Lab Members: Christina Kim & Evan Oskierko-Jeznacki

Group: 17

Due Date: 22 September 2018

# Writeup

## 1.2

Take a screenshot of the output 440Hz square wave waveform for your lab report

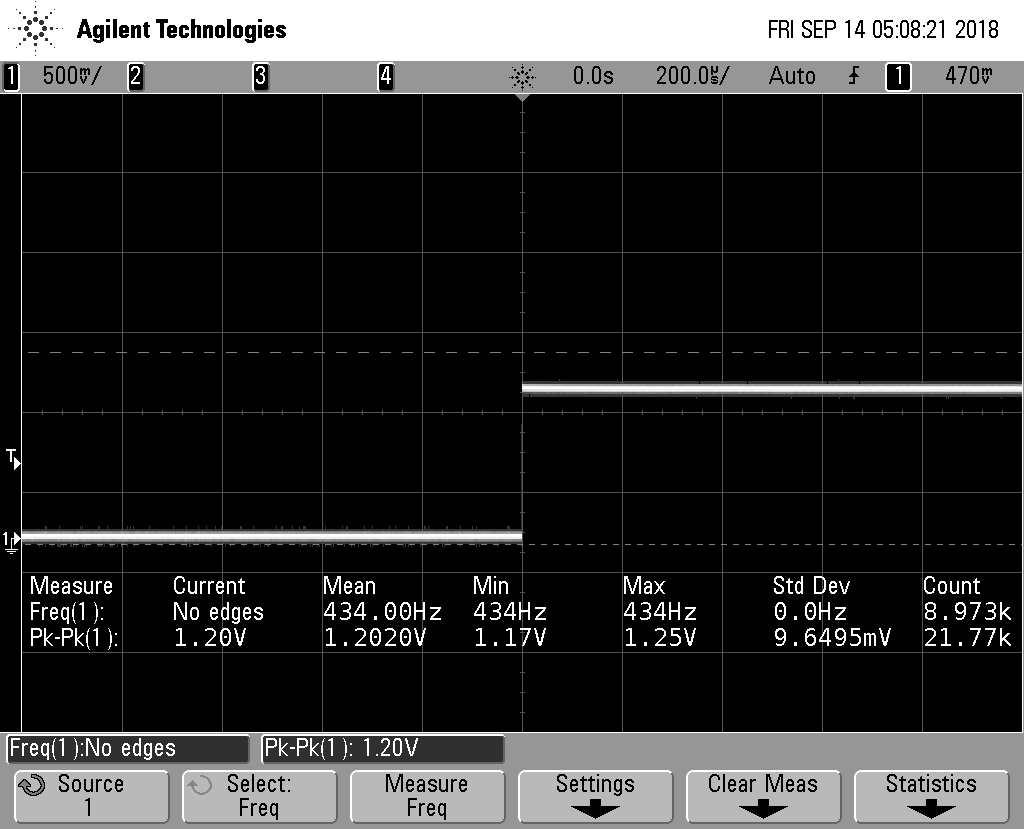
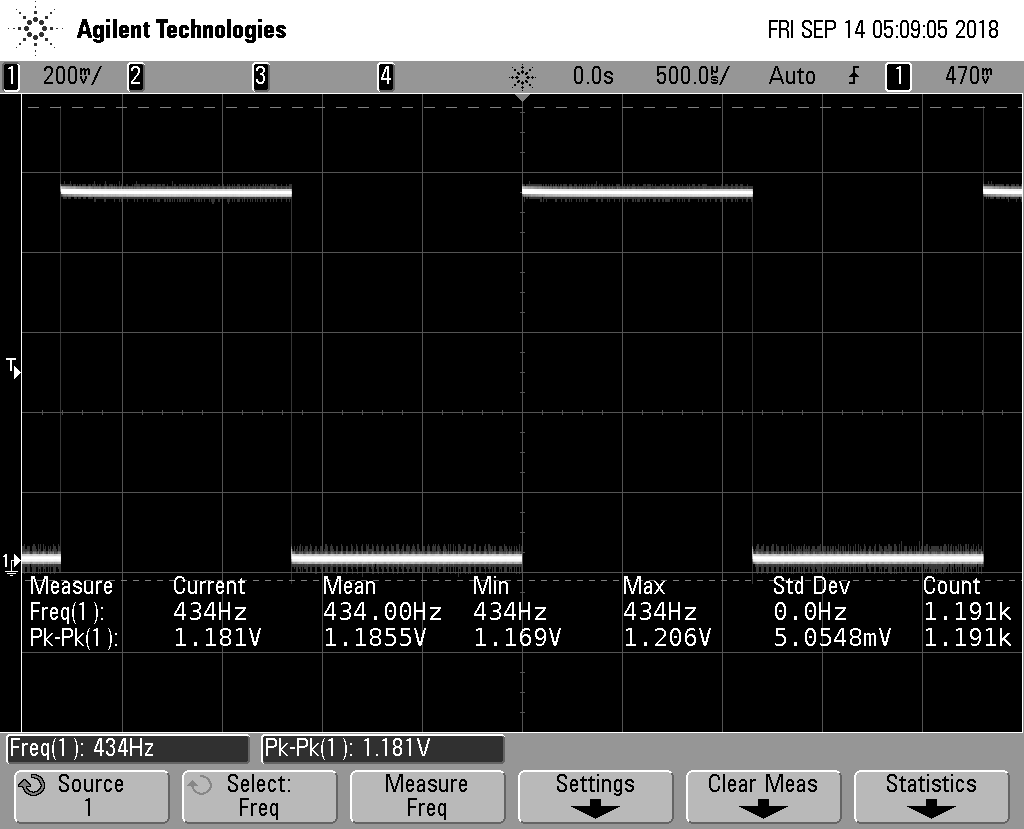


Figure (Left) Image of 440Hz square wave. (Right) Detail image of 440Hz square wave.

## 2.1

Note that the timer can only generate one frequency when in CTC mode. Why is that?

The timer can only generate one frequency when in CTC mode because TCNT0 is toggling (resetting) between OCROA value (frequency) and zero such that TCNT0 will reset after compare. This is true as long as OCROA is fixed and set equal to TCNT0.

## 2.2

Record the largest and smallest pulses you can generate with the rangefinder consistently.

|  |  |  |
| --- | --- | --- |
|  | **Distance** | **Pulse Width** |
| **Minimum** | 2cm  *(recommended as per device specifications)* | 95 μs |
| **Maximum** | 160cm  *(limited by table/lab space)* | 827 μs |

## 2.3

For the frequencies of each note in the table, calculate a proper value for the prescaler and OCR0A register with Timer0 in CTC mode. You will have to round a little bit. Choose a prescaler that will allow you to minimize the rounding for the whole range.

Prescaler was chosen to be 64. While 1024 prescaler would work best based on our calculations, it did not work for other lab parts requiring us to use a 64 prescaler. These numbers were based using C6 calculations because of it being the lowest note in the range. Thus, driving the choice of the prescaler as it will require the largest number of clock ticks between pulses (i.e. lower frequency, larger wavelength). Please see below for process.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measurements** | **C6** | **D6** | **E6** | **F6** | **G6** | **A6** | **B6** | **C7** |
| **Frequency** | 1046.5 | 1174.66 | 1318.51 | 1496.91 | 1567.98 | 1760 | 1975.53 | 2093 |
| **Pulse Width** | 95 | 199.57 | 304.14 | 408.71 | 513.28 | 617.85 | 722.42 | 826.99 |
| **OCROA** | 7643.5 | 6809.5 | 6066.5 | 5725.9 | 5105.1 | 4544.5 | 4048.5 | 3821.3 |

**Calculation Process:**

>7643.5294 therefore 1024 would work for this part

>7643.5294 therefore 64

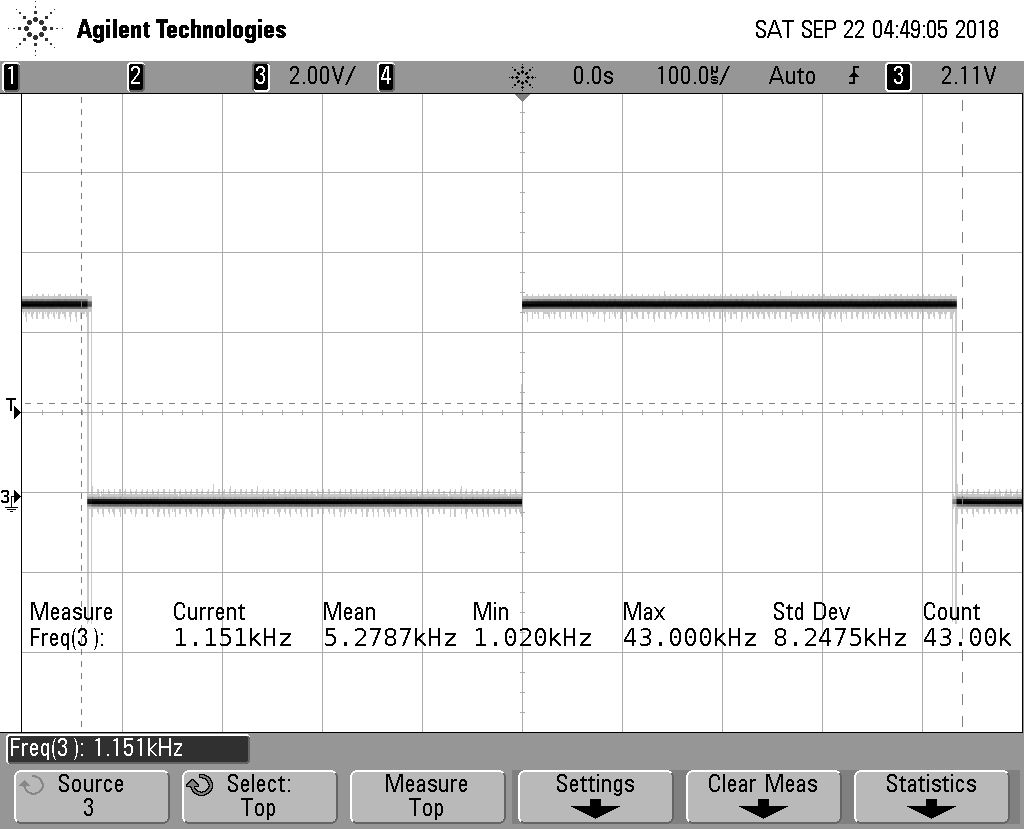
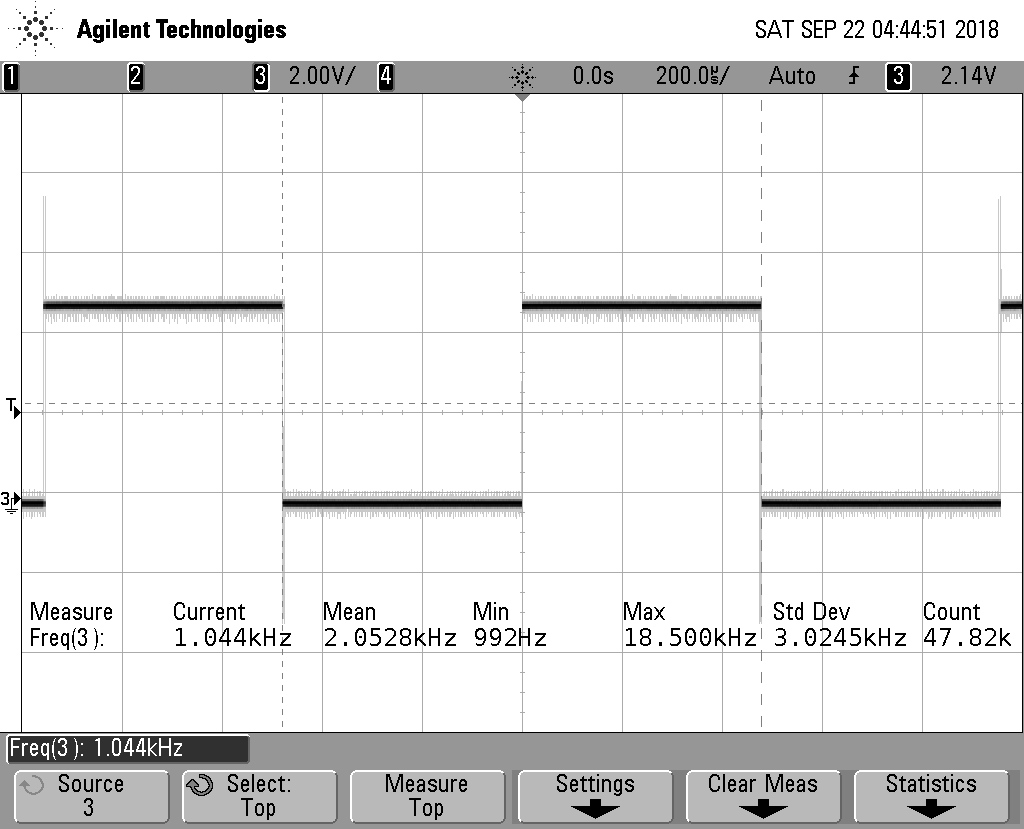
\*Note: All values were rounded as seen in the table after all calculations

## 2.4

For each of the notes in the octave, take a screenshot of the output waveform.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measurements** | **C6** | **D6** | **E6** | **F6** | **G6** | **A6** | **B6** | **C7** |
| **OCROA** | 7643.5 | 6809.5 | 6066.5 | 5725.9 | 5105.1 | 4544.5 | 4048.5 | 3821.3 |
| **Clock Ticks/2** | 8 | 9 | 10 | 11 | 13 | 14 | 16 | 17 |

\*Note: Clock Ticks were divided by 2 for lowering the octave (different buzzer used causing same notes an octave higher)



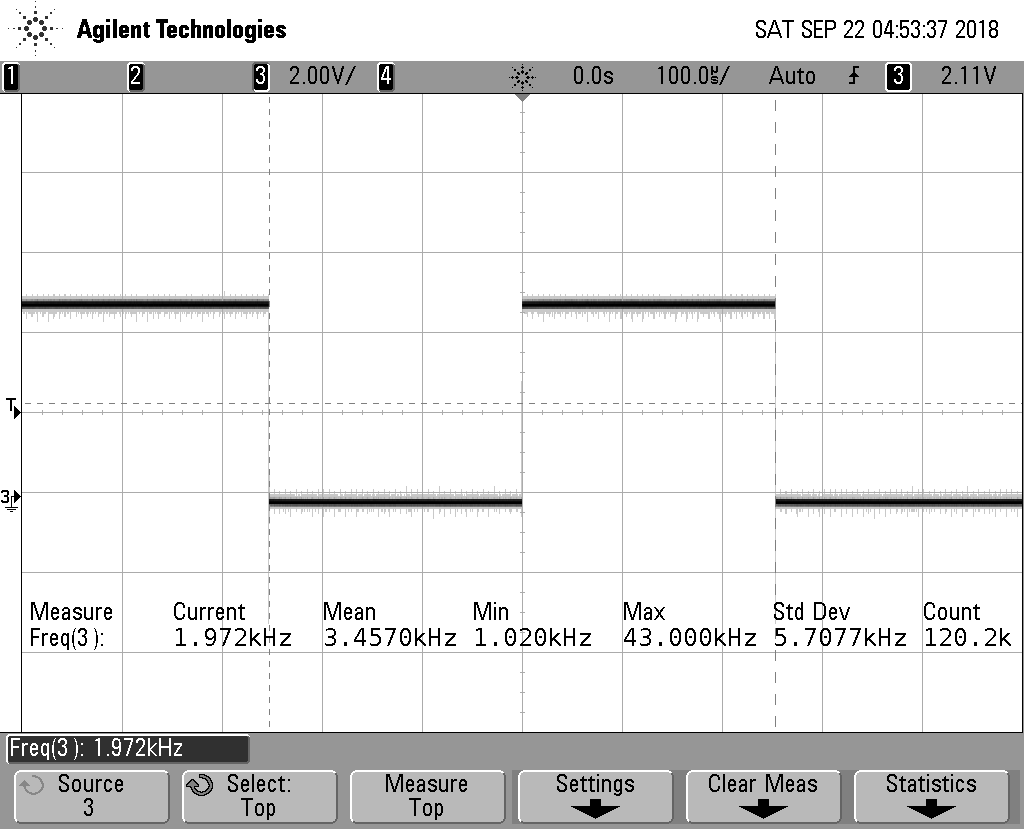
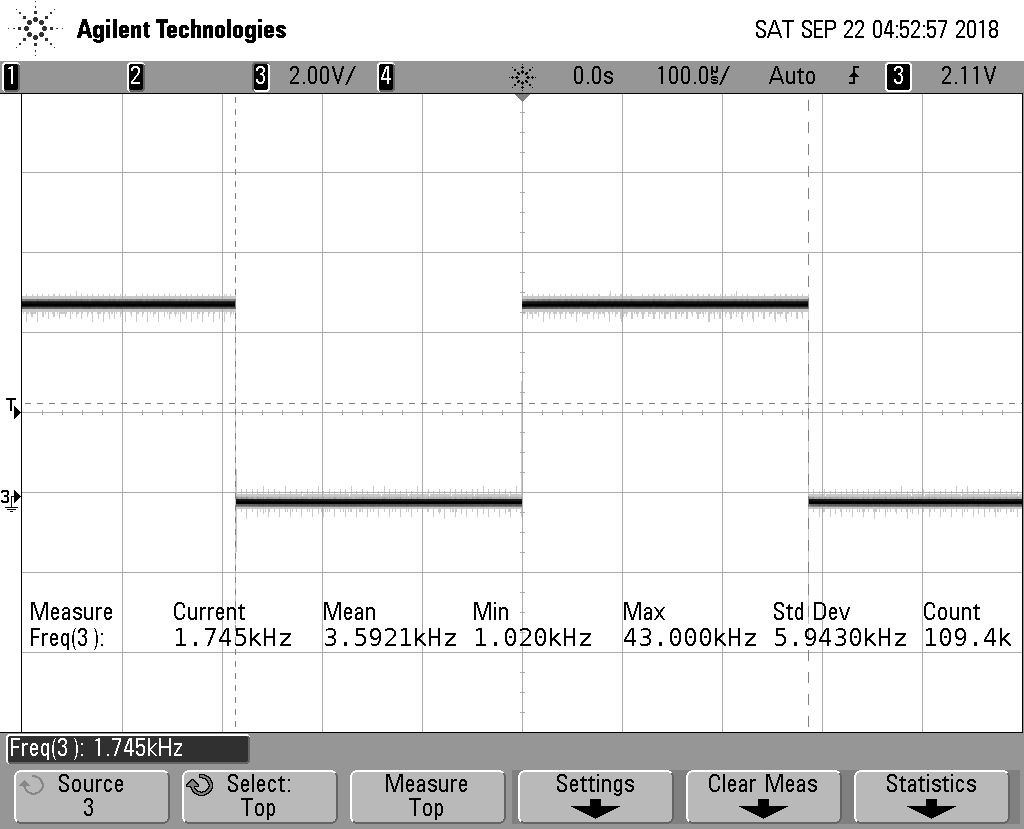
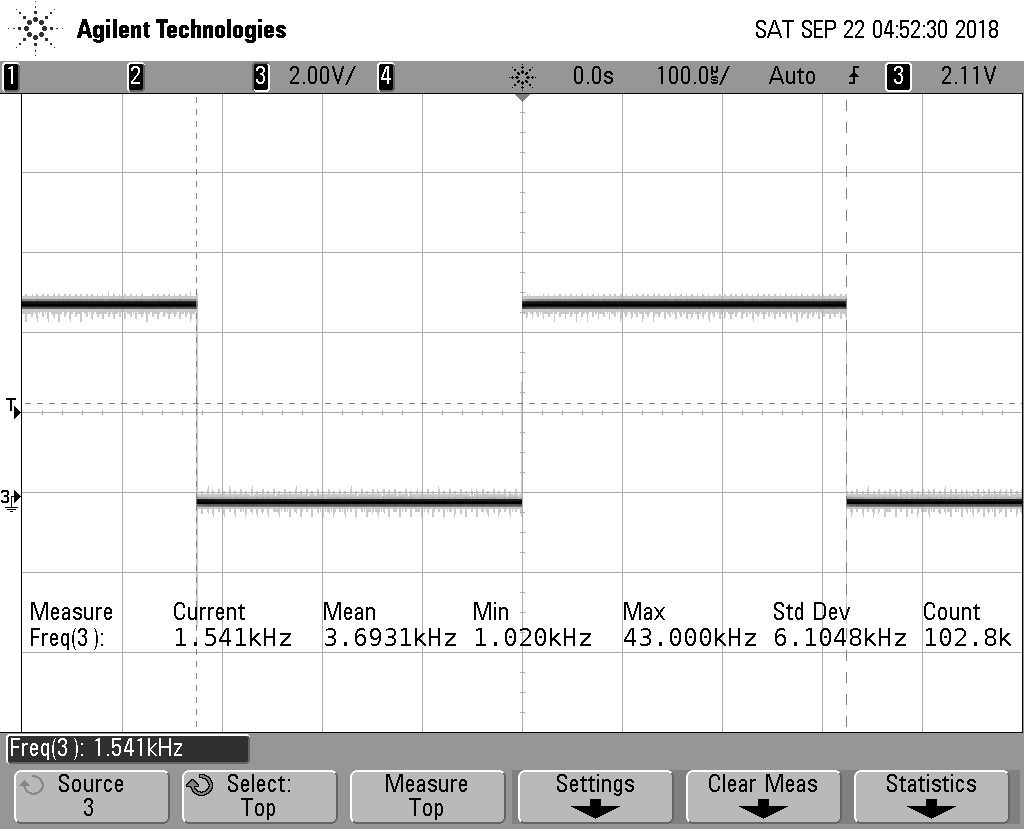
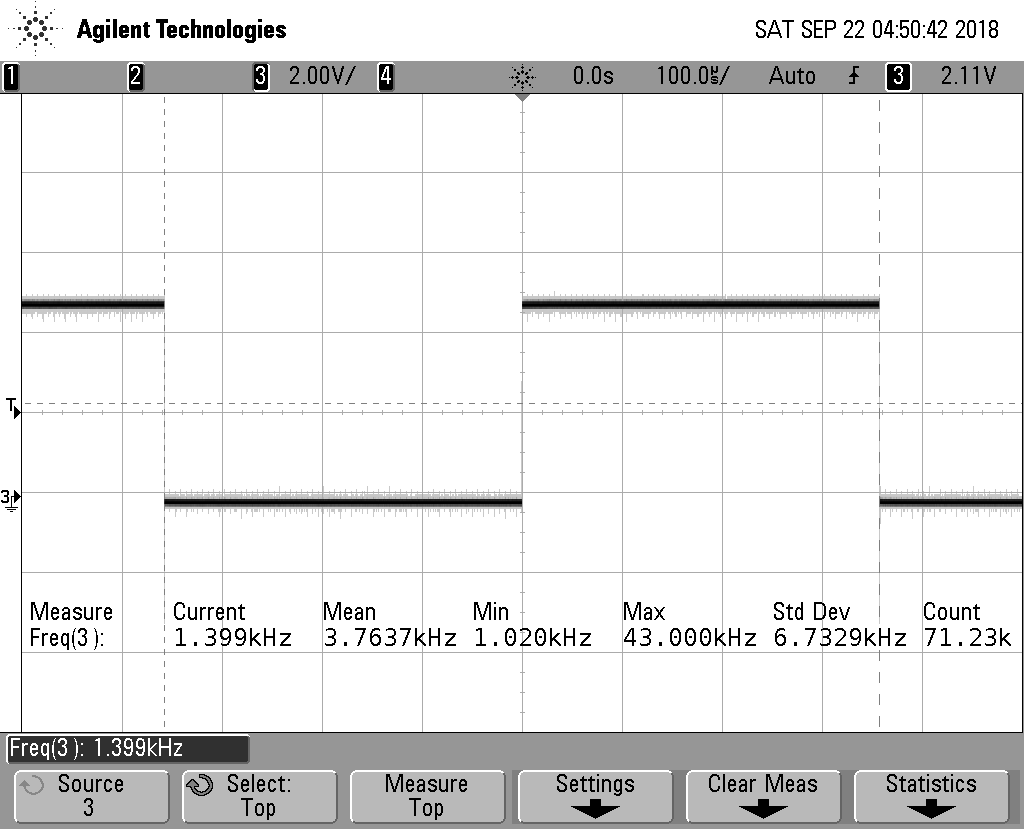
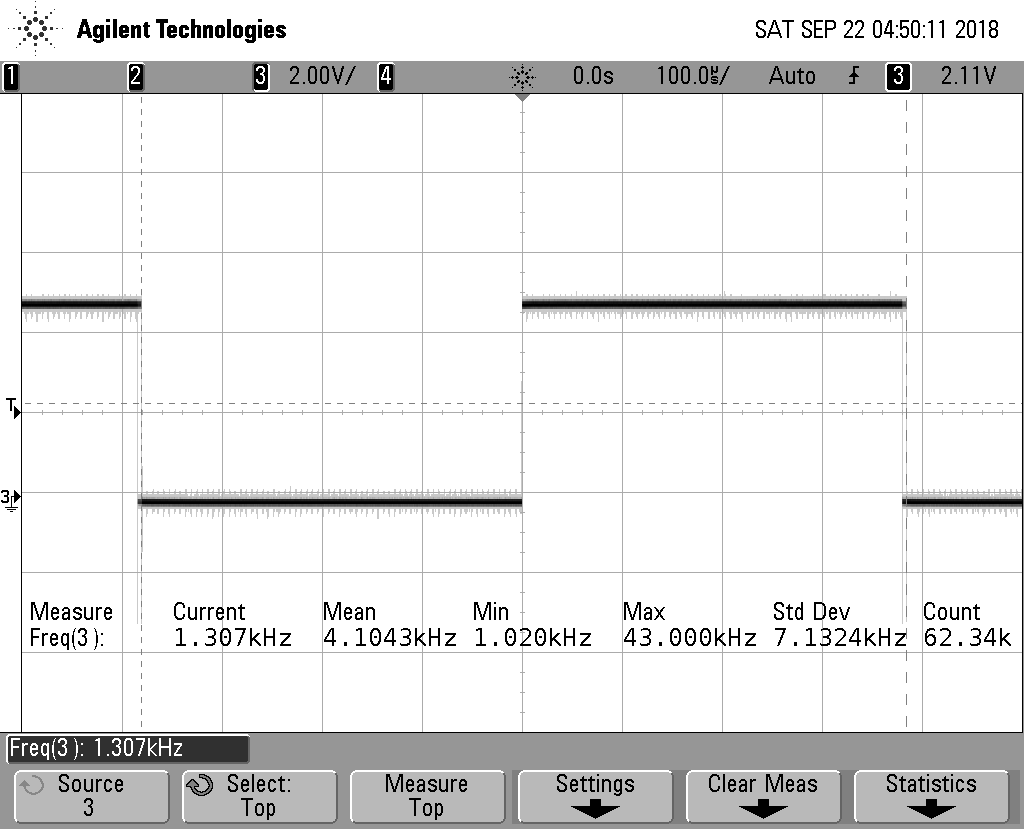
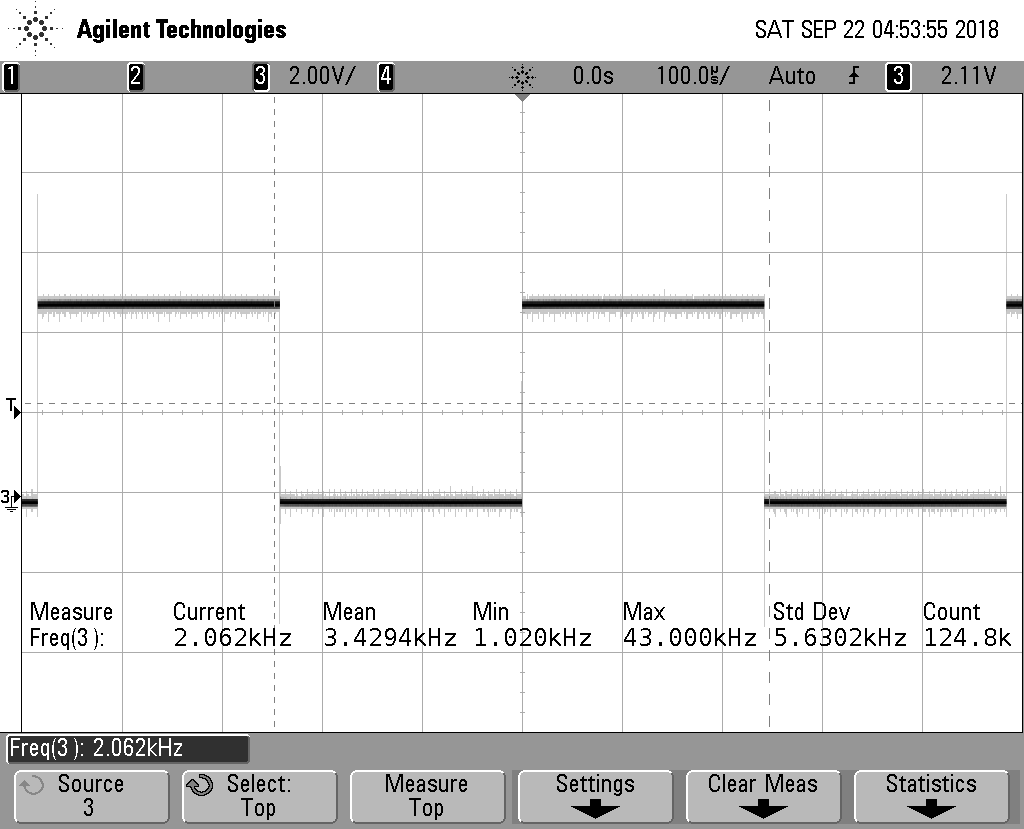
****

Figure 2: Waveform observed at (top left) C6 (top right) D6 (middle top left) E6 (middle top right) F6 (middle bottom left) G6 (middle bottom right) A6 (bottom left) B6 (bottom right) C7

## 3.1

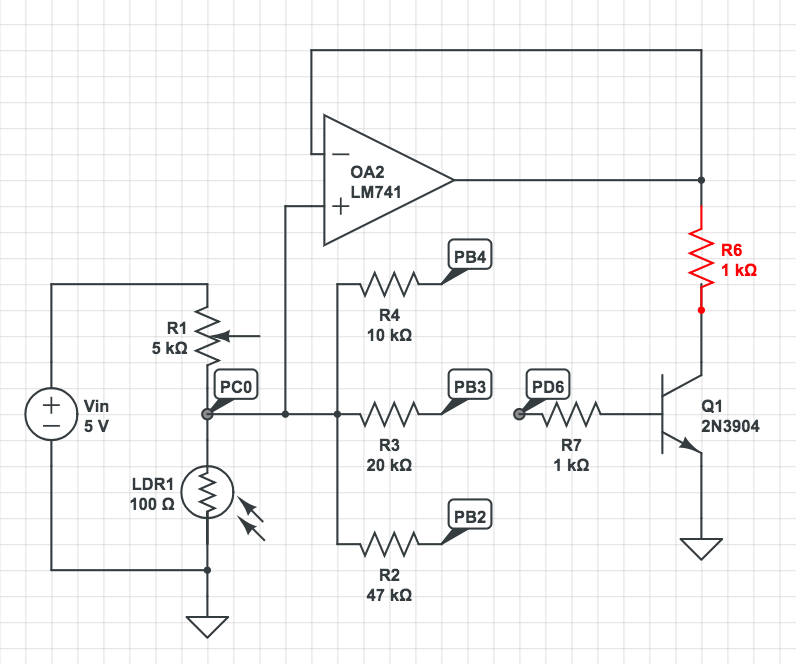
Record the minimum and maximum values you can measure from the light sensor connected to the ADC

Figure 3 Minimum recorded value =336; Maximum recorded value = 765

## 3.4 (EC)

Describe your solution (including circuit diagrams and any code changes as required)

Our solution to controlling volume was to substitute a potentiometer in the voltage divider as R1 of the light-sensitive resistor circuit (shown below). Because we used a 5Ohm potentiometer, we had a large range of change in the voltage output of the voltage divider, and as a result changed the sensitivity of the volume control (light-sensitive resistor). \*Note: R6 is the buzzer and some connections is not shown. All grounds including Arduino GND are common.



**GitHub Link:**