilloscope.

R&V: LEDs

Requirement

❑ The LED has noticeable light at input voltage 3.3v.

Verification

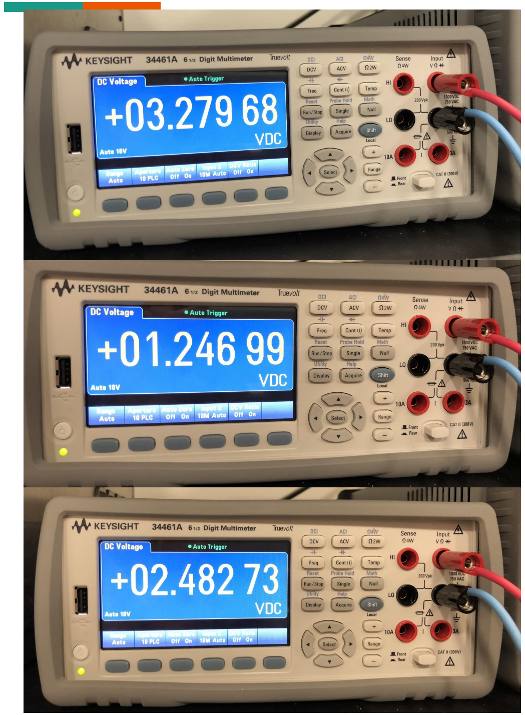
❑ Hard code FPGA, set the output GPIO pins to logic 1, to control LED blinking.

R&V: LEDs

Result

❑ The LEDs are successfully driven by FPGA. Those LEDs work as a Volume Unit meter (VU meter).

R&V: Power supply



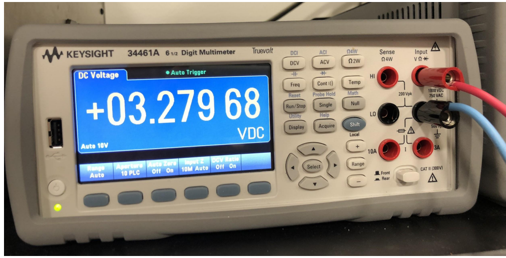
Requirement

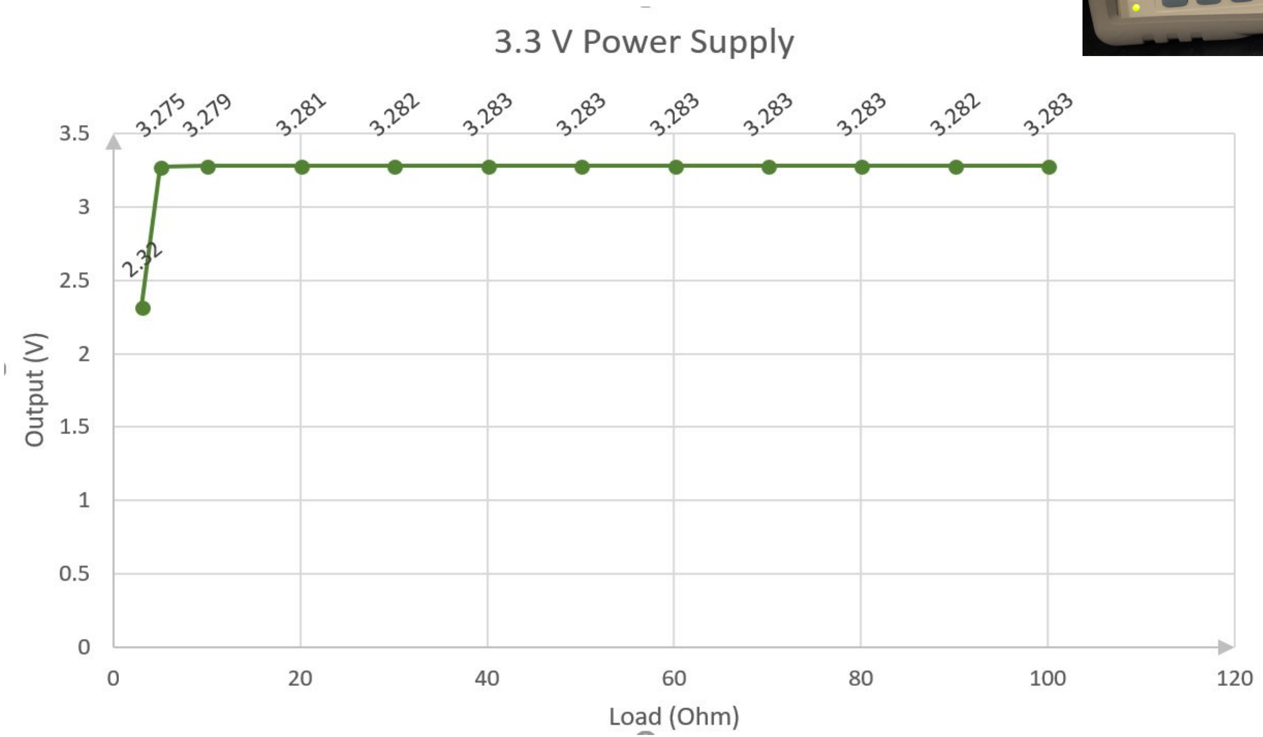
❑ Three power supply
1.25v +/- 5%, maximum 1A
2.5v +/- 5%, maximum 800mA
3.3v +/- 5%, maximum 800mA

Verification

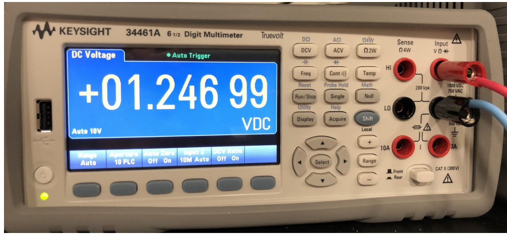
❑ Connect output to different load. Read voltage across output with multimeter.

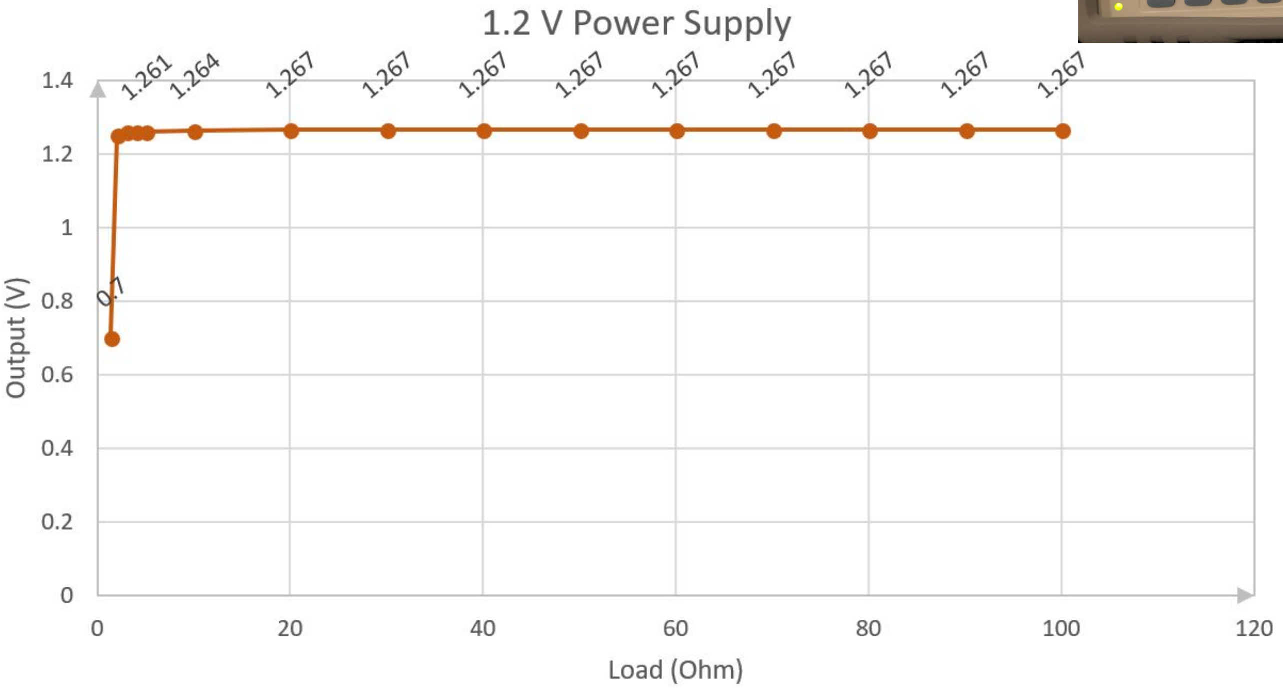
3.3V


KEYSIGHT
34461A 6½ Digit Multimeter
Truevolt
DC Voltage
Auto Trigger
+03.279 68
VDC
Auto 10V

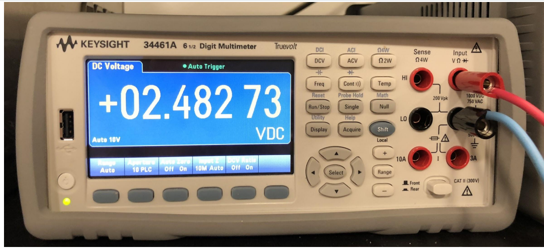

3.3 V Power Supply
Output (V)
Load (Ohm)
0
0.5
1
1.5
2
2.5
3
3.5
0
20
40
60
80
100
120
2.32
3.275
3.279
3.281
3.282
3.283
3.283
3.283
3.283
3.283
3.282
3.283

2.5V

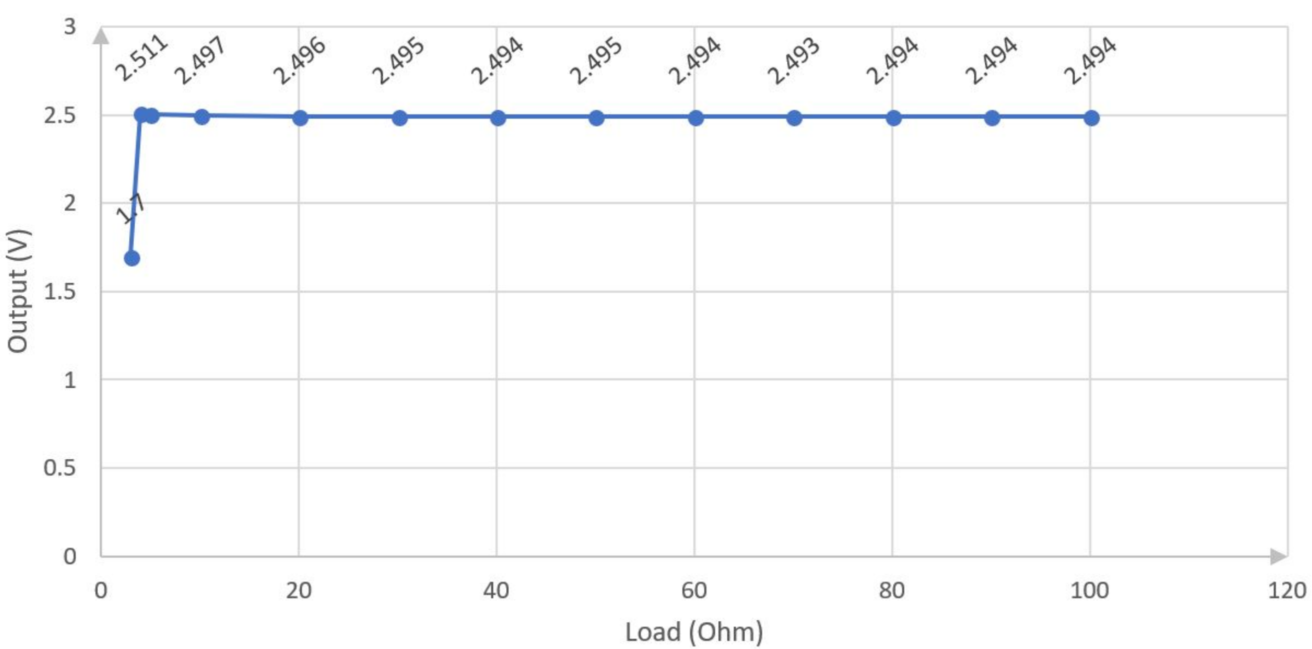

KEYSIGHT 34461A 6½ Digit Multimeter
DC Voltage
Auto Trigger
+01.246 99
VDC
Auto 10V


1.2 V Power Supply
Output (V)
Load (Ohm)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
20
40
60
80
100
120
0.7
1.261
1.264
1.267
1.267
1.267
1.267
1.267
1.267
1.267
1.267
1.267

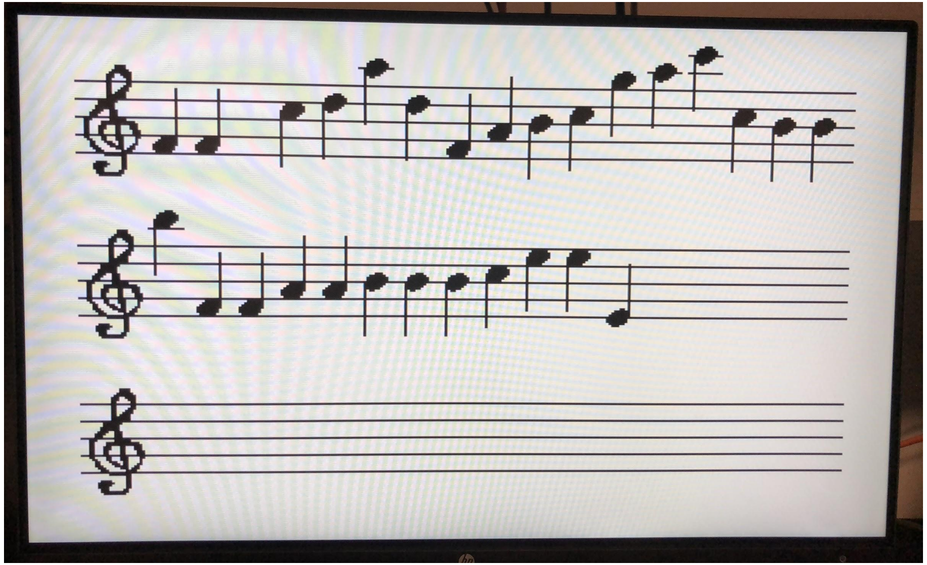
1.25V



2.5 V Power Supply



R&V: VGA



Requirement

❏ VGA output @ resolution 640 * 480.

Verification

❏ Connect a display through VGA port.

R&V: MCU

Requirement

❑ Sampling sound data with A/D module at sampling frequency 22.1kHz.

Verification

❑ Connect the Wave generator to MCU through GPIO.

R&V: MCU continue

Requirement

❑ Perform pitch detection, delay < 250 ms

Verification

❑ Perform the pitch detection algorithm on the collected data, print result through serial port

R&V: MCU continue

Restul

❑ 22.1khz sampling frequency

❑ The 4096 points FFT is performed very fast. Less than 5ms.

R&V: FPGA

Requirement

❑ Output the correct imageand display it on VGA monitor.

Verification

❑ Ensure the output VGA signal is 640 * 480 at 60Hz frame

Conclusion & Discussion

❑ Correct output voltage in power supply

❑ MCU tested and verified

☑ Detect pitch for monotone ☒ No duration and not good for multitone sound

❑ FPGA tested and verified

☑ Draw notes on stave and output on monitor

☒ Only one type of note

Future Work

❑ Try to use autocorrelation algorithm

❑ Use energy of frequency component

❑ Multiple music input compatibility. Eg, piano, human, guitar

❑ Add music rhythm into transcript

Allegro.
p
Allegro.
p

Credit:

❑ Professor Chen

❑ Dongwei Shi

❑ All 445 staff

❑ ECE department

Thank you!

Questions?