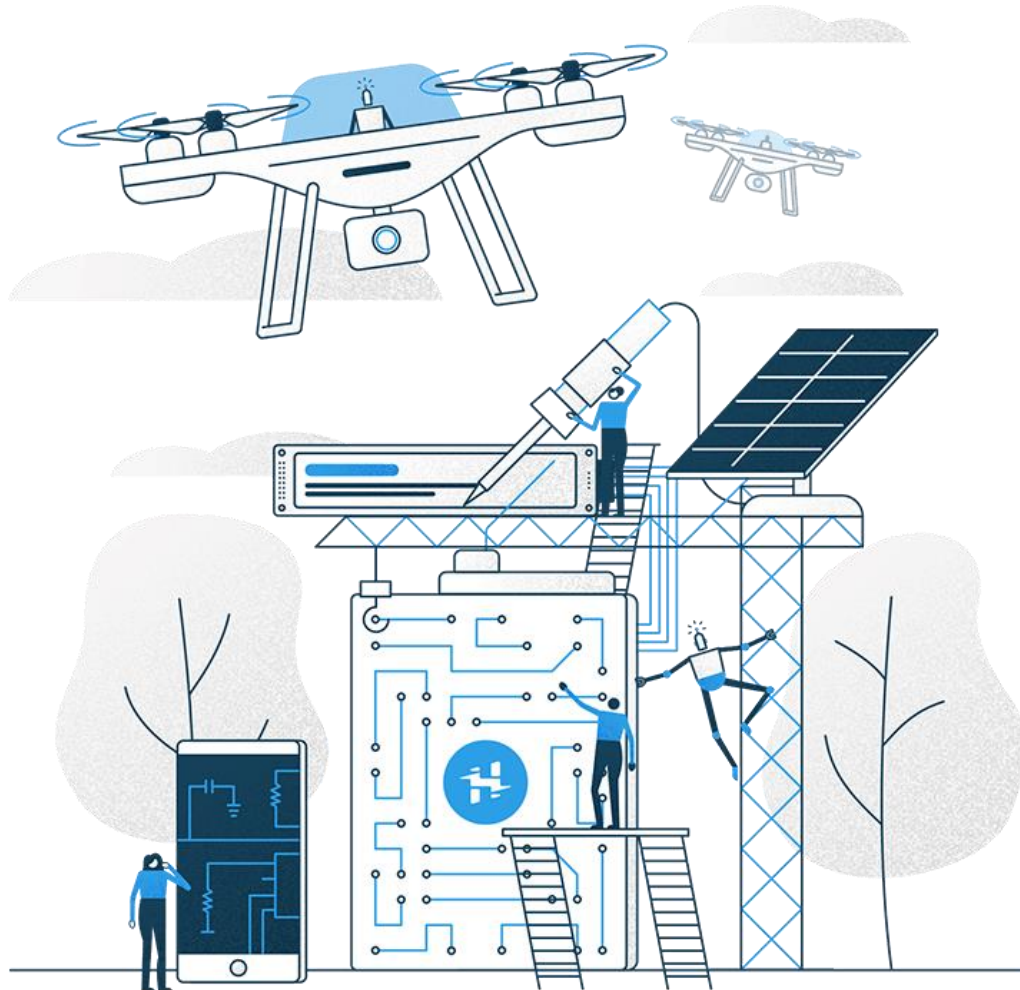


МІКРОПРОЦЕСОРНІ СИСТЕМИ УПРАВЛІННЯ



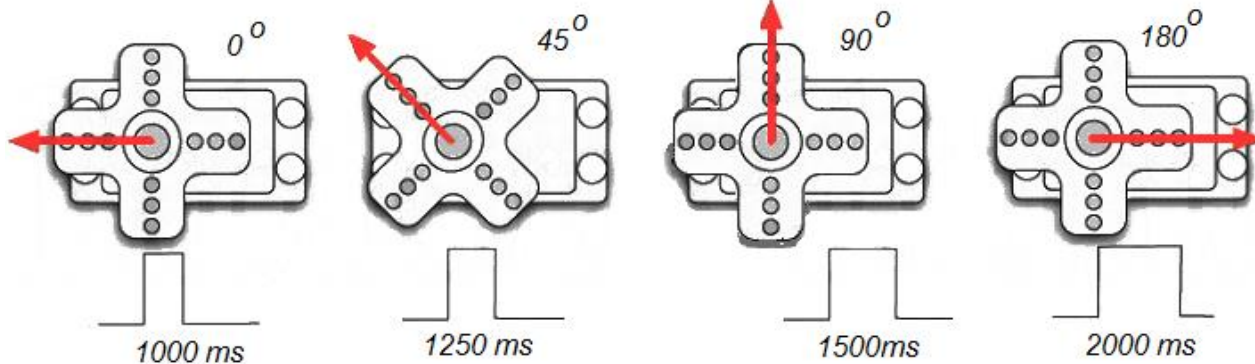
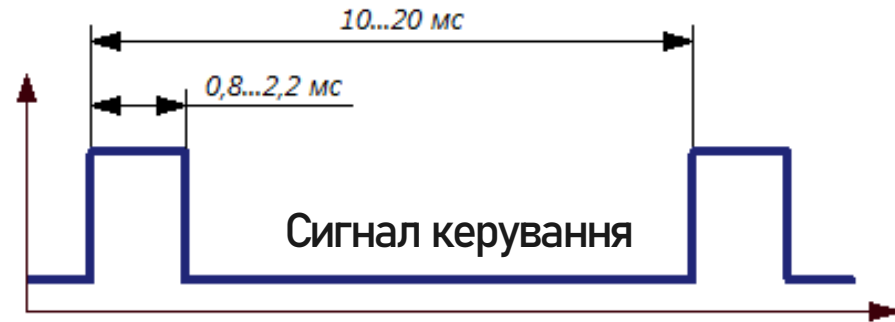
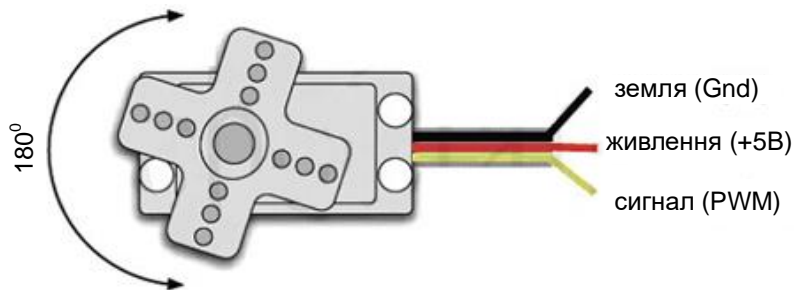
Lesson 13

Серводвигун

Сервопривід – це тип механічного приводу, до складу якого входить датчик (положення, швидкості, сили) та блок керування приводом, який автоматично підтримує необхідні параметри на датчику та на пристрої згідно заданому зовнішньому значенню.

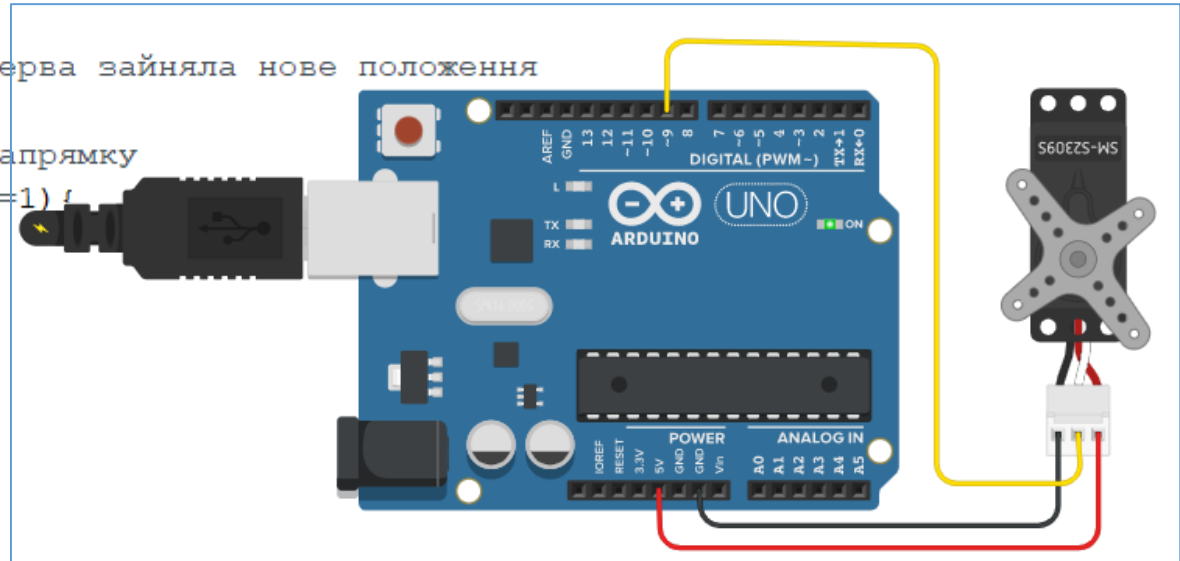
Сервопривід – мотор-редуктор дозволяє повертати вихідний вал строго у задане положення (на кут) та утримати його там.

Сервопривід живиться від постійної напруги в діапазоні від 4,8 В до 6В. Підключається сервопривід через універсальний з'єднувач (рис. 7.18) з трьома контактами: чорний – загальний провід або земля (GND), червоний – напруга живлення (+Vcc), жовтий – сигнал керування.



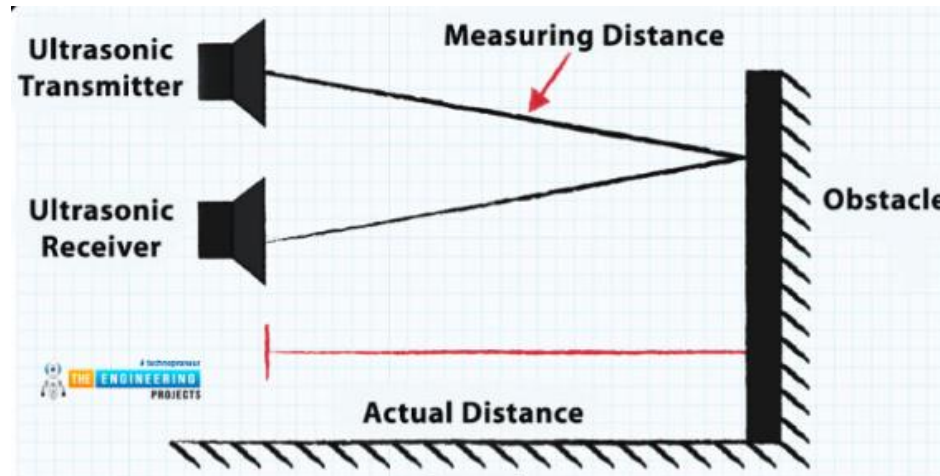
Серводвигун

```
1 #include <Servo.h>
2 Servo myservo;
3 // створюємо об'єкт для контролю серводвигуна
4 // максимальна кількість таких об'єктів — 8
5 int pos = 0;
6 // змінна для зберігання позиції серви
7 void setup()
8 {
9   myservo.attach(9); // серводвигун підключений до D9
10 }
11 void loop()
12 {
13   for(pos = 0; pos < 180; pos += 1) {
14     // від 0 до 180 градусів
15     myservo.write(pos);
16     // встановлюємо положення
17     delay(15);
18     // чекаємо 15 мс, щоб серва зайняла нове положення
19   }
20 //обертаємо у зворотньому напрямку
21 for(pos = 180; pos>=1; pos--=1){
22   myservo.write(pos);
23   delay(15);
24 }
25 }
```



Ультразвуковий датчик

Ультразвуковий датчик складається з двох частин: передавача та приймача. Найпоширеніший варіант, коли вони розташовані поруч якомога ближче (рис. 1). Менші похибки вимірювання досягаються, коли приймач знаходиться поблизу випромінювача, оскільки шлях звуку від джерела до місця призначення більш прямий. Крім того, функції передавача та приймача деяких ультразвукових датчиків об'єднані в один пристрій, що максимально знижує неточність і одночасно зменшує розмір друкованої плати пристрою.



Ультразвукові датчики випромінюють звук, частота якого набагато вище, ніж діапазон людського слуху, тому їх називають «ультразвуковими». Цей звук використовується для обчислення тривалості відбиття звуку від предмета. Цей метод заснований на принципах ехолокації, які використовують кажани для виявлення своєї жертви. З огляду на це, легко перетворити час ультразвукової хвилі у відстань, оскільки звук поширюється зі швидкістю 343 м/с в повітрі при температурі навколишнього середовища.

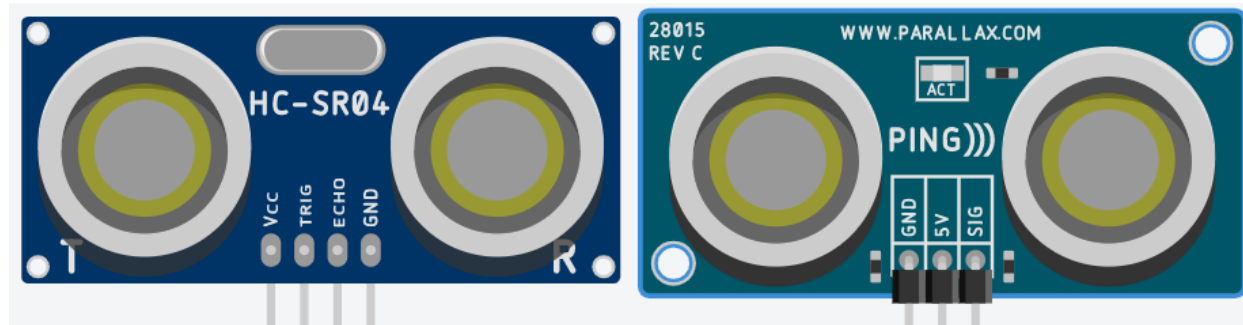
Ультразвуковий датчик

$$\text{Distance (meters)} = (\text{time elapsed [seconds]} * 343 \text{ [meters/second]}) / 2$$

Наприклад, ультразвуковому датчику, спрямованому на коробку, потрібно 0,025 секунди, щоб відбитися від поверхні та повернутися назад, що вказує на відстань між датчиком і коробкою:

$$D = 0,5 \times 0,025 \times 343 = 4,2875\text{м}$$

Модуль HC-SR04 підключається чотирма дротами. Контакти VCC та GND служать для підключення живлення, а Trig та Echo– для відправлення та прийому сигналів давача.

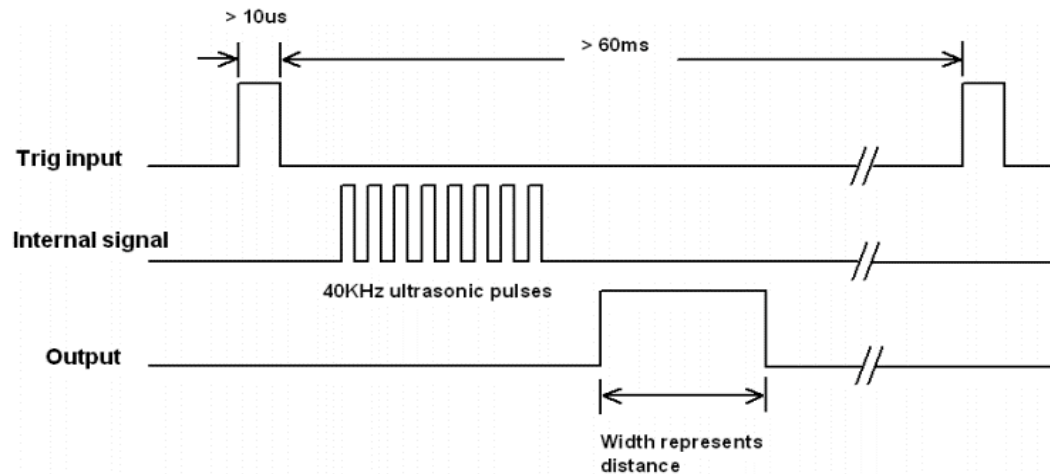


Характеристики датчика: напруга живлення 5В, ефективний кут <math><15^\circ</math>, діапазон вимірювання відстані 2–450 см, максимальна точність датчика 0.3 см.

Ультразвуковий датчик

Для того, щоб почати відправлення сигналу датчиком, необхідно подати високий сигнал тривалістю 10 μs на пін **Trig**, після чого модуль генерує пучок із восьми сигналів частотою 40 кГц та встановлює високий рівень на пині **Echo**.

Після отримання відбитого сигналу модуль встановлює на пині **Echo** низький рівень



Знаючи тривалість високого сигналу на пині Echo, можемо обчислити відстань, помноживши час, який витратив звуковий імпульс, перш ніж повернувся до модуля, на швидкість поширення звуку в повітрі (343 м/с).

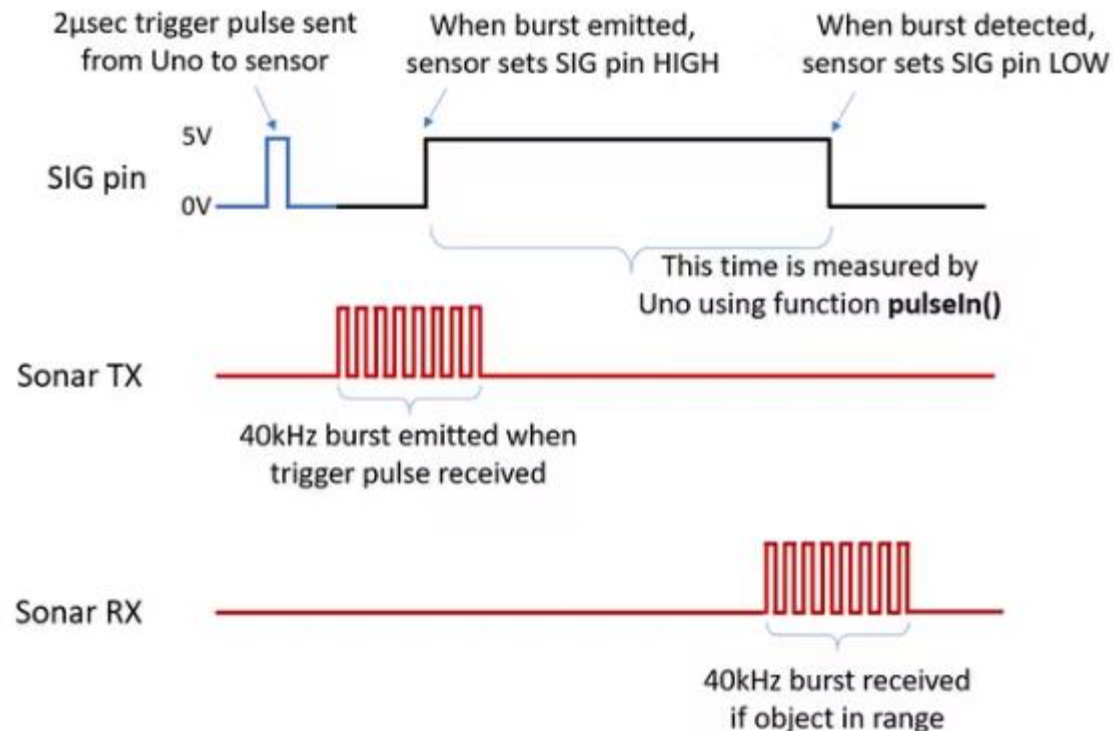
Функція `pulseIn` дозволяє дізнатися тривалість імпульсу μs . Результат роботи цієї функції збережемо у змінну `duration`. Обчислимо відстань:

$$\begin{aligned} \text{distance} &= \text{duration} * 343 \text{ м/с} = \text{duration} * 0.0343 \text{ см/мкс} \\ \text{distance} &= \text{duration} * 1/29 = \text{duration} / 29 \end{aligned}$$

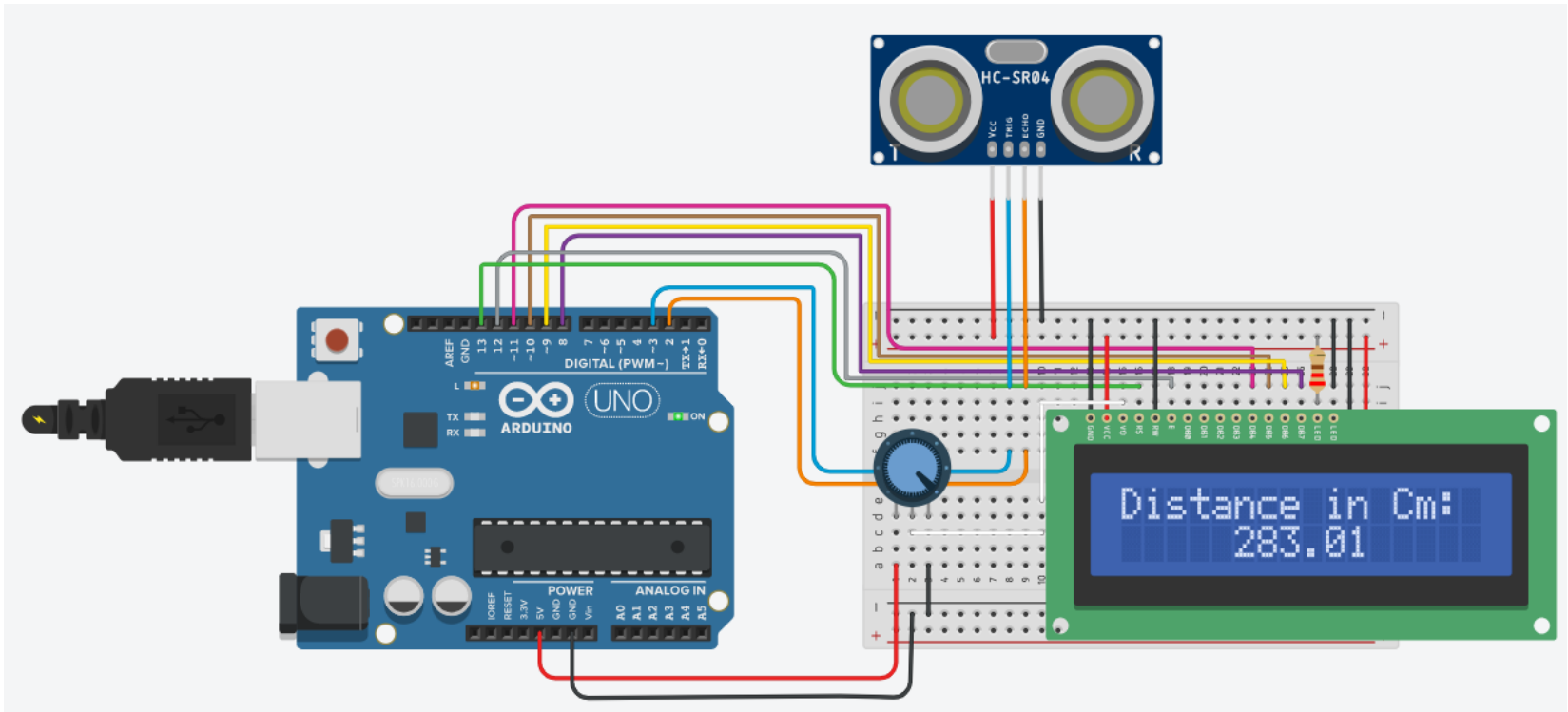
$$\text{distance} = \text{duration} / 58$$

Ультразвуковий датчик

Ультразвуковий датчик SEN136B5B (Ping) Parallax складається з 3-х контактів: GND, 5V і SIG. Вивід SIG використовується як комбінований вхід/ вихід. Щоб ініціювати початок вимірювання, MCU повинен надіслати HIGH імпульс протягом 2 мкс. Потім датчик надсилає вісім імпульсів, встановлює вивід SIG в HIGH і чекає, поки ультразвукові хвилі повернуться назад (луна). Коли відлуння виявлено, датчик встановлює на виводі SIG LOW. Ширина цього імпульсу HIGH на виводі SIG відповідає подвійній відстані до цілі. Різниця в кольорі форми сигналу SIG вказує, який пристрій керує контактом. Наприклад, Arduino керує контактом, щоб надіслати імпульс (синій), а потім датчик керує контактом, щоб надіслати імпульс HIGH (чорний).



Ультразвуковой датчик



```
#include <LiquidCrystal.h> //LCD library
```

```
#define echo 2
```

```
#define trig 3
```

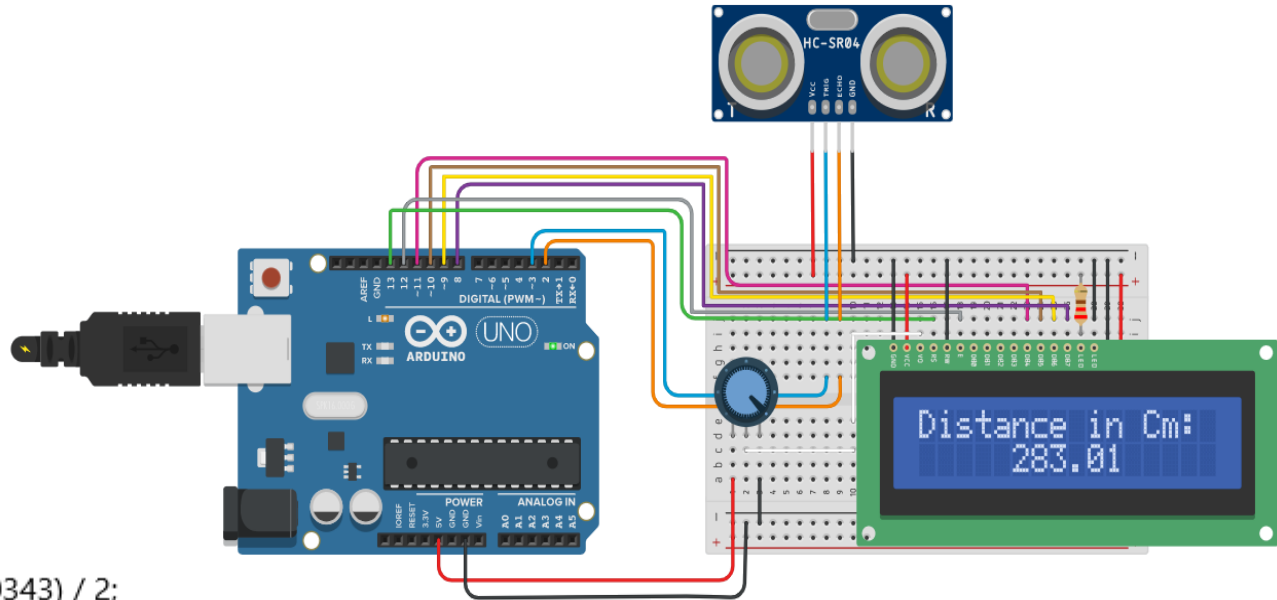
```
float duration; // time taken by the pulse to return back
```

```
float distance; // oneway distance travelled by the pulse
```

```
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); //lcd(RS,EN,D4,D5,D6,D7)
```

```
void setup() {  
  pinMode(trig, OUTPUT);  
  pinMode(echo, INPUT);  
  Serial.begin(9600);  
  lcd.begin(16, 2);  
}
```


Ультразвуковий датчик



```
void loop() {  
  time_Measurement();  
  distance = duration * (0.0343) / 2;  
  // calculate the oneway distance travelled by the pulse  
  display_distance();  
}
```

```
void time_Measurement() {  
  //function to measure the time taken by the pulse to return bar  
  digitalWrite(trig, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trig, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trig, LOW);  
  
  duration = pulseIn(echo, HIGH);  
}
```

```
void display_distance() {  
  //function to display the distance on LCD/Serial Monitor  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  Serial.print("Distance in Cm: ");  
  Serial.print(distance);  
  Serial.println();  
  lcd.print("Distance in Cm: ");  
  lcd.setCursor(5, 1);  
  lcd.print(distance);  
  delay(1000);  
}
```

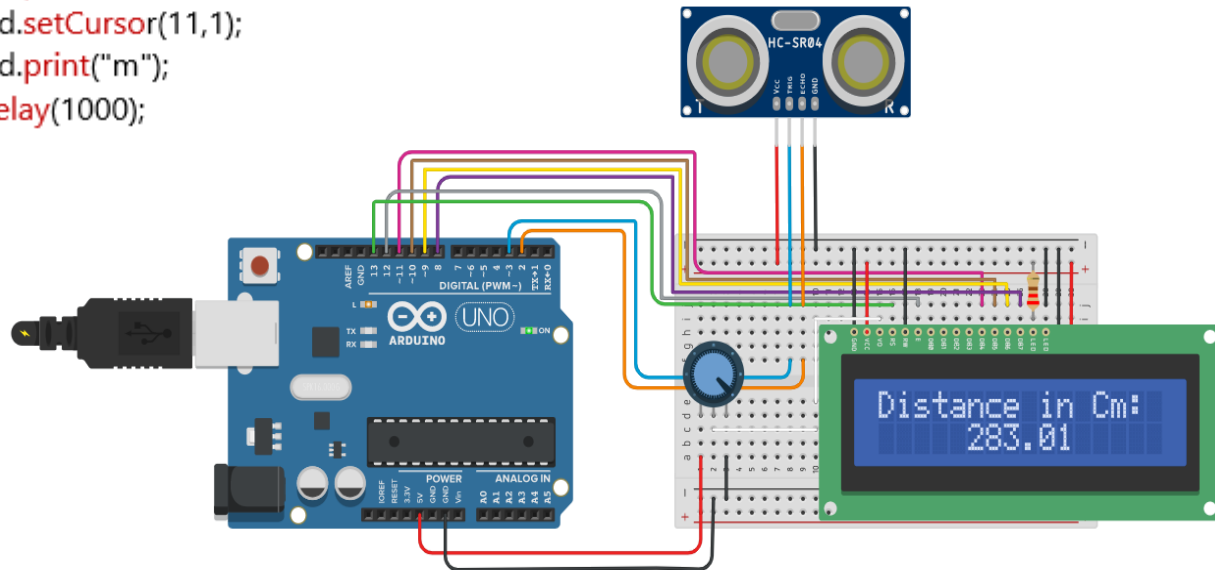
Ультразвуковий датчик

```
#include <LiquidCrystal.h>
#define ECHO 2
#define TRIGGER 3
float time = 0, distance=0;
LiquidCrystal lcd(13,12,11,10,9,8);
int num=1;

void setup() {
  pinMode(TRIGGER, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("Ultrasonic");
  lcd.setCursor(0,1);
  lcd.print("Distance Meter");
  delay(2000);
  lcd.clear();
  lcd.print("Circuit Digest");
  delay(2000);
}
```

```
void loop() {
  lcd.clear();
  digitalWrite(TRIGGER, HIGH);
  delay(2);
  digitalWrite(TRIGGER, LOW);

  time = pulseIn(ECHO,HIGH);
  distance = 0.01716*time;
  Serial.println(time);
  lcd.setCursor(5,0);
  lcd.print(float(distance));
  lcd.setCursor(5,1);
  lcd.print(float(distance/100));
  lcd.setCursor(11,0);
  lcd.setCursor(11,1);
  lcd.print("cm");
  lcd.setCursor(11,1);
  lcd.print("m");
  delay(1000);
}
```



Ультразвуковой датчик

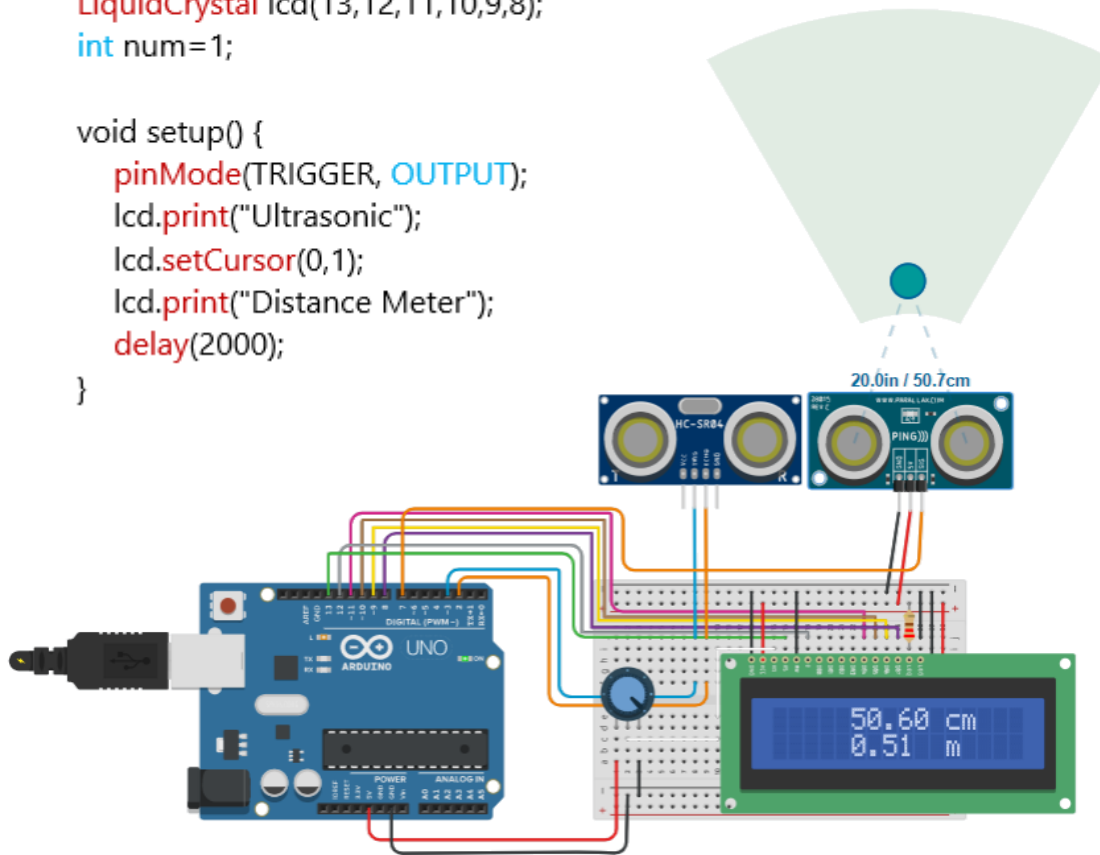
```
#include <LiquidCrystal.h>
#define PING 7
```

```
float time = 0, distance=0;
LiquidCrystal lcd(13,12,11,10,9,8);
int num=1;
```

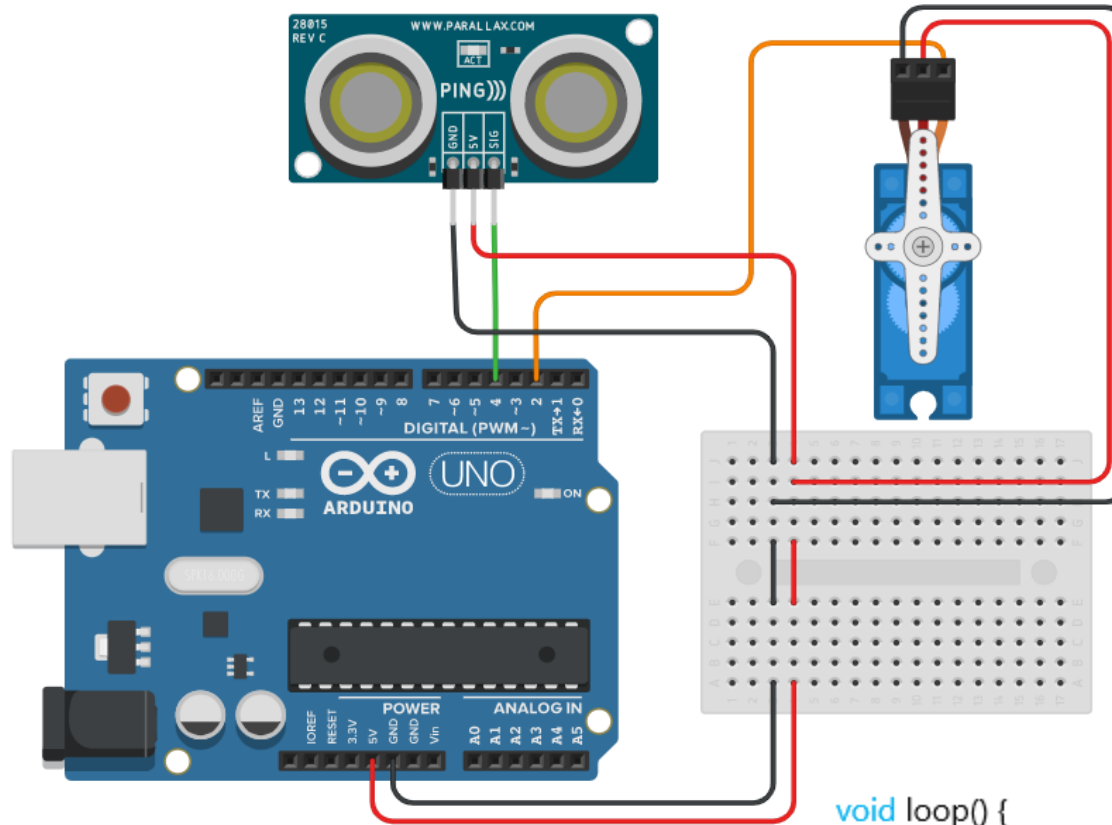
```
void setup() {
  pinMode(TRIGGER, OUTPUT);
  lcd.print("Ultrasonic");
  lcd.setCursor(0,1);
  lcd.print("Distance Meter");
  delay(2000);
}
```

```
void loop() {
  lcd.clear();
  pinMode(PING, OUTPUT); // Clear the trigger
  digitalWrite(PING, LOW);
  delayMicroseconds(2);
  digitalWrite(PING, HIGH);
  delayMicroseconds(10);
  digitalWrite(PING, LOW);
  pinMode(PING, INPUT);
  time = pulseIn(PING,HIGH);
  distance = 0.01716*time;
  Serial.println(time);
  lcd.setCursor(5,0);
  lcd.print(float(distance));
  lcd.setCursor(5,1);
  lcd.print(float(distance/100));
  lcd.setCursor(11,0);
  lcd.print("cm");
  lcd.setCursor(11,1);
  lcd.print("m");

  delay(1000);
}
```



Ультразвуковой датчик



```
#include <Servo.h>  
const int pingPin = 4;  
const int servoPin = 2;  
Servo servoMotor;
```

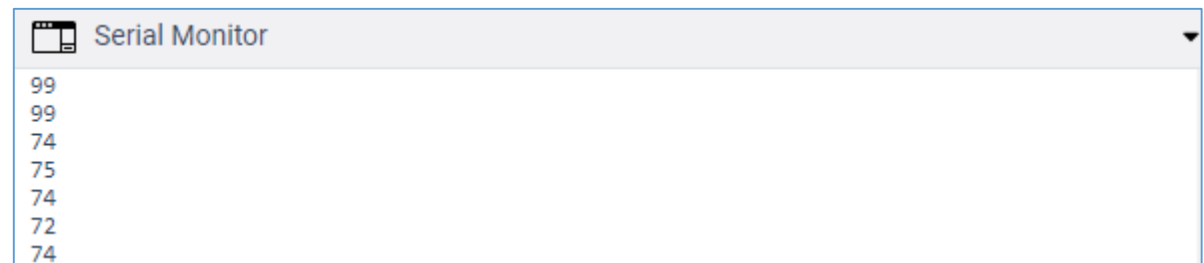
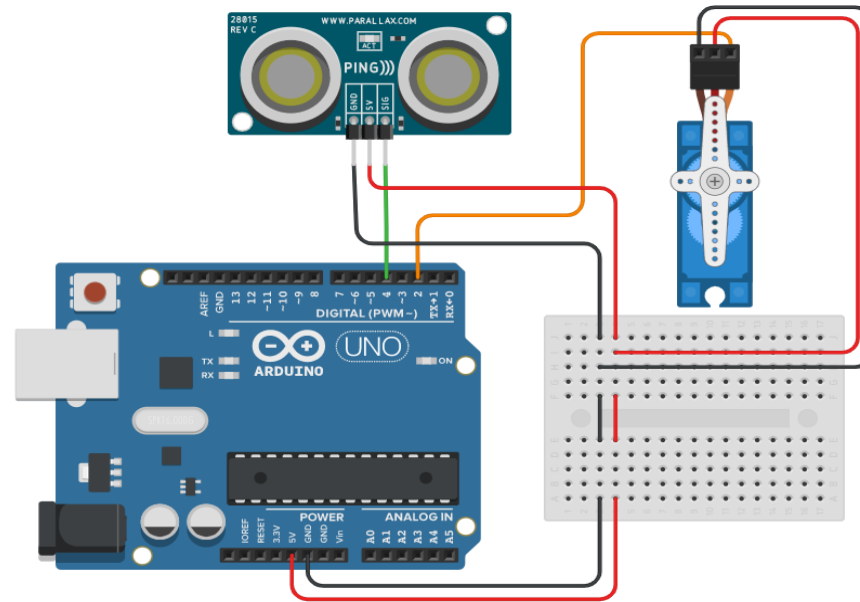
```
void setup() {  
  Serial.begin(9600);  
  servoMotor.attach(servoPin);  
}
```

```
void loop() {  
  long duration, cm;  
  pinMode(pingPin, OUTPUT);  
  digitalWrite(pingPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(pingPin, HIGH);  
  delayMicroseconds(5);  
  digitalWrite(pingPin, LOW);  
  pinMode(pingPin, INPUT);  
  duration = pulseIn(pingPin, HIGH);
```

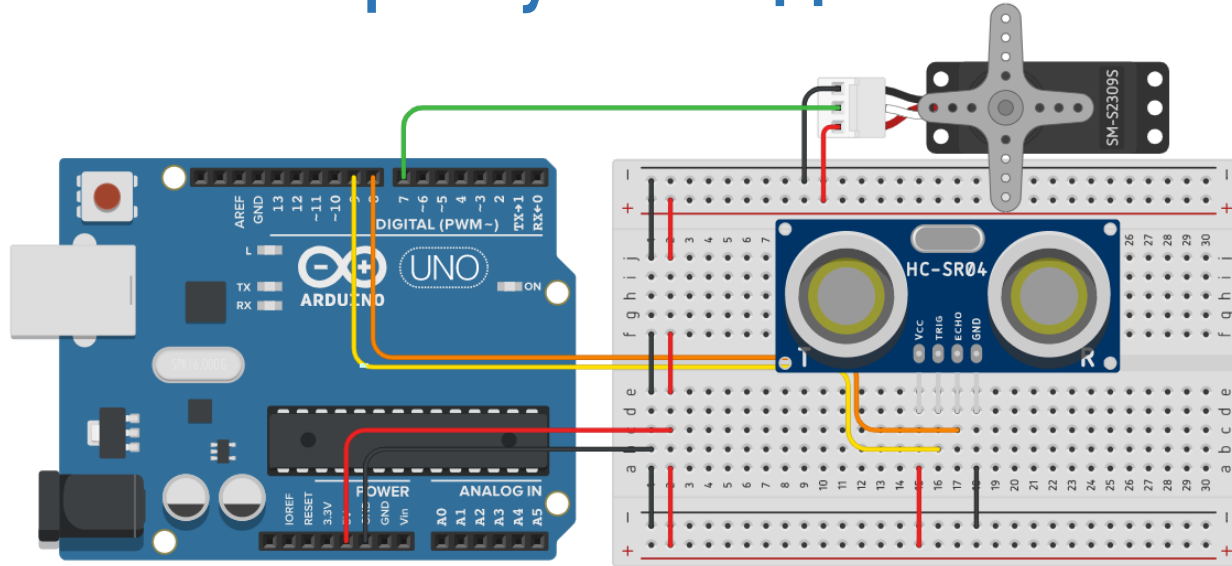
Ультразвуковой датчик

```
// convert the time into a distance
```

```
cm = duration/ 29 /2;  
if(cm <=25){  
  servoMotor.write(0);  
}  
else if(cm <= 50){  
  servoMotor.write(30);  
}  
else if(cm <= 100){  
  servoMotor.write(50);  
}  
else if(cm <= 150){  
  servoMotor.write(70);  
}  
else if(cm <= 170){  
  servoMotor.write(100);  
}  
else if(cm <= 200){  
  servoMotor.write(130);  
}  
else if(cm <= 250){  
  servoMotor.write(150);  
}  
else if(cm > 250){  
  servoMotor.write(180);  
}  
}  
delay(100);  
Serial.println(cm);  
}
```



Ультразвуковий датчик



```
#include <Servo.h>
```

```
Servo servo;
```

```
int trigPin = 9;
```

```
int echoPin = 8;
```

```
int pinServo = 7;
```

```
int angle;
```

```
long distance;
```

```
long duration;
```

```
void setup() {
```

```
    servo.attach(pinServo);
```

```
    pinMode(trigPin, OUTPUT);
```

```
    pinMode(echoPin, INPUT);
```

```
}
```

```
void loop() {
```

```
    ultra();
```

```
    servo.write(0);
```

```
    angle = map(distance, 0,335,0,180);
```

```
    servo.write(angle);
```

```
}
```

```
void ultra() {
```

```
    digitalWrite(trigPin, LOW);
```

```
    delayMicroseconds(2);
```

```
    digitalWrite(trigPin, HIGH);
```

```
    delayMicroseconds(10);
```

```
    digitalWrite(trigPin, LOW);
```

```
    duration = pulseIn(echoPin, HIGH);
```

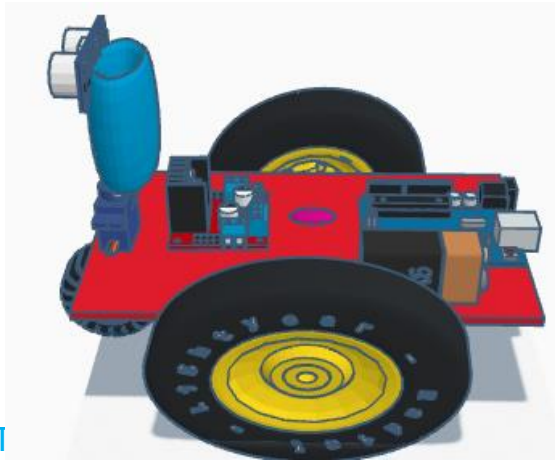
```
    distance = duration*0.034/2;
```

```
}
```

Ультразвуковой датчик

```
int echoPin = 4;
int trigPin = 3;
int MOTORB_IN1=9;
int MOTORB_IN2=10;
int MOTORA_IN1=7;
int MOTORA_IN2=8;
void setup()
{
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(MOTORA_IN1, OUTPUT); //НАСТРОЙКА МОТОРА
  pinMode(MOTORA_IN2, OUTPUT);
  pinMode(MOTORB_IN1, OUTPUT); //НАСТРОЙКА МОТОРА
  pinMode(MOTORB_IN2, OUTPUT);
}

void loop()
{
  Serial.print(distance());
  Serial.println(" cm");
  delay(100);
  if(distance()<25){
    forward();
    delay(3000);
  }
}
```



```
float distance(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int t = pulseIn(echoPin, HIGH);
  float cm = t/58; //час в см
  return cm;
}

void forward()
{
  digitalWrite(MOTORA_IN1, LOW);
  digitalWrite(MOTORA_IN2, HIGH);
  digitalWrite(MOTORB_IN1, HIGH);
  digitalWrite(MOTORB_IN2, LOW);
}

void backward()
{
  digitalWrite(MOTORA_IN1, HIGH);
  digitalWrite(MOTORA_IN2, LOW);
  digitalWrite(MOTORB_IN1, LOW);
  digitalWrite(MOTORB_IN2, HIGH);
}
```

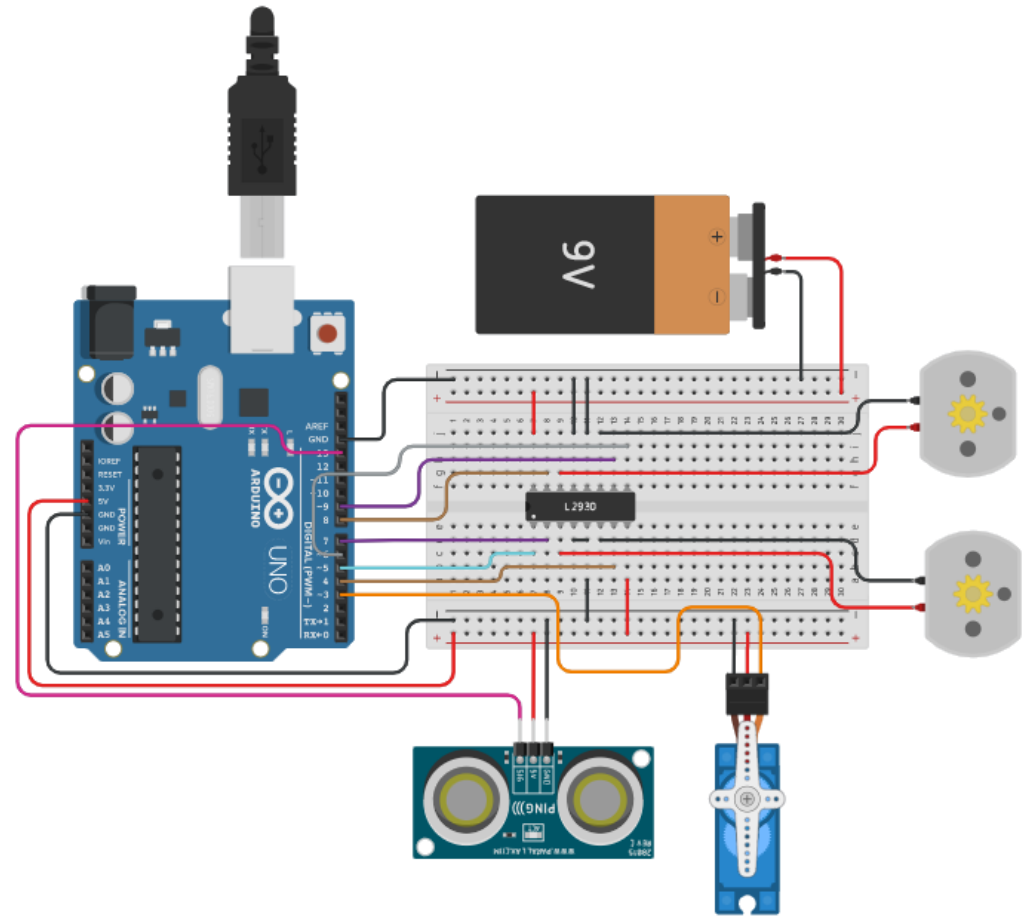
Ультразвуковой датчик

```
void Turn_right()
{
    digitalWrite(MOTORA_IN1, LOW);
    digitalWrite(MOTORA_IN2, LOW);
    digitalWrite(MOTORB_IN1, LOW);
    digitalWrite(MOTORB_IN2, HIGH);
}

void Turn_left()
{
    digitalWrite(MOTORA_IN1, LOW);
    digitalWrite(MOTORA_IN2, HIGH);
    digitalWrite(MOTORB_IN1, LOW);
    digitalWrite(MOTORB_IN2, LOW);
}

void RobotStop()
{
    digitalWrite(MOTORA_IN1, LOW);
    digitalWrite(MOTORA_IN2, LOW);
    digitalWrite(MOTORB_IN1, LOW);
    digitalWrite(MOTORB_IN2, LOW);
}

void turnrobot()
{
    digitalWrite(MOTORA_IN1, LOW);
    digitalWrite(MOTORA_IN2, HIGH);
    digitalWrite(MOTORB_IN1, LOW);
    digitalWrite(MOTORB_IN2, HIGH);
}
}
```



Домашнє завдання

Розробити програму ультразвукового радару (датчик HC-SR04) який в залежності від зони Green, Mid, Danger включає відповідні світлодіоди та подає звуковий сигнал

Green Zone - >250 LED-G1

Green Zone - <250 та ≥ 200 LED-G1, LED-G2

Mid Zone - <200 та ≥ 150 LED-Y1, Beep -tone1

Mid Zone - <150 та ≥ 125 LED-Y1, LED-Y2, Beep-tone1

Mid Zone - <125 та ≥ 100 LED-Y1, LED-Y2, LED-Y3, Beep-tone1

Mid Zone - <100 та ≥ 50 LED-Y1, LED-Y2, LED-Y3, LED-Y4, Beep-tone1

Danger Zone <50 та ≥ 25 LED-R1, Beep-tone2

Danger Zone <25 LED-R1, LED-R2, Beep-tone2

