

Opinnäytetyö (AMK)

Elektroniikan koulutusohjelma

Elektroniikkasuunnittelu

2016

Juuso Metsävuori

SARJAPORTIN OHJAAMINEN GRAAFISELLA KÄYTTÖLIITTYMÄLLÄ

OPINNÄYTETYÖ (AMK / YAMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikan koulutusohjelma | Elektroniikkasuunnittelu

2016 | 14

TkT Timo Tolmunen

Juuso Metsävuori

SARJAPORTIN OHJAAMINEN GRAAFISELLA KÄYTTÖLIITTYMÄLLÄ

Mittalaitteen ja käyttäjän välille tarvitaan jokin rajapinta. Tavallisesti tämä tapahtuu joko laitteella itsessään olevilla fyysisillä kontrolleilla, tai erillisellä tietokoneella sarjaportin ylitse. Sarjaporttia käyttäessä täytyy sopia käskykanta, jonka avulla tietokone ja mittalaite voivat kommunikoida keskenään. Tämän pohjalta voidaan kehittää ohjelma, joka lähettää tarvittavat käskyt sarjaportin ylitse vaatimatta loppukäyttäjältä erillistä osaamista kommunikaatioprotokollasta.

Työssä kehitettiin Python -ohjelmointikielellä graafinen käyttöliitymä sarjaportin yli kommunikoimiselle. Ohjelman kehitys ja testaaminen toteutettiin kosketusnäytöllä varustetulla Raspberry Pi -tietokoneella. Käyttöliitymän ulkoasu saatiin jäljittelemällä etukäteen toimitettua luonnosta, josta tulkittiin halutut ominaisuudet ja toimintatavat. Lopputuloksena saatiin kehitettyä käyttöliitymä konseptilaitteeseen, jonka avulla voitiin demnstroida tuotekonseptia.

ASIASANAT:

Raspberry Pi, Python, Sarjakommunikaatio, Käyttöliitymä

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics degree programme | Electronics Design

2016 | 14

Timo Tolmunen, D.Sc(Tech)

Juuso Metsävuori

CONTROLLING A SERIAL PORT WITH A GRAPHICAL USER INTERFACE

Measuring equipment needs some type of user interface. Usually this is done with physical controls on the device. Another option is to control the equipment with a computer over a serial connection. This requires there to be a known set of commands in order to allow the computer and device to communicate. With these commands one can create a piece of software which translates what the user is desiring to do in to commands which the device understands.

The goal of this thesis was to create a graphical user interface for driving a proof of concept device over serial port. The software was developed and tested on a Raspberry Pi embedded computer. The desired layout of the software was given in advance which from the desired functions and appearance were taken. The resulting software mimicked the agreed layout and was used to demonstrate the functions of a possible future product.

KEYWORDS:

Raspberry Pi, Python, Serial communication, User interface

SISÄLTÖ

1 JOHDANTO	6
2 LAITE	7
2.1 Raspberry Pi	7
2.2 Sarjaportti	7
2.3 Kehitysympäristö	8
3 OHJELMOINTI	9
3.1 Sarjakommunikaatio	9
3.2 Painikkeet	10
3.3 Kuvaaja	10
4 VALMIS OHJELMA	12
5 LOPUKSI	13
LÄHTEET	14

KUVAT

Kuva 1. Lohkokaavio laitteen kommunikaatorajapinnoista	7
Kuva 2. Sarjaportin Käyttäjäoikeuksien selvittäminen	8
Kuva 3. Luonnos halutusta käyttöliittymästä	9
Kuva 4. Kuvankaappaus lopullisesta ohjelmasta	12

KÄYTETYT LYHENTEET TAI SANASTO

UART	Universal Asynchronous Receiver/Transmitter, de-facto sarjakommunikaatiometodi
ASCII	American Standard Code for Information Interchange, minimaalinen merkkistandardi, sisältää numerot 0-9, englanninkieliset aakkoset, sekä kokoelman erikoismerkkejä
BaudRate	Bittä sekunnissa, sisältää aloitus-, lopetus-, pariteetti-, sekä databitit
SSH	Secure Shell, ohjelma, jonka jakaa terminaalin ethernetin ylitse
RS232	Recommended Standard 232. Laitteiden väliseen sarjakommunikaation fyysisen kytkennän standardi.
MCU	MicroController Unit, sulautetun järjestelmän ohjausyksikkö
GPIO	General Purpose Input/Output, sulautetun kontrollerin ohjelmoitava sisään-/ulostulo portti

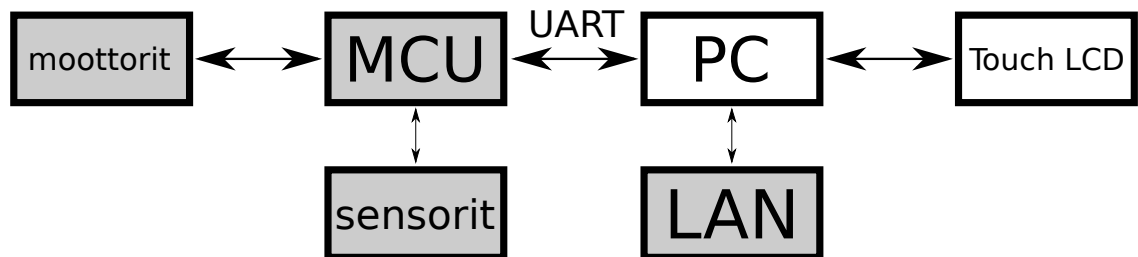
1 JOHDANTO

Labrox oy:n valmistamien mittalaitteiden ohjaamiseen tarvitaan yleensä erillinen tietokone käynnistämään mittauksia ja tulkitsemaan niiden tuloksia. Tutotetut mittaustulokset ovat kuitenkin yksinkertaista taulukkoataa, josta mittauksen onnistuminen on helppo tulkita piirtämällä kuvaaja. Tämän vuoksi halutaan tutkia mahdollisuutta datan käsittelyyn itse laitteessa ilman ulkoista tietokonetta.

Työn tavoitteena on luoda kehitysympäristö laitteen sisälle asennettavalle tietokoneelle, jonka avulla voidaan kokeilla yksinkertaisten käyttöliittymien kehittämistä. Käyttöliittymän, sekä muiden mahdollisten rajapintojen kehittämiseen tarvittavien työkalujen täytyy olla avoimesti tarjolla ilman lisäkustannuksia. Valittua ympäristöä sovelletaan konseptituotteen kehittämisessä, jonka avulla voidaan kerätä palautetta potentiaalisilta asiakkailta.

2 LAITE

Mittalaitteen sisälle valittiin asentaa Raspberry Pi -tietokone. Sen HDMI -porttiin on kyteetty 5” kosketusnäyttö, jonka kosketuspuoli toimii USB:n ylitse ja tunnistuu hiirenä sen käyttöjärjestelmässä. Tietokoneelta voidaan myös tarvittaessa jakaa kerätty data lähiverkkoon jatkokäsittelyä varten. Tietokoneen sarjaporttiin on kytkettynä mikrokontrolleri joka ohjaa laitteen mittapäätä, sekä laitteen edessä olevaa luukkua, joka avautuu vastaanottamaan mitattavan näytteen (Kuva 1).



Kuva 1. Lohkokaavio laitteen kommunikaatorajapinnoista

2.1 Raspberry Pi

Raspberry Pi valittiin laitteen sisäiseksi tietokoneeksi sen hyvän saatavuuden ja pienen kokonsa vuoksi. Tietokone aseteltiin laitteeseen siten, että sen Ethernet, sekä USB liitännät saatiin kytkettyä laitteen ulkopuolelta. Kosketusnäytön tarvitseman HDMI:n ja USB kommunikaation piilottamiseksi jouduttiin tietokoneesta poistamaan yksi sen kahdesta USB -liitimestä ja juottamaan sen tilalle erillinen USB piuha, joka jätettiin laitteen sisälle. Tietokoneen tarvitsema 5V käyttöjännite, sekä laitteen ohjaamiseen tarvittava sarjaportti saatiin kytkettyä suoraan tietokoneen piirilevyllä löytyvään GPIO -liittimeen. Käyttöjärjestelmäksi Raspberry Pi -tietokoneelle valittiin sille räätälöity Raspbian Linux.

2.2 Sarjaportti

Laitteen kanssa kommunikoidaan ASCII -enkoodatulla tekstillä sarjaportin ylitse. Raspbianissa sarjaportti on normaalisti varattuna terminaaliksi, joten sen käyttö tavallisena sarjaporttina vaatii sarjaporttiterminaalin poistamisen käynnistyskäskyistä.

Raspberry pi -tietokoneen sisäinen sarjaportti löytyy käyttöjärjestelmästä sijainiista /dev/ttyAMA0 ja sen käyttäjäryhmä on "tty"(Kuva 2). Tämän vuoksi ohjelmaa suorittava käyttäjä lisättiin ryhmään "tty" ennen kuin sarjaporttia voitiin käyttää.

```
netl@raspberrypi:~ $ ls -l /dev/ttyAMA0  
crw--w---- 1 root tty 204, 64 Apr  6 14:29 /dev/ttyAMA0
```

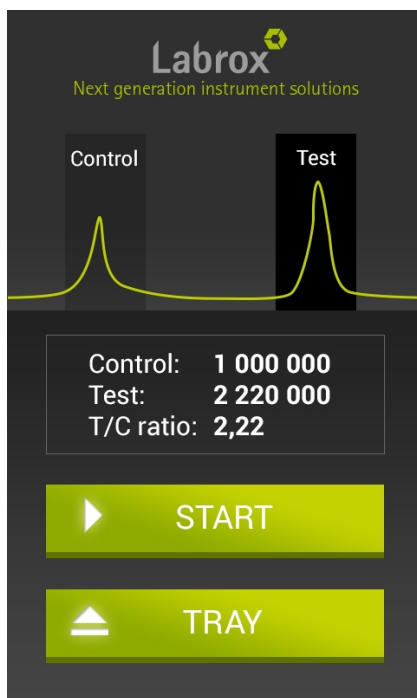
Kuva 2. Sarjaportin Käyttäjäoikeuksien selvittäminen

2.3 Kehitysympäristö

Ohjelman kehittäminen toteutettiin laitteen sisäisellä tietokoneella ssh yhteyden ylitse. Tämä mahdollisti laitteen kosketusnäytön, sekä sarjakommunikaation hyödyntämistä ohjelman toiminnan testaamiesssa. Ohjelmointikieleksi valittiin Raspbianista valmiiksi löytyvä Python niminen reaaliaikainen olio-ohjelmointikieli [1]. Käyttöjärjestelmästä löytyvän Pythonin ja ssh:n lisäksi jouduttiin erkiseen asentamaan Git -versiohallintaohjelma [2], joka myös mahdollistaa projektin haarauttamisen eri konfiguraatioihin, mikäli tulevaisuudessa laitteesta päädyttäisiin tekemään variaatioita eri ominaisuuksilla. Tämän lisäksi työssä tehtävää käyttöliittymän ohjelmaversioita voidaan tarvittaessa helposti päivittää ulkoiselta palvelimelta.

3 OHJELMOINTI

Laitteeseen halutusta käyttöliittymästä saatiin luonnos, josta näkyy halutut ominaisuudet, sekä niiden asettelun näytöllä (Kuva 3). Ohjelman päätarkoituksena on lähettää käyttäjän haluamia käskyjä sarjaportin ylitse hyödyntäen selkeää ja helppokäyttöistä käyttöliittymää. Käyttöliittymän visuaalinen puoli valittiin toteuttaa Python -ohjelmointikielen Tkinter -kirjastolla [3].



Kuva 3. Luonnos halutusta käyttöliittymästä

3.1 Sarjakommunikaatio

Laitteen sarjakommunikaatio muodostuu yleensä kahdesta osasta: yhdestä tai useammasta arvosta, joka on yleensä numero ja käskystä, joka kertoo mihinkä arvo viittaa. Kaikki käskyt myös päättyvät rivinvaihtoon. Tämä pitää paikkansa molempiin suuntiin. Esimerkiksi käsky "100. 10. 50.meas" käskee laitteen aloittaa 100:n mittauksen suorituksen odottaen 10 millisekuntia ja siirtäen mittapäätä 50 askelta joka mittauksen välissä. Mittaustulos tulee viestinä "n. result", jossa n vastaa mittaustulosta desimaalilukuna. Mittauksen päättymistä ilmaisee käsky "donemeas".[4]

Vastaanotettua viestiä tulkitessa Python -kielen tekstiluokan `split()` -käsky [5] muodostaa taulukon, jossa numerot ja teksti erotellaan välimerkkien kohdalta erilliseen taulukkoon. Tämän lisäksi eroteltu numeroarvo täytyy vielä muokata tekstistä muuttujaksi joka tapahtuu `int()` -funktioilla. Esimerkiksi: käsky `int(data.split()[0][:-1])`, jossa `data` muuttuja sisältää merkkijonon `"100.result"` palauttaa numeron 100. PySerial -kirjasto sisältää tarvittavat funktiot sarjaportin avaamiseen, sekä sillä kommunikoidaan [6].

3.2 Painikkeet

Tavallisesti graafisen käyttöliittymän halutaan täsmäävän väriteemaltaan käyttöjärjestelmän työpöytäympäristöön. Tämän työn tapauksessa kuitenkin ohjelma pyörii yksinään näytöllä, eikä sen haluta näyttävän käyttöjärjestelmän tarjoamilla painikkeilta. Tämä tarkoittaa, että myös painikkeiden ei haluta näyttävän raspbian -käyttöjärjestelmältä, vaan samalta kuin luonnoksessa. Käyttöliittymän painikkeet valittiin korvata kuvilla, joita voi helposti muokata tulevaisuudessakin. Tkinterillä voi piirtää kuvan nappi -elementtiin [7], mutta painikkeelle piirtyy kehykset, joita ei pysty poistamaan. Tämän vuoksi painikkeesta tehtiin canvas -elementillä [8] kuva, johonka sidottiin funktiokutsuja, joidenka avulla vaihdetaan piirrettävää kuvaa ja kutsutaan lisäfunktioita tarpeen mukaan.

Näytteen vastaanottamiseen laitteen tarvitsee suorittaa luukun avaus- tai sulkukäsky riippuen onko laitteen luukku auki vai kiinni. Laitetta ei saa käskä toteuttamaan mittausta mikäli luukku on auki, eikä luukku saa avata, mikäli mittaus on käynnissä. Luukun tilaa varten tehtiin muuttuja, jota päivitetään avaus- ja sulkukäskyjen ohessa. Kyseisen muuttujan tilaa valvomalla voitiin estää mittauksen aloittaminen. Vastaavasti luukun avaaminen voitiin estää tarkastamalla onko odotettu määrä mittauksia suoritettuna.

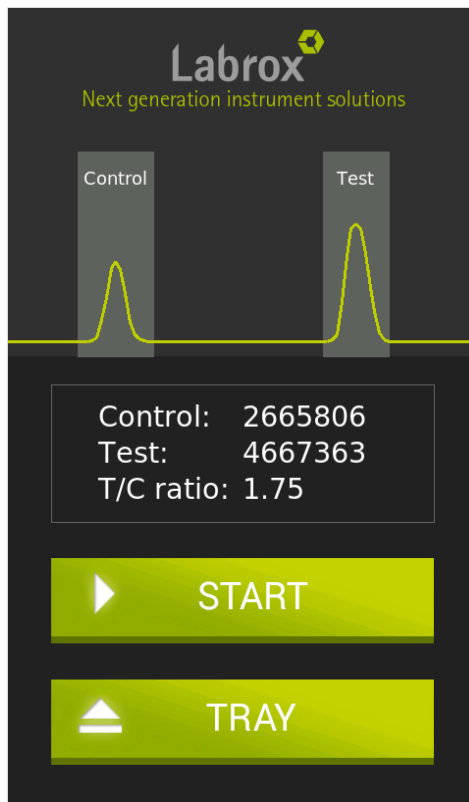
3.3 Kuvaaja

Koska Tkinterissä ei ole valmista funktiota kuvaajan piirtämiseen, jouduttiin sitä varten tekemään erillinen funktio joka muodostaa kuvaajan useammasta viivasta mittauspisteiden väleihin [8]. Ohjelman selkeyttämiseksi näytölle piirrettävästä kuvaajasta tehtiin oma taulukko, johonka ohjelma syöttää mittalaitteen antamia

vastauksia. Ohjelman ulkoasun vuoksi haluttiin, että vielä mittaamatonta osaa taulukosta ei haluta piirtää. Tämä toteutettiin muokkaamalla piirtofunktiota siten, että mittaustulosta jonka arvo on pienempi kuin nolla ei piirretä näytölle. Tämän myötä voidaan mittausta aloittaessa piilottaa kuvaaja asettamalla kaikki sen taulukon arvot -1:een. Mittaustulosten määrä oli myös käyttäjän valitsema parametri, jotenka mittapisteiden välitys leveyssuunnassa saatiin jakamalla mittaustulosten määrä näyttöalueen leveydellä.

4 VALMIS OHJELMA

Konseptilaitteeseen saatiin toteutettua selkeä graafinen käyttöliittymä, joka jäljittelee työn alussa saatua konseptikuvaa (Kuva 4). Käytetyillä kirjastoilla päästiin varsin lähelle haluttua tulosta, mutta grafiikkakirjaston vaatima .gif -formaatti ja käytetyn LCD näytön katselukulma vaati värien pelkistämistä, sekä kontrastin lisäämistä. Näytöllä olevia painikkeita koskettamalla saadaan laitteen luukku avattua, suljettua, sekä tehtyä mittauksen, joka päivittyy reaaliaikaisesti laitteen näytölle. Laitteen tietokoneen käyttöjärjestelmän käynnistyminen myös konfiguroitiin siten, että näytölle ei piirretä muuta kuin käyttöliittymä käyttöjärjestelmän käynnistyttyä.



Kuva 4. Kuvankaappaus lopullisesta ohjelmasta

5 LOPUKSI

Tässä työssä toteutettiin graafinen käyttöliittymä laitteelle, joka tulkitsee käyttäjän haluamat käskyt ja lähettää ne sarjaportin kautta laitteelle. Käyttöliittymään ulkoasu ja ominaisuudet toteutettiin jäljittelemällä annettua luonnosta. Kirjoitettua ohjelmaa on tulevaisuudessa mahdollista muokata soveltumaan muihinkin vastaavantyyliisiin käyttötarkoituksiin.

Jatkokehityksenä käytetylle grafiikkakirjastolle voisi tutkia vaihtoehtista kirjastoja, jotka mahdollisesti tukisivat useampaa kuvaformaattia, sekä mahdollisesti muita medioita kuten videota.

LÄHTEET

- [1] [www-dokumentti]. Saatavilla: <https://www.python.org/>
- [2] [www-dokumentti]. Saatavilla: <https://git-scm.com/>
- [3] [www-dokumentti]. Saatavilla: <https://docs.python.org/2/library/tkinter.html>
- [4] Labrox Oy, henkilökohtainen keskustelu, 20.7.2015
- [5] [www-dokumentti]. Saatavilla: <https://docs.python.org/2/library/re.html>
- [6] [www-dokumentti]. Saatavilla: <https://pyserial.readthedocs.org/>
- [7] [www-dokumentti]. Saatavilla: <http://effbot.org/tkinterbook/button.htm>
- [8] [www-dokumentti]. Saatavilla: <http://effbot.org/tkinterbook/canvas.htm>