임베디드컴퓨팅
Embedded Computing
(0009488)
Ultrasonic distance sensor

2022년 2학기

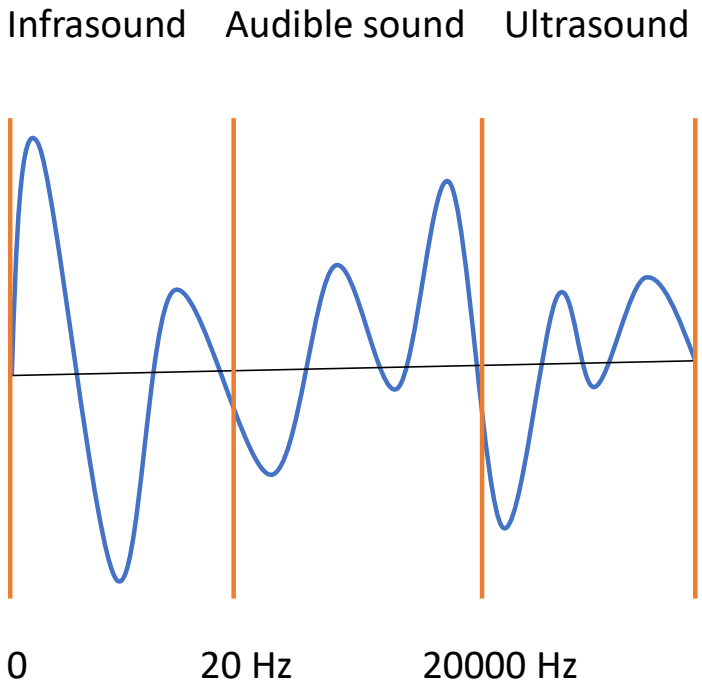
정보기술대학 정보통신공학과
김 영 필
ypkim@inu.ac.kr

INU 인천대학교
INCHEON NATIONAL UNIVERSITY

Ultrasonic distance sensor

- **(HC-SR04 sensor)**
  Provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm.
- Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.
- cf) Ultrasound
  - High-pitched sound waves with a frequency of over (inaudible to humans).

HC-SR04

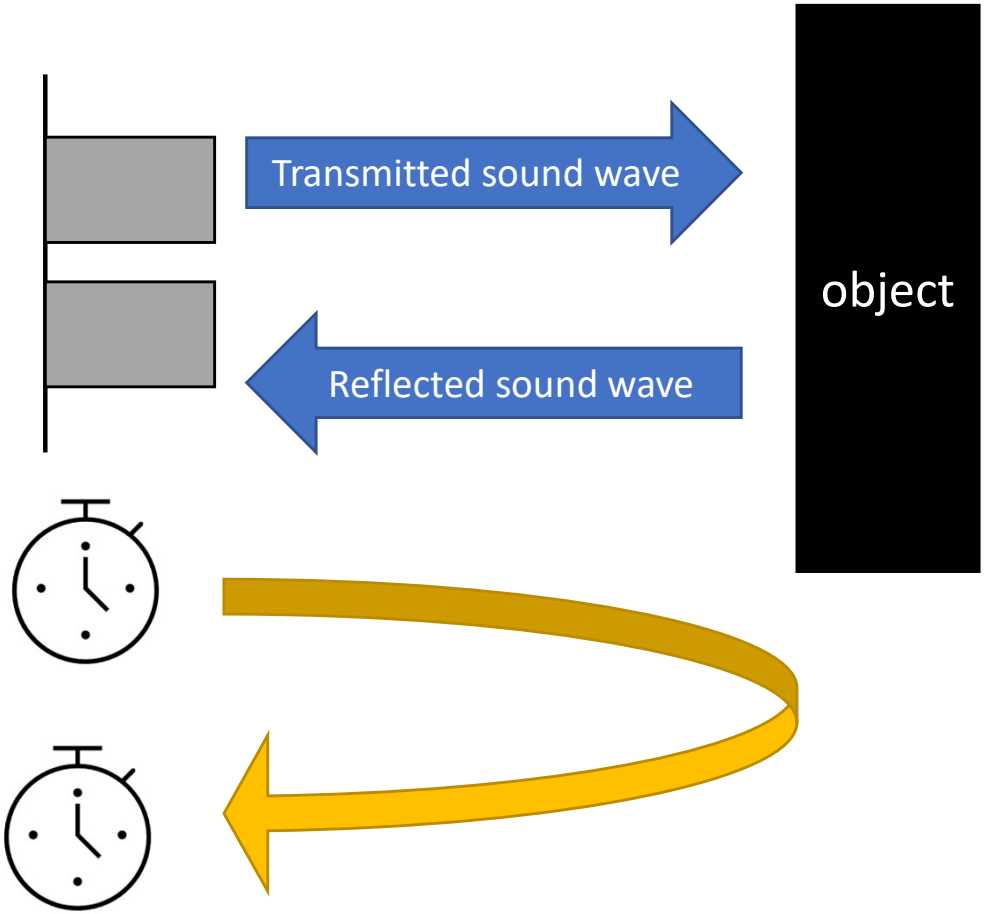


HC-SR04 module pins

- Consists of four pins.
  - **VCC**: A pin that applies + power.
    - Operating voltage = 4.5 ~ 5.5V, connect it to the **5V pin** of the Arduino.
  - **Trig** : A pin that generates a short 10 microsecond (μs or us) pulse
    - Connects it to the **digital pin** of the Arduino.
  - **Echo**: A pin that generates a pulse when ultrasonic waves are detected.
    - Connected to the **digital pin** of the Arduino.
  - **GND**: A grounding pin
    - Connected to the **GND pin** of the Arduino.

How to measure a distance?

- Program
  - Write a signal via Trigger pin
    - at least 10 us in duration
- HC-SR04
  - Transmit a sonic burst of 8 pulses (40 KHz)
  - Wait echo-back signal
    - Echo pin goes HIGH after the burst
  - When receiving echo-back signal
    - Echo pin goes LOW
    - Timeout after 38 ms == No obstruction
- Program
  - Read a Echo duration (T)
    - It's RTT (round time time);
  - One-way Distance = One-way time x Speed
    - cm =



Speed of Sound =
331.5 m/s ➔
0.0343 cm/us

How to get the duration?

- Use Arduino built-in function, **pulseIn()**
  - Reads a pulse (either HIGH or LOW) on a pin.
    - **HIGH** = Elapsed Time (LOW ->HIGH)
    - **LOW** = Elapsed Time (HIGH->LOW)
  - Works on pulses from 10 us to 3 min. in length.
- Syntax
  - pulseIn(pin, value)
  - pulseIn(pin, value, timeout)
- Parameters
  - **pin**: the number of the Arduino pin
  - **value**: type of pulse to read: either HIGH or LOW.
  - **timeout** (optional): the number of us to wait for the pulse to start; default = 1 sec. **(unsigned long)**
- Returns
  - The length of the pulse (in us) or 0 if no pulse started before the timeout. **(unsigned long)**.

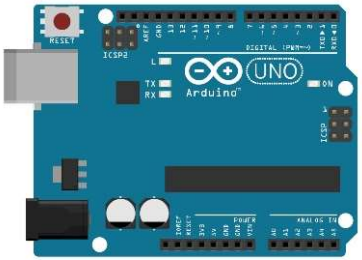
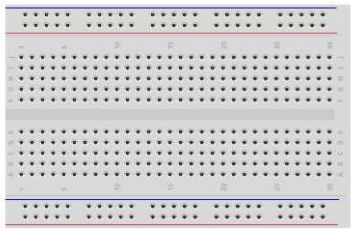
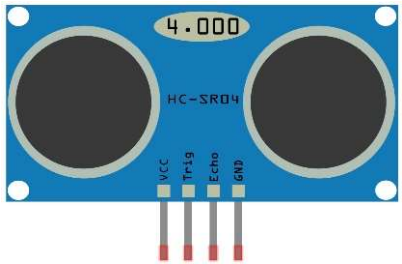
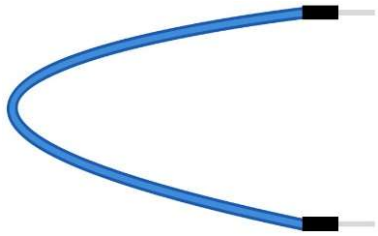
```
int pin = 7;
unsigned long duration;

void setup() {
  Serial.begin(9600);
  pinMode(pin, INPUT);
}

void loop() {
  duration = pulseIn(pin, HIGH);
  Serial.println(duration);
}
```

Ref. - https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/

Lab: Distance finder

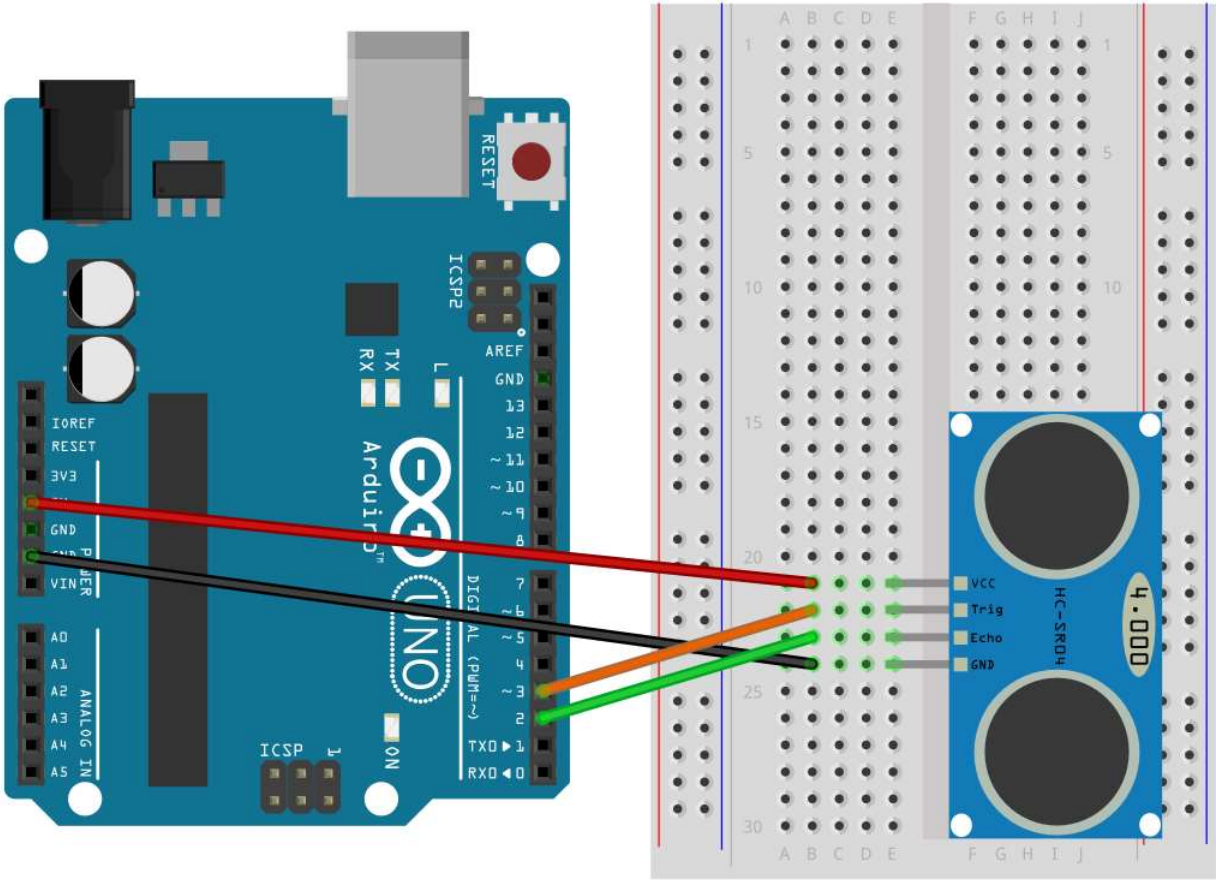
- After calculating the distance using the ultrasonic sensor, let's check the distance data through serial communication.

- Required H/W components

Arduino Uno board	Breadboard	Ultrasonic sensor (HC-SR04)	Jumper cable (Male-Male)
			
x 1	x 1	x 1	x 4

Circuit wiring setup

HC-SR04	Arduino pin
VCC	5V
Trig	digital 3
Echo	digital 2
GND	GND



Basic setup for distance finder

```
#define ECHO 2
#define TRIG 3


void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
}
```

Two digital pins are used:
no. 2 for ECHO, no. 3 for TRIG

TRIG is for a digital output
ECHO is for a digital input

Use serial communication for
log messages

Main loop for distance finder

```
void loop() {
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  long duration =

  if(duration == 0) return;

  long distance =
  Serial.print("Distance : ");
  Serial.print(distance);
  Serial.println("cm");
  delay(2000);
}
```

Start to measure by a Trigger signal;

Wait for 10 us by delayMicroseconds ()

Finish a Trigger signal

Calculate an one-way distance (cm)

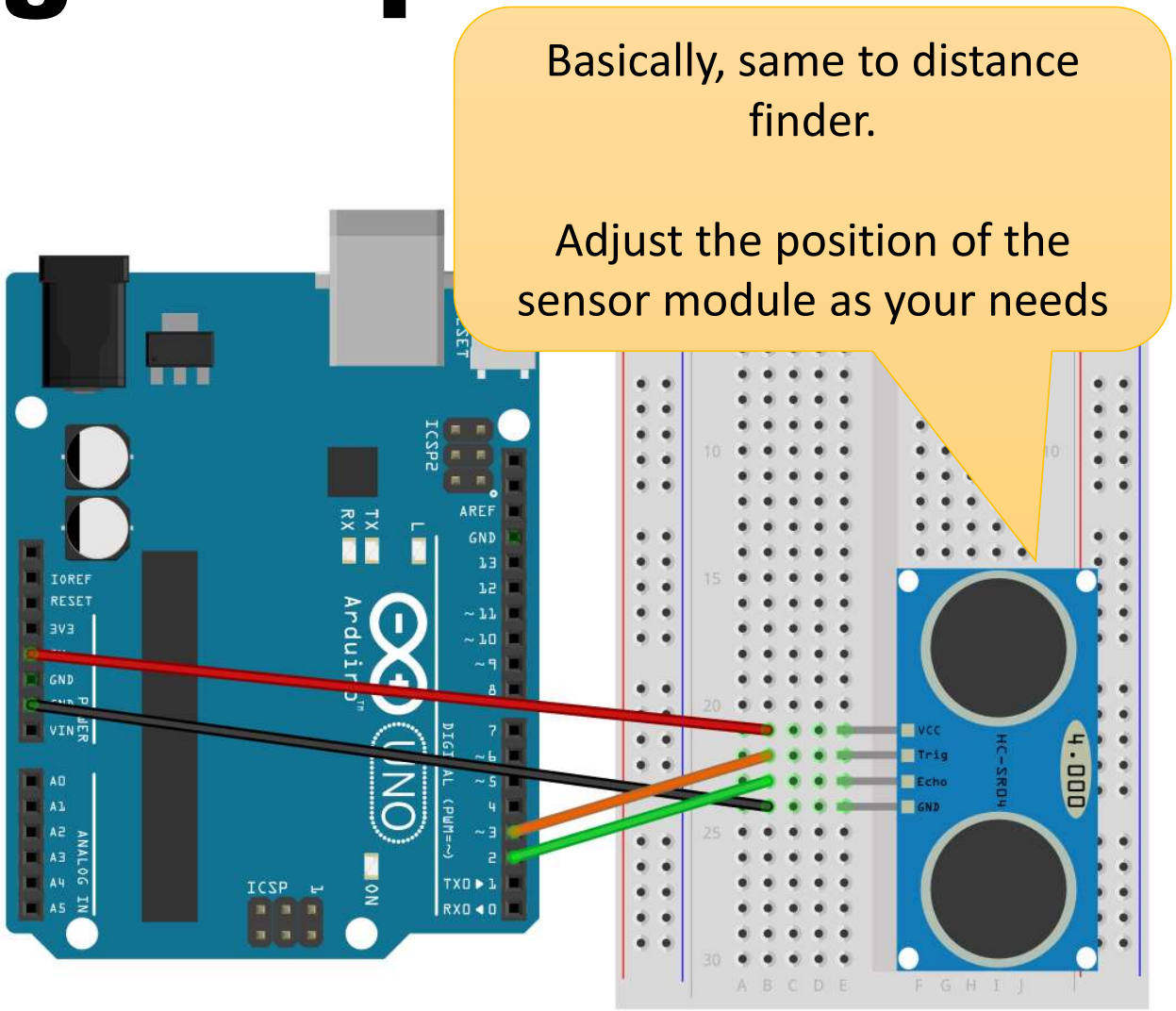
Repeat measuring every 2 secs

Lab: Simple navigator

- One of the core technologies for self-driving is the intelligent navigation.
- Let's make a prototype for the navigation!
- Our self-navi Mark-0 has the following features:
  - Assume the car speed is under 2 cm/sec
  - Only checking forward direction every 2 seconds
  - When no obstruction with in 15 cm, say "Go!" and the car moves
  - Otherwise, say "Stop!" and the car stops
  - Display the cumulative distance of the car every 2 second

Circuit wiring setup

HC-SR04	Arduino pin
VCC	5V
Trig	digital 3
Echo	digital 2
GND	GND



Basic setup for Mark-0

```
#define ECHO 2
#define TRIG 3
#define MARK0_GO 1
#define MARK0_STOP 0
#define CAR_SPEED 2

int askMark0(int dist);
long totalDistance;

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
  totalDistance = 0;
}
```

Two values for Mark0 decision

Car speed is set to 2 cm / s

declaration of a function for Mark0

Need a global variable for total distance

Initialize total distance variable

Main loop for Mark-0

```
void loop() {
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  long duration = pulseIn(ECHO, HIGH);
  if(duration == 0) return;
  long distance = duration * 0.034 / 2;

    Serial.println("Go!");

  } else Serial.println("Stop!");
  Serial.print("Total distance : ");
  Serial.print(totalDistance);
  Serial.println("cm");
  delay(2000);
}
```

Follow Mark0's decision

When GO, calculate cumulative distance of the car

When NO, just show a message

Core function for Mark-0

```
int askMark0 (int dist) {

}
```

Check the driving condition

This function can be extended as needed

Limitations of HC-SR04

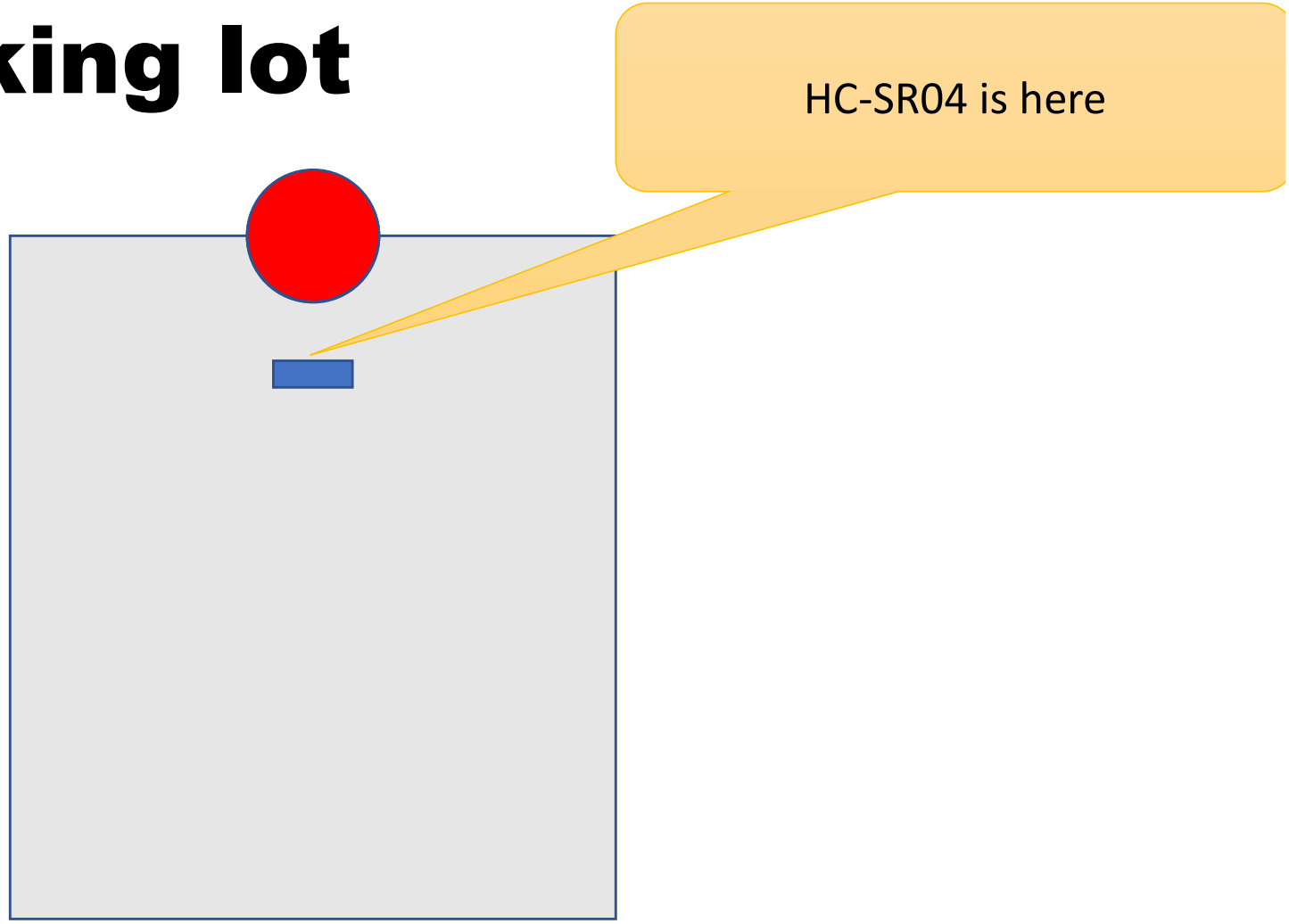
- Generally, work well. But, HC-SR04 cannot measure the distance in the following cases

- 1) Cannot measure the distance of more than 400 cm or less than 2 cm
  - As technical specification

- 2) Sound reflection can fail under the specific condition
  - At a too shallow angle
  - Too small object
  - Too soft object with irregular surface (such as stuffed animals)

Ref - https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/

Assignment:
Smart Parking lot



INU 인천대학교
INCHEON NATIONAL UNIVERSITY

Assignment: Smart Parking lot

- Requirements
  - Based on the ultrasonic and Tri-color examples, write a sketch program as follows.
    - Setup Tri-color LED and HC-SR04 on breadboard
    - Occupied condition = Within 10 cm distance between car and HC-SR04
    - When not occupied, turn on Green LED; otherwise, turn on Red LED
    - Display log messages via Serial communication
  - Write block-type comments in the top of your source code, which includes "your student no., your name, writing date, what you feel about this assignment, etc."
- Results
  - (a source code file) sketch source code (***"sketchfilename.ino"***)
  - (a Arduino board capture file) a photo capture showing how you setup your circuit (max. 1GB file).