Lab Members: Christina Kim & Evan Oskierko-Jeznacki

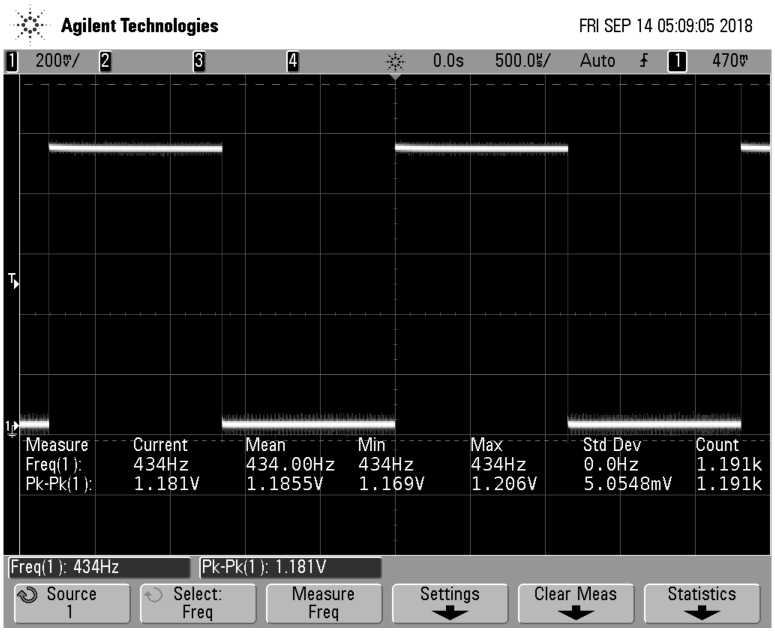
Group: 17

Due Date: 22 September 2018

Writeup

1.2

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



```
Measure    Current   Mean       Min      Max      Std Dev     Count
Freq(1 ):  434Hz     434.00Hz   434Hz    434Hz    0.0Hz       1.191k
Pk-Pk(1 ): 1.181V    1.1855V    1.169V   1.206V   5.0548mV    1.191k
```

```
Freq(1 ): 434Hz    Pk-Pk(1 ): 1.181V
```

Figure 1 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 a single 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. A 1024 prescaler does provide enough resolution between notes, given that C6 and C7 would correspond to 1 and 2 clock cycles, respectively. 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:

$$PWM_{interval} = \frac{max_{PWM} - min_{PWM}}{\#bins - 1} = \frac{827 - 95}{8 - 1} = 104.57$$

$$PWM_{new_{C6}} = PWM_{previous} + PWM_{interval} = 95 + 104.57 = 199.57$$

$$OCROA_{C6} = \left(\frac{1}{f_{OCROA}} \Big/ \frac{1}{f_{clock}} \right) - 1 = \left(\frac{1}{1046.5} \Big/ \frac{1}{8 * 10^6} \right) - 1 = 7643.5294$$

$$Prescaler\_value = \frac{f_{clock}}{prescaler} = \frac{8*10^6}{1024} = 7812.5 > 7643.5294 \text{ therefore 1024 would work for this part}$$

$$Prescaler\_value = \frac{f_{clock}}{prescaler} = \frac{8*10^6}{64} = 125{,}000 > 7643.5294 \text{ therefore 64}$$

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

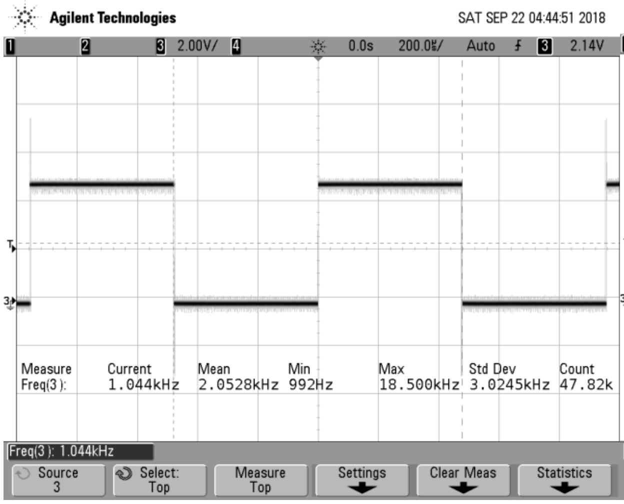
2.4

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

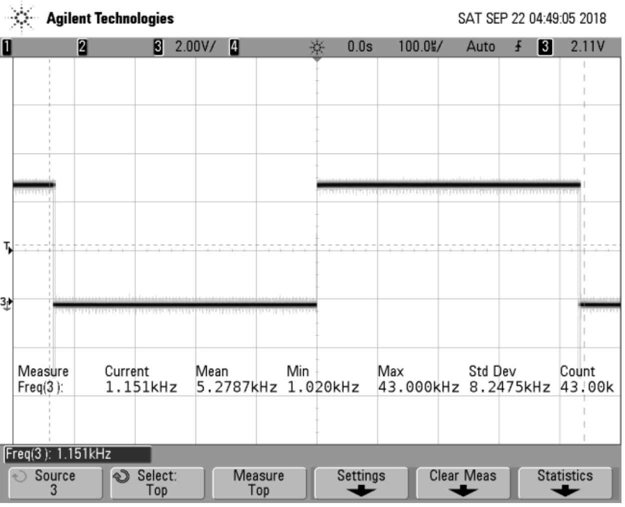
Measurements	C6	D6	E6	F6	G6	A6	B6	C7
OCROA	7643.5	6809.5	6066.5	5725.9	5105.1	4544.5	4048.5	3821.3
Clock Ticks/2	8	9	10	11	13	14	16	17

*Note: Clock Ticks were divided by 2 to lower the octave, per Chris Foley's suggestion (different buzzer was used, causing same notes to be an octave higher)

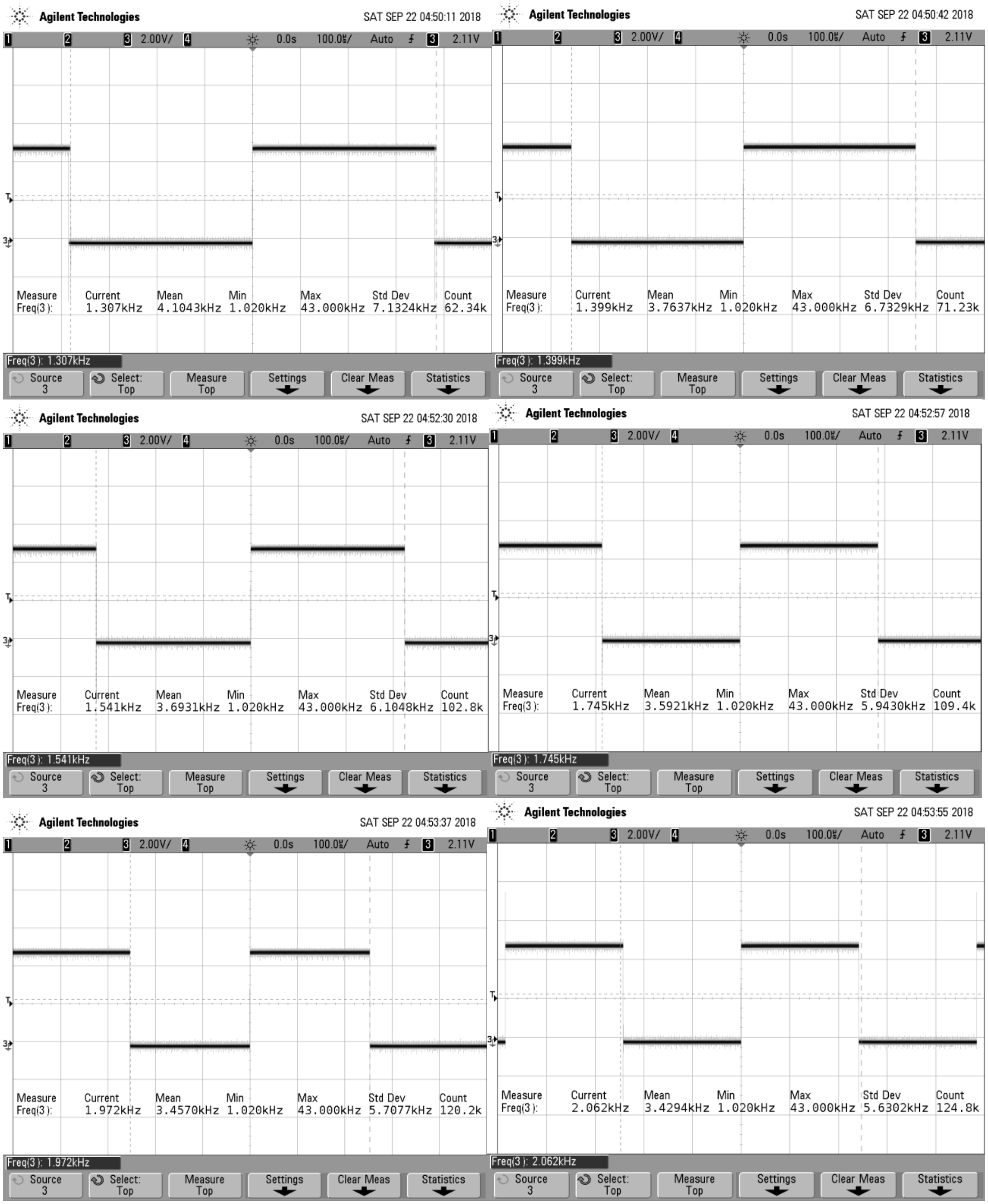
$$Clock\ Ticks = \frac{Prescaler\_value}{OCROA} = \frac{125 * 10^3}{7643.5294} = 16.3537 \simeq 16$$



```
Measure   Current   Mean       Min    Max        Std Dev    Count
Freq(3):  1.044kHz  2.0528kHz  992Hz  18.500kHz  3.0245kHz  47.82k
```



```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.151kHz  5.2787kHz  1.020kHz  43.000kHz  8.2475kHz  43.00k
```



Agilent Technologies SAT SEP 22 04:50:11 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.307kHz  4.1043kHz  1.020kHz  43.000kHz  7.1324kHz  62.34k
```

Freq(3): 1.307kHz

Agilent Technologies SAT SEP 22 04:50:42 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.399kHz  3.7637kHz  1.020kHz  43.000kHz  6.7329kHz  71.23k
```

Freq(3): 1.399kHz

Agilent Technologies SAT SEP 22 04:52:30 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.541kHz  3.6931kHz  1.020kHz  43.000kHz  6.1048kHz  102.8k
```

Freq(3): 1.541kHz

Agilent Technologies SAT SEP 22 04:52:57 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.745kHz  3.5921kHz  1.020kHz  43.000kHz  5.9430kHz  109.4k
```

Freq(3): 1.745kHz

Agilent Technologies SAT SEP 22 04:53:37 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  1.972kHz  3.4570kHz  1.020kHz  43.000kHz  5.7077kHz  120.2k
```

Freq(3): 1.972kHz

Agilent Technologies SAT SEP 22 04:53:55 2018

```
Measure   Current   Mean       Min       Max        Std Dev    Count
Freq(3):  2.062kHz  3.4294kHz  1.020kHz  43.000kHz  5.6302kHz  124.8k
```

Freq(3): 2.062kHz

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

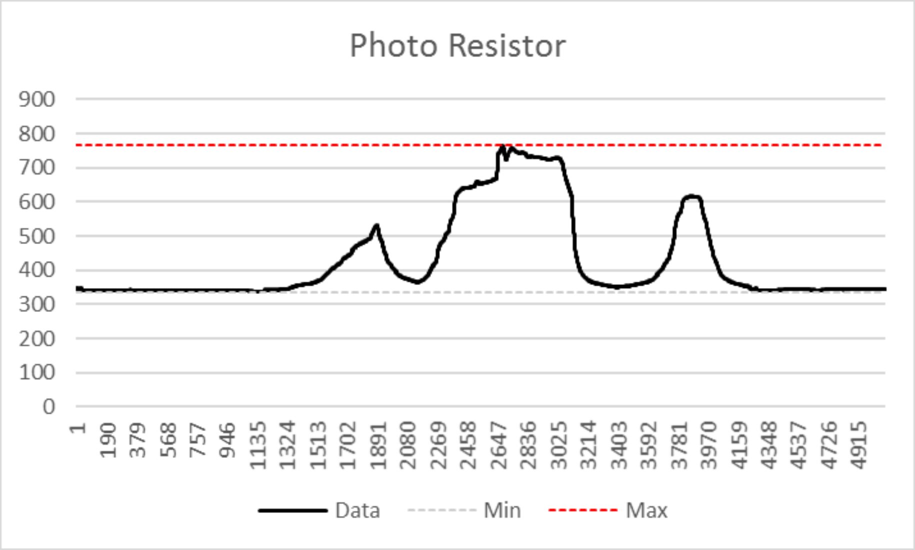


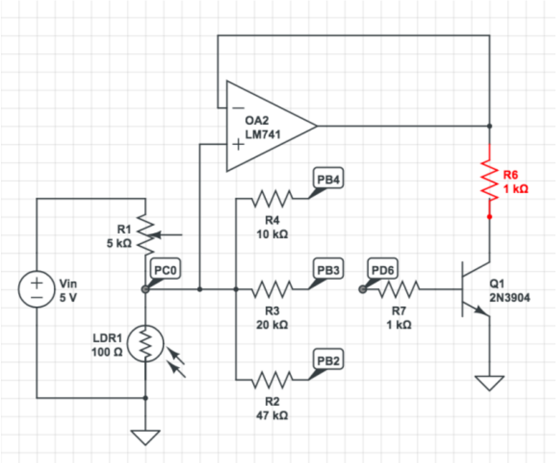
Figure 3 Minimum recorded value =336; Maximum recorded value = 765

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 larger range of change in the voltage output of the voltage divider, and as a result increased the sensitivity of the volume (amplitude) control (light-sensitive resistor).

**Note: R6 is the buzzer and some connections is not shown. All grounds including Arduino GND are common.*



GitHub Link: https://github.com/EvanOJ/ESE-519.git