

SAVONIA

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

PIIRIKAAVIOIDEN SUUNNITTELUN AUTOMATISOINTI

Ohjelmiston luominen ja kehitys

TEKIJÄ Jami Hartikainen

ESA18SP

21.04.2022

| | |
|--|----------------------------|
| Koulutusala Tekniikan ja liikenteen ala | |
| Tutkinto-ohjelma Sähkö- ja automaatiotekniikan tutkinto-ohjelma | |
| Työn tekijä Jami Hartikainen | |
| Työn nimi Piirikaavioiden suunnittelun automatisointi | |
| Päiväys 21.04.2022 | Sivumäärä/Liitteet 25/0 |
| Toimeksiantaja ja Yhteistyökumppani Suomen Automaatiopalvelu Oy | |
| <p>Tämän opinnäytetyön tarkoituksena oli luoda sähkösuunnittelussa käytettävä piirikaavioita suunnitteleva ja piirtävä ohjelmistoprototyyppi. Ohjelmisto kerää piirien lähtötiedot Excel-muodon IO-luettelosta mitoittaen, piirtäen ja lopuksi tallentaen kaaviot omilla positiotunnuksilla. Työn määrittelyyn kuului, että ohjelmointikieli on Python ja kehitettävä ohjelmisto ei saa ohjata mitään kolmannen osapuolen ohjelmistoa, vaan kyetä luomaan ja muokkaamaan kuvia itsenäisesti.</p> <p>Työ aloitettiin selvittämällä, että onko moottoripiirejä mahdollista piirtää ohjelmallisesti ja miten se voisi tapahtua. Itsenäisen ohjelmiston suunnitteluun löytyi Python kirjasto, joka oli tarkoitettu DXF tiedostomuotojen luomiseen ja muokkaamiseen. DXF tiedostomuoto sopi tilanteeseen hyvin, koska kuvat tulee lopuksi julkaista PDF- ja DWG-muodoissa, joihin kääntäminen DXF-muodosta on mahdollista.</p> <p>Työn tuloksena saavutettiin ohjelmistoprototyyppi, joka kykenee lukemaan tiedot IO-luettelosta, suorittamaan mitoituslaskennan, mutta tavoitteesta poiketen piirtämään ja täyttämään vain piirikaavion päävirtapiiriosuuden. Ohjausvirtapiirin luominen ja kuvien kääntäminen ja tallennus PDF ja DWG muotoihin jäi aikataulusyistä tekemättä. Ohjausvirtapiirin koneellista piirtoa on työn kehityksen aikana suunniteltu ja se vaikuttaa mahdolliselta, mutta paljon haasteellisemmalle kuin päävirtapiirin luominen. Ohjelmiston kehitystä voidaan jatkaa muillakin osa-alueilla, kuten automaattisten komponenttiluetteloiden luomisessa, komponenttien valinnassa ja moottorien käynnistysvirtojen sekä jännitteenalenumien analysoinnissa. Ohjelmaa tullaan kehittämään tulevaisuudessa lisäämällä siihen ominaisuuksia, mutta ensin työn alkuperäinen määrittely pitää saada tehtyä.</p> | |
| Avainsanat Piirikaaviot, Piirikaaviosuunnittelu, Ohjelmointi, Automaatio, Kehitys, Python | |

| | |
|---|--------------------------|
| Field of Study Technology, Communication and Transport | |
| Degree Programme Degree Programme in Electrical and Automation Engineering | |
| Author Jami Hartikainen | |
| Title of Thesis Automatization of Designing Circuit Diagrams | |
| Date 21 April 2022 | Pages/Appendices 25/0 |
| Client Organisation and Partner Automation Service of Finland Ltd. | |
| <p>The aim of the thesis was to create a program prototype with the ability to plan and draw circuit diagrams used in electrical engineering. The program gathers data about the circuits from an Excel type IO-list making necessary calculations, drawing, and saving the diagram files by their position names. The predetermined features included programming language to be Python and the program was also supposed to be a standalone system without the need of any third-party application or program.</p> <p>The project begun by studying if it was possible to create circuit diagrams automatically with software and how it could be done. For a standalone system a method was found in the form of a Python library, which was designed for creating and modifying DXF files. The DXF filetype fit the application well because it could easily be converted to PDF and DWG.</p> <p>As a result, a working prototype program was created which was able to read data from the IO list, run calculations, but deviate from the goal by drawing and filling only the main circuit part of the circuit diagram. Drawing the control circuit and converting files to DWG and PDF were left undone because of tight schedule. Creating the control circuit as a part of the circuit diagram was planned and seems possible but is clearly more challenging than the main circuit. Further development of the program can be done on different areas such as adding features like an automatically generated component list, help choosing the components and analyzing current spikes and voltage drops related to starting of a motor. The development of the program will continue by adding more features but first the initial objective of the project needs to be done.</p> | |
| Keywords Circuit diagram, Designing circuit diagrams, Programming, Automation, Development, Python | |

SISÄLTÖ

| | |
|--|----|
| 1. JOHDANTO | 6 |
| 2. PIIRIKAAVIO..... | 7 |
| 2.1. Mikä piirikaavio on?..... | 7 |
| 2.2. Mihin piirikaaviota käytetään? | 7 |
| 3. PIIRIKAAVIOIDEN SUUNNITTELU | 8 |
| 3.1. Suunnitteluprosessi | 8 |
| 3.2. Piirin tekemiseen tarvittavat tiedot | 9 |
| 3.3. Mitoitus | 9 |
| 3.4. Piirtäminen | 9 |
| 3.5. Tallennus ja arkistointi | 10 |
| 4. SUUNNITTELUN KEHITTÄMINEN | 11 |
| 4.1. Virheiden seuraukset ja niiden vähentäminen | 11 |
| 4.2. Automatisointi | 11 |
| 5. KEHITETTÄVÄ OHJELMISTO..... | 12 |
| 5.1. Määrittely ja tavoite | 12 |
| 5.2. Ohjelmistoympäristö ja ohjelmointikieli..... | 12 |
| 5.2.1. Visual Studio 2019 | 12 |
| 5.2.2. Python | 13 |
| 5.3. Ohjelmoinnissa käytetyt kirjastot ja tietokanta | 13 |
| 5.3.1. Python kirjasto, Pandas..... | 13 |
| 5.3.2. Python kirjasto, Ezdxf | 13 |
| 5.3.3. Tietokanta, Excel | 14 |
| 6. OHJELMISTON TOIMINNOT | 15 |
| 6.1. Luetaan tiedot Excelistä Python muuttujiin | 15 |
| 6.2. Mitoituslaskenta | 16 |
| 6.2.1. Taajuusmuuttajan valinta | 16 |
| 6.2.2. Suojalaitteen valinta | 17 |
| 6.2.3. Moottorikaapelin valinta | 18 |
| 6.3. Piirretään kuva..... | 19 |
| 6.3.1. Valmistelut..... | 19 |
| 6.3.2. Symbolien asettelu | 20 |

| | | |
|--------|----------------------------|----|
| 6.3.3. | Symbolien yhdistys | 21 |
| 6.3.4. | Kaapelimerkinnät..... | 21 |
| 6.3.5. | Tietokenttien täyttö | 22 |
| 6.3.6. | Tallennus | 22 |
| 7. | YHTEENVETO..... | 23 |
| 7.1. | Työn kulku | 23 |
| 7.2. | Tulokset..... | 23 |
| 7.3. | Kehitysmahdollisuudet..... | 24 |
| 8. | LÄHTEET | 25 |

KUVALUETTELO

| | | |
|----------|---|----|
| KUVA 1. | Esimerkki, piirikaavio (moottoripiiri)..... | 7 |
| KUVA 2. | Suunnitteluprosessin vuokaavio | 8 |
| KUVA 3. | Piirikaaviosuunnittelun vuokaavio | 12 |
| KUVA 4. | Excel taulukko, mistä luetaan tiedot muuttujiin | 15 |
| KUVA 5. | Pythonin "Lista" tyyppisiä muuttujia | 15 |
| KUVA 6. | Esimerkki, yhden listatyyppisen muuttujan sisältö (kuuden eri piirin positiot) | 15 |
| KUVA 7. | Ohjelmisto tekee ehdotelman mahdollisista valinnoista mitoitusvirran perusteella..... | 16 |
| KUVA 8. | Johdonsuojakatkaisija (ABB, 2022)..... | 17 |
| KUVA 9. | Moottorinsuojakytkin (ABB, 2022) | 17 |
| KUVA 10. | Kahvasulake (ABB, 2022)..... | 17 |
| KUVA 11. | Esimerkki, kytkinvarokkeen pohja käsin piirrettyinä | 20 |
| KUVA 12. | Päävirtapiiri koostuu yhdeksästä (9) osiosta | 20 |
| KUVA 13. | Esimerkki, johdonsuoja ja kontaktori | 21 |
| KUVA 14. | Esimerkki, kaapelimerkintä (tunnus: 1AP006-W1, tyyppi: MCMK 3x6/6)..... | 22 |
| KUVA 15. | Esimerkki, Johdonsuoja (tunnus: 6F1, tyyppi: C, koko: 20 A) | 22 |

1. JOHDANTO

Sähkö- ja automaatio suunnittelun keskeisiä työtehtäviä ovat järjestelmien, koneiden ja laitteiden kytkentäkuvien luominen, sekä muokkaaminen. Kytkentäkuva, eli viralliselta nimeltään piirikaavio on kuva, jolla ohjeistetaan ja selvennetään miten systeemin kytkennät ovat tehty.

Suomen Automaatiopalvelu Oy:ssä on mietitty, että olisiko mahdollista suunnitella ja piirtää piirikaavioita ohjelmallisesti, käyttäen ainoastaan riittävän kattavan Excel-muotoisen IO-luettelon sisältöä piirien lähtötietoina.

Piirikaaviot piirretään tavallisesti käsin, käyttäen suunnitteluohjelmia kuten CADMATIC Electrical, Microsoft Visio tai Eplan Electric. Näissä ohjelmissa on erilaisia toimintoja, jotka on suunniteltu helpottamaan ja nopeuttamaan piirtämistä, kuten valmiita symbolivalikkoja ja piiripohjia, sekä automaattinen symbolien yhdistys. Nämä toiminnot ovat kuitenkin vain avustavia, jonka vuoksi kuvien piirtäminen on aikaa vievää ja siksi myös kallista.

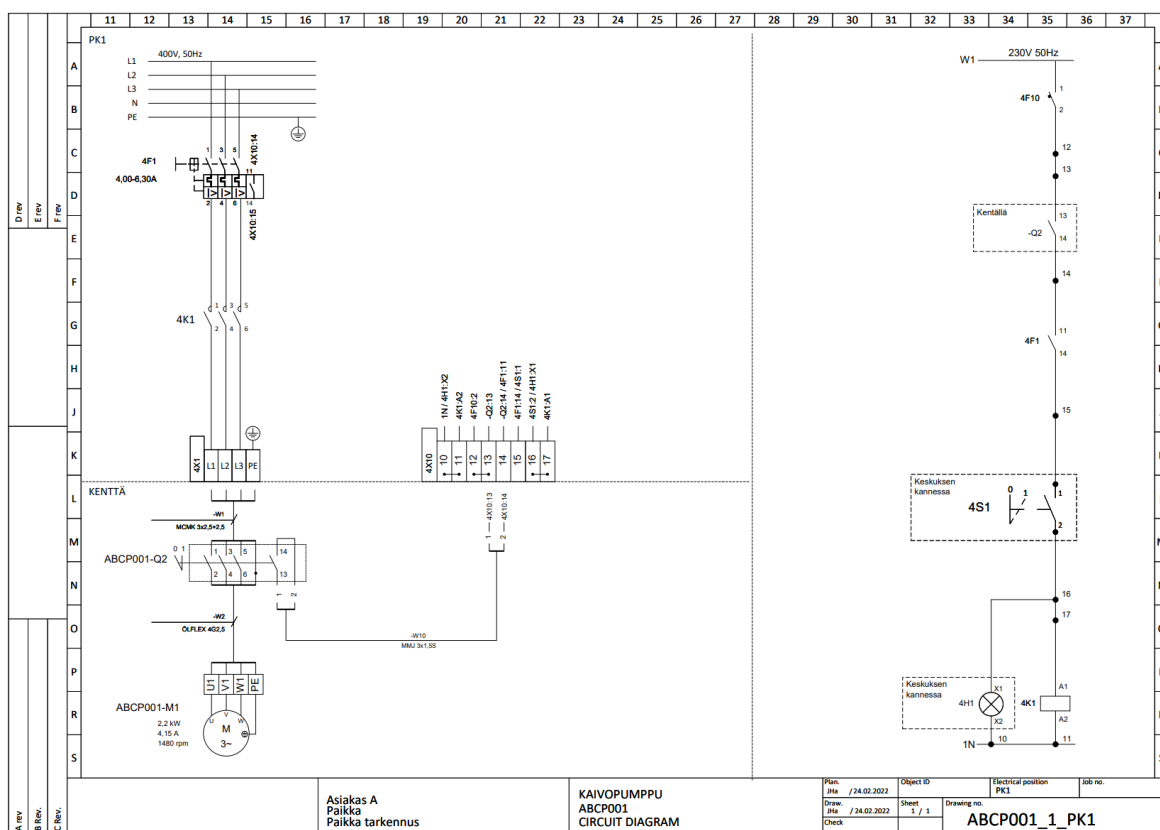
Tämän opinnäytetyön aiheena on kehitellä sähkösuunnittelussa käytettävä piirikaavioiden piirto-ohjelmiston prototyyppi, jonka tarkoitus on eliminoida tavanomainen käsin piirto kokonaan. Ohjelma lukee piirin tiedot suoraan Excel-taulukosta ja piirtää niiden perusteella halutun kuvan. Ohjelmointikielenä toimii Python.

2. PIIRIKAAVIO

2.1. Mikä piirikaavio on?

Kytkentäkaavio tai viralliselta nimeltään piirikaavio kuvaa laitteiden tai järjestelmien sähköisiä kytkentöjä ja tietoja. Kaaviosta nähdään kyseisen systeemin komponentit, laitteet ja niitä yhdistävät johtimet. Piirikaavioita on lukuisia erilaisia, riippuen kuvattavasta kohteesta ja käyttötarkoituksesta. Tämän vuoksi, tässä työssä keskitytään vain niin kutsuttujen ”moottoripiirikaavioiden” suunnitteluun.

Tyypillisen moottoritoimilaitteen piirikaavio koostuu pää- ja ohjausvirtapiiristä. Kuvassa 1, vasemmalla puolella sijaitsee päävirtapiiri ja oikealla puolella ohjausvirtapiiri. Päävirtapiirissä kuvataan piirin sähkön syötön tiedot, kuten suoja- ja ohjuslaite sekä kaapelityypit. Vuorostaan ohjauspiirissä kuvataan nimensä mukaisesti laitteen ohjauslogiikka, eli miten ja mistä laitetta ohjataan. Piirikaavioon kuuluu myös kehys, josta nähdään muun muassa asiakkaan ja piirin nimi, positio, kohteen sijainti, sekä suunnittelijan yritys ja nimikirjaimet.



KUVA 1. Esimerkki, piirikaavio (moottoripiiri)

2.2. Mihin piirikaaviota käytetään?

Piirikaaviot ovat käytännössä ohjeita sähköasentajille. Projektin sähköasennusten alkaessa tulisi kaikkien kytkettävien piirien kaaviot ja kaapelivetoluettelo olla valmiina, koska ilman niitä sähköasentajien on hankala tehdä yhtään mitään. Piirikaavioista on hyötyä myös silloin, jos jonkin piirin kytkentää halutaan jälkikäteen muuttaa tai korjata. Mikäli jotain piiriä muokataan, tulisi sen kuvat päivitettäväksi vastaamaan tehtyjä kytkentöjä.

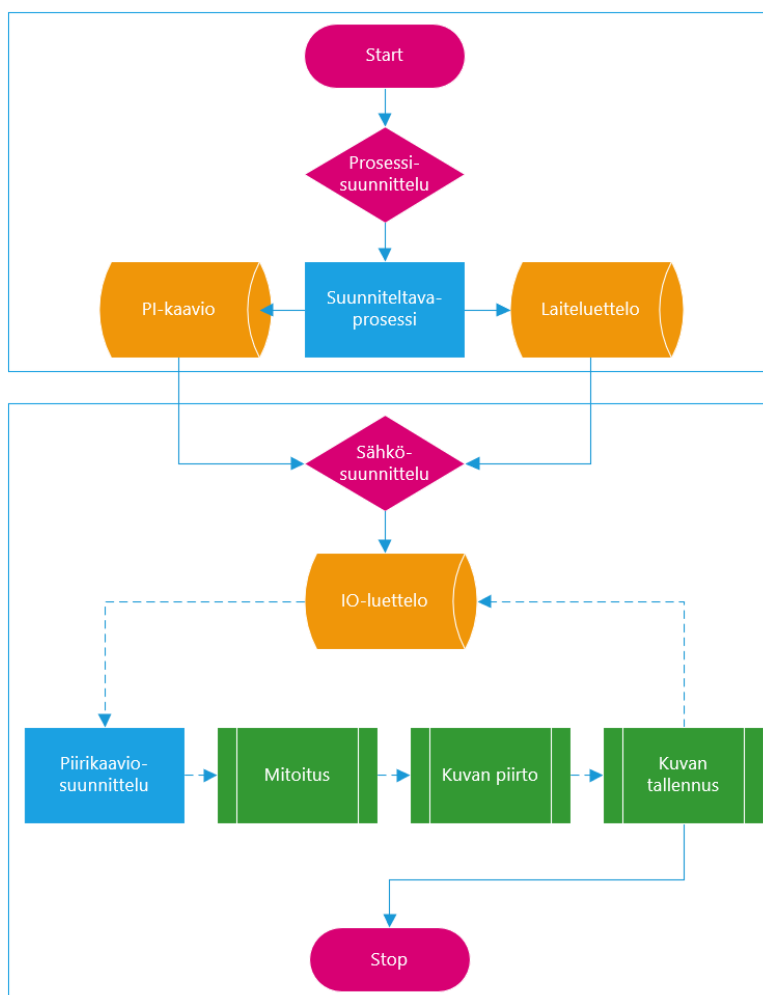
3. PIIRIKAAVIOIDEN SUUNNITTELU

3.1. Suunnitteluprosessi

Keskitytään tässä esimerkissä teollisen prosessin suunnitteluun, kuten elintarvikkeita valmistavan linjastoon. Suunnittelu alkaa prosessisuunnittelusta, missä luodaan PI-kaavio, eli putkitus- ja instrumentointikaavio, sekä laiteluettelo. PI-kaaviossa esitetään prosessin sähkösuunnittelun kannalta keskeiset laitteet ja säätöpiirit.

Sähkösuunnittelu alkaa valmiin PI-kaavion sekä laiteluettelon seurauksena. Ensin täytetään IO-luettelo. IO-luetteloon kerätään kaikki prosessissa esiintyvät mittaukset ja ohjaukset, sekä niiden tarkat tiedot.

Valmiin IO-luettelon jälkeen voidaan siirtyä piirikaaviosuunnitteluun, joka koostuu kolmesta eri alaprosessista. Mitoituslaskenta, kuvan piirtäminen ja lopuksi kuvan tallentaminen. Tämä kolmen alaprosessin työ suoritetaan jokaista IO-luettelon piiriä kohti.



KUVA 2. Suunnitteluprosessin vuokaavio

3.2. Piirin tekemiseen tarvittavat tiedot

Sähkösuunnittelun alussa käydään läpi laitemäärittely, joka on monesti IO-luettelon muodossa. IO-luettelosta nähdään kunkin laitteen positio, tyyppi, koko, ominaisuudet ja muut tiedot. Mitä laajemmin piireistä on tietoa, niin sitä helpompi sähkösuunnittelijan on alkaa suunnittelemaan piirikaavioita.

Listaan merkitään yleensä muun muassa laitteiden tarkat tyypit ja niiden avulla kytkentäohjeita kannattaa etsiä laitetyypin koodilla netistä, esimerkiksi suoraan laitevalmistajan omilta sivuilta.

TAULUKKO 1. IO-luettelon tietoja

| Paikka tarkennus | Piirin nimi | Positio | Kuva tunnus | Suunnittelija | Piirtäjä | Tarkastaja | Sivunumero | Nimellisjännite V | Teho kW | Virta A | Cosphi |
|------------------|-----------------|---------|-------------|---------------|----------|------------|------------|-------------------|---------|---------|--------|
| Verstas yläkerta | Sorvausruuvi | 1AP001 | 0001_01 | Jha | Jha | | 1 / 1 | 400 | 18 | 36 | 0,8 |
| Verstas yläkerta | Hihnakuljetin 1 | 1AP002 | 0001_02 | Jha | Jha | | 1 / 1 | 400 | 5 | 13 | 0,8 |
| Verstas yläkerta | Hihnakuljetin 2 | 1AP003 | 0001_03 | Jha | Jha | | 1 / 1 | 400 | 5 | 13 | 0,8 |
| Verstas alakerta | Hiomakone 1 | 1AP004 | 0001_04 | Jha | Jha | | 1 / 1 | 400 | 9 | 18 | 0,8 |
| Verstas alakerta | Hiomakone 2 | 1AP005 | 0001_05 | Jha | Jha | | 1 / 1 | 400 | 9 | 18 | 0,8 |

3.3. Mitoitus

Mitoituksella tarkoitetaan piirin johdonsuojan, kaapelien ja mahdollisten ohjauslaitteiden tyyppien valintaa.

Kaapeli valitaan piirin kuormitusvirran, jännitteenaleneman ja asennustavan mukaisesti. Kuormitusvirta määritellään piirin toimilaitteen, tai sen ohjauslaitteen mukaan, kuten taajuusmuuttaja. Jännitteenalenemaan vaikuttaa kuormitusvirta, sekä kaapelin tyyppi. Asennustapa vaikuttaa kaapelin lämpenemään ja sitä kautta kuormitusvirran keston.

Piirin suojalaite, kuten johdonsuojakatkaisija, tulee mitoittaa siten, että se suojaa koko kaapelia toimilaitteelle asti oikosulku- tai kuormitusvirran ylittyessä.

Mikäli ohjauslaitteena toimii taajuusmuuttaja, tulee sen mitoitus laskea moottorin virran ja sen käytötavan mukaisesti. Jos moottoria ohjataan jatkuvasti muuttuvilla kierroksilla tai pidetään jotain kuormaa paikallaan, voi olla, että taajuusmuuttaja tulee ylivoimaisesti moottorin nimellisvirtaa suuremmaksi. Mikäli näin tehdään, tulee kaapelin ja johdonsuojan koko mitoittaa taajuusmuuttajan nimellisvirran mukaisesti. Muiden ohjauslaitteiden tietoja, kuten kontaktorin tyyppi, tulee määrittää piirin nimellisvirran mukaan.

3.4. Piirtäminen

Laitteen piirikaavio piirretään IO-listan ominaisuuksien mukaisesti käyttäen haluttua suunnitteluohjelmaa, kuten esimerkiksi CADMATIC Electric tai Microsoft Visio.

Jos piiri on nimeltään "Pumppu" ja sen ohjaustapa on "Suora" niin kyseessä on kontaktorilla ohjattu yhteen suuntaan, yhdellä nopeudella pyörivä moottori. Muita yleisiä ohjaustapoja ovat muun muassa "Tamu", eli taajuusmuuttaja. Taajuusmuuttaja käytössä moottoria voi ohjata molempiin suuntiin eri nopeuksilla. Ohjaustapa voi olla myös "Suora+suunnanvaihto", eli kahdella kontaktorilla toteutettu kytkentä, joka mahdollistaa moottorin ohjauksen molempiin suuntiin, mutta vain yhdellä nopeudella. Piirin muut ominaisuudet määrittelevät, että mistä tai millä logiikalla pumppua ohjataan päälle tai pois.

Kaavioiden piirtäminen tapahtuu järjestyksessä ylhäältä alas ja vasemmalta oikealle, välttämällä symbolien yhdistämistä muilla kuin suorilla viivoilla. Käsillä piirtäessä käyttäen suunnitteluohjelmaa kuten CADMATIC, kuvia ei ole järkeä piirtää aina alusta asti, vaan hyödyntää mahdollisimman paljon edellisiin projekteihin piirrettyjä kuvia ja valmiita kirjastoja. Suunnittelutyö ja piirikaavioiden piirtäminen on paljon sitä, että kopioidaan edellinen piiri ja tehdään siihen vähän muutoksia.

3.5. Tallennus ja arkistointi

Valmiit kaaviot tallennetaan yleensä sähköisesti DWG- ja PDF-muotoihin. PDF-muoto on yleisesti käytetty koska niitä voi avata ja tarkastella kaikilla nykytietokoneilla ja puhelimilla. Vaikka DWG on Autocad tiedostomuoto niin sitä voi kuitenkin avata ja muokata myös monilla muilla piirustusohjelmilla, kuten Microsoft Visiolla. Tämän vuoksi se on hyvä tallennusmuoto arkistointiin, mikäli piirikaavioita tarvitsee tulevaisuudessa muokata.

4. SUUNNITTELUN KEHITTÄMINEN

4.1. Virheiden seuraukset ja niiden vähentäminen

Suurissa projekteissa, kuten tehtaiden automaatiolinjastojen päivityksissä tehdään lukuisia täysin samankaltaisia piirejä, joiden eroja ovat käytännössä vain positiot, nimet ja komponenttien tunnukset. Nykyään tämän tyyppiset suunnitelmat tehdään siten, että piirretään ensin yksi piiri valmiiksi, jonka jälkeen tästä samasta piiristä tehdään käsin kymmenen kopiota eri positioilla, nimillä ja tunnuksilla. Tällaiseen kopiokoneen virkaan ei sähköinsinööri ole koulutettu, ja syystäkin. Kopiointityö on puuduttavaa ja kaikki tunnukset ja nimet alkavat näyttää muutaman tunnin kuluttua samalta ja siksi virheet ovat väistämättömiä.

Esimerkiksi, mikäli tapahtuisi virhe missä kaapelityyppessä ei ole muutettu kuvissa oikeaksi, voi pahimmillaan mennä siihen, että väärät kaapelit vedetään virheellisten kuvien mukaisesti kentälle ja vika huomataan vasta myöhemmin, pahimmillaan jälkitarkastuksissa. Tällaisessa tilanteessa kaapelien vaihtotyö on erittäin hankalaa ja kallista. Kaapelireittien pituus ja monimutkaisuus lisäävät vain hintaa.

4.2. Automatisointi

Työtehtävät, joissa noudatetaan aina tiettyjä kaavoja, laskelmia ja päätelmiä voidaan hoitaa koneiden avulla ja tämä on jotakuinkin automaation määritelmä. Eli luodaan järjestelmiä, ohjelmia ja koneita, jotka kykenevät työskentelemään ihmisten avustuksella tai täysin itsenäisesti.

Piirikaavioiden suunnittelu on melko yksiselitteistä ja siihen tarvittavia lähtötietoja luetaan monesti samalla kaavalla. Vaikuttaisi siis siltä, että piirikaavioiden piirtämisprosessin työvaiheita voidaan siirtää koneiden tehtäväksi.

Piirikaavion tekemiseen ei käytännössä katsottuna tarvitse kuin riittävän kattavan IO-luettelon. Mikäli tällainen luettelo löytyy, niin piirien tiedot voidaan lukea ohjelmallisesti suoraan siitä ja piirtää näiden muuttujien perusteella kaavio. Näin työtahti nopeutuu ja virheiden mahdollisuus laskee.

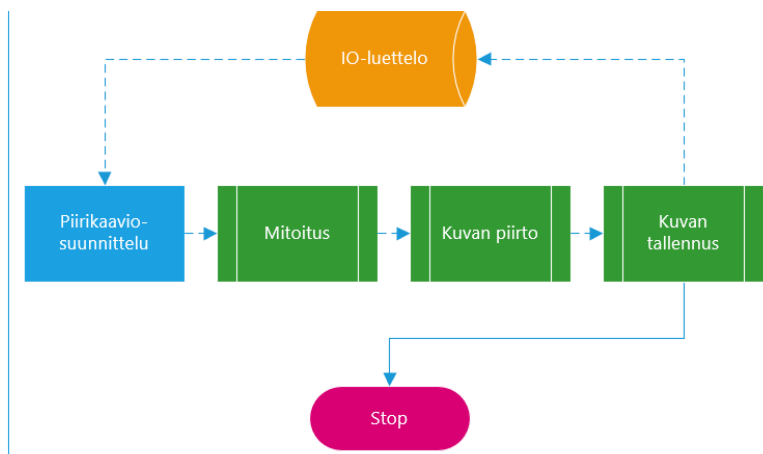
5. KEHITETTÄVÄ OHJELMISTO

5.1. Määrittely ja tavoite

Työnantaja määritteli työn tavoitteeksi moottoripiirikaavioiden ohjelmallisen piirtämisen ja täyttämisen, pelkän IO-luettelon avulla. Ohjelmistokielenä täytyi olla Python ja lisäksi ohjelmisto ei saanut käyttää mitään kolmannen osapuolen ohjelmistoa, kuten Autocadia, vaan ohjelmiston piti pystyä luomaan ja muokkaamaan kuvia täysin itsenäisesti.

Ohjelman määritelty toiminta:

1. Excel-taulukkoon (IO-luetteloon) täytetään käsin tarvittavat tiedot, kuten moottorin koko, positio, käytettävät komponentit, ylimerkinnän tarve ja kaapelityypit.
2. Ohjelma lukee tiedot luettelosta ja tekee mitoituslaskennan mm. sulakkeelle, kaapeleille ja taajuusmuuttajalle, jos tällainen on valittu.
3. Seuraavaksi ohjelma alkaa luonnostelemaan moottoripiiriä ja täyttää komponenttien tekstikenttiin oikeat tiedot.
4. Lopuksi kuvat nimetään position mukaisesti ja tallennetaan DWG ja PDF muodoissa.



KUVA 3. Piirikaaviosuunnittelun vuokaavio

Ideana on, että sähkösuunnittelija tarvitsee vain täyttää tarvittavat tiedot piirikohtaisesti Excel-taulukkoon, jonka jälkeen painetaan vain käynnistyspainiketta. Ohjelmisto hoitaa piirien suunnittelun ja piirtämisen. Piirit tallentuvat itsenäisinä tiedostoina ja nimetään position mukaisesti.

5.2. Ohjelmistoympäristö ja ohjelmointikieli

Ohjelmistoympäristöksi valittiin Microsoftin Visual Studio 2019 ja ohjelmointikieleksi Python, versio 3.9. Tätä kombinaatiota on käytetty yrityksen aiempien sisäisten ohjelmistojen kehityksessä ja on muun muassa siksi järkevä valinta jatkaa myös uusien ohjelmistojen kehitystä.

5.2.1. Visual Studio 2019

Visual Studio on Microsoftin kehittämä ohjelmistoympäristö, joka on suunniteltu graafisten käyttöliittymien, komentokonsolien, web-sovellusten, mobiilisovellusten, pilvipalveluiden ja ohjelmistojen

kehitykseen. Se käyttää hyödyksi lukuisia Microsoftin omia sovelluksia kuten Windows store, Microsoft Silverlight ja Windows API. Visual Studiota ei ole rajattu tiettyyn ohjelmistokieleen ja mahdollistaa koodausta 36 eri kielellä, kuten C#, C++, VB (Visual Basic), Python ja JavaScript. (GeeksforGeeks, 2019)

5.2.2. Python

Python on oliopohjainen ja helposti ymmärrettävä korkean tason ohjelmointikieli. Python julkaistiin vuonna 1992 ja se on rakennettu siten, että se olisi suhteellisen intuitiivisesti eli vaistonvaraisesti kirjoitettava ja ymmärrettävä. Python on erinomainen valinta aloittelijoille, jotka haluavat nopeaa kehitystä. Suuret ja tunnetut yhtiöt kuten esimerkiksi NASA, Google, Netflix ja Spotify käyttävät Pythonia tuotteiden ja palveluiden ylläpidossa. (FutureLearn, 2021)

Pythonilla voidaan tehdä esimerkiksi nettisivuja, ohjelmistoja, automaattioratkaisuja, data-analytiikkaa ja datan visualisointia. Pythonia käytetään paljon myös tekoälyn kehityksessä. (Coursera, 2021)

Python on avoin lähdekoodi ja sen on vapaasti muokattavissa, sekä uudelleen jaettavissa. Sen lataus ja käyttö ovat myös ilmaisia. Lisäksi Python toimii kaikkialla, Mac OS, Windows, Linux, Unix ja epäviralliset versiot ovat saatavilla myös Androidille ja iOS:lle. (PythonWiki, 2019)

5.3. Ohjelmoinnissa käytetyt kirjastot ja tietokanta

5.3.1. Python kirjasto, Pandas

Pandas on data-analytiikassa käytettävä Python kirjasto. Vuonna 2008 Wes McKinney näki tarpeen kvantitatiiviselle analytiikka työkalulle ja aloitti silloin kirjaston kehityksen. Siitä lähtien Pandas on kasvanut yhdeksi suurimmista Python kirjastoista. (Mode, -)

Pandas on rakennettu kahden muun pääkirjaston päälle, tiedon visualisoinnissa käytettävä Matplotlib ja NumPy matemaattisissa operaatioissa. Pandas toimii niin sanotusti ”kääreenä” näille kahdelle kirjastolle ja mahdollistaa niiden ominaisuuksien käytön vähemmällä koodilla. (Mode, -) Tässä opinnäytetyössä Pandas kirjastoa käytetään muun muassa tiedon keräykseen Excel-taulukosta ja sen käsittelyyn.

5.3.2. Python kirjasto, Ezdxf

Ezdxf on Python kirjasto Autodesk:n suunnittelema DXF tiedostomuodolle. Kirjasto mahdollistaa DXF tiedostojen ominaisuuksien lukemisen, muokkauksen sekä uusien tiedostojen luomisen.

Pääasiana Ezdxf kirjaston suunnittelussa pidettiin turhien ylikompleksisten DXF ominaisuuksien piilottamisessa ohjelmoijalta, mutta silti mahdollistaen suurimman osan DXF formaatin muokkaustoiminnoista. Kaikkia DXF ominaisuuksia ei vielä tueta, mutta kirjastoa ja sen ominaisuuksia päivitetään tulevaisuudessa. Kirjastoa käytetään tässä työssä kaikessa piirtämiseen liittyvässä, esimerkiksi, symbolien piirto ja niiden yhdistys, sekä tietokenttien muokkaus. (Moitzi, 2021)

5.3.3. Tietokanta, Excel

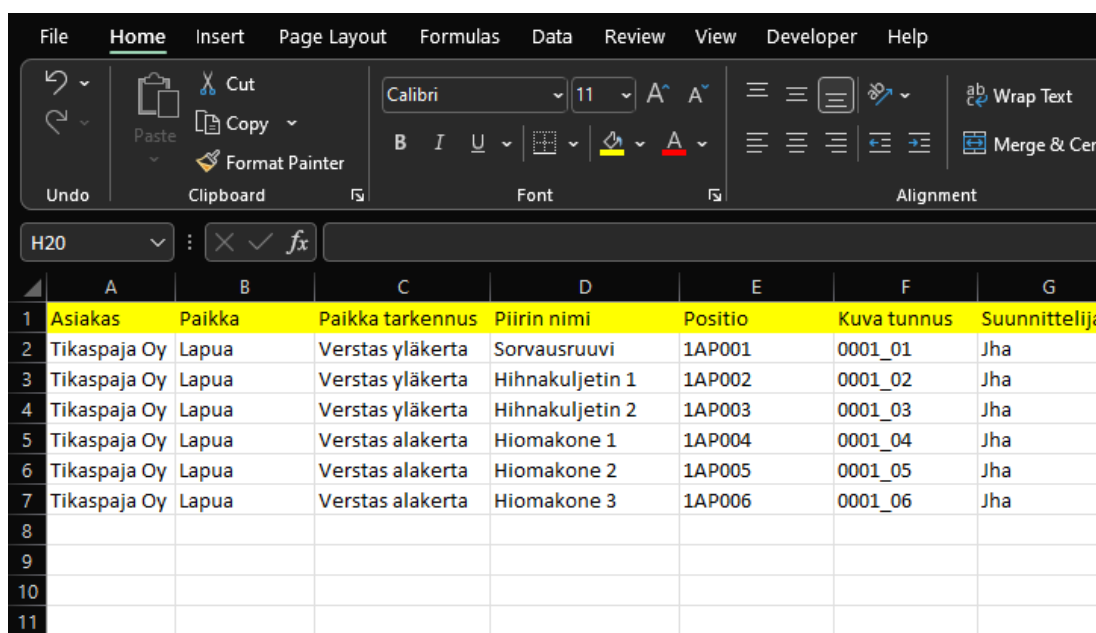
Ohjelmistolla ei ole virallista käyttöliittymää tai operointipaneelia, vaan kaikki käsiteltävät tiedot luetaan suoraan IO-listan tyylisestä Excel-tilukosta. Luettelo toimii siten ohjelmiston käyttäjän ja koneen välisenä rajapintana.

6. OHJELMISTON TOIMINNOT

6.1. Luetaan tiedot Excelistä Python muuttujiin

Kaikki piirtämiseen tarvittavat tiedot luetaan ensin Pythonin "lista" tyyppisiin muuttujiin. Muuttujat luetaan Excel-taulukosta rivi kerrallaan siten, että listojen ensimmäiset muuttujat ovat aina Excel-taulukon rivillä yksi. Seuraavan piirin muuttujat luetaan riviltä kaksi ja niin edelleen. Tähän tiedon-siirto ja käsittely toimintoon käytetään "Pandas" nimistä Python kirjastoa.

Lista-muuttujassa tiedot on nimensä mukaisesti listattu ja eroteltu pilkulla. Ohjelmisto käsittelee listan tietoja järjestyksessä kutsumalla listalta tietoja, tässä tapauksessa, oikealta vasemmalta.

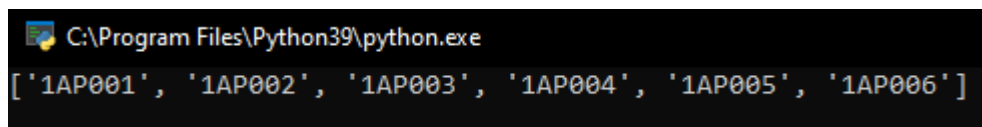


| | A | B | C | D | E | F | G |
|----|--------------|--------|------------------|------------------|---------|-------------|---------------|
| 1 | Asiakas | Paikka | Paikka tarkennus | Piirin nimi | Positio | Kuva tunnus | Suunnittelija |
| 2 | Tikaspaja Oy | Lapua | Verstas yläkerta | Sorvausruuvi | 1AP001 | 0001_01 | Jha |
| 3 | Tikaspaja Oy | Lapua | Verstas yläkerta | Hihnakuuljetin 1 | 1AP002 | 0001_02 | Jha |
| 4 | Tikaspaja Oy | Lapua | Verstas yläkerta | Hihnakuuljetin 2 | 1AP003 | 0001_03 | Jha |
| 5 | Tikaspaja Oy | Lapua | Verstas alakerta | Hiomakone 1 | 1AP004 | 0001_04 | Jha |
| 6 | Tikaspaja Oy | Lapua | Verstas alakerta | Hiomakone 2 | 1AP005 | 0001_05 | Jha |
| 7 | Tikaspaja Oy | Lapua | Verstas alakerta | Hiomakone 3 | 1AP006 | 0001_06 | Jha |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |

KUVA 4. Excel taulukko, mistä luetaan tiedot muuttujiin

```
# kehys
Write_Asiakas = Moottoripiirit_lista["Asiakas"].tolist()
Write_Paikka = Moottoripiirit_lista["Paikka"].tolist()
Write_Paikka_tarkennus = Moottoripiirit_lista["Paikka tarkennus"].tolist()
Write_Nimi = Moottoripiirit_lista["Piirin nimi"].tolist()
Write_Positio = Moottoripiirit_lista["Positio"].tolist()
Write_Suunnittelija = Moottoripiirit_lista["Suunnittelija"].tolist()
Write_Piirtaja = Moottoripiirit_lista["Piirtaja"].tolist()
Write_Tarkastaja = Moottoripiirit_lista["Tarkastaja"].tolist()
Write_Sivunumero = Moottoripiirit_lista["Sivunumero"].tolist()
Write_Kuva_tunnus = Moottoripiirit_lista["Kuva tunnus"].tolist()
```

KUVA 5. Pythonin "Lista" tyyppisiä muuttujia



```
C:\Program Files\Python39\python.exe
['1AP001', '1AP002', '1AP003', '1AP004', '1AP005', '1AP006']
```

KUVA 6. Esimerkki, yhden listatyyppisen muuttujan sisältö (kuuden eri piirin positiot)

6.2. Mitoituslaskenta

Tiettyjä lähtötietojen muuttujia tarvitaan ohjelmiston seuraavassa osiossa, eli mitoituslaskennassa. Mitoituksessa ohjelma päättää, laskee ja lukee standardin mukaisista mitoitusaulukoista, että mitä komponentteja ja kaapeleita piirikaavioon tulee valita.

Ohjelmisto on suunniteltu laskemaan kuormitukset vain 400 voltin jännitteellä. Tämän vuoksi laskennassa ei tarvitse tietää kuin moottorin virta, piirin suoja- ja ohjauslaite, taajuusmuuttajan mahdollinen yli- tai alimitoitus, kaapelityyppi, sekä asennustapa.

6.2.1. Taajuusmuuttajan valinta

Taajuusmuuttajan valinnassa tarkastellaan ensin moottorin virtaa ja haluttua yli-/alimitoituksen tarvetta. Lopuksi tehdään ehdotelma mahdollisesti valittavista taajuusmuuttajista, joista käyttäjä valitsee mieleisensä esimerkiksi valmistajan tai hinnan perusteella.

Ohjelmisto lukee moottorille asetetun virran ja halutun yli-/alimitoituksen prosentteina. Mikäli taajuusmuuttajia on valittu useampaan piiriin, ohjelmisto ehdottaa piirikohtaisesti jokaisen mahdollisesti valittavan taajuusmuuttajatyypin. Käyttäjä valitsee halutun taajuusmuuttajan kirjoittamalla komentokenttään halutun "ehdotus numeron", kuten esimerkiksi "2" (valitaan: "ehdotus nro 2").

```

C:\Program Files\Python39\python.exe
Jännite: 400 V
Teho: 18 kW
Virta: 36 A
Cosphi: 0.8
Käytettävä suojalaite: gG
Käytettävä ohjauslaite: Tamu
Tamun yli-/alimitoitus: 10.0 %

Ehdotus nro 1: ABB ACS880:
ACS880-01-045A-3

Ehdotus nro 2: Vacon 100 Industrial (IP21):
VACON0100-3L-0046-5
Listahinta: 2556 euroa

Ehdotus nro 3: Vacon 100 Industrial (IP54):
VACON0100-3L-0046-5+IP54
Listahinta: 2812 euroa

Ehdotus nro 4: Vacon 100 HVAC (IP21):
VACON0100-3L-0046-5-HVAC
Listahinta: 2222 euroa

Ehdotus nro 5: Vacon 100 HVAC (IP54):
VACON0100-3L-0046-5-HVAC+IP54
Listahinta: 2445 euroa

Ehdotus nro 6: Vacon 100 FLOW (IP21):
VACON0100-3L-0046-5-FLOW
Listahinta: 2516 euroa

Ehdotus nro 7: Vacon 100 FLOW (IP54):
VACON0100-3L-0046-5-FLOW+IP54
Listahinta: 2758 euroa

Valitse ehdotus #:

```

KUVA 7. Ohjelmisto tekee ehdotelman mahdollisista valinnoista mitoitusvirran perusteella

TAULUKKO 2. Esimerkki, lista Vacon 100 Industrial mallin taajuusmuuttajista

| | A | B | C | D |
|----|--|---|-----------------------------|--------------------|
| 1 | Tyyppi: | | | Virta: |
| 2 | Vacon 100 Industrial | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | Vacon 100, 380-500 V, IP21/Type 1, IP54/Type 12 and IP00/Open type, air-cooled wall-mounted drive, EMC Class C2, Gr | | | |
| 6 | PRODUCT CODE | | Low Overload Ratings | |
| 7 | | | Motor Power | Continuous |
| 8 | | | Low Overload | Current |
| 9 | IP54/Type 12 | | 400V [kW] | I _L [A] |
| 10 | VACON0100-3L-0003-5+IP54 | | 1,1 | 3,4 |
| 11 | VACON0100-3L-0004-5+IP54 | | 1,5 | 4,8 |
| 12 | VACON0100-3L-0005-5+IP54 | | 2,2 | 5,6 |
| 13 | VACON0100-3L-0008-5+IP54 | | 3 | 8 |
| 14 | VACON0100-3L-0009-5+IP54 | | 4 | 9,6 |
| 15 | VACON0100-3L-0012-5+IP54 | | 5,5 | 12 |
| 16 | VACON0100-3L-0016-5+IP54 | | 7,5 | 16 |
| 17 | VACON0100-3L-0023-5+IP54 | | 11 | 23 |
| 18 | VACON0100-3L-0031-5+IP54 | | 15 | 31 |
| 19 | VACON0100-3L-0038-5+IP54 | | 18,5 | 38 |
| 20 | VACON0100-3L-0046-5+IP54 | | 22 | 46 |
| 21 | VACON0100-3L-0061-5+IP54 | | 30 | 61 |
| 22 | VACON0100-3L-0072-5+IP54 | | 37 | 72 |
| 23 | VACON0100-3L-0087-5+IP54 | | 45 | 87 |
| 24 | VACON0100-3L-0105-5+IP54 | | 55 | 105 |
| 25 | VACON0100-3L-0140-5+IP54 | | 75 | 140 |
| 26 | VACON0100-3L-0170-5+IP54 | | 90 | 170 |
| 27 | VACON0100-3L-0205-5+IP54 | | 110 | 205 |
| 28 | VACON0100-3L-0261-5+IP54 | | 132 | 261 |
| 29 | VACON0100-3L-0310-5+IP54 | | 160 | 310 |

6.2.2. Suojalaitteen valinta

Piirin suojalaitteeksi on valittavana moottorinsuojakytkin, johdonsuoja tai gG-tyyppin kahvasulake. Ohjelmisto lukee ensin piirin muuttujista mitoitusvirran sekä valitun suojalaitetyypin.



KUVA 8. Johdonsuojakatkaisija (ABB, 2022)



KUVA 9. Moottorinsuojakytkin (ABB, 2022)



KUVA 10. Kahvasulake (ABB, 2022)

Ohjelmisto vertaa valittua mitoitusvirtaa taulukoihin, mihin on listattu erityyppisiä ja kokoisia suoja-laitteita. Ohjelmisto valitsee pienimmän mahdollisen suojan listalta ampeerien perusteella ja kirjoit-taa valitun laitetyypin uuteen muuttujaan, seuraavaa osiota varten.

TAULUKKO 3. Esimerkki, lista valittavista moottorinsuojakytkimistä

| | A | B |
|----|----------------|--------|
| 1 | Tyyppi: | Virta: |
| 2 | ABB MS132-0.16 | 0.16 |
| 3 | ABB MS132-0.25 | 0.25 |
| 4 | ABB MS132-0.4 | 0.4 |
| 5 | ABB MS132-0.63 | 0.63 |
| 6 | ABB MS132-1.0 | 1.0 |
| 7 | ABB MS132-1.6 | 1.6 |
| 8 | ABB MS132-2.5 | 2.5 |
| 9 | ABB MS132-4.0 | 4.0 |
| 10 | ABB MS132-6.3 | 6.3 |
| 11 | ABB MS132-10 | 10 |
| 12 | ABB MS132-12 | 12 |
| 13 | ABB MS132-16 | 16 |
| 14 | ABB MS132-20 | 20 |
| 15 | ABB MS132-25 | 25 |
| 16 | ABB MS132-32 | 32 |

Mikäli käsiteltävään piiriin on valittu taajuusmuuttaja, jonka nimellisvirta on suurempi kuin toimilait-teen, suojalaitteen koon valinta tehdään silloin taajuusmuuttajan koon perusteella. Esimerkiksi, jos taajuusmuuttajalle on haluttu 20 prosentin ylimitoitus, tulee suojalaitteen mitoitusvirta valita taa-juusmuuttajan huomattavasti suuremman virran mukaan, eikä pienemmän pyöritettävän moottorin virran mukaan.

6.2.3. Moottorikaapelin valinta

Kaapelin valinnassa otetaan huomioon valitun suojalaitteen määrittelemä mitoitusvirta, asennustapa, sekä valittu kaapelityyppi.

Ohjelmisto vertaa suojalaitteen määrittämää mitoitusvirtaa ja valittua asennustapaa, ja valitsee nii-den perustella taulukosta kaapelille sopivan poikkipinta-alan. Esimerkiksi, jos suojalaitteen koko on 50 ampeeria ja asennustapa on "E", valitaan johtimen poikkipinta-alaksi 10 mm² koska, 10 mm² ku-parikaapeli kestää 63 ampeerin kuormitusvirran "E" luokan asennustilanteessa.

TAULUKKO 4. Kuparikaapelin poikkipinta-alan valintataulukko

| | A | B | C | D | E |
|----|--------------|-----|------|-----|-----|
| 1 | Poikkipinta: | A | C | D | E |
| 2 | 1.5 | 14 | 18.5 | 26 | 19 |
| 3 | 2.5 | 19 | 25 | 35 | 26 |
| 4 | 4 | 24 | 34 | 46 | 36 |
| 5 | 6 | 31 | 43 | 57 | 45 |
| 6 | 10 | 41 | 60 | 77 | 63 |
| 7 | 16 | 55 | 80 | 100 | 85 |
| 8 | 25 | 72 | 102 | 130 | 107 |
| 9 | 35 | 88 | 126 | 160 | 134 |
| 10 | 50 | 105 | 153 | 190 | 162 |
| 11 | 70 | 133 | 195 | 240 | 208 |
| 12 | 95 | 159 | 236 | 285 | 252 |
| 13 | 120 | 182 | 274 | 325 | 292 |
| 14 | 150 | 208 | 317 | 370 | 338 |
| 15 | 185 | 236 | 361 | 420 | 386 |
| 16 | 240 | 278 | 427 | 480 | 456 |
| 17 | 300 | 316 | 492 | 550 | 527 |

Katsotaan mikä kaapelityyppi piirille on valittu ja tehdään valinta kaapelityyppi kohtaisesta taulukosta, valitsemalla tyyppi vaihejohtimen koon mukaisesti. Esimerkiksi jos poikkipinta-alaksi on valittu 10 mm² ja kaapelityypiksi MCMK, niin kaapeliksi valitaan taulukosta MCMK 3x10/10.

TAULUKKO 5. Esimerkki, MCMK kaapelityypin valintataulukko

| | A | B |
|----|-----------|---------------------|
| 1 | Tyyppi: | Vaihejohtimen koko: |
| 2 | 3x1.5/1.5 | 1.5 |
| 3 | 3x2.5/2.5 | 2.5 |
| 4 | 3x6/6 | 6 |
| 5 | 3x10/10 | 10 |
| 6 | 3x16/16 | 16 |
| 7 | 3x25/16 | 25 |
| 8 | 3x35/16 | 35 |
| 9 | 3x50/25 | 50 |
| 10 | 3x70/35 | 70 |
| 11 | 3x95/50 | 95 |
| 12 | 3x120/70 | 120 |

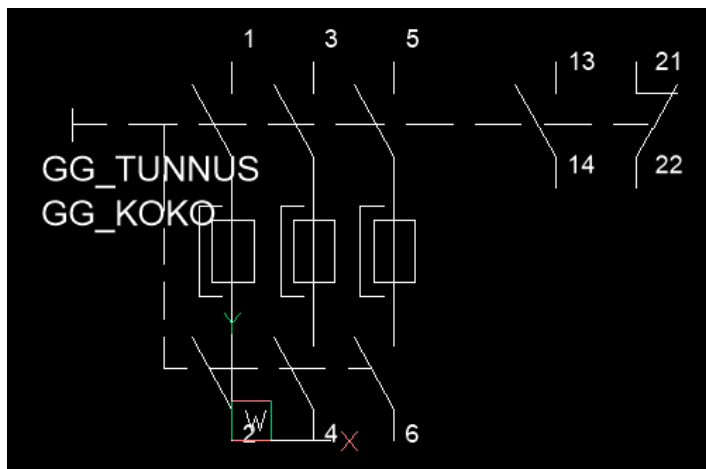
6.3. Piirretään kuva

Kuvien piirrossa käytetään Python kirjastoa nimeltä "Ezdx". Tämä on .DXF tiedostomuotojen luontiin ja muokkaukseen suunniteltu kirjasto ja on tärkein osa ohjelmiston kokonaisuuden toiminnan kannalta.

6.3.1. Valmistelut

Kaikki kuvissa käytettävät komponentit, kehykset, ja symbolit tulee ensin piirtää käsin ja tallentaa ne omina tiedostoinaan "Components" nimiseen kansioon .DXF muodossa. Ohjelmisto asettelee symbolit lopulliseen kuvaan muuttamalla niiden koordinaatteja ja siksi käsin piirtämisessä tulee varmistaa, että symbolien nollakoordinaatit sattuvat aina samaan kohtaan. Piirtämisessä tulee myös huomioida,

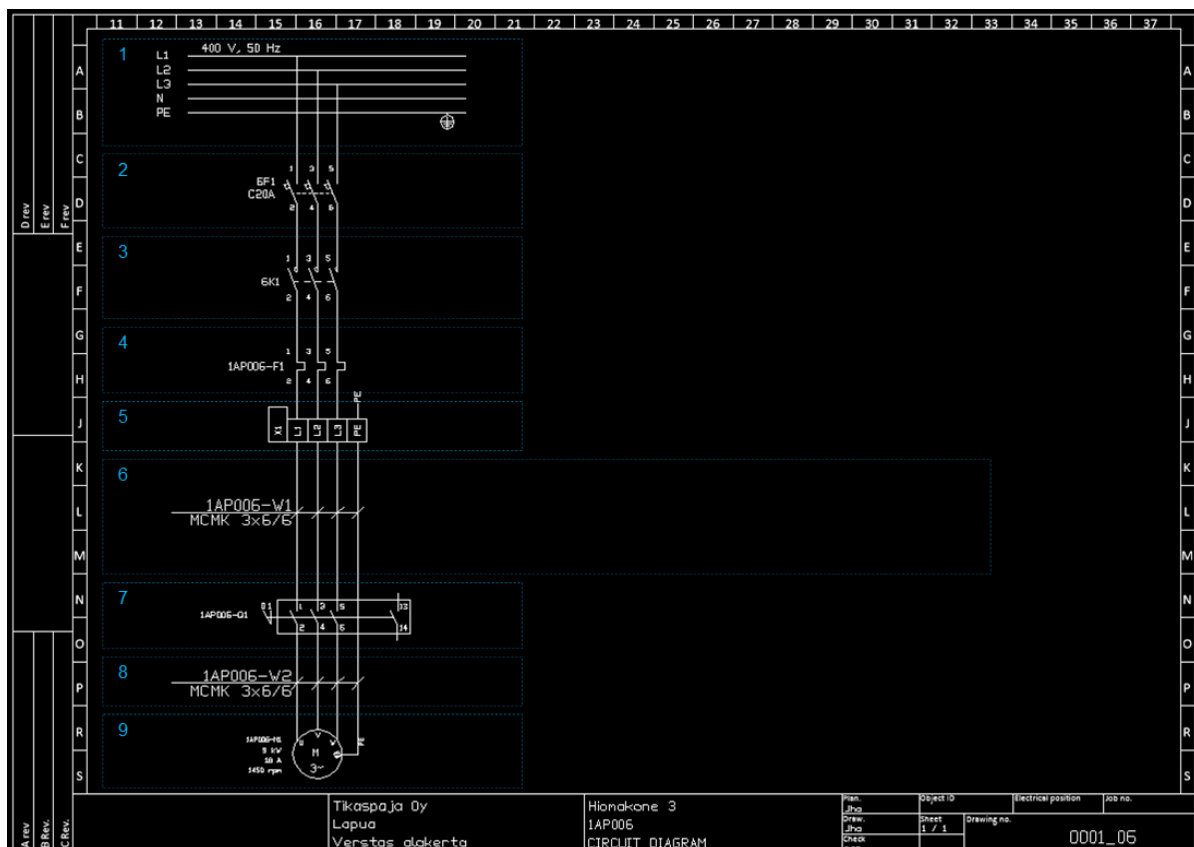
että komponenttien johtimien liityntäpisteiden välit tulee olla samalla leveydellä ja yleinen kokoluokka on samankaltainen. Komponenttien muokattavat tietokentät, kuten tunnukset ja laitetiedot kirjoitetaan kaikki isoilla kirjaimilla.



KUVA 11. Esimerkki, kytkinvarokkeen pohja käsin piirrettynä

6.3.2. Symbolien asettelu

Päävirtapiiri on jaettu yhdeksään eri osioon, johon ohjelmisto asettelee eri symboleita. Jokaiseen osioon on valittavissa tiettyjä symboleita, mitkä tulee ensin piirtää ja tallentaa käsin "Components" nimiseen kansioon. Osioden koordinaatit eivät ole aina samoja. Ne voivat elää hieman riippuen mitä komponentteja kuvaan on valittu. Esimerkiksi, mikäli piirissä on taajuusmuuttaja, niin silloin riviliittimiä ja muita ylempänä olevia komponentteja siirretään vähän ylöspäin. Näin kuvasta ei tule liian ahdas ja se on helpommin tulkittavissa.



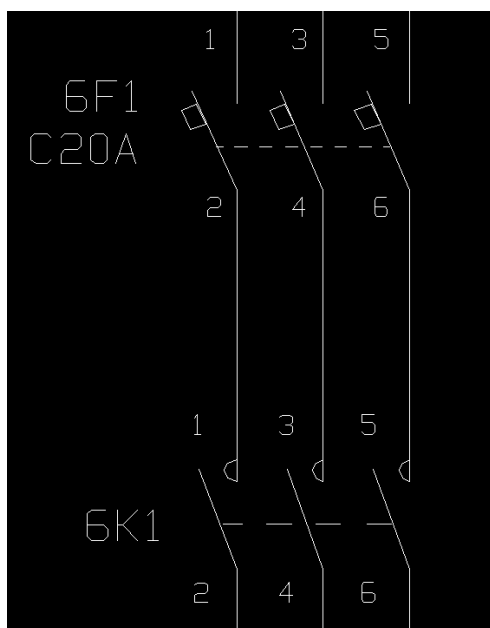
KUVA 12. Päävirtapiiri koostuu yhdeksästä (9) osiosta

Kuvan yhdeksän osiota ja niiden sisältö:

1. Päävirtakisko
2. Suojalaite (Moottorinsuojakytin, johdonsuoja tai kahvasulake)
3. Ohjauslaite (Kontaktori, kontaktorit suunnanvaihdolla tai ei mitään)
4. Lisälaite (Lämpörele tai ei mitään)
5. Riviliittimet
6. Taajuusmuuttaja (tai ei mitään)
7. Turvakytin (tai ei mitään)
8. Pistoke (tai ei mitään)
9. Toimilaite (Moottori)

6.3.3. Symbolien yhdistys

Ohjelmisto käynnistää symbolien yhdistystoiminnon. Symboliviivojen koordinaattien mukaan tehdään päätelmä, että mitkä viivojen päät tulee yhdistää seuraavaan komponenttiin. Luetaan esimerkiksi piirin suojalaitteen kolme alinta y-koordinaattia järjestyksessä vasemmalta oikealle ja laitetaan nämä muistiin. Sen jälkeen luetaan ohjauslaitteen kuten kontaktorin kolme ylintä y-koordinaattia järjestyksessä vasemmalta oikealle. Lopuksi piirretään yhdistysviivat suojalaitteen alimmista koordinaateista ohjauslaitteen ylimpiin koordinaatteihin. Yhdistysviivoja ei siis asetella aina tietyille paikoille, vaan ne ovat joustavia ja venyvät aina oikean mittaisiksi, mikäli symbolien koordinaatit muuttuvat.



KUVA 13. Esimerkki, johdonsuoja ja kontaktori

6.3.4. Kaapelimerkinnot

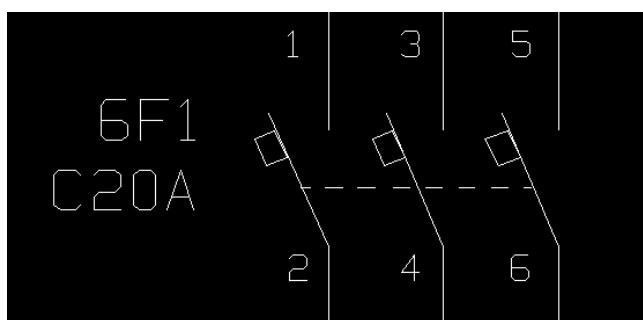
Merkinnät asetellaan keskuksen ulkopuolelle tuleviin kaapeleihin. Merkinnot ovat aina samannäköisiä mutta niiden tunnuksot, tyypit ja koordinaatit muuttuvat. Merkintöjen koordinaatit päätellään piirissä käytettävien komponenttien perusteella. Tunnuksot kirjoitetaan seuraavasti, piirin positio + "-W1", "-W2", "-W3" tai "-W4". Tunnuksen loppuviite riippuu siitä, että monentena kaapeli sijaitsee keskukselta päin luettuna. Kaapelityyppi määritellään ohjelmiston alkuvaiheissa "Moottorikaapelin valinta" nimisessä toiminnossa.



KUVA 14. Esimerkki, kaapelimerkintä (tunnus: 1AP006-W1, tyyppi: MCMK 3x6/6)

6.3.5. Tietokenttien täyttö

Kaikilla komponenteilla on ainakin yksi tietokenttä, josta näkyy komponentin tunnus. Useilla komponenteilla on myös lisäkenttiä, joihin määritellään tarkemmat tiedot, kuten esimerkiksi sulakkeen tyyppi. Ohjelmisto täyttää laitteiden tunnuskentät IO-luetteloon kirjoitetuilla tiedoilla, mutta komponenttien lisätiedot luetaan yleensä ohjelmiston "mitoitus" toiminnon kautta. Tämä siksi, että "mitoitus" toiminto määrittelee laitteiden tarkemmat tiedot, kuten esimerkiksi sulakkeen koon, riippuen piirin muista ominaisuuksista.



KUVA 15. Esimerkki, Johdonsuoja (tunnus: 6F1, tyyppi: C, koko: 20 A)

6.3.6. Tallennus

Työn määrittelyssä oli tarkoitus, että ohjelmisto tallentaa valmiit kuvat DWG ja PDF muodoissa. Tämä asia on tehtävissä, mutta tehdään vasta myöhemmin. Ohjelmisto tallentaa kuvat tällä hetkellä vain DXF-muotoon, eikä kääntämistä PDF- ja DWG-muotoihin tulla vielä tekemään. Tämä siksi, että ohjelmiston muita piirtämiseen liittyviä osa-alueita ei aikataulusyistä tehty vielä loppuun ja kuvien tallentaminen julkaistavaan muotoon on viimeisenä vuorossa.

7. YHTEENVETO

7.1. Työn kulku

Työ aloitettiin selvittämällä, onko moottoriipiirejä mahdollista piirtää ohjelmallisesti ja miten se voisi tapahtua. Ohjelmointikieleksi valittu Python ei vaikeuttanut menetelmien etsintää, mutta ongelmana oli itsenäisen ohjelmiston kehittäminen, joka ei saanut ohjata mitään kolmannen osapuolen ohjelmistoa. Esimerkiksi Microsoft Visiota olisi voinut ohjata Pythonin avulla käyttämällä ohjelmiston makroja. Tätä menetelmää testattiin mutta hylättiin lopulta työnantajan toimesta.

Itsenäisen ohjelmiston suunnitteluun löytyi kuitenkin Python kirjasto nimeltä "ezdxf", joka oli tarkoitettu DXF tiedostomuotojen luomiseen ja muokkaamiseen. Tämä kirjasto ei vaatinut mitään kolmannen osapuolen ohjelmistoa, mikä oli tärkeä ominaisuus työn määrittelyssä. Menetelmä esitettiin ja hyväksyttiin työnantajalla, jonka jälkeen työn virallinen tekeminen alkoi.

Ohjelmointityö sujui pitkälti todella hyvin, mutta ohjelmiston eri osioiden valmistuessa alkoi hahmottua, että työn laajuus on aivan liian suuri, eikä sitä saada pidettyä tämän opinnäytetyön laajuuden rajoissa.

7.2. Tulokset

Opinnäytetyö vei ohjelmallista piirikaavioiden suunnittelua huomattavasti eteenpäin, ja tässä saatiin todistettua menetelmiä, kuinka se voisi olla mahdollista. Työ jäi kuitenkin kesken aikataulusyistä.

Tässä versiossa ohjelmisto kykenee lukemaan tarvittavat tiedot IO-luettelosta, sekä piirtämään moottoriipiirikaavion päävirtapiirin, muttei vielä ohjausvirtapiiriä. Lisäksi kuvien kääntäminen ja tallentaminen DXF muodoista PDF ja DWG muotoihin jäi myös tekemättä. Tämä siksi, että kuvat eivät ole vielä valmiita tai muutenkaan käyttökelpoisia ilman ohjausvirtapiiriä, jolloin kuvien automaattinen tallentaminen julkaistaviin muotoihin ei ole vielä ajankohtaista. DXF muodon kuvat voidaan kuitenkin kääntää haluttuihin muotoihin nopeasti ohjelmilla kuten Open Design Alliancen julkaisema ODA-FileConverter ja AnyDWG:n julkaisema Any DWG to PDF Converter. (OpenDesign, 2022) (AnyDWG, 2022)

Ohjelmiston tämänhetkiset ominaisuudet toimivat hyvin ja esimerkiksi mitoitus on pitänyt paikkansa jokaisella testaus keralla. Mitoitusohjelmaosuuden lukemat taulukot ja muut tiedot ovat sisällöltään identtisiä työntekijöiden käyttämien taulukoiden kanssa.

Päävirtapiirin ulkoasu on samanlainen kuin käsin piirretyllä, ja on todella hankala erottaa koneen piirtämäksi. Tulevaisuudessa tehtävä koneen piirtämä ohjausvirtapiiri tulee olemaan parin testauksen mukaan todennäköisesti hieman erilaisempi kuin ihmisen piirtämä, mutta sisältäen tietenkin samat tiedot.

Ohjelmiston toiminta on vielä käytännössä katsottuna suppea eikä sillä ole tässä vaiheessa käyttöä työelämässä. Tämä oli kuitenkin täysin ennustettavissa, koska ohjelmiston määritelty laajuus on valtava, että myös haastava ja vaatii todennäköisesti vuosien mittaista kehittämistä. Tässä opinnäytetyössä kehitetty ohjelmisto on osana suurempaa kokonaisuutta, joka koostuu monista muista ohjelmistoista ja sitä on tarkoitus kehittää tulevaisuudessa.

7.3. Kehitysmahdollisuudet

Kuten aiemmassa osiossa mainittiin, tämä ohjelmiston versio on toiminnoiltaan vielä suppea, eikä sillä sen vuoksi ole työelämässä käyttöä. Ohjelmisto kykenee vasta piirtämään piirikaavioiden päävirtapiiriin, muttei ohjausvirtapiiriä. Useimmiten piirikaavioiden kompleksisin osa on nimenomaan ohjausvirtapiiri ja tällä hetkellä se joudutaan edelleen piirtämään käsin.

Ohjausvirtapiirin koneellista piirtoa on työn kehityksen aikana suunniteltu ja se vaikuttaa mahdolliselta, mutta paljon haasteellisemmalle kuin päävirtapiirin luominen. Ohjelmiston kehitystä voidaan jatkaa muillakin osa-alueilla, kuten automaattisten komponenttiluetteloiden luomisessa, komponenttien valinnassa ja moottorien käynnistysvirtojen sekä jännitteenalenemien analysoinnissa.

8. LÄHTEET

- ABB. (2022). MO496-50. Noudettu osoitteesta ABB:
<https://new.abb.com/products/1SAM590000R1006/mo496-50>
- ABB. (2022). OFAA00GG80. Noudettu osoitteesta ABB:
<https://new.abb.com/products/1SCA022700R9440/ofaa00gg80>
- ABB. (2022). SH201-C16. Noudettu osoitteesta ABB:
<https://new.abb.com/products/2CDS211001R0164/sh201-c16>
- AnyDWG. (2022). Any DWG to PDF Converter. Noudettu osoitteesta <https://anydwg.com/dwg2pdf/>
- Coursera. (22. Syyskuu 2021). What Is Python Used For? A Beginner's Guide. Haettu 02. Helmikuu 2022 osoitteesta Coursera: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- FutureLearn. (09. Huhtikuu 2021). What is Python used for? 10 practical Python uses. Haettu 02. Helmikuu 2022 osoitteesta FutureLearn: <https://www.futurelearn.com/info/blog/what-is-python-used-for>
- GeeksforGeeks. (04. Joulukuu 2019). Introduction to Visual Studio. Haettu 02. Helmikuu 2022 osoitteesta GeeksforGeeks: <https://www.geeksforgeeks.org/introduction-to-visual-studio/>
- Mode. (-). Pandas. Haettu 02. Helmikuu 2022 osoitteesta Mode: <https://mode.com/python-tutorial/libraries/pandas/>
- Moitzi, M. (2021). Introduction. Haettu 06. Kesäkuu 2021 osoitteesta ezdx: <https://ez-dxf.readthedocs.io/en/stable/introduction.html>
- OpenDesign. (2022). ODAFileConverter. Noudettu osoitteesta https://www.open-design.com/guestfiles/oda_file_converter
- PythonWiki. (18. Syyskuu 2019). BeginnersGuide/Overview. Haettu 02. Helmikuu 2022 osoitteesta Python Wiki: <https://wiki.python.org/moin/BeginnersGuide/Overview>