

SUMMARY

With the current thesis, a new version of the software used in the lab works of the web based ISC learning environment has been created. It offers the possibility to be utilized along different platforms, thus being a very convenient option for the students. This new version provides an easier maintenance than the previous one written in VB, since it is coded in Python, a more modern programming language with larger support. Python itself already handles most of the changes along different OS, simplifying the maintenance process for all the different OS. It has proved that with very few changes the software can be easily implemented.

The software has been developed integrating different Python libraries for the different functions of the program: the pyserial library for the serial communication; matplotlib for the continuous plotting of the oscilloscope; the Tkinter library for the overall GUI; the bitstring library for some incoming data parsing and casting; and some other standard Python libraries. The software together with the signal generator and oscilloscope device cover the main functions such as basic signal waveforms from the signal generator part, the calculation of voltage in different ways (RMS, dBm...) and the phase shift between 2 measured signals; that are needed in the study of AC.

The GUI has been tested using different circuits attached to the signal generator part of the hardware and measured with the oscilloscope part. The software handles errors that might arise when there is a serial communication problem between the device and the computer such as being unplugged. It has also been tested in various OS (Windows, different implementations of Linux, and iOS) with positive results.

During this thesis the implementation in the labs has been left out of the scope. That would be the final step in the development of the software. After that, there are some improvements that could be implemented as well as the maintenance of the software. A big improvement that could be made is doing multi-threading for separate tasks. For instance, that would be to have the serial communication in a thread parallel to the main loop of Tkinter; together with that, the plotting could be handled also in a separate thread.

Further advancement would be related to the interaction between the hardware and the software. For this project the hardware program has not been changed but there are certain things that could be optimized, like the data packets size and the delay in between them. Also, the usage of the asyncio library in the Python framework should be studied as a possible improvement for the serial communication using the different OS.

KOKKUVÕTE

Käesolevas lõputöös on loodud veebipõhise ISC õpikeskkonna laboritöodes kasutatava tarkvara uus versioon. See võimaldab võimaluse kasutada seda erinevatel platvormidel, olles seega õpilastele väga mugav võimalus. Arendatud uut versiooni on lihtsam hooldada kui eelmist, mis oli kirjutatud VB-s, kuna see on programmeeritud Pythonis, moodsamas programmeerimiskeeles, millel on suurem tugi. Python hoolitseb ise enamiku muudatuste eest, mis tulenevad erinevatest OS-idest, lihtsustades kõigi erinevate OS-ide hooldusprotsessi. On tõestatud, et väga väheste muudatustega saab tarkvara hõlpsasti rakendada.

Tarkvara on välja töötatud nii, et see integreerib erinevad Pythoni teegid programmi erinevateks funktsioonideks: jadakommunikatsiooni teek; matplotlib ostsiloskoobi pidevaks joonistamiseks; Tkinteri teek üldise GUI jaoks; bitstringi teek sissetulevate andmete parsimiseks ja ülekandmiseks; ja mõned muud Pythoni standardteegid. Tarkvara koos signaaligeneraatori ja ostsiloskoobi seadmega katavad peamised funktsioonid, näiteks põhilised signaali lainekujud signaali generaatori osast, pinge arvutamine erinevatel viisidel (RMS, dBm...) ja faasi nihkumine kahe mõõdetud signaali vahel, mida on vaja vahelduvvoolu uurimisel.

GUI-d on testitud riistvara signaaligeneraatori osa külge kinnitatud erinevate vooluahelate abil ja seda on mõõdetud ostsiloskoobiga. Tarkvara tegeleb tõrgetega, mis võivad tekkida seadme ja arvuti vahelise jadaühendusega seotud probleemide korral, näiteks ühenduse lahtiühendamine. Seda on testitud ka erinevates OS-ides (Windows, Linux ja iOS-i erinevad rakendused), ning tulemused on olnud positiivsed.

Selle lõputöö käigus on laboratooriumides rakendamine välja jäetud. See oleks tarkvara arendamise viimane samm. Pärast seda oleks võimalik rakendada mõndasid täiustusi ja samuti tarkvara hooldada. Suur edasiminekuks oleks mitme töö lõimimine eraldi ülesannete jaoks. Näiteks tähendaks see, et seeriakommunikatsioon toimub keermes, mis on paralleelne Tkinteri peaahelaga; koos sellega saaks süvendit käsitleda ka eraldi keermes.

Edasine täiustamine oleks seotud riistvara ja tarkvara vastastikmõjuga. Selle projekti jaoks pole riistvaraprogrammi muudetud, kuid on ka teatud asju, mida saaks optimeerida, näiteks andmepakettide suurus ja nendevaheline viivitus. Samuti tuleks uurida asyncio teegi kasutamist Pythoni raamistikus, kui võimalikku täiustust erinevat operatsioonisüsteemi kasutatava seeriakommunikatsiooni jaoks.