# Robotics

## Buzzer / Ultrasonic sensor

School of Computing, Gachon University

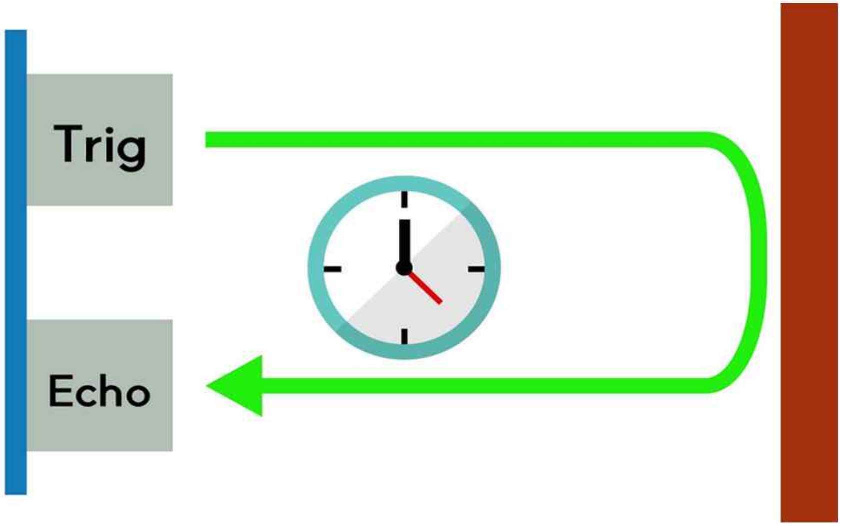
Kang, Sangwoo

# Basics

- Buzzer - Piezo Speaker
  - **Piezo buzzer** is basically a tiny speaker and is an electronic device that can be used to play tones.

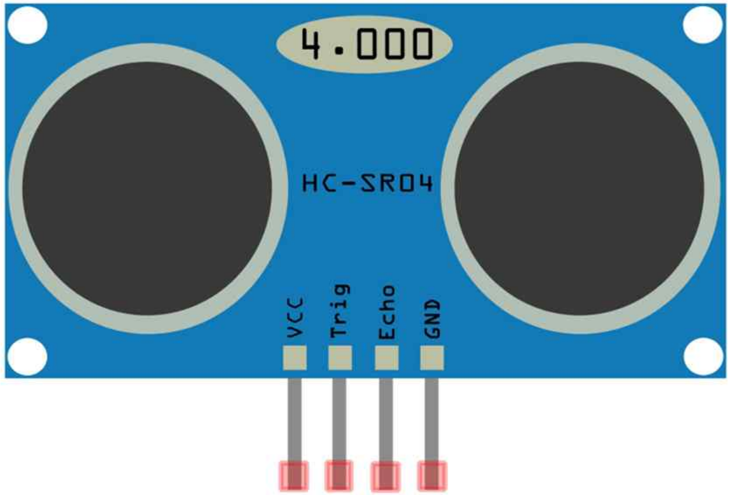
# Basics [cont'd]

- Ultrasonic Sensor
  - It emits an ultrasound at 40Hz which travels through the air and if there is an object or obstacle on its path, It will bounce back to the module
  - Considering the travel time and the speed of the sound you can calculate the distance



# Basics [cont'd]

- The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo
  - The **Ground** and the **VCC** pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the **trig** and **echo** pins to any Digital I/O pin on the Arduino Board
    - Trig (signal output pin → High state)
    - Echo (signal input pin → High state)



# Functions

- tone()
  - It can play tones from a piezo buzzer or other speaker. A duration of a buzzer can be specified, otherwise the wave continues until a call to noTone(). Only one tone can be generated at a time. If a tone is already playing on a different pin, the call to tone() will have no effect. If the tone is playing on the same pin, the call will set its frequency.
  - Syntax
    - tone(*pin, frequency*)
    - tone(*pin, frequency, duration*)
  - Parameters
    - *pin*: the pin on which to generate the tone
    - *frequency*: the frequency of the tone in hertz
    - *duration*: the duration of the tone in milliseconds (optional)

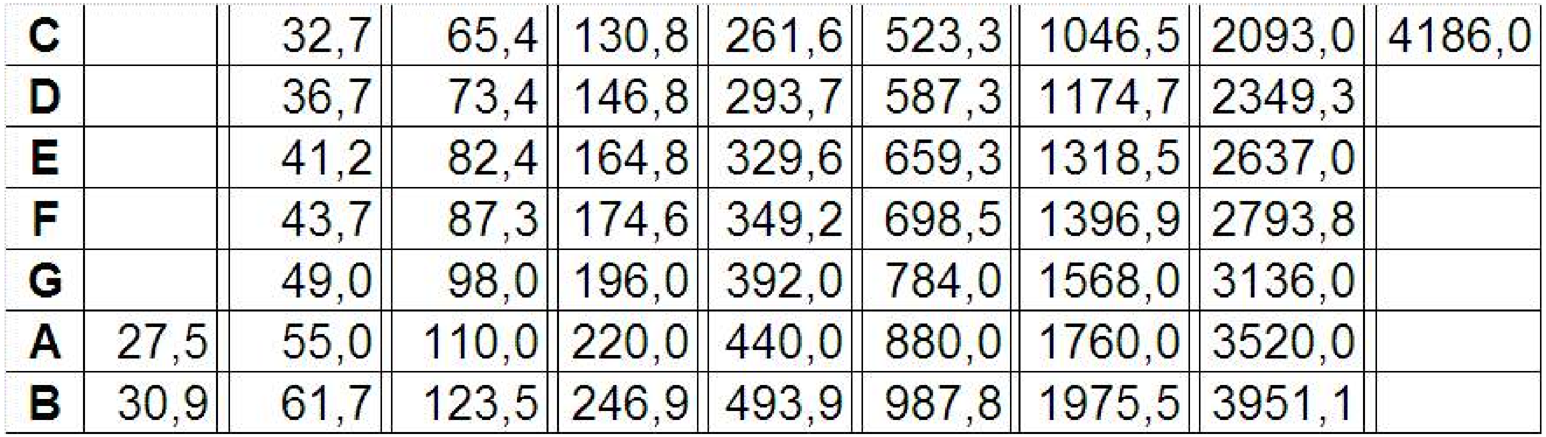
# Functions [cont'd]

- noTone()
  - Stops the generation of a square wave triggered by tone()
  - Has no effect if no tone is being generated.
- Syntax
  - noTone(*pin*)
- Parameters
  - *pin*: the pin on which to stop generating the tone

# Functions [cont'd]

○ example

```
void setup() {
pinMode(9, OUTPUT);
tone (9, 262);
delay(500);
noTone(9);
}
```



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **C** | | 32,7 | 65,4 | 130,8 | 261,6 | 523,3 | 1046,5 | 2093,0 | 4186,0 |
| **D** | | 36,7 | 73,4 | 146,8 | 293,7 | 587,3 | 1174,7 | 2349,3 | |
| **E** | | 41,2 | 82,4 | 164,8 | 329,6 | 659,3 | 1318,5 | 2637,0 | |
| **F** | | 43,7 | 87,3 | 174,6 | 349,2 | 698,5 | 1396,9 | 2793,8 | |
| **G** | | 49,0 | 98,0 | 196,0 | 392,0 | 784,0 | 1568,0 | 3136,0 | |
| **A** | 27,5 | 55,0 | 110,0 | 220,0 | 440,0 | 880,0 | 1760,0 | 3520,0 | |
| **B** | 30,9 | 61,7 | 123,5 | 246,9 | 493,9 | 987,8 | 1975,5 | 3951,1 | |

# Functions [cont'd]

- pulseIn()
  - ○ Reading a pulse (either HIGH or LOW) on a pin.
    - It returns the length of the pulse in microseconds or 0 if no complete pulse was received within the timeout.
  - ○ Syntax
    - pulseIn(pin, value)

      pulseIn(pin, value, timeout)
  - ○ Parameters
    - pin: the number of the pin on which you want to read the pulse (int)
    - value: type of pulse to read: either HIGH or LOW (int)
    - timeout (optional): the number of microseconds to wait for the pulse to be completed: Default is one second.

# Functions [cont'd]

- delayMicroseconds()
  - Pausing the program for the amount of time (in microseconds) specified as parameter
    - There are a thousand microseconds in a millisecond, and a million microseconds in a second
  - Syntax
    - delayMicroseconds(us)
  - Parameters
    - us: the number of microseconds to pause (unsigned int)

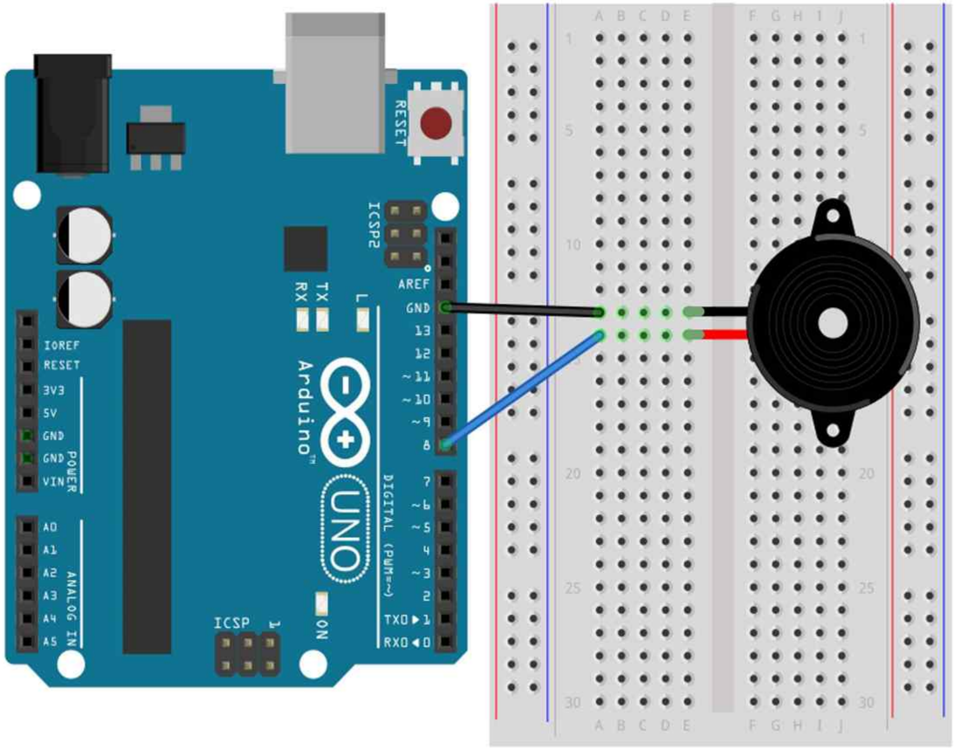
# Functions [cont'd]

○ Example

```
int outPin = 2;
int inPin = 3;
void setup(){
  Serial.begin(9600);
  pinMode(outPin, OUTPUT);  // sets the digital pin as output
  pinMode(inPin, INTPUT);      // sets the digital pin as input
}
void loop(){
  digitalWrite(outPin, HIGH);   // sets the pin on
  delayMicroseconds(10);       // pauses for 10 microseconds
  digitalWrite(outPin, LOW);    // sets the pin off
  int duration = pulseIn(inPin, HIGH)
  Serial.print(“duration =  ”);
  Serial.print(duration);
}
```

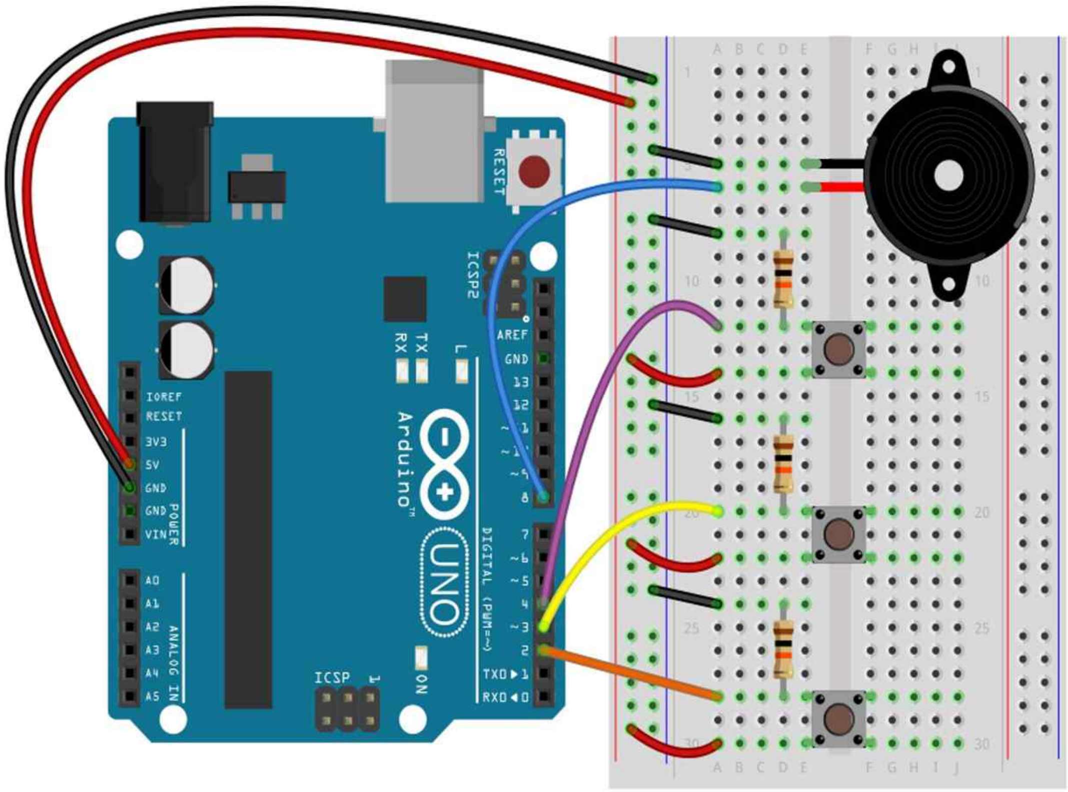
# Lab. 1 – Piezo buzzer (speaker) I

- Play “School bell's ringing Ding Ding Dong”
  - ○ use arrays
- Circuit diagram:



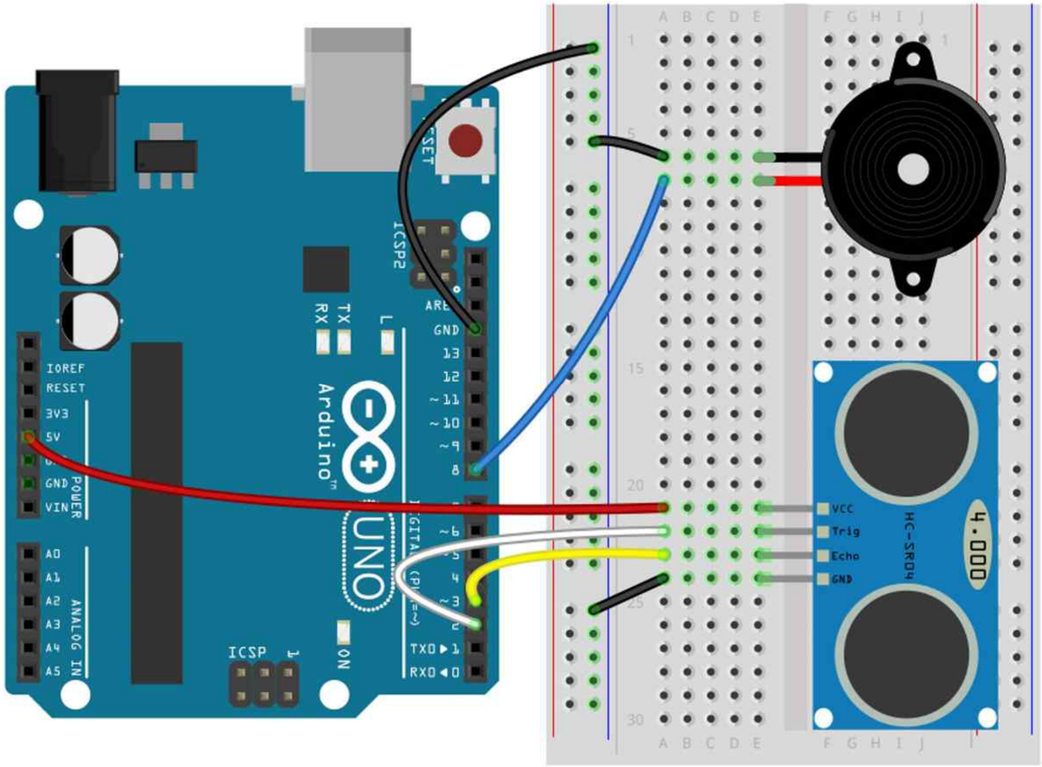
# Lab. 2 – Piezo buzzer (speaker) II

- **Make a sound when the pushbutton is pressed.**
  - ○ e.g. first pushbutton ➔ C(do), second pushbutton ➔ D(re), …
- **Circuit diagram:**



# Lab. 3 - Ultrasonic sensor + Piezo buzzer

- Play the notes (more than 3) depends on a distances.
  - ○ centimeter = duration / 58


Arduino
UNO
IOREF
RESET
3V3
5V
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
ICSP
ICSP2
TX
RX
L
ON
AREF
GND
13
12
~11
~10
~9
8
~6
~5
4
~3
2
TX0 ▸ 1
RX0 ◂ 0
DIGITAL (PWM=~)
VCC
Trig
Echo
GND
HC-SR04
4.000
A B C D E
F G H I J
1
5
15
20
25
30