



Käyttöönotto- ja koestuspöytäkirjojen kehitys

Kimmo Mäki

OPINNÄYTETYÖ
Joulukuu 2021

Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Sähkövoimatekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Sähkövoimatekniikka

MÄKI, KIMMO
Käyttöönotto- ja koestuspöytäkirjojen kehitys

Opinnäytetyö 29 sivua.
Joulukuu 2021

Tässä opinnäytetyössä kehitetään ja modernisoidaan käyttöönotto- ja koestuspöytäkirjojen täyttämistä ja luomista, sekä käsitellään keinoja saada aikaan säästöjä työtunneissa ja vakauttaa asiakkaille lähtevien raporttien laatu ja muotoilu yhdenmukaiseksi. Työtunneissa saadaan säästöä automatisoimalla toistuvat työvaiheet Excel-taulukkolaskennan VBA-koodausta hyödyntäen ja luodaan Excel-työkalu, joka sisältää kaikki nykyisin käytettävät koestuspöytäkirjat ja johon on helppo lisätä uusia laitteita.

Excel-työkalun avulla on mahdollista, yhdestä ikkunasta, luoda sadoille eri sähköjakelun suojalaitteille, katkaisijoille, erottimille ja vastaaville koestuspöytäkirjat. Uudet pöytäkirjat luodaan tehokkaasti, yrityksen vaatimusten ja nimeämiskäytäntöjen mukaisesti, sekä ilman eriyvää ulkoasua käyttäen opinnäytetyössä käsiteltyjä menetelmiä apuna.

Asiasanat: käyttöönottopöytäkirja, koestuspöytäkirja, Excel, VBA, suojarele, katkaisija, erotin, kojeisto

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical Engineering, Bachelor of Engineering
Power Engineering

MÄKI, KIMMO
Developing of the Commissioning Reports

Bachelor's thesis 29 pages.
December 2021

This bachelor's thesis addresses how to develop and modernize the filling and creating of commissioning test reports. The finished tool saves labor hours and improves quality of the reports that are sent to clients. The tool eliminates repetitive tasks in Excel Worksheets by taking advantage of VBA coding build into the Excel. The report tool contains all currently used commissioning reports in the company and new sheets can be easily added with macros.

The Excel tool enables producing commissioning reports for hundreds of power distribution devices including protective relays, circuit breakers, disconnectors, and switchgear. Operation is done from a single sheet to make it efficient and easy for the end-user. Making new files follows all the naming and quality guidelines the company has set for the reports.

Key words: commissioning, report, Excel, VBA, protective relay, circuit breaker, disconnector, switchgear

SISÄLLYS

1	JOHDANTO	6
2	KOESTAMINEN JA KÄYTTÖÖNOTTO	7
2.1	Huollettavat ja koestettavat komponentit.....	7
2.1.1	Erottimet	7
2.1.2	Katkaisijat	9
2.1.3	Suojareleet	10
2.1.4	Kompensointi.....	12
2.1.5	Maadoitukset ja eristykset	12
2.2	Raportointi.....	13
3	VBA-KOODAUKSEN PERUSTEET	14
3.1	VBA-koodin kehitys	14
3.2	Koodin rakenne.....	14
4	PÖYTÄKIRJATYÖKALUN RAKENTAMINEN	16
4.1	Vaatimusten suunnittelu	16
4.2	Toteuttaminen	16
4.2.1	Taustatyöt.....	17
4.2.2	Tiedostopolun valinta.....	18
4.2.3	Tietojen kopioiminen uuteen tiedostoon	19
4.2.4	Hallinnointisivun kehittäminen	21
4.2.5	Pöytäkirjapohjien lisäys	23
4.2.6	Pöytäkirjapohjien laskentakaavat ja laitetiedot	25
5	POHDINTA	27
	LÄHTEET	28

LYHENTEET JA TERMIT

VBA	Excel ohjelmointikieli (Visual Basic for Applications)
SF ₆	Hyvin sähköä eristävä kaasu (Rikkiheksafluoridi)
VAC	Vaihtojännite
VDC	Tasajännite
PJ	Pienjännite, joka on välillä 50–1000 VAC (1.)
KJ	Keskijännite, joka on välillä 1–36 kVAC (1.)
SJ	Suurjännite, joka on yli 36 kVAC (1.)

1 JOHDANTO

Opinnäytetyötä lähdettiin kehittämään, koska koestus- ja käyttöönottopöytäkirjojen täyttö sisältää paljon itseään toistavia työvaiheita. Itseään toistavat työvaiheet eivät motivoi työntekoon ja lisäksi ne ovat yritykselle iso kuluerä vuositasolla. Lisäksi kasvavassa yrityksessä ongelmaksi muodostui asiakkaalle lähtevien raporttien laatueroamat, kun raporttien tekijöitä on toistakymmentä ja kaikki tekevät pohjiin omia pieniä muutoksiaan tai vanhojen projektien pohjat olivat jo vanhentuneita.

Pöytäkirjat toteutettiin jo valmiiksi pääasiassa Excel-taulukkolaskennalla, joten Excel ja sen sisältämä ohjelmointikieli VBA, eli Visual Basic for Applications, oli luonnollinen valinta opinnäytetyön toteutukseen. Osaamista VBA:n käyttöön oli hieman, sekä kurssien, että aiempien projektien muodossa, mutta näin laajaa kokonaisuutta toteuttaessa täytyy huomattava osa projektista toteuttaa etsien Google-hauilla ja Microsoftin VBA-ohjeista tehokkaita tapoja toteuttaa toimintoja.

Valmis Excel-työkalu sisältää raportointipohjia yli kymmeneltä sähköjakeluverkon komponentteja valmistavalta yritykseltä, eri vuosikymmeniltä. Kokonaislaitemäärä nousee siis noin sataan erilaiseen laitepöytäkirjaan, joissa on erilaisia ominaisuuksia, toimintoja ja suojauksia. Laitevalmistajien ohjekirjoihin täytyi perehtyä ja kokemuseräistä osaamista vanhoista laitteista siirtää raportointipohjiin.

2 KOESTAMINEN JA KÄYTTÖÖNOTTO

2.1 Huollettavat ja koestettavat komponentit

Sähkönjakeluverkon laitteistojen koestamisessa ja käyttöönotossa erilaisia laitteistoja, niiden osia ja toimintaperiaatteita täytyy tuntea erittäin suuri määrä perusteellisesti. Suomessa sähkönjakeluverkossa on edelleen laitteita 50-luvulta aina täysin uusiin kohteisiin, joten vain uusimpien laitteiden osaamisella ei pysty toimimaan.

Varsinkin käyttöönotoissa on myös tärkeää testata kaikkien laitteiden välisten luokitusten toimivuus, jotta käyttäjän virheiden mahdollisuus on eliminoitu tehokkaasti ja esimerkiksi maadoituserotinta vasten ei ole mahdollista kytkeä jännitteitä. Alla on eriteltynä tärkeimmät huollettavat komponentit, mutta ei läheskään kaikkia huoltoa ja koestusta vaativia sähkönjakeluverkon osia.

2.1.1 Erottimet

Erottimet nimensä mukaisesti erottavat kaksi sähköä johtavaa osaa toisistaan. Tavalliset erottimet eivät omaa virrankatkaisukykyä ja ne voivat erottaa jännitteellisiä osia toisistaan vain, kun virtapiiri on katkaistu muulla komponentilla.

Kuormanerotin taas on yleensä jousivoimainen erotin, joka omaa myös virrankatkaisukyvyyn, mutta ei yleensä kovin suurta oikosulkuvirran katkaisukykyä. Kuormanerotin jousi on aina vireessä, kun erotin on yhdistetty ja avaus tapahtuu jousen avustuksella, jotta toiminta olisi nopeaa. Lisäksi kuormanerotimet on varustettu valokaaren sammutuksen mahdollistavilla osilla. Kuormanerotimet saattavat olla SF₆-kaasueristeisessä tilassa, joka pienentää valokaaren sammumismatkaa.

Varokekuormanerotin taas on toiminnaltaan kuten kuormanerotinkin, mutta lisäksi se on varustettu sulakkeilla, joissa on iskuri sisällä. Liian suuri ylivirta polttaa sulakkeen ja laukaisee iskurin, joka avaa koko varokekuormanerotin ja näin

katkaisee ylikuormituksen tai oikosulun pois syöttösuuntaa kuormittamasta. Kuvassa 1 on vanhassa betonikennokojeistossa sijaitseva varokekuormanerotin. Vaaleanruskeat muoviset osat erottimen edessä ovat sammutuskammiot, jotka sammuttavat valokaaren turvallisesti.



Kuva 1. 24 kV varokekuormanerotin vanhassa betonikennokojeistossa

Maadoituserotin toimii kuten kuormanerotinkin, eli yleensä jousivoimaisesti, mutta se maadoittaa siihen liitetyn sähkölaitteiston kaikki vaiheisesti. Näin saadaan työskentely osien läheisyydessä turvalliseksi ja estetään jännitteen vahinkokytkeä laitteistoon, koska maadoitus aiheuttaisi suoran oikosulun ja jokin verkon suojalaite toimisi. (2. ABB:n TTT-käsikirja. Kytkinlaitteet ja eristimet.)

Erottimien huollossa mm. tarkistetaan laite silmämääräisesti, rasvataan mekanismit ja kosketuspinnat, puhdistetaan eristeet, tarkistetaan kaasunpaineet, testataan laukaisujen ja ulkoisten laukaisukelojen toiminta, ohjataan laitetta ja vaihdetaan mahdollisesti rikkoontuneet osat.

2.1.2 Katkaisijat

Katkaisijat ovat yleensä suojareleohjattuja sähkönjakeluverkon komponentteja, joilla on hyvä virrankatkaisukyky myös oikosulkutilanteissa. Keski- ja suurjännite katkaisijoissa vanhemmat katkaisijat ovat usein vähäöljykatkaisijoita, joissa öljy toimii eristeenä katkaisutilassa ja uudemmat tyhjiö- tai SF₆-eristeisiä. Pienjännitekatkaisijat ovat ilmaeristeisiä, koska pienellä jännitteellä valokaari sammuu helpommin. Katkaisijat ovat joko kiinteästi asennettuja tai vaunukatkaisijoita, jotka toimivat myös erottimina, kun katkaisija voidaan auki-tilassa irrottaa jännitteisistä osista täysin molemminpuolisesti. (3. Elovaara ja Haarla. Sähköverkot 2. s. 162–163)

Myös katkaisijoiden ohjauslaitteet toimivat yleensä jousitoimisesti ja jousi virittää joko käsin tai moottorilla. Katkaisijoissa on joko painonapit, ohjauskelat tai molemmat katkaisijan auki- ja kiinniohjaamiseen, sekä laukaisukela suojaukselle. Alla olevassa kuvassa 2 on Strömbergin valmistama vähäöljykatkaisija ohjaimen kansi avattuna.



Kuva 2. Strömbergin vähäöljykatkaisija ohjaimen kansi avattuna.

Katkaisijoiden koestamisessa tarkastetaan laite silmämääräisesti, rasvataan ohjaimen liikkuvat osat, puhdistetaan eristeet, tarkastetaan kaasunpaineet tai öljynpinta, testataan laukaisujen ja ulkoisten laukaisukelojen toiminta, sekä ohjataan laitetta. Lisäksi saatetaan mitata ylimenovastukset kaikista vaiheista, toimintaajat auki- ja kiinniohjauksessa, kelojen toimintajännitteet, yms.

2.1.3 Suojareleet

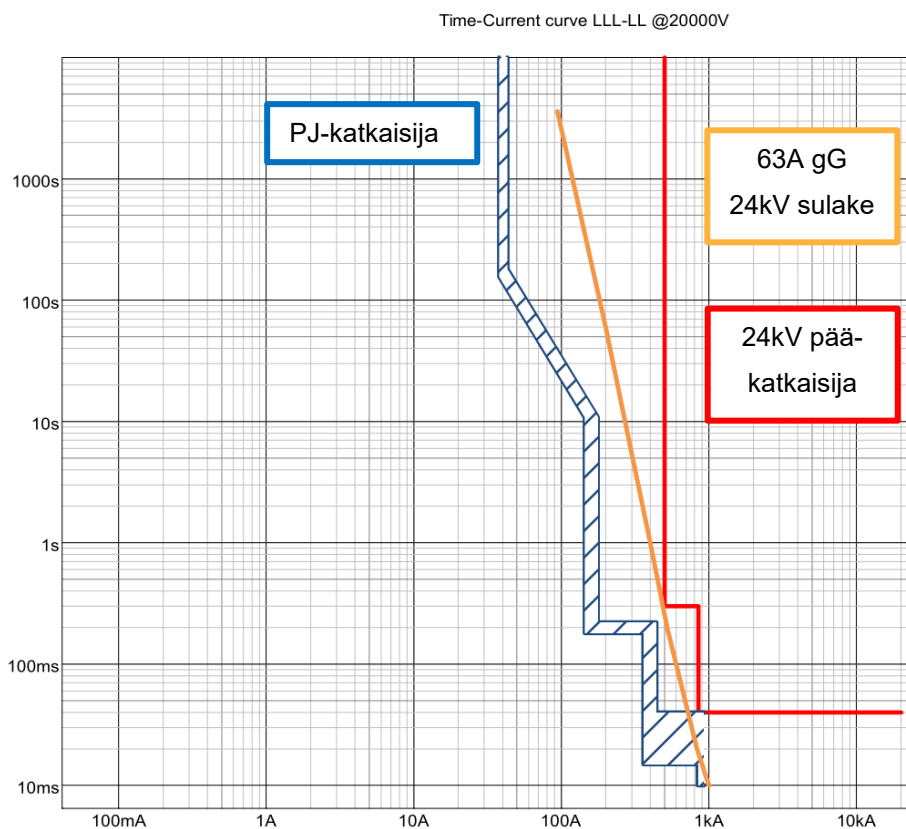
Suojareleitä on useille erilaisille sähkönjakelun suojauksille. Esimerkiksi yli- ja alijännitteille; yli-, ali- ja erovirroille; yli- ja alitaajuuksille; maasulkuvirroille ja -jännitteille ja valokaarisuojaukselle. Suojareleiden mittaukset tulevat yleensä mittamuuntajien kautta, jolla saadaan suojareleissä käytetyt mitattavat suureet tasolle, jota elektroniikka kestää. Vanhoja suojareleitä on myös täysin sähkömagneettisina oikosulkuvirran aiheuttamasta magneettikentästä tai ylivirrasta lämpenevinä mekaanisesti toimivina keskijännitepuolellakin. (3. Elovaara ja Haarla. Sähköverkot 2. s. 342–344)

Suojareleiden koestuksessa täytyy kaikki releessä käytössä olevat suojausportaat koestaa esimerkiksi Megger SVERKER -reletesterillä tai vastaavalla. Kaikki mittaustulokset kirjataan ylös ja verrataan, että nämä sopivat aseteltuihin arvoihin ja releen laukaisukärjet toimivat oikein. Kuvassa 3 on ABB:n REF615-kennotermiinaali, jota ohjelmoidaan kuvaushetkellä etuportin Ethernet-liitännän kautta tietokoneella PCM600-ohjelmistolla.



Kuva 3. ABB REF615 -kennotermiäli.

Suojareleissa varsinkin ylivirtaportaissa on myös tärkeää tarkastella asetteluportaiden selektiivisyyttä. Selektiivisyydellä varmistetaan se, että vikatilanteessa ensin laukeaa alemmat vikaantuneet haarat ja vasta viimeisenä syöttöä suojaava komponentti. Kuvassa 4 on kuvattu selektiivisyyden tarkastelun kuvaajat 20 000 / 400 V muuntajan molemmin puolin.



Kuva 4. 20 kV jännitteellä laukaisuaikojen kuvaajat suhteessa virtaan.

Kuvan sininen, eli vasemmanpuoleinen käyrä, on PJ-katkaisija, jonka ensimmäinen porraskäyrä on käänteisaikainen ja kaksi seuraavaa porrasta ovat suora-aikaisia. Oranssilla, eli keskellä on 63 A 24 kV keskijännitesulake, joka on muuntajaa syöttävässä lähdössä suojalaitteena. Punainen oikeanpuoleinen käyrä on keskijännitekojeiston pääkatkaisija.

2.1.4 Kompensointi

Kompensointilaitteistoilla kompensoidaan asiakkaan omistamasta PJ tai KJ verkosta yleensä loistehoa pois kytkemällä kondensaattoriparistosta portaita säätielmille annettujen parametrien mukaisesti, jotta sähköä toimittava yhtiö ei laskuta loistehomaksuista huomattavasti pätötehon kulutusta suurempaa hintaa. Kompensointi saattaa joskus olla myös yliaaltoja varten.

Kompensointien koestuksessa täytyy kaikkien kondensaattoreiden kunto mitata, portaita ohjaavien kontaktorien toiminta testata ja tarkistaa, että sulakkeita ei ole palanut. Laitteisto tarkastetaan silmämääräisesti lämpenemien ja hajonneiden komponenttien varalta, sekä ilmanvaihdon tuulettimien, termostaattien ja suodatimien kunto tarkastetaan.

2.1.5 Maadoitukset ja eristykset

Maadoitusten koestamisessa mitataan maaperän resistanssia päämaadoituskiskolta tai jostain muusta maadoitusverkon osasta lähtien, jotta suojauslaitteet toimisivat maasuluissa tarpeeksi tehokkaasti. Lisäksi mitataan maadoitusten jatkuvuutta maadoituskiskolta lähtevistä maadoituksista niiden toisiin päihin, jotta laitteiden runkoihin ei pääsisi muodostumaan vaarallisia jännitteitä, vaan suojauslaitteet toimisivat heti, jos laitteen runkoon on yhteydessä jännitteinen osa. (4. SFS 6001:2018. 10 Maadoitusjärjestelmän suunnittelu. s. 87–89)

Eristysvastuksia mitataan esimerkiksi keskuksista, kojeistoista, kaapeleista, muuntajista ja kiskosilloista, jotta niiden eristysten eheys pystytään varmistamaan. Näin varmistumaan siitä, että jännitteet eivät pääse johtimien ulkopuolelle ja aiheuta vaaraa tai tuhoa muita laitteita.

2.2 Raportointi

Suoritettavat mittaus-, huolto- ja tarkastustoimenpiteet tulee kaikki raportoida hyvin, jotta asiakkaalle voidaan lähettää koestuspöytäkirjat kaikista tehdyistä toimenpiteistä. Näin pystytään myös seuraamaan poikkeamia määräaikaisten huoltojen välillä ja puuttumaan niihin mahdollisesti ennen laitteen täydellistä hajoamista tai viimeistään siinä vaiheessa.

Hyvät raportit myös auttavat suunnittelemaan kohteen määräaikaiskoestuksia ja laiterikon sattuessa niistä saadaan lähes kaikki tarvittavat tiedot uuden laitteen suojausten ja mitoitusten valinnassa. Laitteiden ja laitteistojen uudistuksia on myös helpompi suunnitella ja perustella kiinteistön omistajille, kun niiden kunnosta ja toiminnasta on kattavat dokumentit.

Pöytäkirjoista tulee käydä ilmi kaikki kohteen tiedot, laitteen tiedot, suojalaitteiden asetteluarvot, nimellisvirrat ja -jännitteet, sekä näiden testauksessa käytetyt virrat ja ajat. Lisäksi pöytäkirjassa täytyy olla käytetyt koestuslaitteet, koestajat ja päivämäärä.

3 VBA-KOODAUKSEN PERUSTEET

3.1 VBA-koodin kehitys

VBA-koodi pohjautuu Microsoftin Visual Basic -ohjelmointikieleen. VBA:n ensimmäinen version on julkaistu jo vuonna 1993 ja siitä on päivitetty uusia versioita tasaisesti uusien Office-pakettien mukana. Viimeisin versio Visual Basic 6:sta, johon VBA pohjautuu, on julkaistu jo 1998 ja tuki ohjelmointikielelle on loppunut täysin vuonna 2008. VBA-koodia ei suoriteta yksinään, vaan se suoritetaan pääasiassa Microsoft Excel-ohjelmiston sisältä. Tällä hetkellä uusin versio on VBA 7.1, jota myös työssä käytetään. VBA 7.1 sisältyy Office 2013 versiosta aina kirjoitushetkellä uusimpaan Office 2019 versioon. (5. Jussi Merensalmi. Excel VBA yrityskäytössä.)

VBA:n avulla on mahdollista automatisoida erittäin laajasti Microsoft Officen käyttöä ja toimintoja. Sen käyttökohteet ovatkin pääasiassa yritysmaailmassa, jossa pyritään pääsemään eroon itseään toistavista tehtävistä ja tehostamaan työn tekoa. VBA-koodia ei suoriteta .XLSX-tiedostoissa, vaan Excel-tiedostoissa tiedostopäätteellä .XLSM. Oletuksena Excel myös kysyy varmistusta makrokoodien suorittamiseen tietoturvan vuoksi. Makrokoodilla on mahdollista toteuttaa myös haitallista koodia ja viruksia, joten tuntemattomien tiedostojen makrokoodia ei tulisi koskaan sallia. (6. Avast. Macro Virus.)

3.2 Koodin rakenne

VBA-koodin rakenne muodostuu moduuleista koodin sisällä ja näitä moduuleita voidaan taas kutsua toisten moduuleiden sisältä tai suorittaa yksinään. Ohjelman runko kirjoitetaan Sub-komennon jälkeen ja runko suljetaan lopussa End Sub-komennolla. Private komennolla estetään koodin kutsumista kyseisen sivun tai moduulin ulkopuolelta. (7. Microsoft. Sub statement)

Kuvassa 5 on esimerkki ohjelman luomisesta. Ohjelma täyttää painiketta, joka on nimetty btnPvmTanaan, painettaessa soluun T33 nykyisen päivämäärän.

```
Private Sub btnPvmTanaan_Click()
    Range("T33").Value = Date
End Sub
```

Kuva 5. Ohjelma painikkeen takana, joka tuo nykyisen päivämäärän soluun T33.

Koodissa voidaan määritellä erilaisia muuttujia, jotka voivat sisältää esimerkiksi pelkkiä kokonaislukuja, numeroita desimaaleilla, kirjaimia tai yhden bittitiedon 0 tai 1. Nämä kaikki määritellään komennolla Dim "muistipaikan nimi" As "Tyyppi", eli esimerkiksi Dim kokonaisluku As Integer. Erilaisia datatyyppejä on yli kymmenen ja myös käyttäjän määrittämät datatyypit ovat mahdollisia. (8. Microsoft. Data type summary)

Koodissa voidaan myös käyttää For- ja While-ohjelmakiertoja, jotka käyvät läpi esimerkiksi ison alueen kaikki tiedot. Nämä voidaan yhdistää If-komentoihin, jolloin saadaan tarkistettua vastaako tietyn alueen mikään datapiste annettuja ehtoja ja tarvittaessa vaikka muokattua kaikkia, jotka vastaavat kyseistä/kyseisiä ehtoja. Esimerkkinä kuvassa 6 ohjelma, joka kutsuttaessa etsii kaikki tietyllä nimellä nimetyt kuvat ja muodot avoimesta sivusta ja poistaa ne.

```
Private Sub PoistaKuvat()
Dim shp As Shape
    For Each shp In ActiveSheet.Shapes
        If shp.Name = "Logo" Or shp.Name = "Allekirjoitus" Then
            shp.Delete
        End If
    Next shp
End Sub
```

Kuva 6. For-kierrolla toteutettu ohjelma, joka poistaa tietyn nimiset kuvat sivulta.

VBA-koodia pystyy myös nauhoittamaan makrotyökalulla, joka tallentaa koodina kaiken mitä Excel-työkirjassa tehdään sen jälkeen, kun nauhoitus on käynnistetty. Tämän työkalun avulla on helppo etsiä toimivia kaavoja ja saada suuntaa miten toimiva koodi tulisi kirjoittaa. Makrolla nauhoitetut koodit eivät ole tehokkaita, mutta ne ovat useimmiten kuitenkin suoraan toimivia ja pienellä viimeistelyllä niistä saadaan erittäin hyviä.

4 PÖYTÄKIRJATYÖKALUN RAKENTAMINEN

4.1 Vaatimusten suunnittelu

Työn toteuttaminen hyvin vaatii hyvän pohjatyön kaikista toivotuista ominaisuuksista. Ensin asiat jaettiin kolmeen kategoriaan, joista ensimmäisessä oli ominaisuudet, jotka ovat lähes pakollisia tehokkaassa työkalussa. Esimerkiksi uusien tiedostojen nimeäminen täysin oikein yrityksen nimeämiskäytäntöjen mukaisesti syötettyjen projektin tietojen pohjalta. Lisäksi kaikki pöytäkirjan luominen ja täyttäminen tapahtuisi yhdestä välilehdestä, jotta käyttäminen olisi tehokasta ja yksinkertaista. Eri laitteiden koestuspöytäkirjojen täytyy myös olla muotoilultaan yhteneviä, jotta raportoinnista välittyy laadukas kuva asiakkaalle.

Toisena olivat ominaisuudet ja muotoilu, joka olisi hyvä sisällyttää toteutukseen, mutta jotka eivät ole täysin pakollisia. Tähän kategoriaan kuuluu esimerkiksi kielen vaihto englanniksi pöytäkirjoissa ja logon tuonti vain yhdestä paikasta tulevaisuuden muutosten varalta.

Kolmantena olivat ominaisuudet, jotka eivät ole olennaisia, mutta olisivat toivottava lisä, jos niitä on mahdollista toteuttaa ilman älyttömän suurta lisätyötä tai vaarantamatta muita ominaisuuksia ja niiden toimintaa. Eräs ominaisuuksista tässä kategoriassa oli mobiililaitteissa käytettävä ohjelmisto, mutta tämä ei ole mahdollista, koska VBA-koodia ei voida suorittaa Android- tai IOS-laitteilla. (9. Microsoft. Miksi Excel-tiedostoa ei voi avata?)

4.2 Toteuttaminen

Toteuttaminen aloitettiin etsimällä internetistä ja kirjallisuudesta ensin vinkkejä, että mitä kaikkea VBA-koodilla voidaan tehdä ja toteuttaa. Lisäksi selvitettiin, että millaisia asioita sillä ei kannattaisi tavoitella, jotta muut ominaisuudet eivät kärsi. Microsoft ylläpitää VBA-koodauksen ohjeistusta sivuillaan ja tätä tietokantaa kannattaakin hyödyntää koodia kirjoitettaessa.

4.2.1 Taustatyöt

Ensimmäisenä työkirjaan täytyi luoda kohdat, joihin voidaan täyttää tiedoston nimeämisessä ja kohteiden tunnistuksessa tarvittavat kohdat. Näistä tiedot tallennetaan apuohjelmalla VBA-koodilla muistipaikkoihin ja hyödynnetään muita toimintoja suoritettaessa. Tarvittavia kohtia olivat muun muassa työnnumero, asiakas, kohde, osoite, muuntamo ja laitteen yksilöivät tiedot. Nämä kaikki tallennettiin VBA-koodissa String-muuttujina, eli tekstiä sisältävinä tietueina.

Työkirjaan tarvittiin myös ActiveX-komponentteja, eli painikkeita, joita hiirellä painamalla saadaan suoritettua koodia, joka on painikkeen taakse luotu. Painikkeita saadaan Excelissä luotua kehitystyökaluista, jota kautta saadaan myös tallennettua makroja. Työkalun muutoksien ja päivitysten, sekä versiohistorian seurannan takia täytyi luoda välilehti, jolle kirjataan päivämääräileimalla mitä päivityksiä ja ominaisuuksia on kulloinkin lisätty tai muokattu.

Kaikki käyttäjän tarvitsemat ominaisuudet luotiin yhdelle välilehdelle, jotta tulevaisuudessa loppukäyttäjän ei tarvitsisi työkalua käyttäessä vaihdella välilehtiä, vaan toiminta suoritettaisiin tehokkaasti yhdeltä sivulta alusta loppuun. Ensimmäinen koodattu ominaisuus oli uuden tiedoston luominen ohjelmakoodin avulla, kun käyttäjä painaa painiketta ”Luo uusi koestuspöytäkirja”. Tämän ominaisuuden toteuttamisessa tarvittiin apuna makron nauhoitusta ja tällä saatua koodia siistittiin.

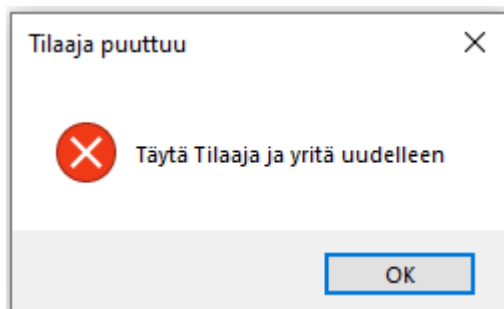
Luotavan tiedoston nimeäminen täytyi tapahtua aiemmin täytettyjen tietojen pohjalta ja huomioida tai varoittaa puuttuvista tiedoista. Osa vaadituista tiedoista ovat sellaisia, että jos ne puuttuvat, niin uutta tiedostoa ei saa luoda. Tällainen tieto on esimerkiksi tilaaja. Tilaaja-kohdan tarkistuksen ohjelmakoodi on alla kuvassa 7. Suoritettava koodi luo kuvan 8 mukaisen varoituksen, joka myös keskeyttää uuden tiedoston luomisen.

```

Dim Tilaaja As String:      Tilaaja = Range("T4").Value
If Tilaaja <> Empty Then
    Tiedostonimi = Tiedostonimi + " " + Tilaaja
Else
    MsgBox "Täytä Tilaaja ja yritä uudelleen", vbCritical, "Tilaaja puuttuu"
    Tiedostonimi = Empty
    Exit Sub
End If

```

Kuva 7. Tilaajan hakeminen ja varoitus, jos tieto puuttuu.



Kuva 8. Varoitus puuttuvasta tilaajan tiedosta.

Tiedostopäätteeksi valittiin oletuksena aina .xlsx, koska uusien luotavien asiakirjojen ei tarvitse itse suorittaa yhtään VBA-koodia, johon vaadittaisiin .xlsm -tiedostopäätte. Lopulta koodattu ohjelma määritettiin suoritettavaksi aina painiketta painettaessa.

4.2.2 Tiedostopolun valinta

Uusien tiedostojen luonti tapahtuu pääasiassa palvelimelle ja vähintäänkin eri projektien tiedostot luodaan eri sijainteihin, joten työkaluun täytyi pystyä myös määrittämään tiedostosijainti ennen tiedoston luontia. VBA-koodilla pystyy onneksi avaamaan suoraan Windowsin resurssienhallinnan, jonka voi määrittää palauttamaan tietona käyttäjän valitseman kansion nimen ja polun. (10. Encodedna. How to open a File Dialog box in Excel using VBA)

Resurssienhallinnalle piti antaa parametrinä msoFileDialogFolderPicker, jolloin palautetaan käyttäjän valitseman kansion polku, eikä esimerkiksi tiedostoa. Resurssienhallinnalle voidaan myös määrittää kansio mihin ohjelma aina avautuu vakiona, jotta käyttäjälle kansion valinta on tehokasta. Valitun kansion tiedosto-

polku ja nimi tallennettiin muistipaikkaan String-muodossa. Käyttäjän virheitä estämään luotiin ominaisuuksia, jotka estävät väärät valinnat ja aiheuttavat virheilmoituksen, jos käyttäjä valitsee jotain, mikä ei käy tallennuspoluksi. Myös resursienhallinnan sulkeminen valitsematta kansiota antaa huomautuksen, että käyttäjä ei valinnut kansiota, jotta työkalu olisi mahdollisimman informatiivinen. Saatu tiedostopolku asetettiin tallennuspaikaksi tiedostonluontitoiminnolle ja toiminnan vakautta testattiin useita kertoja mahdollisten käyttäjän virheiden varalta.

4.2.3 Tietojen kopioiminen uuteen tiedostoon

Työkalua käyttäen oli nyt mahdollista luoda oikein nimetty tiedosto oikeaan tiedostosijaintiin, mutta tiedosto luotiin täysin tyhjänä. Tietojen kopiointiin täytyi ensin kokeilla ja tutkia mahdollisia vaihtoehtoja. Vaihtoehtoista järkeviä oli lopulta kaksi, eli joko koko välilehden kopiointi uuteen työkirjaan tai osan kopioiminen uuteen työkirjaan.

Koko välilehden kopioimisen etuina on erittäin yksinkertainen koodi, jolla saa siirrettyä kerralla kaiken muotoilun, tulostusalueen, tekstin, kaavat ja kuvat ilman mitään ongelmia. Tämä ei kuitenkaan ollut toimiva tapa, koska näin ei ole mahdollista luoda yhdestä välilehdestä käytettävää työkirjaa, joka oli helpon ja tehokkaan käytettävyyden edellytys.

Ratkaisuksi jää siis kopioida kaikki halutut tiedot, kuvat ja muotoilut erikseen koodirivi kerrallaan. Pelkästään halutun alueen valitsemalla ja kopioimalla uuteen työkirjaan jää kopioimatta muun muassa osa kuvista, tulostusalue, rivinleveydet ja sarakeleveydet. PasteSpecial-komennolla saadaan käytettyä liittämisen lisäasetuksia datan kopioimisen apuna. Komennon perään määritellään, että mitä asetteluja tai muotoiluja halutaan tuoda kopioinnin mukana. Alla olevassa kuvassa 9 on käytetty kopiointia, joka tuo mukanaan myös sarakkeiden leveydet. (11. Automate Excel. VBA Value Paste & PasteSpecial.)

```
ActiveSheet.Paste
Selection.PasteSpecial Paste:=xlPasteColumnWidths, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False 'Liittää sarakkeiden leveydet'
```

Kuva 9. Tietojen kopiointi sarakkelevyysien kanssa.

Ensimmäinen rivi liittää vain kaikki kopioidut tiedot ja PasteSpecial-komentoon on määriteltä "Paste:=xlPasteColumnWidths", joka kopioi myös sarakkeiden leveydet. Vihreällä tekstillä olevat ovat kommentteja selkeyttämään koodia ja jotta muutkin käyttäjät ymmärtävät myöhemmin, että mitä eri osat koodia tekevät.

Rivileveyksiä ei pysty kopioimaan millään vastaavalla komennolla, joten niiden tiedot täytyi tallentaa ensin kopioitavasta paikasta VBA-koodilla listamuotoiseen muuttujaan käyttäen For-kiertoa apuna. Tallennetut rivilevydet kopioidaan muuttujasta toista For-kiertoa apuna käyttäen uuteen työkirjaan muokattavalle sivulle. Liittämisen osalta kopiointia on havainnollistettu alla olevassa kuvassa.

```
'Rivin leveyksien muokkaus uudessa asiakirjassa
For r = 1 To Worksheets(1).Range("A1:P200").Rows.Count
    Worksheets(1).Range("A1:P200").Rows(r).RowHeight = Leveydet(r)
Next r
```

Kuva 10. Rivileveyksien kopiointi listamuotoisesta muuttujasta For-kierrolla.

Yllä olevan kuvan 10 koodi käy läpi kaikki rivit A1:P200 välisellä alueella ja muuttaa niiden kaikkien korkeutta vastaavalla listamuodossa olevan muuttujan arvolla. Kopiointi on nyt lähes täydellisesti alkuperäistä vastaava, mutta jos yrittää tulostaa uuden asiakirjan puuttuvat kaikki tulostusasetukset ja tulostus ei näytä haluttunlaiselta. Tulostus saattaa myös tulla useille sivuille, vaikka suurin osa pöytäkirjoista tulisi tulostua yksisivuisena.

Tulostusalueen muokkaus tapahtuu komennolla PageSetup, jonka alle voi määrittää 48 eri tulostusasetusta. With-komentoa apuna käyttäen voidaan tulostusasetusten kopioinnissa jättää pois viittaus joka riviltä erikseen haluttuun sivuun ja komentoon ja näin selkeyttää koodin luettavuutta huomattavasti. (12. Microsoft. PageSetup.)

```
With wsLiitto.PageSetup
```

```
.AlignMarginsHeaderFooter = wsKopio.PageSetup.AlignMarginsHeaderFooter
.BlackAndWhite = wsKopio.PageSetup.BlackAndWhite
.BottomMargin = wsKopio.PageSetup.BottomMargin
```

Kuva 11. Tulostusasetusten kopiointi uuteen työkirjaan.

Kuvan 11 koodilla siirretään uuteen työkirjaan tieto, että kohdistetaanko reunusten mukaan ala- ja ylätunnisteet, tulostetaanko sivu mustavalkoisena ja alareunuksen korkeus. Käytetyssä koodissa siirrettiin huomattavasti enemmän tietoja näiden lisäksi. Ongelmaksi muodostui tulostusalueen siirto, koska suomenkielisessä Excel-ympäristössä käytetään koodierottimenä puolipistettä ja englanninkielisessä pilkkua. Merkkien ero aiheuttaa ongelmia useissa kohdissa ohjelmointia, mutta tässä ongelma oli haastava ja syytä oli erittäin hankala löytää.

Ratkaisuna virheisiin oli vaihtaa ohjelmallisesti puolipisteet pilkuiksi, jolloin tiedot kopioituivat oikein. Tämä muunnos täytyi tehdä aina uudelleen, koska vaikka muunnetut tiedot kopioitaisiin heti uudelleen, niin käytössä olisivat taas puolipisteet, jotka aiheuttivat uudet virheilmoitukset ja estivät ohjelmaa toimimasta. Merkkien muunnoksen jälkeen tulostusalue siirtyi ongelmitta.

4.2.4 Hallinnointisivun kehittäminen

Uuden asiakirjan luomisen kaikki ominaisuudet oli nyt saatu koodattua, mutta hallinnointisivulta puuttuivat kaikki toiminnot ja asiakirjapohjia ei ollut käytössä kuin yksi, jota oli käytetty testaamisen aikana. Seuraavaksi täytyi kehittää hallinnointisivun graafista ulkonäköä, jotta käyttäminen olisi selkeää myös ohjelman uudelle käyttäjälle.

Tulostusalue rajattiin harmaiden palkkien sisälle, jotta myös A-sarakkeessa olevat Excel-ikkunan vasenta reunaa vasten olevat reunaviivat olisivat näkyvissä ja niiden puutoksen huomaisi jo ennen tulostamista. Lisäksi tämä alue olisi kopioitava alue, johon myös muokattaisiin huolto- ja koestusmerkinnät ennen kopiointia uuteen työkirjaan ja tarkastettaisiin ulkonäkö.

Alueen sivuun siirrettiin kaikki tietokentät, joita tarvittiin tiedoston nimeämiseen ja painikkeet, joilla suoritettiin VBA-koodia. Lisäksi tarvittiin laitelistaukset, joiden avulla voidaan valita koestuspöytäkirjan tyyppi ja laitteet eri valmistajien joukosta. Laitelistauksissa käytettiin kolmiulotteisia valikoita, eli ensimmäisen pudotusvalikon valinta vaikuttaa seuraavan pudotusvalikon sisältöön ja sen valikon valinnat taas vaikuttavat kolmanteen valikkoon. (13. Svetlana Chesheva.)



Kuva 12. Koestuspöytäkirjatyökalun välilehtirakenne.

Yllä olevassa kuvassa 12 näkyvään luettelot-välilehteen luotiin yksi valikko, joka sisältää kaikki pöytäkirjatyytit ja tämän valikon alle luotiin jokaisen pöytäkirjatyytin eri laitevalmistajat. Laitevalmistajien alle luotiin vielä kaikkiin erikseen luettelot, jotka sisältävät jokaisen laitevalmistajan kyseisen tyyppisen laitteen. Pudotusvalikon sisältö taas asetettiin viittaamaan luotujen taulukoiden nimiin. Toinen ja kolmas valikko saivat viittauksensa nimen edellisen tai edellisten pudotusvalikon valintojen mukaan.

Nykyisen tiedostopolun näyttämiseksi ja luotavan tiedoston nimen esikatselulle luotiin kentät, joihin tiedot täydentyvät automaattisesti. Näin käyttäjä on aina tietoinen uuden asiakirjan luomista, että mihin kyseinen asiakirja tallennetaan ja millä nimellä.

Pöytäkirjan sisältämien tietojen, jotka ovat myös tiedostonnimeä varten jo täydennettyinä, kopiointia varten luotiin uusi painike, joka täyttää nämä kohdetiedot. Täytettävien kohdetietojen näkymä on kuvassa 13 alla.

Täytä kohdetiedot	
Asiakas: Malli Oy	=JOSVIRHE(PHAKU(STS4; Asiakkaat; 2; EPÄTOSI); STS4 & "Oy")
Paikka: Testi	=SUS4
Osoite: Mannerheimintie 30	=SVS4
Postinro: 00100	https://www.google.fi/search?q=Mannerheimintie 30+Helsinki
Paikkakunta: Helsinki	=SWS4
Muuntamo: 0	=XS4
Keskus: RK	=SYS4
Kenno: 01	=JOS(PITUUS(SZS4)=3; VASEN(SZS4; 2) & "." & OIKEA(SZS4); SZS4)
Tunnus: Q1	
Toiminta: Pääkytkin	Pääkatkaisija tms.
Päivä: 1.1.2021	PVM Tänään
Koestanut: Kimmo Mäki	<input type="checkbox"/> Allekirjoitus puolesta
2. Koestaja:	

Kuva 13. Täytettävät kohdetiedot.

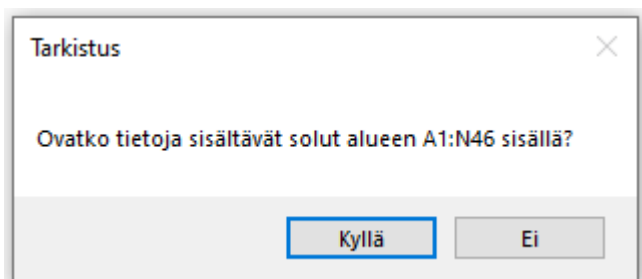
Kohdetiedot punapilkullisiin ruutuihin haetaan automaattisesti jo täytetyistä tiedoista, ja "Asiakas"-kohtaan täydennetään yrityksen koko nimi, jos lyhenne löytyy luettelosta tai muuten täydennetään perään Oy. Keltaiset kohdat vaativat muutoksia käyttäjältä ja postinumeron tarkistusta varten sen viereen on luotu automaattisesti muodostuva linkki Google-hakuun osoitteen ja paikkakunnan pohjalta, jolla postinumeron saa haettua suoraan.

4.2.5 Pöytäkirjapohjien lisäys

Kaikki muut pakolliset toiminnot olivat valmiina, mutta työkalussa oli edelleen käytössä vain yksi koestuspöytäkirja. Uusille sivuille täytyi määrittää perusasetukset kuntoon, kuten tietosolujen sijainnit, tulostusalue ja vastaavat. Tämän prosessin nopeuttamiseksi työkirjaan tehtiin kaksi makroa, joilla saatiin nopeasti muokattua perusominaisuudet kuntoon. Makroja käyttäjä voi suorittaa ALT+F8 näppäinyhdistelmällä ja valita avautuvasta luettelosta yhden työkirjaan sisällytetyistä makroista.

Makrojen käyttöä varten käyttäjille luotiin ohjeet käyttämällä solujen sanomatoimintaa, jolloin yhteen pieneen soluun saa sisällytettyä melko pitkänkin ohjeen,

joka avautuu klikkaamalla solua. Sanoman lisääminen tapahtuu Datatyökaluista tietoisuuden kelpoisuuden tarkastamisen alta. Ensimmäinen ajettava makro kysyy käyttäjältä, että ovatko tietoja sisältävät solut alueen sisällä, johon makro lisää tulostusalueen. Kuvassa 14 alla on esitetty käyttäjälle näytettävä kysymys.



Kuva 14. Uuden asiakirjan lisäämisen apumakron tarkastuskysymys.

Jos käyttäjä hyväksyy yllä olevan ilmoituksen, niin makro poistaa asiakirjasta kaikki kuvat, lisää harmaat reunat alueen ympärille ja lisää tulostusalueen sen sisälle. Lisäksi asiakirjan kaikkien solujen tekstin fontti muutetaan muissakin käytettyyn fonttiin. Tulostusalueen sivuun lisätään yksi punaiseksi värjätty solu, johon tulee sanomana lisäohjeet asioista, jotka käyttäjän täytyy tarkistaa ennen seuraavan makron suorittamista.

Toinen makro lisää tekstit ja objektit kaikkiin kohtiin, joita käyttäjä voisi täydentää automaattisesti ja kysyy käyttäjältä, että tulivatko kaikki täytetyt tiedot oikeisiin kohtiin. Jos käyttäjä vastaa "kyllä", niin kaikki testausta varten täytetyt tiedot poistuvat. Seuraavaksi makro kysyy käyttäjältä, että mikä on tietokentän viimeinen rivi ja muokkaa kaikki rivileveydet tämän mukaan oikeaksi. Lopuksi avataan tulostuksen esikatselu, josta käyttäjä näkee suoraan, että tulostuisiko uusi asiakirja oikein ja yhdelle sivulle.

Makrotyökaluja apuna käyttäen koestuspöytäkirjatyökaluun lisättiin n. 20 erilaista koestuspöytäkirjaa vanhojen pöytäkirjojen pohjalta ja muotoiltiin ne ulkonäöltään täysin työkaluun sopiviksi. Uusia pohjia työkaluun lisätään myöhemmin, kunhan sen käytöstä saadaan riittävästi dataa ja ilmenevät puutteet ja ongelmat ratkaistua, jotta ne eivät monistuisi liian monelle välilehdelle.

4.2.6 Pöytäkirjapohjien laskentakaavat ja laitetiedot

Lisättyjen pöytäkirjojen tietoja ja täytettäviä kenttiä tarkasteltiin laitevalmistajien ohjekirjoista ja verrattiin niiden oikeellisuutta. Eri valmistajien suojareleiden parametrit on nimetty eri lailla ja monipuolisemmissa releissä on useita eri parametrejä esimerkiksi lämpenemän ja sen kertymisen tai jäähtymisen asetteluille, joiden nimeämiset täytyy tarkistaa suojareleiden manuaaleista. Laitevalmistajien suojareleiden toiminnat ja varsinkin niiden laukaisukäyrät eivät ole keskenään täysin vertailukelpoisia, vaan suojauksen toiminta-aikoja tietyillä virroilla lasketaan hieman eri kaavoilla.

ABB | SACE Emax 2

Summary table of protections

ABB	ANSI ⁽⁵⁾	Threshold ⁽¹⁾	Threshold tolerance ⁽²⁾	Time ⁽¹⁾	Calculation formula t_1 ⁽²⁾	Calculation example t_1 ⁽²⁾	Tolerance t_1 ⁽³⁾
L	49	$I1 = 0.4...1 I_n$	activation for I_f in the range (1.05...1.2) x $I1$	$t1 = 3...144$ s	$t_1 = (9 t1) / (I_f / I1)^2$	$t_1 = 6.75$ s with: $I1 = 0.4 I_n$; $t1 = 3$ s; $I_f = 0.8 I_n$	$\pm 10\%$ with $I_f \leq 6 I_n$ $\pm 20\%$ with $I_f > 6 I_n$
S ($t = k$)	50 TD	$I2 = 0.6...10 I_n$	$\pm 7\%$ with $I_f \leq 6 I_n$ $\pm 10\%$ with $I_f > 6 I_n$	$t2 = 0.1...0.8$ s	$t_1 = t2$	-	The better of the two values: $\pm 10\%$ or ± 40 ms
S ($t = k / I^2$)	51	$I2 = 0.6...10 I_n$	$\pm 7\%$ with $I_f \leq 6 I_n$ $\pm 10\%$ with $I_f > 6 I_n$	$t2 = 0.1...0.8$ s	$t_1 = (100 t2) / (I_f)^2$	$t_1 = 5$ s con: $I2 = 1 I_n$; $t2 = 0.8$ s; $I_f = 4 I_n$	$\pm 15\%$ with $I_f \leq 6 I_n$ $\pm 20\%$ with $I_f > 6 I_n$
I	50	$I3 = 1.5...15 I_n$	$\pm 10\%$	Not adjustable	$t_1 \leq 30$ ms	-	-
G ($t = k$)	50N TD	$I4^{(4)} = 0.1...1 I_n$	$\pm 7\%$	$t4 = 0.1...0.8$ s	$t_1 = t4$	-	The better of the two values: $\pm 10\%$ or ± 40 ms
G ($t = k / I^2$)	51N	$I4^{(4)} = 0.1...1 I_n$	$\pm 7\%$	$t4 = 0.1...0.8$ s	$t_1 = 2 / (I_f / I4)^2$	$t_1 = 0.32$ s with: $I4 = 0.8 I_n$; $t4 = 0.2$ s; $I_f = 2 I_n$	$\pm 15\%$
Inst	-	Defined by ABB	-	Instantaneous	-	-	-

Kuva 15. ABB SACE Emax2 ylivirtaportaiden toiminta-aikojen kaavat. (14. ABB. SACE Emax2)

Edellisestä kuvasta 15 nähdään ABB:n SACE Emax2 katkaisijatyypissä käytettyjen suojareleiden ylivirtaportaiden toiminnan pohjana olevat laskukaavat ja esimerkit. S-portaassa käyrän tyyppiä voidaan vaihtaa suora- ja käänteisaikaisen välillä. Suora-aikaisen, eli $t = k$, laukaisuaikat ovat aina vakioita ja rele laukaisee virtarajan ylityttyä aina samassa ajassa. Käänteisaikaisella toiminnalla, eli $t = k / I^2$, laukaisuaika nopeutuu mitä korkeampi virta on suhteessa suojausportaan asetteluarvoon.

Ylivirtaportaita on myös useilla erilaisilla laukaisukäyrillä ja erilaisilla standardeilla. Esimerkiksi ABB:n REF615 -suojarleessä on 16 erilaista suojauskäyriä

ylivirtasuojaukselle. ANSI-standardilla käänteisaikaisena Extremely Inverse (EI), Very Inverse (VI), Normal Inverse (NI) ja Moderately Inverse (MI), sekä lisäksi suora-aikainen Define Time (DT). Lisäksi releessä on IEC-standardia noudattavat lähes vastaavat käyrät ja IEEE C37.112-standardilla 7 käyrää. (15. ABB. 615 series Technical Manual. s. 65)

Kaikkea tätä tietoa ei sisällytetä koestuspöytäkirjoihin, mutta niiden täyttäminen on huomioitu dokumenteissa lisäämällä sopivat kohdat. Eri laitevalmistajien pöytäkirjat poikkeavat toisistaan, koska eri valmistajilla on eri merkintätapoja suo-
jauksilleen, vaikka toiminnot pohjautuisivatkin samoihin standardeihin.

5 POHDINTA

Työkalun käyttö on loppukäyttäjälle ohjeiden kanssa melko selkeää ja täysin uusien pöytäkirjojen luonti nopeutuu sitä käyttämällä huomattavasti, joten työkalu on kokonaisuutena erittäin onnistunut. Jos kohteesta on jo vanhat pöytäkirjat edellisestä määräaikaiskoestuksesta, niin silloin nopeampi on todennäköisesti käyttää vanhoja pöytäkirjoja pohjana. Myös uusien pöytäkirjojen lisääminen työkaluun on melko vaivatonta ja nopeaa, kun apuna käyttää makroja. Työkalussa on kuitenkin vielä kehitettävää, mutta kehitystä varten tarvitsee käyttäjiltä tietoja, että mikä on hyvää ja missä on puutteita.

Työkalua voisi kehittää lähes loputtomiin, koska VBA-koodi mahdollistaa erittäin monimutkaistenkin asioiden tekemisen Windows-ympäristössä. Tärkeintä kehitysaluetta on kuitenkin saada työkaluun kaikki mahdolliset käytettävät laitepöytäkirjat, jotta työkalu olisi käyttäjälle mahdollisimman tehokas, kun ei tarvitse välillä etsiä tai muokata pöytäkirjaa tietylle laitteelle sopivaksi. Työkalun ulkonäköä voisi muokata myös vielä visuaalisesti näyttävämmäksi, mutta tärkeintä siinä on selkeys ja helppokäyttöisyys ja visuaalinen ilme ei ole kovinkaan olennainen muuten kuin lopullisissa pöytäkirjoissa.

VBA-koodin tuottaminen ja ominaisuuksien luominen sillä on nyt itselle huomattavasti selkeämpää, kuin ennen aloitusta, mutta edelleen mielestäni ohjelmoijan paras työkalu on käyttää hakukoneita ja ottaa mallia muiden tekemistä koodeista, mutta tehdä ne täysin itse, jotta ymmärrys koodin toiminnasta säilyy. Excel tuntuu olevan insinöörin tärkeimpiä työkaluja, joten sen käyttökokemuksen monipuolistaminen palvelee varmasti hyvin tulevaisuudessa.

LÄHTEET

1. SFS-EN 50160 standardi.
2. ABB. TTT-käsikirja 2000-07. Luku 12: Kytkinlaitteet ja eristimet. Helsinki: ABB.
3. Elovaara, J ja Haarla, L. 2011. Sähköverkot 2 – Verkon suunnittelu, järjestelmät ja laitteet. Helsinki: Otatieto.
4. SFS 6001:2018. Suurjännitesähköasennukset. 5. painos.
5. Merensalmi, J. 2007. Excel VBA yrityskäytössä. 1. painos. Porvoo: WSOY.
6. Microsoft. Miksi Excel-tiedostoa ei voi avata? Luettu 28.10.2021.
<https://support.microsoft.com/fi-fi/office/miksi-excel-tiedostoa-ei-voi-avata-2daa6bd6-8db5-4521-9f37-7e23b3e4bdab>
7. Avast. Macro Virus: What Is It and How to Remove It. Luettu 14.11.2021.
<https://www.avast.com/c-macro-virus>
8. Microsoft Office VBA Reference. Sub statement. Luettu 14.11.2021.
<https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/sub-statement>
9. Microsoft Office VBA Reference. Data type summary. Luettu 14.11.2021.
<https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/data-type-summary>
10. Encodedna. How to open a File Dialog box in Excel using VBA. Luettu 14.11.2021.
<https://www.encodedna.com/excel/how-to-open-a-file-dialog-box-in-excel-using-vba.htm>
11. Automate Excel. VBA Value Paste & PasteSpecial. Luettu 27.11.2021.
<https://www.automateexcel.com/vba/value-paste-pastespecial/>
12. Microsoft Office VBA Reference. PageSetup. Luettu 27.11.2021.
<https://docs.microsoft.com/en-us/office/vba/api/excel.pagesetup>
13. Svetlana Cheusheva. How to make a dynamic dependent dropdown list in Excel an easy way. Luettu 27.11.2021.
<https://www.ablebits.com/office-addins-blog/2020/09/02/create-dynamic-dependent-drop-down-excel/>

- 14.ABB. SACE Emax 2. Low voltage air circuit-breakers.
DOC. N°: 1SDH001330R0002 – ECN000086018. Rev. B.

- 15.ABB. 615 Series Technical Manual.
DOC. N°: 1MRS756887. 02.12.2009. Rev. B.