

# Ultraäänietäisyysmittari

Sulautetut prosessorijärjestelmät 2013

4.2.2014

Joni Rämö, Teemu Tähtinen ja Minna Isomäki

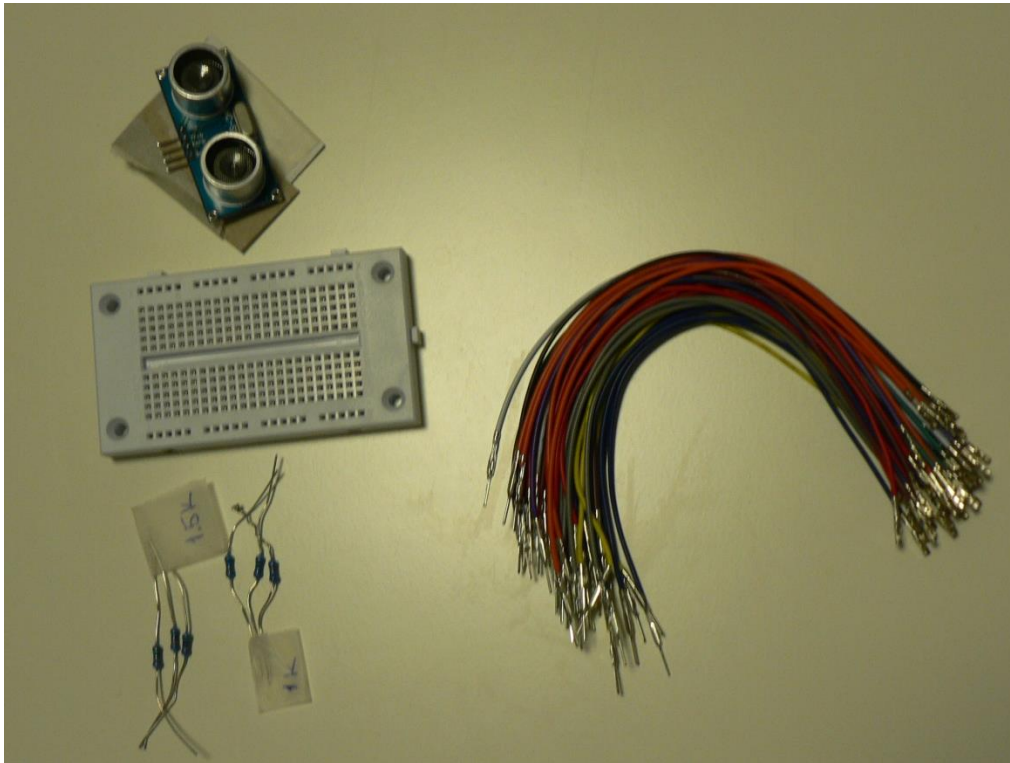
## Tehtävän kuvaus ja analysointi

Projektinamme oli tehdä ultraäänellä toimiva etäisyysmittari, joka käyttää alustanaan Raspberry Pi -pienoistietokonetta. Käyttöjärjestelmänä on Raspbian, koska kaikki osat toimivat siinä. Ohjelmointikielenä toimii RasPin natiivikieli Python ja mittaamiseen käytetään ultraäänisensoria.

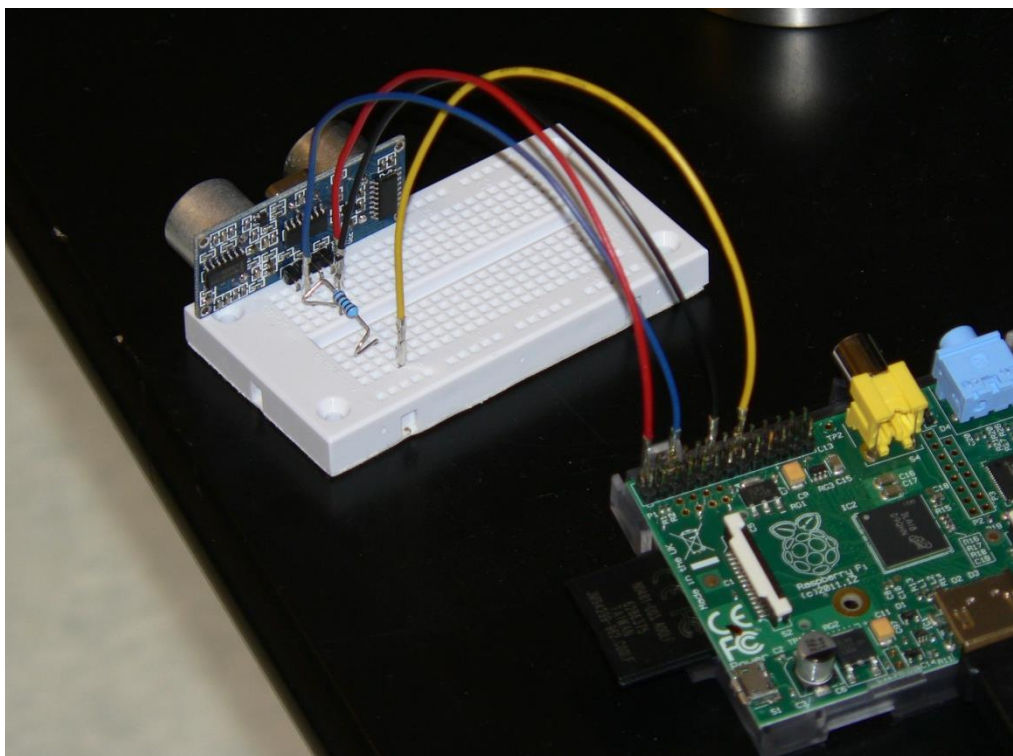
Komentoriviltä haetaan kansio, jossa mittausohjelma sijaitsee ja suoritetaan se. Ohjelman käyttöliittymä kysyy, että halutaanko mitata ja halutaanko mitata metreinä vai jalkoina. Tämän jälkeen ohjelma antaa sensorille käskyn, että se mittaa käytännössä ja sitten se käsittelee saadun tiedon ja muuttaa sen metreiksi tai jaloiksi ja tulostaa sen. Ensimmäisen mittaustuloksen jälkeen ohjelma kysyy: halutaanko mitata uudelleen. Ohjelma kysyy jokaisen mittauksen jälkeen halutaanko mitata uudelleen, kunnes sille ilmoitetaan ettei haluta mitata uudelleen, jolloin ohjelman suoritus päättyy.

Laitteistossa ei vielä ole omaa näyttöä, mutta se on tarkoitus lisätä myöhemmin (kun saamme osan, tästä hetkestä +2 viikkoa). Tarkoituksena olisi lisätä nappulat ja näyttö, jotta siitä saataisiin tietokoneriippumaton kokonaisuus. Alkuperäisestä suunnitelmasta poiketen laitteistossa ei ole erillistä paristokoteloä, vaan sen pitää olla verkkovirrassa kiinni.

## Ratkaisuperiaate



Kuva 1. Pääkomponentit: ultraäänisensori, koekytkentälevy, vastukset ja hyppylangat.



Kuva 2. Lopullinen kytkentä

Projektityön toteutus alkoi lopullisen kytkennän (kuvassa 2) kasaamisesta, kuvan 1 komponentteja käyttäen. Valmis laitteisto kytkettiin televisioon (HDMI), verkkovirtaan sekä Internet-kaapeliin. Tämän jälkeen laitteistoon ajettiin mittausta varten koodattu ohjelma ja testaus voitiin aloittaa.

## Ohjelman ja sen osien kuvaaminen

```
self.remeasure()
File "UI.py", line 36, in remeasure
    dialog2 = raw_input("Would you like to measure aga
KeyboardInterrupt
pi@raspberrypi ~/Documents/Filut $ sudo python UI.py
Would you like to measure distance? (Y/N) y
Do you want to measure in meters or feet? (M/F) m
Prepare the device. Are you ready? (Y/N) y
Measuring...
Distance measured: 0.59 m
Would you like to measure again? (Y/N) y
Measuring...
Distance measured: 0.59 m
Would you like to measure again? (Y/N) y
Measuring...
Distance measured: 0.59 m
Would you like to measure again? (Y/N) n
Shutting down...
pi@raspberrypi ~/Documents/Filut $ sudo python UI.py
Would you like to measure distance? (Y/N) y
Do you want to measure in meters or feet? (M/F) f
Prepare the device. Are you ready? (Y/N) y
Measuring...
Distance measured: 1 ft 11 in
Would you like to measure again? (Y/N) y
Measuring...
Distance measured: 1 ft 11 in
Would you like to measure again? (Y/N) y
Measuring...
Distance measured: 1 ft 11 in
Would you like to measure again? (Y/N) y
Measuring...
Distance measured: 1 ft 11 in
Would you like to measure again? (Y/N) _
```

Kuva 3. Mittauksen esimerkkitulostus metreinä ja jalkoina

Kuvasta 3 nähdään, että ohjelma kysyy: haluatko mitata etäisyyden (Y/N), Tämän jälkeen ohjelma kysyy mitataanko jalkoina ja metreinä. Ennen ensimmäistä mittausta ohjelma kysyy vielä varmistuskysymyksen: oletko aivan

varma. Ennen kuin kysymykseen vastataan on sensorin paikkaa/suuntaa vielä mahdollista muuttaa.

Koodinpätkä on liitteenä, mutta seuraavassa on koodin toimintaselostus.

Aluksi ohjelma suorittaa `main` metodin, jossa luodaan `ui`-olio ja kutsutaan `userinterface()`-metodia. `Userinterface`-metodissa luodaan `input` luokan olion `'putin'` ja tämän jälkeen alkaa ensimmäinen `if`-silmukka ja metodin kysyy: halutaanko mitata? Vastattaessa kyllä se kysyy missä mittayksikössä tulos halutaan, vaihtoehtoina metrit ja jalat. Mittayksikön valinnan jälkeen ohjelma varmistaa käyttäjän olevan valmis mittaukseen ja tämän aikana sensorin asentoa voidaan vielä muuttaa. Vastaus "y" varmistuskysymykseen kutsuu `input`-luokan `putin`-olion `input()`-metodia. `Userinterface`-metodissa olevat `while True` silmukat estävät muiden kuin kysyttyjen argumenttien antamisen. Väärän argumentin saadessaan ohjelma ilmoittaa tästä tulostamalla `"invalid argument"` ja kysyy viime kysymyksensä uudelleen.

`Input`-metodissa luodaan ensin `Measurement`-luokan olio tulevaa kutsumista varten. Raspberry Pi:n GPIO-pinnien käyttö on tuotu ohjelman alussa komennolla `'import RPi.GPIO as GPIO'` ja tässä asetetaan seuraavaksi käskyllä `'setwarnings'` GPIO:n pinnien käytöstä aiheutuvat virheilmoitukset pois päältä. Tämän jälkeen asetetaan ohjelma käyttämään Raspberry Pi:n GPIO-pinnien BCM (Broadcom) PIN-numeroita, jotta niitä voidaan myöhemmin kutsua oikein. `Setup`-komennoilla määritetään, mitä pinnejä käytetään laudalta sisään- ja ulostuloon. Seuraavalla käskyllä asetetaan GPIO-pinni 18 lähettämään pulssia, mutta käskyllä `low` vielä epäaktiiviseksi. Komennolla `'GPIO.output(18, True)'` määrätään sensori lähettämään pulssia. Sen jälkeen tulevalla kymmenen mikrosekunnin tauolla saadaan sensorin ohjeiden mukainen ultraäänipulssi, josta voidaan laskea kaikuun kuluneen ajan perusteella etäisyys kertomalla aika puolella äänennopeudella. Vain puoli äänennopeutta, koska mitattu aika on pulssin edestakaiseen matkaan kulunut aika. `Cleanup`-käsky tyhjentää pinnien asetukset, koska niitä ei enää käytetä. Tämän jälkeen kutsutaan `Measurement`-luokan `measure()`-metodia saadun arvon käsittelyä varten.

`Measurement()`-metodissa luodaan `UI()`-luokan olio `'face'`. Mikäli alussa valittu mittayksikkö oli metrit, ohjelma tulostaa etäisyyden metreissä kahden desimaalin tarkkuudella. Mikäli mittayksikkö oli jalat, metodi laskee ensin etäisyyden jaloissa kokonaislukuna ja sitten ylijääneen tuloksen tuumissa ja tulostaa etäisyyden muodossa `'jalat tuumat'`. Etäisyyden tulostamisen jälkeen

metodi kysyy mitataanko uudestaan kutsumalla `remeasure()`-metodia. Remeasure käyttää `input`-luokkaa uudelleen mittaamiseen.

Ohjelman suorituksen aikana ohjelman suorittamisen lopettaminen on mahdollista aina kun kysytään 'kyllä' tai 'ei'. Kun vastataan riittävän monta (1 tai 2) kertaa 'ei' niin ohjelman suorittaminen lopetetaan.

## Testausjärjestelmä

Aluksi tarkastelimme komponentteja silmämääräisesti ja totesimme ne laadukkaiksi. Tämän jälkeen aloitimme ohjelman kehittämisen ja testauksen/kääntämisen tietokoneella. Laitteiston valmistuttua ajoimme ohjelman siihen ja aloitimme ohjelmavirheiden poistamisen ja ohjelman optimoinnin. Riittävällä testauksella saimme halutun lopputuloksen. Tarkistimme sensorin mittaustuloksen tarkkuuden manuaalisella rullamitalla ja havaitsimme tarkkuuden olevan parhaimmillaan 30cm – 5m välillä.