**’Parkkitutka’ dokumentaatio**

**Tässä etäisyysmittarin liittämisen ohjeet ja koodi**

https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

\* Ultrasonic Sensor HC-SR04 and Arduino Tutorial

\* by Dejan Nedelkovski,

\* www.HowToMechatronics.com

#include <Button.h> **Otetaan käyttöön kirjasto painikkeelle**  
#include <LiquidCrystal.h> **Otetaan käyttöön LCD näytön kirjasto**

#define ledV 9

#define ledK 10

#define ledP 11

#define kaiutin 8

#define lednappi 14

#define kaiutinnappi 15 **Definet määrittävät tarvittavat pinnit ja luovat niille vakiot**

#define NOTONE 0

#define TONE 1

#define trigPin 13

#define echoPin 12

Button S1(0, 200); **Määritetään nappi**

char noise=TONE;

LiquidCrystal lcd(2, 3, 4, 5, 6, 7); **Määrittaa LCD näytön**

long duration;

int distanceCm, distanceInch;

int viive = 60; **Määrittää käytettävät viiveet**

void setup() { **Järjestelmän alustus asetukset**

lcd.begin(16,2);  **Asettaa näytön koon (16x2)**

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode (ledV, OUTPUT);

pinMode (ledK, OUTPUT); **Pinmodet määrittää onko käytetyt pinnit sisään vai ulostuloja**

pinMode (ledP, OUTPUT);

pinMode (kaiutin, OUTPUT);

pinMode (lednappi, INPUT);

pinMode (kaiutinnappi, INPUT);

Serial.begin(9600); **Alustaa sarjaportin**

}

void loop() { **Tässä on itse laitteen ohjelma, joka toistuu ikuisesti laitteen käynnissä ollessa**

digitalWrite(trigPin, HIGH); **Etäisyysmittari lähettää ja vastaanottaa näillä ääntä, ja matkaan kuluneesta ajasta se laskee etäisyyden**

delayMicroseconds(10); **Mittauksen kesto**

digitalWrite(trigPin, LOW); **Lopettaa lähetyksen**

duration = pulseIn(echoPin, HIGH); **Tallentaa mitatun matkan ’duration’ muuttujaksi**

distanceCm= duration\*0.034/2;  **Muutetaan mittarin mittaama matka millimetreiksi**

distanceInch = duration\*0.0133/2; **Muutetaan mittarin mittaama matka tuumiksi**

delay(viive); **Viive että ihminen pysyy mittauksissa mukana**

lcd.clear();  **Tämä tyhjentää näytön kerran loopissa, lisäsin tämän koska cm ja inch jäi näkymään taululle eri kohtiin.**

lcd.setCursor(0,0); **Määrittää mihin kohtaan kirjasin tulee**

lcd.print("Sentit: "); **Kirjoittaa näytölle ‘Sentit’ sanan**

lcd.print(distanceCm); **Kirjoittaa näytölle mikä arvo muuttujassa ’distanceCm’ on**

lcd.print(" cm"); **Kirjoittaa näytölle ‘cm’ sanan**

lcd.setCursor(0,1); **Liikuttaa kirjasimen toiselle riville**

lcd.print("Matka: "); **Kirjoittaa näytölle ‘Tuumat’ sanan**

lcd.print(distanceInch); **Kirjoittaa näytölle mikä arvo muuttujassa ’distanceInch’ on**

lcd.print(" inch"); **Kirjoittaa näytölle ‘inch’ sanan**

delay(viive); **Viive, että ihmissilmä pysyy nopeasti muuttuvan tekstin perässä**

if(S1.pressed()){ **Tämä if testi mittaa onko nappia painettu**

if(noise==TONE){ **Jos nappi on painettu (muuttuja on on TONE)**

noise=NOTONE; **muuttuja vaihdetaan NOTONE**

Serial.println("NOTONE"); **Ja kirjoitetaan sarjaporttiin NOTONE**

noTone(kaiutin); **Kautin hiljennetään**

}else{ **Jos ei niin=**

noise=TONE; **Sama kun ylhäällä, mutta toisinpäin**

Serial.println("TONE");  **Ja kirjoitetaan sarjaporttiin TONE**

}

}

if (distanceCm < 20){ **Testataan onko mittarin mittaama matka senttimetreinä alle 20**

digitalWrite(ledV, HIGH); **Jos on, niin vihreä ledi syttyy**

digitalWrite(ledK, LOW); **Jos on, niin keltainen ledi ei syty**

digitalWrite(ledP, LOW); **Jos on, niin punainen ledi ei syty**

if(noise==TONE){ **Ja toistetaan ääni, jos muuttujan arvo on TONE**

tone(kaiutin, 1600); **Määritetty pinni ja hertzit**

}

}

else if (distanceCm < 50){ **Testataan onko mittarin mittaama matka senttimetreinä alle 50**

digitalWrite(ledV, LOW); **Jos on, niin vihreä ledi ei syty**

digitalWrite(ledK, HIGH); **Jos on, niin keltainen ledi syttyy**

digitalWrite(ledP, LOW); **Jos on, niin punainen ledi ei syty**

if(noise==TONE){ **Ja toistetaan ääni, jos muuttujan arvo on TONE**

tone(kaiutin, 900); **Määritetty pinni ja hertzit**

}

}

else if(distanceCm < 100){ **Testataan onko mittarin mittaama matka senttimetreinä alle 100**

digitalWrite(ledV, LOW); **Jos on, niin vihreä ledi ei syty**

digitalWrite(ledK, LOW); **Jos on, niin keltainen ledi ei syty**

digitalWrite(ledP, HIGH); **Jos on, niin vihreä ledi syttyy**

if(noise==TONE){ **Ja toistetaan ääni, jos muuttujan arvo on TONE**

tone(kaiutin, 300); **Määritetty pinni ja hertzit**

}

}

else{ **Jos mikään ehdoista ei täyty, toistetaan seuraava, eli sammutetaan kaikki**

digitalWrite(ledV, LOW); **Jos on, niin vihreä ledi ei syty**

digitalWrite(ledK, LOW); **Jos on, niin keltainen ledi ei syty**

digitalWrite(ledP, LOW); **Jos on, niin punainen ledi ei syty**

noTone(kaiutin); **Ei ääntä**

}

delay(viive); **Viive, ettei vaan mentäisi liian nopeasti 😊**

}

BY: *JJK*