Lab 04: Sound

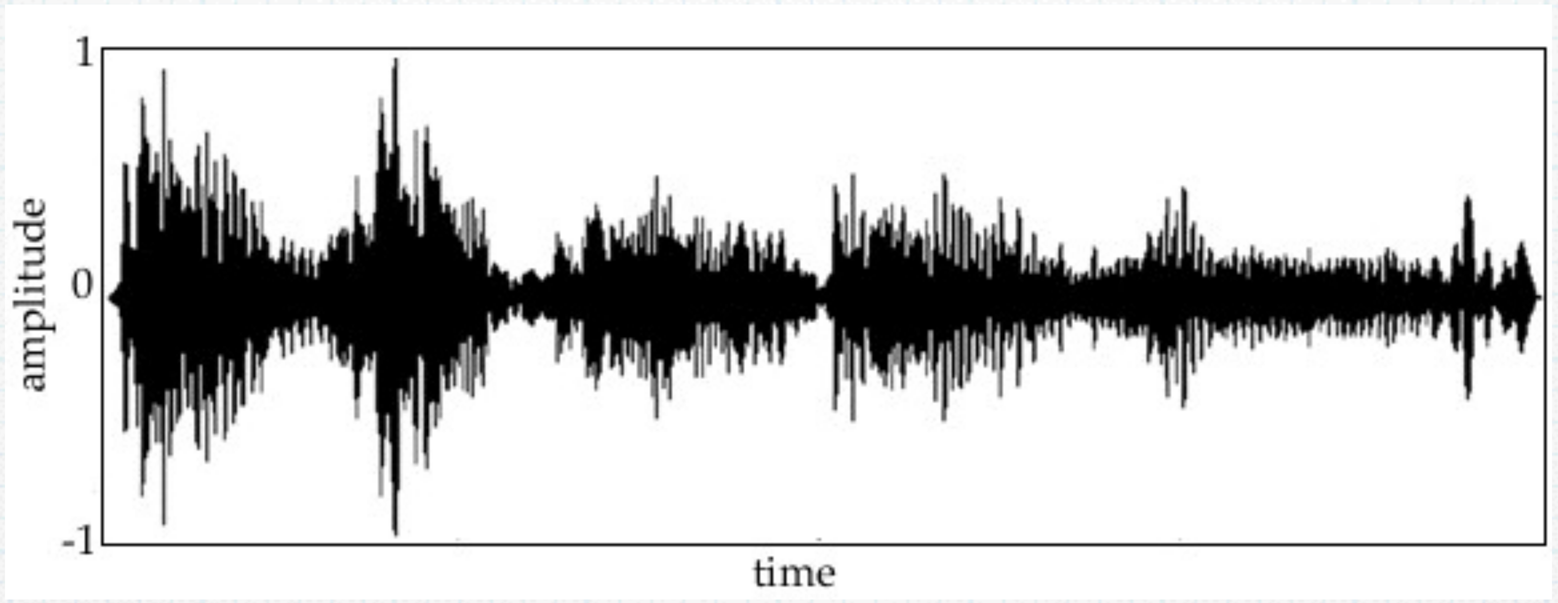
by Dr. Sethavidh Gertphol

Outline

* Sound
* Sound sensor
* Ultrasonic Ranger
* Buzzer
* tone() ใช้สร้างเสียง

การบีบ-อัดตัวของอากาศ

Sound characteristics



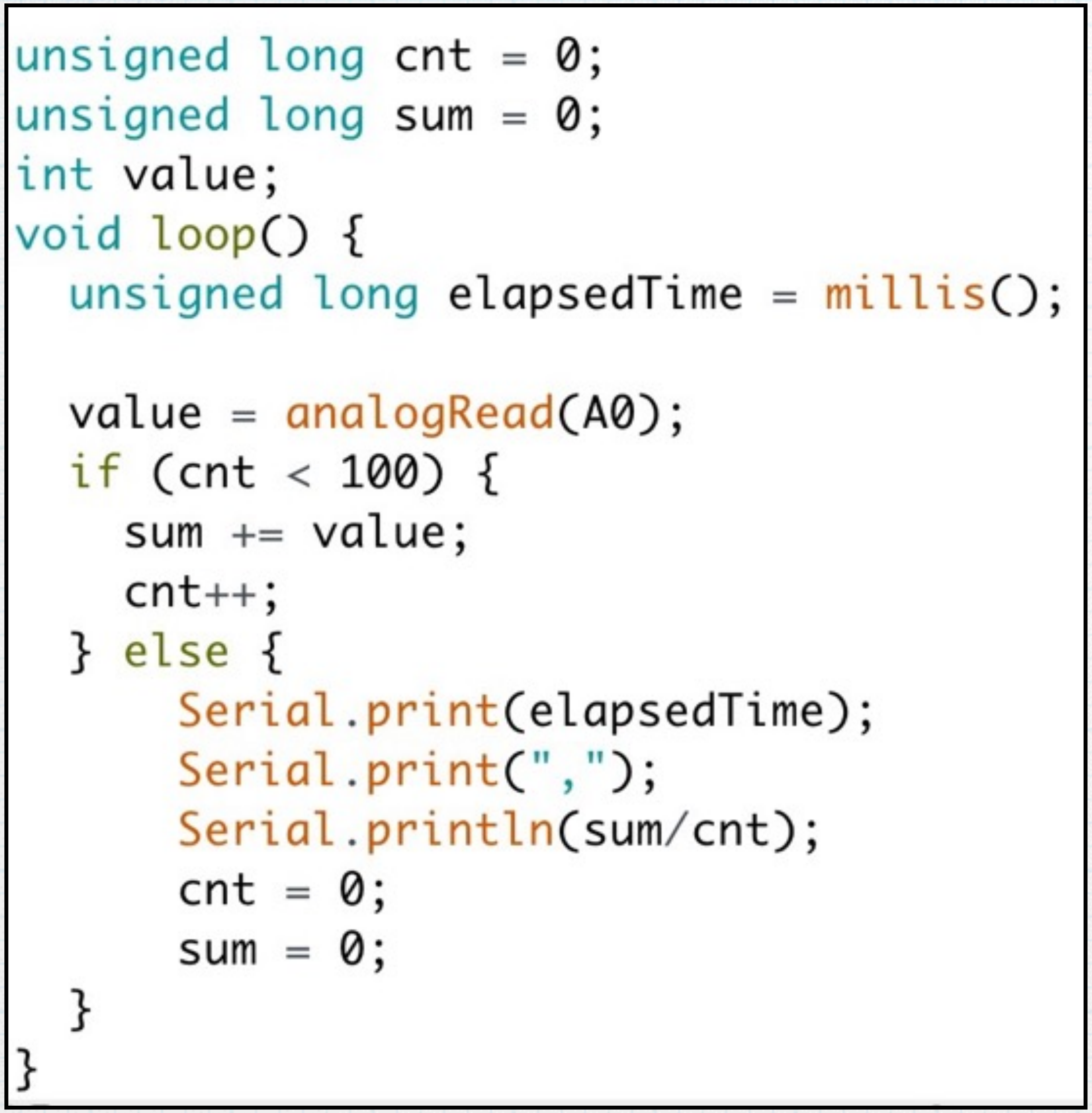
http://music.columbia.edu/cmc/MusicAndComputers/chapter1/01_01.php

* amplitude
* frequency ทำให้เกิดเสียงสูง-ต่ำ

Sound sensor

* simple analog input using `analogRead()`
* measure amplitude of sound wave
* sample sound very frequently
  * higher fidelity
* should process raw readings
  * e.g. average over time

Example



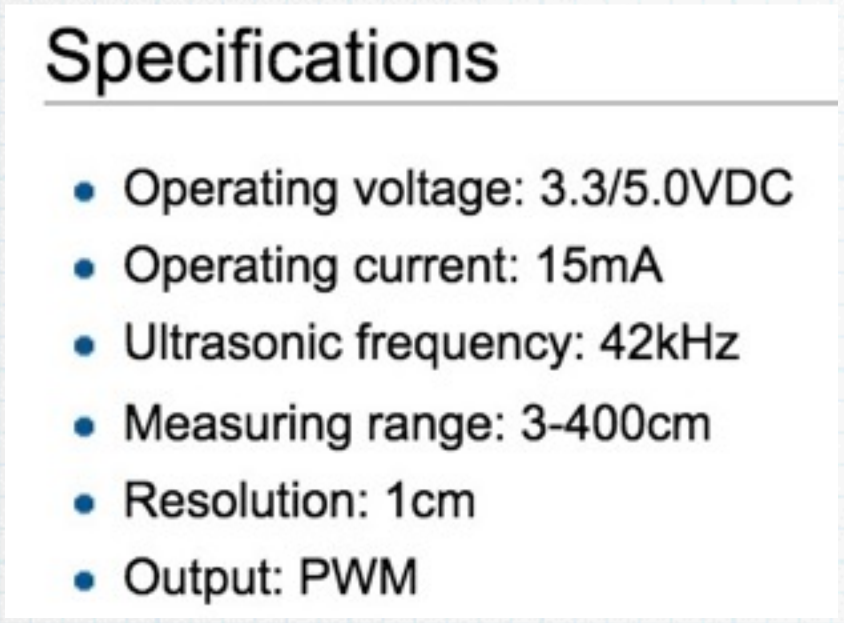
```
unsigned long cnt = 0;
unsigned long sum = 0;
int value;
void loop() {
  unsigned long elapsedTime = millis();

  value = analogRead(A0);
  if (cnt < 100) {
    sum += value;
    cnt++;
  } else {
      Serial.print(elapsedTime);
      Serial.print(",");
      Serial.println(sum/cnt);
      cnt = 0;
      sum = 0;
  }
}
```

* average sound volume over 100 samples
* send time and averaged value by Serial

Ultrasonic Rangefinder

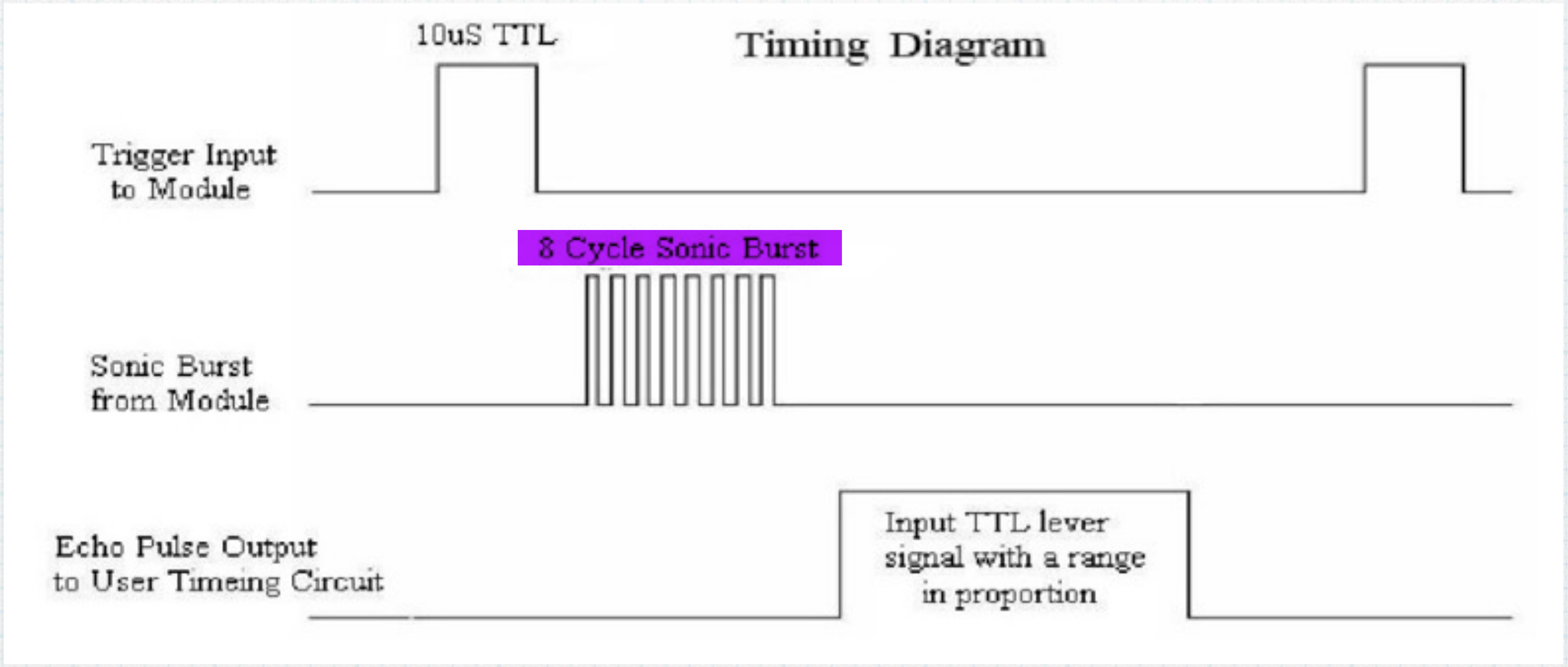
* use ultrasonic sound to measure distance
    * high frequency sound human can't hear
* send sound out and measure how long to hear back echo
    * calculate distance from speed of sound
    * similar to sonar or bat

Datasheet

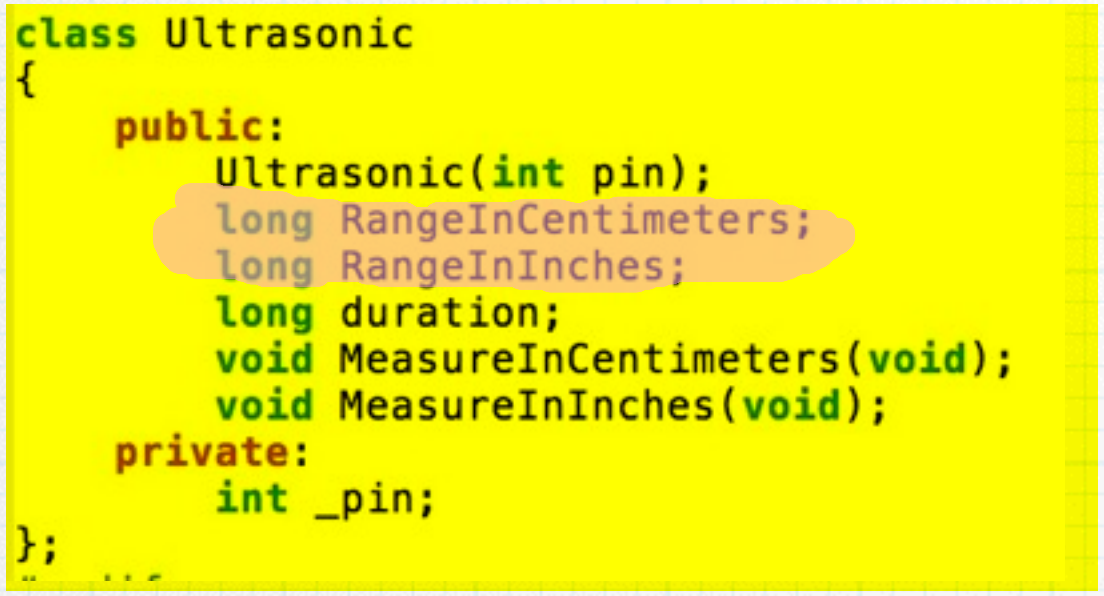


Specifications

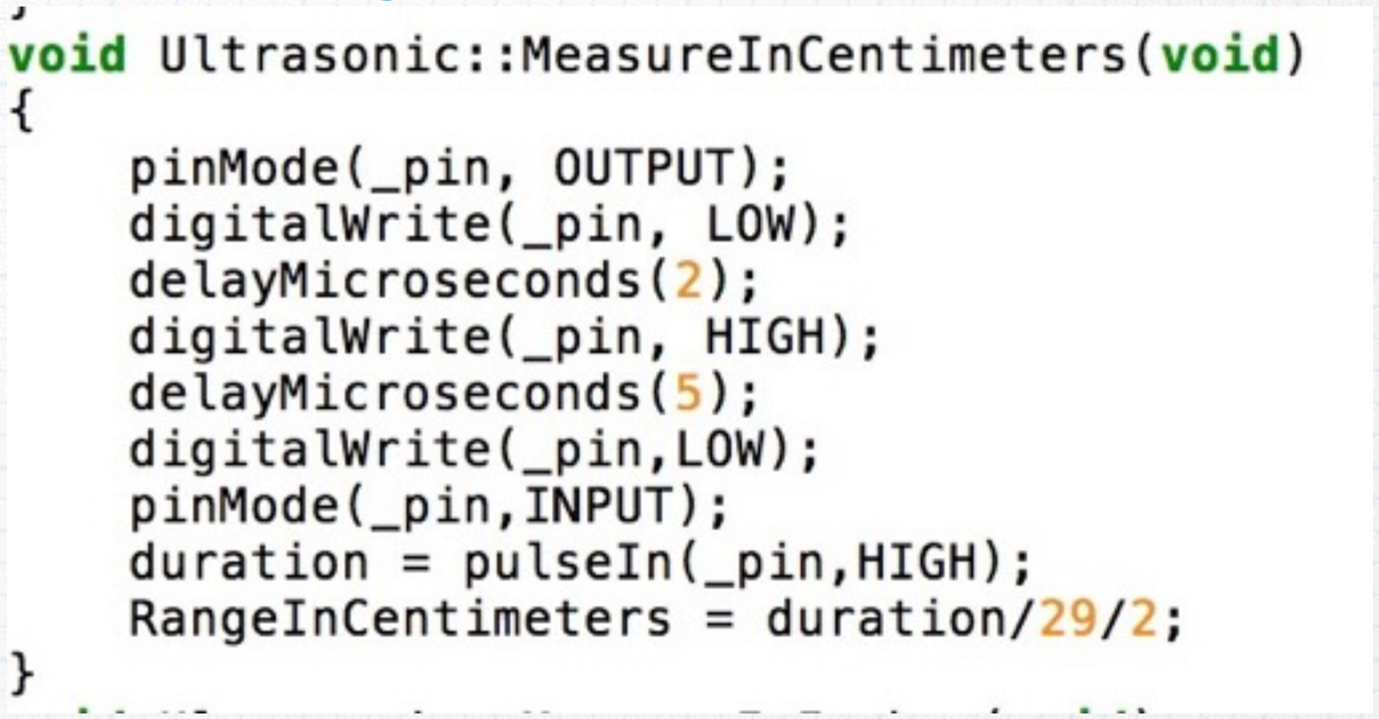
- Operating voltage: 3.3/5.0VDC
- Operating current: 15mA
- Ultrasonic frequency: 42kHz
- Measuring range: 3-400cm
- Resolution: 1cm
- Output: PWM


10uS TTL
Timing Diagram
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timeing Circuit
Input TTL lever signal with a range in proportion

Library



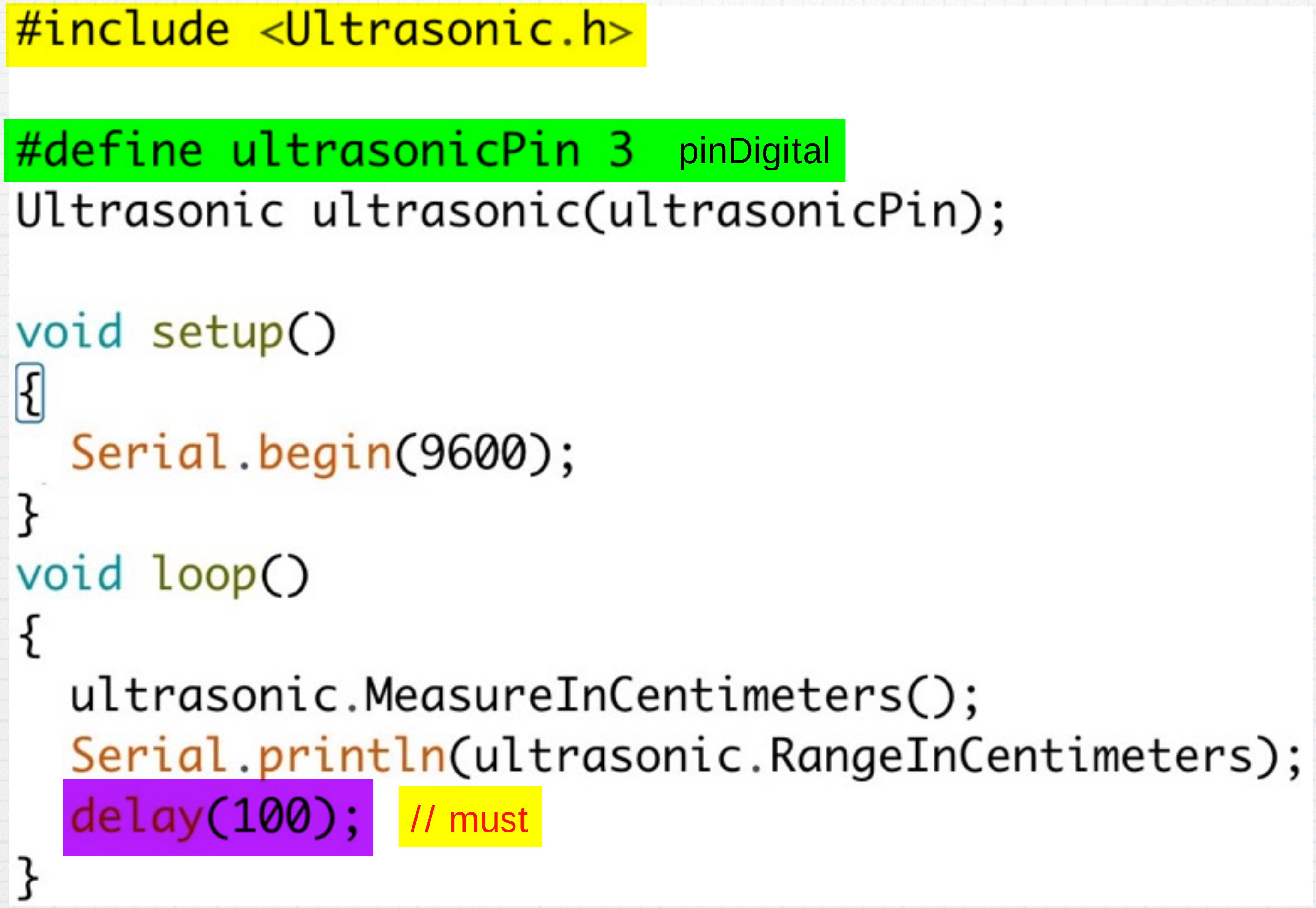
```
class Ultrasonic
{
    public:
        Ultrasonic(int pin);
        long RangeInCentimeters;
        long RangeInInches;
        long duration;
        void MeasureInCentimeters(void);
        void MeasureInInches(void);
    private:
        int _pin;
};
```



```
void Ultrasonic::MeasureInCentimeters(void)
{
        pinMode(_pin, OUTPUT);
        digitalWrite(_pin, LOW);
        delayMicroseconds(2);
        digitalWrite(_pin, HIGH);
        delayMicroseconds(5);
        digitalWrite(_pin,LOW);
        pinMode(_pin,INPUT);
        duration = pulseIn(_pin,HIGH);
        RangeInCentimeters = duration/29/2;
}
```

* use MeasureInCentimeters() or MeasureInInches() methods to find range
* result in RangeInCentimeters or RangeInInches variable
* duration variable contain how long until the pulse echo back
    * used in case you want to calibrate sensor

Example

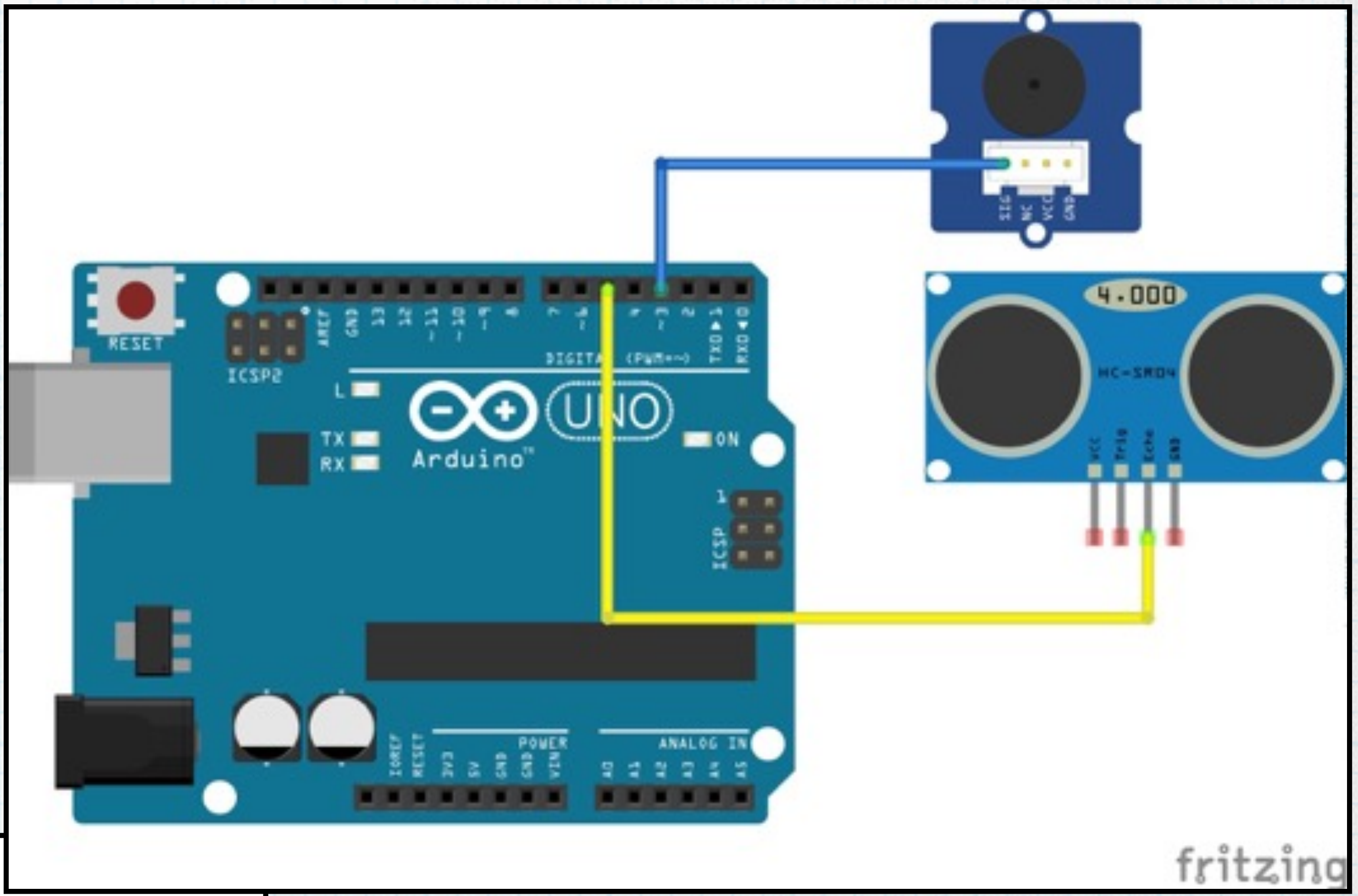


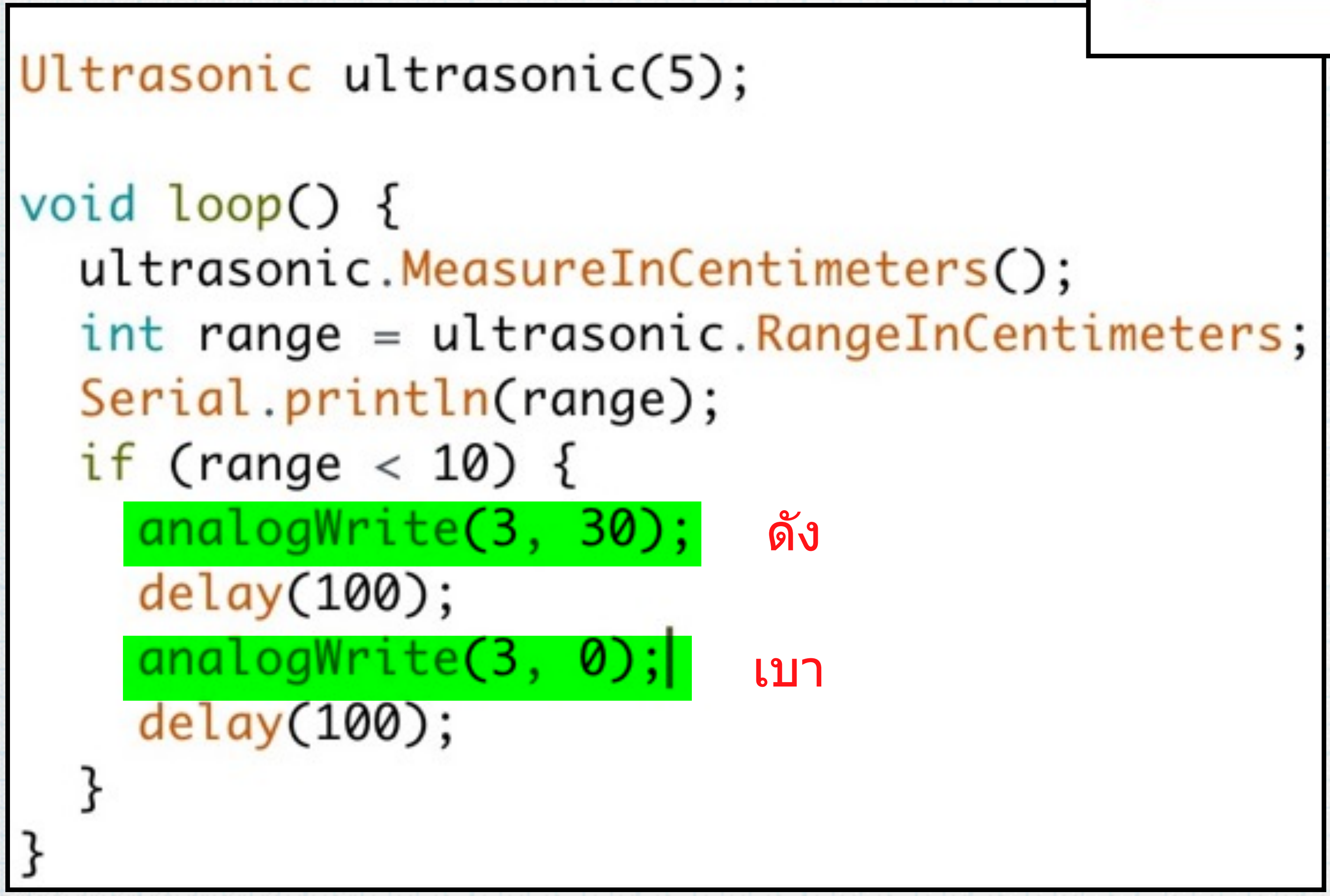
```
#include <Ultrasonic.h>

#define ultrasonicPin 3  pinDigital
Ultrasonic ultrasonic(ultrasonicPin);

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  ultrasonic.MeasureInCentimeters();
  Serial.println(ultrasonic.RangeInCentimeters);
  delay(100);  // must
}
```

Buzzer

- simple device to generate sound
- piezoelectric material
  - vibrate when subjected to electricity
- can be used with PWM
  - duty cycle affects loudness (not obvious at high duty cycles)
  - only one tone (beep)

Example





```
Ultrasonic ultrasonic(5);

void loop() {
  ultrasonic.MeasureInCentimeters();
  int range = ultrasonic.RangeInCentimeters;
  Serial.println(range);
  if (range < 10) {
    analogWrite(3, 30);        ดัง
    delay(100);
    analogWrite(3, 0);         เบา
    delay(100);
  }
}
```

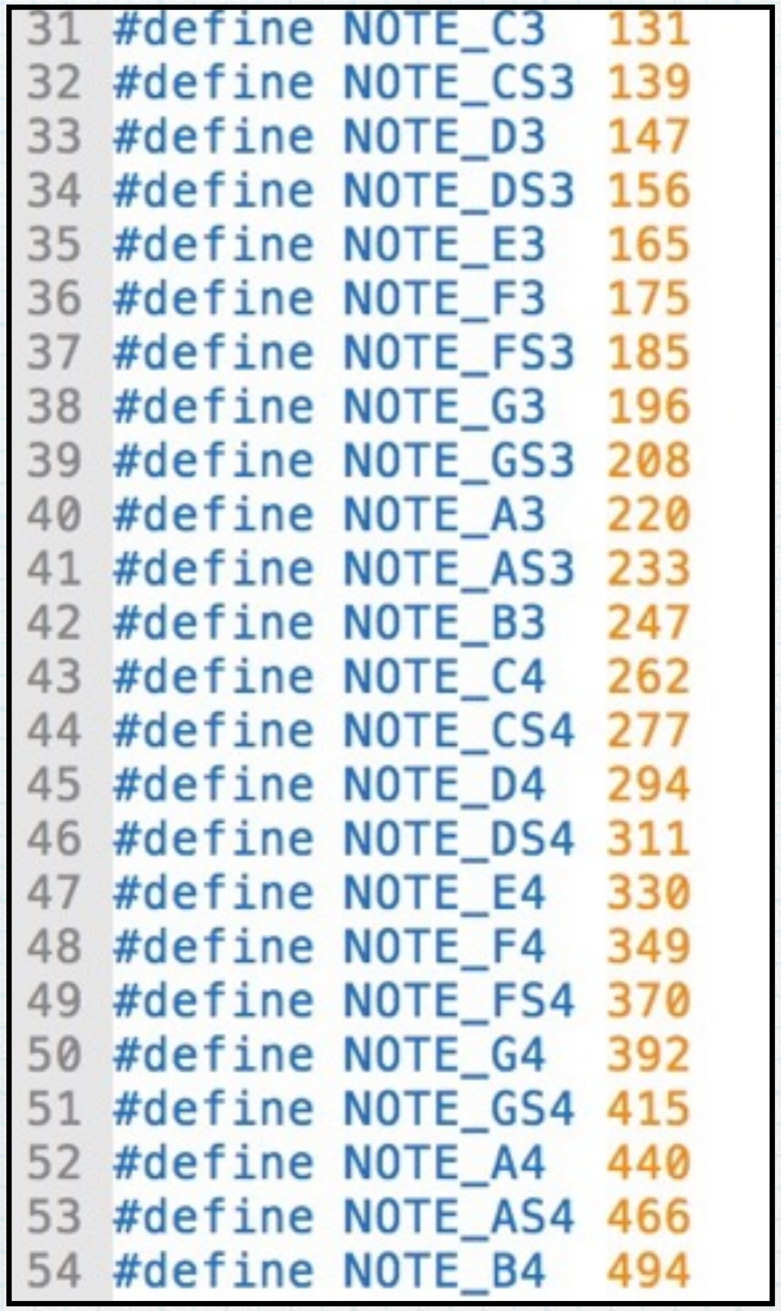
* beep when range is under 10 cm

tone()

* Arduino function for easy sound generation
* no need to include any library or use PWM
* `tone(pin, frequency [, duration])`
  * pin: any digital pin
  * frequency: of sound to play
  * duration: millisec of sound duration
* noTone(): turn off sound

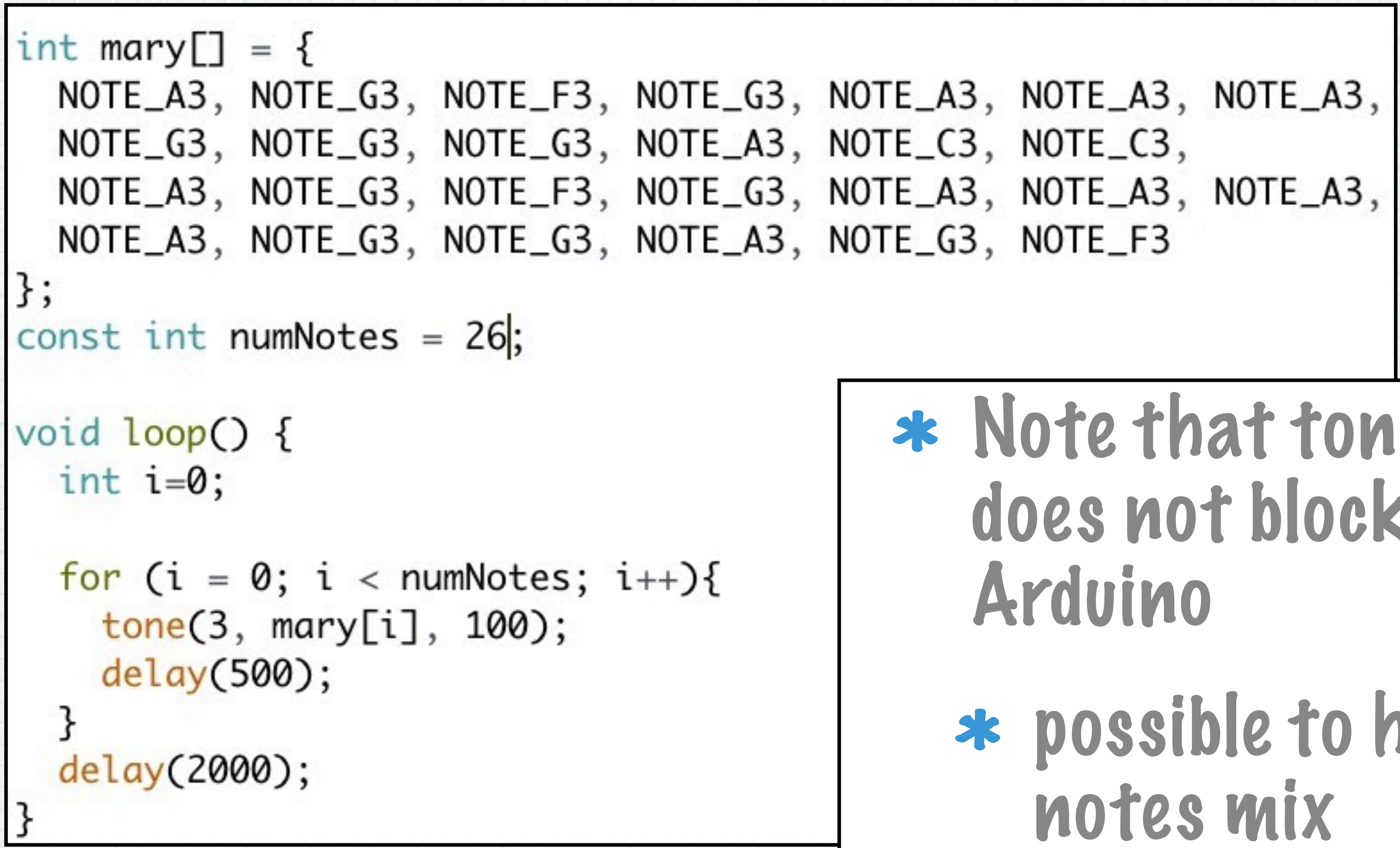
Musical Note frequencies

* musical notes have associated frequency
    * e.g. middle C is ≈ 262 Hertz
* notes are defined in external header file "pitches.h"



```
#define NOTE_C3   131
#define NOTE_CS3  139
#define NOTE_D3   147
#define NOTE_DS3  156
#define NOTE_E3   165
#define NOTE_F3   175
#define NOTE_FS3  185
#define NOTE_G3   196
#define NOTE_GS3  208
#define NOTE_A3   220
#define NOTE_AS3  233
#define NOTE_B3   247
#define NOTE_C4   262
#define NOTE_CS4  277
#define NOTE_D4   294
#define NOTE_DS4  311
#define NOTE_E4   330
#define NOTE_F4   349
#define NOTE_FS4  370
#define NOTE_G4   392
#define NOTE_GS4  415
#define NOTE_A4   440
#define NOTE_AS4  466
#define NOTE_B4   494
```

Example



```
int mary[] = {
  NOTE_A3, NOTE_G3, NOTE_F3, NOTE_G3, NOTE_A3, NOTE_A3, NOTE_A3,
  NOTE_G3, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_C3, NOTE_C3,
  NOTE_A3, NOTE_G3, NOTE_F3, NOTE_G3, NOTE_A3, NOTE_A3, NOTE_A3,
  NOTE_A3, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, NOTE_F3
};
const int numNotes = 26;

void loop() {
  int i=0;

  for (i = 0; i < numNotes; i++){
    tone(3, mary[i], 100);
    delay(500);
  }
  delay(2000);
}
```

* Note that tone() does not block Arduino
  * possible to have notes mix together

Reference

1. Grove - Sound Sensor, Seeedstudio, retrieved from http://www.seeedstudio.com/wiki/Grove_-_Sound_Sensor

2. Grove - Ultrasonic Ranger, Seeedstudio, retrived from http://www.seeedstudio.com/wiki/Grove_-_Ultrasonic_Ranger

3. tone, Arduino Reference, retrieved from https://www.arduino.cc/en/Reference/Tone

4. Grove - Buzzer, Seeedstudio, retrieved from http://www.seeedstudio.com/wiki/Grove_-_Buzzer