ELEN30013 Workshop 5: Rangefinder and Servo

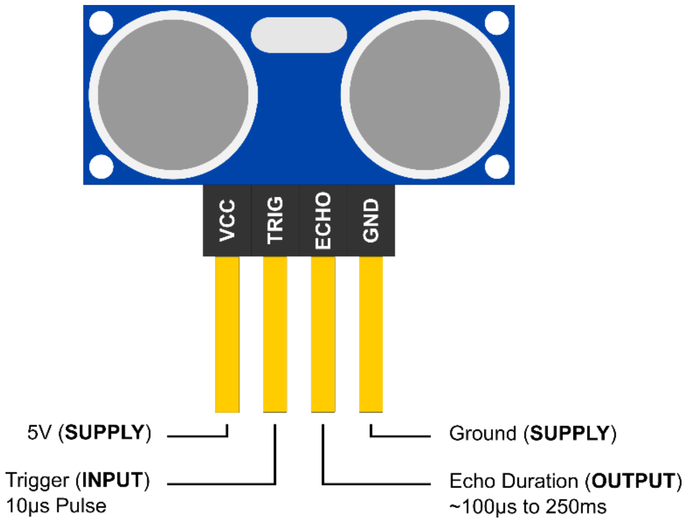
- You will need to bring: A ruler, Optionally a protractor
- Assessment: show your work for each to the demonstrator to get marked off. (You can get marks in Week 6 workshop too in case you couldn't finish in Week 5)
- Attendance (Max 20 points)

Part 1. Ultrasonic Rangefinder

An ultrasonic rangefinder is a sensor that can measure a distance by timing the return time of a sound. While there are different sensors such as IR rangefinders, Time of Flight and Stereoscopic or other image-based approaches, the ultrasonic sensor is often a robust and low cost option. Just like the other sensors each type and model of sensor will have different strengths and weaknesses.

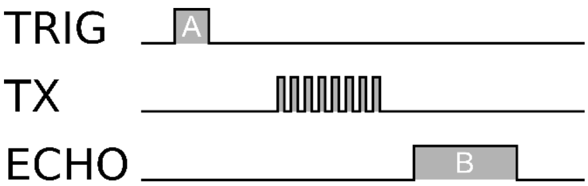
The ultrasonic rangefinder module that will be used in this workshop is the HC-SR04.

Datasheet: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf



The HC-SR04 has four pins; a 5V and ground supply pins, a trigger input (TRIG) and an echo output (ECHO). The nominal use of the device is as follows:

1. A pulse of 10 μs is sent to the trigger input,
2. The device then sends out 8 acoustic pulses at 40KHz,
3. The device then receives the echo of these acoustic pulses reflected back at the HC-SR04,
4. The echo pin then outputs a pulse with a duration equal to the round trip time; from the HC-SR04 to the device and back to the HC-SR04.
5. The speed of sound is then used to convert the echo pulse duration to a distance. This may be altered by using a calibration curve generated using known measurements or temperature, humidity and pressure sensor data.



Time A: 10μs

Time B: ~100μs to 250ms
Distance = Time B × SPEED_OF_SOUND / 2
Distance (cm) ≈ Time B / 58

Task 1 (10 Points)

The first task is to create a **function** that will send a 10 μs pulse to the TRIG pin using the Arduino.

Task 2 (10 points)

Research the following Arduino function PulseIN() and micros() and decide which function to use for measuring the duration of the ECHO pulse.

(Digitalread, Delay based, Clock based)

Task 3 (10 Points)

Put your code together so that every 500 ms the Arduino will measure the distance and prints via serial at a baud rate of 9600 echo duration in microseconds as an integer and (optional: displays the distance on the LCD shield). Test the code by placing a large flat object every 10 cm from 10 cm to 1m in front of the sensor and plot your results. Use a linear calibration curve using least square regression.

Distance (M)	**Echo time (μs)**

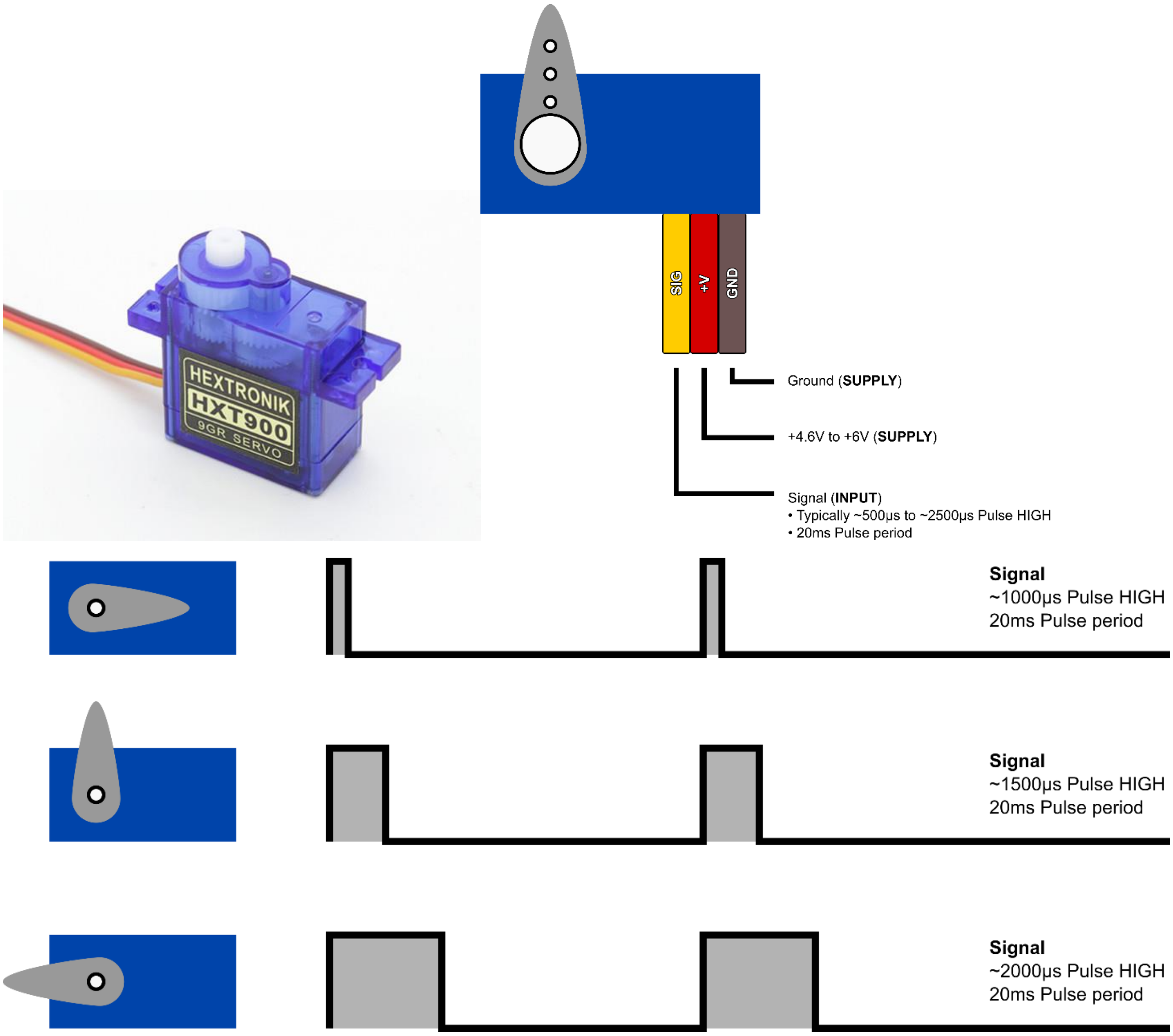
Questions:

1. Based on your measurements using the ruler data vs time duration data calculate the speed of sound in ms^-1 ? **(5 points)**
2. Consider and list what factors might have an impact on the distance measurements of the HC-SR04. **(5 points)**

Part 2. Servo Motors

A servo motor is a motor with feedback to control the position of the motor which is done within the device. They commonly are limited to 180 degrees of rotation with position feedback, although there are some variations that have different ranges of rotation, and some can freely rotate continuously with velocity control instead of position control.

NOTE: Serial print is a slow function so be careful when using serial print and delays in this workshop



A longer pluse duration HIGH should correspond to movement in the counter clockwise direction. This does not have to be the case although for consistantcey this will be the case for the workshop.

Task 1 (10 points)

Write a function that will generate the input signal to the servo generates a plot of Pulse duration vs position. Use the following connections:

Servo +V to Arduino 5V

Servo GND to Arduino GND

Servo Sig to Arduino pin 9

You may wish to use a small duration such as 300 to make the motor move to zero degrees. The plastic connector can be removed and placed in a coinvent. Eyeball the angle or use a protractor.

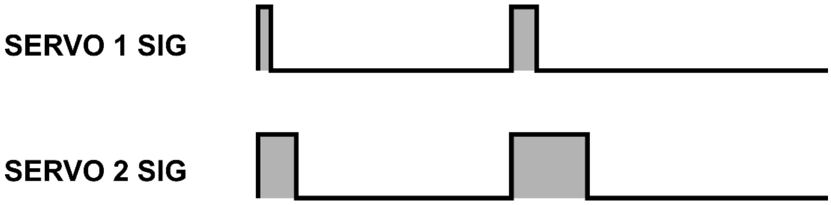
Task 2 (10 points)

Write a small simple servo controller function than can move 2 servo motors simultaneously. Test it by creating a minimal serial interface that accepts the two motors position in degrees. Eg.

```
120,15\n
```

One approach you could use is to make use of delays so that the servos receive a HIGH signal at the same time and one delay is called until the shortest HIGH pulse duration has been sent then that servo signal is set to low and a delay for the difference in HIGH time is called before setting both servo signals LOW and waiting for the remaining time to make the signal period 20ms. This would then repeat each time making use of the current desired position of each servo.

Be careful using serial when using delay as it may cause timing errors. Using delays for a proof of concept is sufficient for this workshop. You may wish to make use of a time based approach that frequently checks the time using millis(), a stored time and the desired period.



An alternative approach is to poll micros() or use timers in a similar fashion. Note that the overflow of micros will occur but so long as the period is less than micros overflow can be delt with as a result of it being unsigned and using micros()-time > period.

Task 3 (10 points) Write a library

After completing the previous tasks and the Library demonstration convert your dual servo code to a library. While there exists a very good Arduino servo library already this will not always be the case. Being able to create and modify libraries is a critical skill when dealing with less common microcontrollers and ICs.

```
int t_total  = 20*1000;
int t_1      = 1.5*1000;
int t_2      = t_total-t_1;
int pinServo = 9;
void setup() {
    pinMode(pinServo,OUTPUT);
}

void loop() {
    digitalWrite(pinServo,HIGH);
    delayMicroseconds(t_1);
    digitalWrite(pinServo,LOW);
    delayMicroseconds(t_2);
}
```