

Kasutajalugu

Korterielanikuna tahan ma enda elektriseadmeid ise juhtida, et kuluks minimaalselt elektrit.

Olen avanud korterielanikule suunatud rakenduse ning sisestanud enda elektripaketi.

Vajutan "Kinnita"

Järgmiseks saan sisestada enda soovitud tingimused elektri kasutamiseks.

Saan valida:

- 1) mitu minutit peab elektriseade kindlalt ööpäevas olema sisse või välja lülitatud.
- 2) mitu minutit järjest peab elektriseade vähemalt korraga sisse lülitatud olema.
- 3) millisel ajavahemikul peab elektriseade kindlasti sisse või välja lülitatud olema.
- 4) milline on maksimaalne kulu, milleni jõudes elektriseade ennast välja peab lülitama.
- 5) millise minimaalse kulu korral peab elektriseade ennast sisse lülitama.

Teen enda soovitud valikud ja vajutan "Kinnita"

Peale tingimuste kehtestamist saan näha tarbija hetkeseisu ja saan ka nupuvajutusega :tarbijat sisse ja välja lülitada.

Saan näha elektri kulu praeguse seisuga, järgmise nädala prognoosi ja ka eelmiste kuude kulu.

Rakendus näitab veel visuaalselt järgmise ööpäeva ja järgmise nädala prognoositavat tarbija töögraafikut, millal ja kui kaua on tarbija olnud sisse või välja lülitatud, saan importida ja eksportida tarbija töögraafikute aluseks olevaid reegleid, tegelikku sisse ja välja lülitamise ajalugu.

Korteriomanikuna soovin ma, et elektri kulud oleksid minimaalsed ja et saaksin neid kontrollida ning jälgida.

Olen avanud kortermaja omanikule mõeldud kasutajaliidese ning sisestanud. Kasutajaliidese on võimalik juhitavaid tarbijaid lisada või muuta tarbija lülitusviisi ilma teiste tarbijate lülitamist segamata.

Sisestan elektripaketi, kui majapidamises on erinevad elektripaketid, saan ka sisestada mitu.

Saan vaadata kõikide majapidamiste tarbijate olekut ning neid sisse ja välja lülitada.

//**Küsimus:** kas kortermaja omanik saab korteri elanike seadmeid ise sisse ja välja lülitada ning mis saab siis kui korteri omanik lülitab mõne tarbija välja aga elanik soovib, et see sees oleks? Kumma sõna peaks peale jääma ja kas mõlemad saavad valida neid tingimusi kuidas tarbijad toimivad ja leitakse kuldne kesktee või siiski ainult elanik ise.

Litsents

Soovime, et praktika käigus valminud tarkvara oleks kõigile vabalt kasutatav ning enda vajadustele vastavalt saaks seda ka muuta ning edasi arendada, selleks otsustasime kasutada FOSS litsentsi (permissive foss licence).

Arendusvahendid

Kasutame tarkvara arendamiseks põhiliselt Raspberry Pi-d, mille programmeerime elektritarbijaid vastavalt soovile ja vajadusele sisse ning välja lülitama. Raspberry Pi-d saab lihtsalt USB kaudu arvutiga ühendada ja juhtida. Protsessori kiirus ulatub 700MHz 1.4GBni ja mälu ulatub 256MG 1GB RAMini. See saab suhelda teiste erinevate seadmetega, meie projektist lähtudes erinevate tarbijatena, näiteks jahutusseadmed, erinevad kütteseadmed, pörandasoojendus, boiler, külmkapp. Raspberry Pi-le saab kirjutada koodi näiteks Scratchi või Pythoniga ning Pythoniga oleme esimesel semestril kõik juba kokku puutunud. Raspberry Pi-l on olemas ka erinevad koodinäited, mille abil saab releeploki releesid sisse-välja lülitada. See peab toimima minuti täpsusega. Raspberry Pi lülitab ühte releed sisse või välja vastavalt sellele, millises seisus rele antud ajahetkel olema peab. Vajalik seisund antakse ette faili(de) abil.

Seda programmi on võimalik süsteemis automaatselt igal minutil käivitada (nt cron-i abil).

Iga tarbija kohta oleks üks selle programmi instants (erineva konfiguratsiooniga) Kuna paljud meeskonnaliikmed on kokku puutunud robotikaga ning programmeerinud Arduinoga, arvame, et võiksime ka Raspberry Pi-ga hästi hakkama saada. Oleme kirjutanud koodi erinevate andurite andmete lugemiseks ja siis nende näitude põhjal kas led-lampe või pisemaid mootoreid sisse ja välja lülitanud. Rääkisime ka enda plaanist Jaagup Kippariga, kellel oli klassis isegi üks Raspberry Pi kohe olemas ja saime sellega natuke tutvust teha. Kuna me otseselt enda tarkvara kuskil testida ja kasutada ei saaks, pakkus Jaagup isegi enda suvekodu välja.

Töökindlus

Ideaalis ei tohiks süsteemil tekkida mingisuguseid vigasid, kuid kui näiteks süsteem on ülekoormatud ja ei toimi korrektelt, peab kasutajal olema võimalus kogu süsteem ühe nupuvajutusega välja lülitada. Taaskäivitamisel töötab süsteem ilma vigadeta.

Kasutajat teavitatakse veel siis, kui ta sisestab ebakorrektseid andmeid. (näiteks, kui sisestatud kulu on negatiivne või seade peab olema päevas sisselülitatud 25 tundi)

Ülalpidamine ja tugi

Pi adapteeritavus ja ühilduvus muudavad selle ideaalseks platvormiks algaja programmeerijale, kes on veel kogenemata ning avastab Raspberry lõputuid võimalusi. Selle kasutamine on üle maailmne, sest see on väike, tasku-kohane ning lihtne ülal pidada.

Kuna Raspberry Pi on krediitkaardi suurune seade, siis selle füüsiline hooldamine probleeme ei tekita, tolmu puhastamine on väga lihtne ning vähe aeganõudev. Küll aga tuleb olla äärmiselt ettevaatlik. Seadme eluiga on sama nagu tavalisel arvutil, enamus kestavad üle 10 aasta eeldades, et hoolitsetakse ka seadme eest. Süle- või lauaarvutil läheb

halvaks esimese asjana tavaliselt *capacitor* (kondensaator), mida tavaliselt Pi peal ei ole palju. Samas aga Pi peal olev *sdkaart* võib visata esimese asjana erroreid, kui juhtub näiteks äkiline volukatkestus. Tegijad ise arvavad, et ideaaltingimustes võib ühe seadme eluiga ulatuda kuni 35 eluaastani, kuid 25-30 aastat, kui seade on *overclock*-tud.

Koodipoolse hooldamise, konfigureeritavuse ning Pi edasise arendamise teeme lihtsaks pideva koodi kommenteerimisega, mis on ideaalne ka seadme testimisel, mil võib ilmnedagi vigu. Nagu eelnevalt mainitud oleme me kõik tuttavad Pythoni keelega, mis peaks meie koostööd tegema sujuvamaks ning koodi kirjutamise ladusamaks. Kirjutame koodi inglise keelsete muutujatega, kuid kasutajaliidese teeme eesti keeles. Rahvusvahelise eesmärgiga võime kasutajaliidest ka tõlkida. Leiame, et inglise keeles olevale koodile on seda arendades lihtsam internetipõhist abi leida.

Esmased kasutajaliidese ideed

