# Lab 3: Theremin

ESE519/IPD519: Introduction to Embedded Systems
University of Pennsylvania

Please fill out your name and link your Github repository below to begin. Be sure that your code on the repo is up-to-date before submission!

**Student Name:Varun Mamtani**
**Pennkey:vmamtani**
**GitHub Repository: https://github.com/mowgleee/lab3-vmamtani**

1. For an 8 bit Timer, the maximum ticks count is 0x11111111 = 255 after which it will overflow. I have prescaled it by 256, which means the frequency of the timer is 16MHz/256=0.0625MHz or 62.5 KHz. If the transistor is being triggered at every overflow, the frequency will be 62.5KHz/256=0.245KHz=244Hz. So the pin D6 is toggling at 245 Hz.
2. Yes I prescaled the timer clock by 256.
3. Count= ((fclk/O)/(2*N*fdes))-1
   =((16MHz)/(2*256*440))-1=70.022 that is nearly 70.

4.

```
    //input output setup
    DDRD |=(1<<DDD6);//OUTPUT PD6(OC0A)
    //PORTD&=~(1<<PORTD6);//PD6 LOW

    //Set Timer 0 to normal
    TCCR0A&=~(1<<WGM00);
    TCCR0A&=~(1<<WGM01);
    TCCR0B&=~(1<<WGM02);

    //prescale by 256

    TCCR0B&=~(1<<CS00);
    TCCR0B&=~(1<<CS01);
    TCCR0B|=(1<<CS02);

    //clear COMPA interrupt flag
    TIFR0|=(1<<OCF0A);

    OCR0A=20;

    //enable COMPA interrupt
    TIMSK0 |= (1<<OCIE0A);

    sei(); //Enable global interrupts
}

ISR(TIMER0_COMPA_vect)
{
    PORTD^=(1<<PORTD6);
    TCNT0=0;
}
```

5. Count= ((fclk/O)/(2*N*fdes))-1
   =((16MHz)/(2*256*440))-1=70.022 that is nearly 70 for toggling.

6.
```
void initialize()
{
    //Disable global interrupts
    cli();

    UART_init(BAUD_PRESCALER);

    //input output setup
    DDRD |=(1<<DDD6);//OUTPUT PD6(OC0A)
    //PORTD&=~(1<<PORTD6);//PD6 LOW

    //Set Timer 0 to CTC
    TCCR0A&=~(1<<WGM00);
    TCCR0A|=(1<<WGM01);
    TCCR0B&=~(1<<WGM02);

    //prescale by 256

    TCCR0B&=~(1<<CS00);
    TCCR0B&=~(1<<CS01);
    TCCR0B|=(1<<CS02);

    TCCR0A&=~(1<<COM0A1);//Toggle OC0A on compare match
    TCCR0A|=(1<<COM0A0);

    //clear COMPA interrupt flag
    TIFR0|=(1<<OCF0A);

    OCR0A=high_time;

    //enable COMPA interrupt
    TIMSK0 |= (1<<OCIE0A);

    sei(); //Enable global interrupts
}
```

7. OCR0A should be 35. TOP value is 35 i.e. OCR0A value.

8.

```
void initialize()
{
    //Disable global interrupts
    cli();

    UART_init(BAUD_PRESCALER);

    //input output setup
    DDRD |=(1<<DDD6);//OUTPUT PD6(OC0A)
    //PORTD&=~(1<<PORTD6);//PD6 LOW

    //Set Timer 0 to Phase correct PWM
    TCCR0A|=(1<<WGM00);
    TCCR0A&=~(1<<WGM01);
    TCCR0B|=(1<<WGM02);

    //prescale by 256

    TCCR0B&=~(1<<CS00);
    TCCR0B&=~(1<<CS01);
    TCCR0B|=(1<<CS02);

    TCCR0A&=~(1<<COM0A1);//Toggle OC0A on compare match
    TCCR0A|=(1<<COM0A0);

    //clear COMPA interrupt flag
    TIFR0|=(1<<OCF0A);

    OCR0A=high_time;//70

    //enable COMPA interrupt
    TIMSK0 |= (1<<OCIE0A);

    sei(); //Enable global interrupts
}
```

9. We need to supply a short 10uS pulse to the trigger input to start the ranging.
10. Trig (Trigger) pin is used to trigger the ultrasonic sound pulses.

The sensor sends 8 cyclic sonic bursts after this after which the Echo pin goes high. The pin goes low again when the reflected signal is received. The width of the pulse is proportional to the time it took for the transmitted signal to be detected.

11. 375 cm(maximum obstacle free distance in room)
12. 2 cm
13. For pre-scalar 64 in phase correct PWM and compare with OCR0B,

| Note | **C6** | **D6** | **E6** | **F6** | **G6** | **A6** | **B6** | **C7** |
|---|---|---|---|---|---|---|---|---|
| Freq (Hz) | 1046 | 1174 | 1318 | 1397 | 1568 | 1760 | 1975 | 2093 |
| OCR0A | 118 | 105 | 94 | 88 | 79 | 70 | 62 | 59 |

14.

Taking minimum distance as 5cm and max distance as 100 cm for theremin to keep it practical:

OCR0A = SENSOR_VALUE(in cm) * 0.62 + 56

As slope=(y2-y1/x2-x1)=(118-59/100-5)=0.62; intercept can be calculated by taking any single point and substituting it.

If we take max distance as 375 cm and minimum distance at 2cm,

OCR0A = SENSOR_VALUE(in cm) * 0.158177 + 59

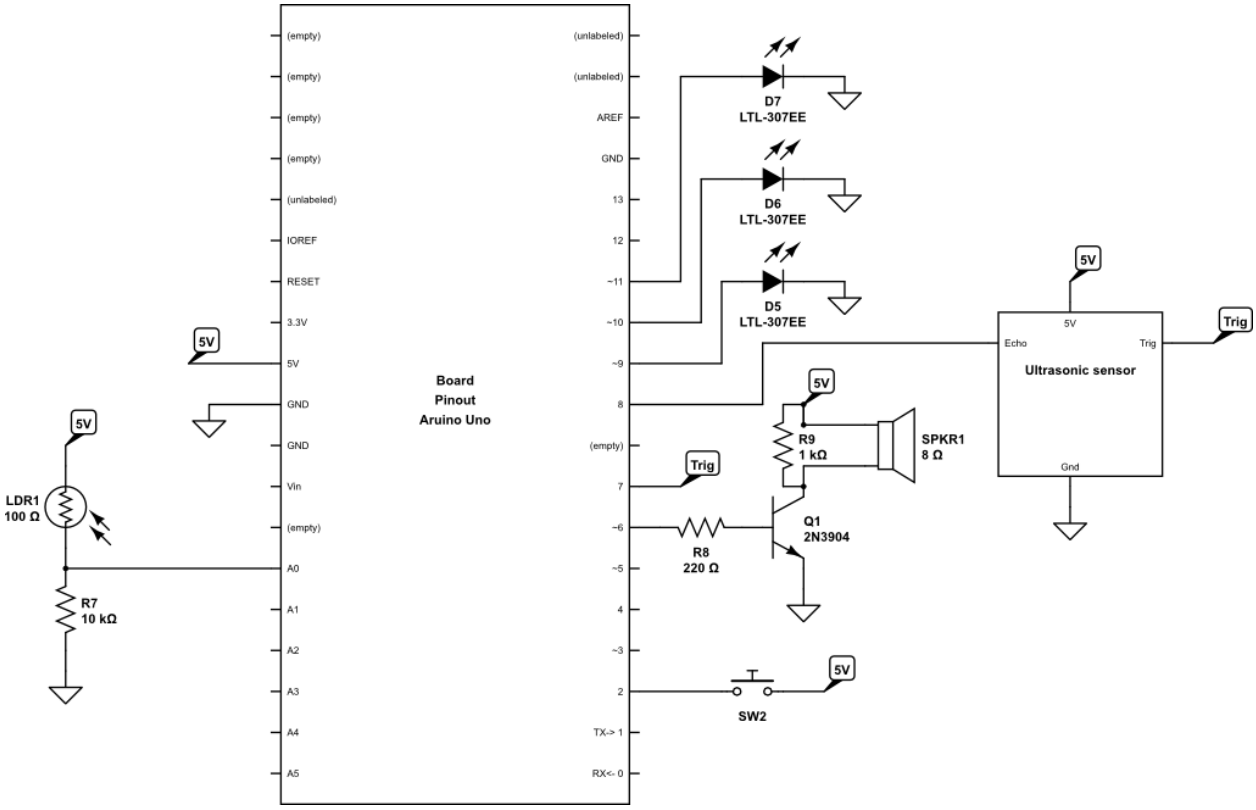
As slope=(y2-y1/x2-x1)=(118-59/375-2)=0.158177; intercept can be calculated by taking any single point and substituting it.

15. Max value: 1015; Min value: 4

16.

| ADC Ranges | Duty Cycle |
|---|---|
| 4-100 | 5% |
| 101-200 | 10% |
| 201-300 | 15% |
| 301-400 | 20% |
| 401-500 | 25% |
| 501-600 | 30% |
| 601-700 | 35% |

| | |
|---|---|
| 701-800 | 40% |
| 801-900 | 45% |
| 901-1024 | 50% |

17. 

18. https://drive.google.com/file/d/1EEAg-toz52m5VXB0F4SylwrM48wd4Crp/view?usp=sharing
19. A base resistor limits the current flowing into the base of the transistor to prevent it from being damaged.
20. BJT can switch faster than MOSFET due to the less capacitance at the control pin which is why we use it here for this relatively high frequency switching.
21. https://drive.google.com/file/d/1l1FBLpleiFtxXAl5P0Dz3SWQVsk-AwZj/view?usp=sharing

    I have used 3 LEDs to denote the 8 modes in the discrete mode of the circuit/program.

22.