

Datan kerääminen Siemensin Comfort -paneeleilla

Joona Kataja-aho

Opinnäytetyö
Marraskuu 2015
Tekniikan ja liikenteen ala
Insinööri (AMK), automaatiotekniikan tutkinto-ohjelma

Tekijä(t) Kataja-aho, Joonas	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 08.12.2015
	Sivumäärä 61	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi Datan kerääminen Siemensin Comfort -paneelilla		
Tutkinto-ohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) Häkkinen, Veli-Matti & Kuisma, Ari		
Toimeksiantaja(t) Pekkanen, Jarmo		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli kehittää toimeksiantajalle systeemi, jolla kerätään ja prosessoidaan teollisuusprosesseista saatavaa dataa. Opinnäytetyön tuloksena muodustunutta ohjelmistoa olisi tarkoitus markkinoida prosessiteollisuuden käyttöön. Toimeksiantajana toimi Jyväskyläläinen suunnittelutoimisto JEEC Oy.</p> <p>Työ toteutettiin hyödyntäen kahta WinCC:n Visual Basic -skriptiä sekä yhtä Excel Visual Basic for Applications -skriptiä. Toinen kehitetyistä VBS-skripteistä hoitaa datan keruun ja toinen valvoo tiedostojen kokoa. Excel VBA -skripti kehitettiin ennalta määrättyjen raporttien luontia varten. Koko työ käsitti noin 1500 riviä Visual Basic koodia.</p> <p>Opinnäytetyön tuloksena saatiin kehitettyä tavoitteita vastaava järjestelmä. Työn aikana ei kuitenkaan saatu kehitettyä ohjelmarakennetta, jolla tallennusvälineen täyttyminen hoidetaan automaattisesti. Myös WinCC VBS-ympäristön rajoitusten takia tiedostojen koon seuraamiseen täytyi kehittää kustomoitu ratkaisu.</p> <p>Johtopäätöksenä työstä löydettiin syntaksiltaan yhtenevien Excel VBA ja WinCC VBS ympäristöjen erilaisuus ohjelmakehityksen kannalta. Teoriaosuus käsittelee raportointia teollisuudessa, Windows CE käyttöjärjestelmää, VB, VBS ja VBA ohjelmointikieliä, flash-muisteja sekä CSV-tiedostomuotoa.</p>		
Avainsanat (asiasanat) Tiedonkeräys, Comfort-paneeli, Visual Basic-skripti, raportti, flash-muisti		
Muut tiedot		

Author(s) Kataja-aho, Joonas	Type of publication Bachelor's thesis	Date 08.12.2015 Language of publication: Finnish
	Number of pages 61	Permission for web publication: x
Title of publication Data logging with Siemens Comfort panels		
Degree programme Automation engineering		
Supervisor(s) Häkkinen, Veli-Matti & Kuisma, Ari		
Assigned by Pekkanen, Jarmo		
Abstract <p>The purpose of the thesis was to design a system, which could gather and process data gained from industrial processes. The designed system would be marketed and sold to be used in process industry. The thesis was assigned by a Jyväskylä based design agency JEEC ltd.</p> <p>The thesis was carried out by designing two WinCC Visual Basic scripts and an Excel Visual Basic for Applications script. One of the VBS scripts gathers data and the other one monitors the size of data log files. The Excel VBA script is designed to create pre-defined reports. The software consists of 1,500 lines of Visual Basic code.</p> <p>The result of the thesis was a system that fulfills the objectives; however, no such system that handles mass storage device space depletion could be created. Also, due to the restrictions existing in WinCC environment a script which monitors file sizes had to be designed in a customized way.</p> <p>As a conclusion of the thesis it was noticed how different program design was when using Excel VBA and WinCC VBS software developing environments. The theoretical part covers reporting in industry, Windows CE operating system, VB, VBA and VBS programming languages, flash memories and CSV file type.</p>		
Keywords/tags (subjects) Data logging, Comfort panel, Visual Basic script, report, flash memory		
Miscellaneous		

Sisältö

Sanasto	4
1 Johdanto.....	6
1.1 Toimeksiantajan esittely	6
1.2 Toimeksiantajan tavoitteet	6
1.3 Henkilökohtaiset tavoitteet.....	7
2 Tietoperusta	8
2.1 Tiedonkeruu teollisuudessa	8
2.2 Siemens WinCC.....	9
2.3 WinCC historical data tool.....	10
2.3.1 WinCC Skriptit.....	12
2.4 Visual Basic	15
2.4.1 Visual Basic for Applications	17
2.4.2 Visual Basic Scripting edition	18
2.5 Windows Embedded CE	18
2.6 Flash-muisti	20
2.7 SD-muistikortti	24
2.7.1 Mini SD.....	25
2.7.2 Micro SD	25
2.7.3 SD-korttien kapasiteettiluokitukset.....	26
2.7.4 SD-korttien nopeusluokitukset.....	26
2.7.5 SPI-Väylä	28
2.8 USB-muistitikku	29
2.9 Tiedostojärjestelmä.....	31
2.9.1 FAT	32
2.9.2 exFAT	32
2.9.3 NTFS	33

2.10	CSV-tiedostomuoto	33
3	Toteutus	35
3.1	Suunnittelu	35
3.2	WinCC-skriptit.....	36
3.2.1	Datalogging-skripti.....	36
3.2.2	FileMonitoring-skripti	38
3.2.3	DLSettings-taulukko.....	39
3.3	Excel VBA skripti	40
3.4	Testaaminen	42
4	Tulokset	45
4.1	Tulosten arviointi.....	45
4.2	Jatkokehitys	47
5	Pohdinta	49
	Lähteet.....	51
	Liitteet	55
	Liite 1. panosraportti	55
	Liite 2. panosperusteinen tuotantoraportti.....	56
	Liite 3. päiväkohtainen tuotantoraportti.....	57
	Liite 4. kulutusraportti	58

Kuviot

Kuvio 1. Historical data -työkalu	10
Kuvio 2. WinCC scripts editor IDE.....	13
Kuvio 3. Windows CE paneeleissa tuetut tiedostonhallintamenetelmät	15
Kuvio 4. Visual Basic syntaksia	17
Kuvio 5. Windows CE 6.0 -työpöytä Siemensin KP700 Comfort -paneelissa	20
Kuvio 6. Flash-muistitransistorin rakenne.....	21
Kuvio 7. NOR-arkkitehtuuri	23
Kuvio 8. NAND-arkkitehtuuri.....	23
Kuvio 9. Class 6-nopeusluokkamerkki	27
Kuvio 10. UHS Class 3-nopeusluokkamerkki	27
Kuvio 11. HVPG4F6 flash-muistipiiri USB-muistitikussa.....	30
Kuvio 12. UP 21 -ohjainpiiri sekä muita komponentteja USB-muistitikussa	30
Kuvio 13. JEEC Datalogging report handling -käyttöliittymä	40
Kuvio 14. WinCC testausnäyttö	44

Taulukot

Taulukko 1. Historical data -työkalun loki.....	12
Taulukko 2. DLSettings-tagitaulukon rakenne	40

Sanasto

ASCII	American standard code for information interchange on numero-merkkitaulukko, jonka avulla binääriarvot koodataan tekstimerkistöksi.
ActiveX	Nimitys Microsoftin uudelleenkäytettävien ohjelmistokomponenttien tekniikalle
Allokaatio-yksikkö	Massamuistivälineen sektoreista koostuva looginen tallennuspaikka
CHEI-ilmiö	Channel hot electron injection on ilmiö, jossa elektronit läpäisevät eristeaineen niihin kohdistetun korkeaenergiaisen sähköpulsin avulla
CPU	Central processing unit on keskusprosessori
CSV	Comma separated values on pilkuilla erotettu taulukkomuotoinen tekstitiedostomuoto.
DS	Default speed on SD-muistikortin väylämoodi, jossa on 3,3 V jännite ja maksimissaan 12,5 MB/s tiedonsiirtonopeus.
EEPROM	Electrically erasable programmable read only memory on sähköisesti pyyhittävässä oleva haihtumaton puolijohdemuisti.
FET	Field effect transistor, kanavatransistori, jossa johtavuuden muutos perustuu sähkökenttään
Flash	Toshiban kaupp nimi EEPROMille, jota tyhjenetään lohko kerrallaan.
F-N-ilmiö	Fowler-Nordheim, ilmiö, jossa elektronit läpäisevät eristeaineen niiden voimakkaasta sähkökentästä saaman liike-energian avulla.
HMI	Human machine interface on automaation käyttöliittymä.
HS	High speed on SD-muistikortin väylämoodi, jossa on 3,3 V jännite ja maksimissaan 25 MB/s tiedonsiirtonopeus.
IDE	Integrated development environment on integroitu ohjelmistonkehitysohjelma.
Massamuisti	Suuren datamäärän tallennukseen sopiva laite
Panos	Prosessin tuote-erä.
Resepti	Automaatiojärjestelmän ymmärtämä tietopaketti, jolla määritetään valmistuksen hallinnassa tarvittavat tiedot.

Runtime	WinCC:n käyntitila, jossa ajetaan valvomonäyttöjä
SCADA	Supervisory control and data acquisition on valvomo
SD	Secure digital on Flash-muistipiirin perustuva muistikorttityyppi.
SDA	Secure digital association on Toshiba, Panasonicin ja SanDiskin vuonna 2000 perustama koaliatio SD-korttien kehittämistä varten.
Sektor	Massamuistivälineen pienin fyysinen tallennuspaikka
Skripti	Ohjelmallinen komentosarja toimintojen automatisointiin
SPI	Serial peripheral interface eli sarjamoottori, kellosignaaliin perustuva tiedonsiirtolinja.
Tag	Tagi on Siemensin Step 7 järjestelmissä käytetty muuttuja, jolla on nimi, tietotyyppi ja arvo.
UHS	Ultra high speed on SD-muistikortin väylämoodi, jossa mahdolliset jännitteet ovat 3,3 V, 1,8 V sekä 0,4 V. Maksimi tiedonsiirtonopeus jopa 312 MB/s.
VB	Visual Basic on Microsoftin kehittämä Basicin kaltainen ohjelmointikieli.
VBA	Visual Basic for Applications on Microsoftin kehittämä Visual Basic murre officeohjelmien skriptikehitykseen.
VBS	Visual Basic Scripting edition on Microsoftin kehittämä Visual Basic murre mobiililaitteiden ja Internet skriptien ohjelmointiin.

1 Johdanto

1.1 Toimeksiantajan esittely

JEEC Oy on Jyväskylän Kirrissä sijaitseva insinööritoimisto, jonka toiminta koostuu korkealaatuista sähkö- ja automaatiotekniikan projekteista suunnittelun ja konsultoinnin saralla. Ydinosaamisen automaatiotekniikan alalla kuuluu kaksi osa-alueita: prosessiteollisuuden sovelluskehitys sekä käyttöönotot. Sähkötekniikan osaamisalueita ovat rakennus- ja prosessisähkösuunnittelu sekä tele- ja turvajärjestelmät. JEEC Oy on perustettu 2009. (JEEC Oy nettisivut 2015.)

Kirrin suunnittelutoimistossa työskentelee tällä hetkellä 15 vakituista työntekijää. Työntekijöistä 12 sijoittuu pääasiassa automaatioon, 1 sähkötekniikkaan, 1 myyntiin ja 1 kirjanpitoon. JEEC Oy varmistaa laatunsa työntekijöidensä monipuolisella osaamisella sekä pitkäaikaisella kokemuksella. Yritys korostaa omassa toiminnassaan seuraavia arvoja: asiakaslähtöisyys, laatu, jatkuva kehittäminen, osaaminen sekä luottamus. JEEC Oy täyttää ISO 9001 laadunhallintastandardin. (JEEC Oy nettisivut 2015.)

1.2 Toimeksiantajan tavoitteet

Nykyisistä automaatiojärjestelmistä saatavan tiedon määrä on kasvanut valtavasti verrattuna aikaisempaan. Lisääntyvien kustannusten takia tuotanto- sekä energiatehokkuuden tarkkailu moderneissa valmistusprosesseissa on tärkeää. Tästä johtuen tiedon jäsentelyn merkitys on lisääntynyt. (Siemens WinCC -esite 2014.)

Toimeksiantaja oli huomannut tarpeen järjestelmälle, joka kerää dataa prosesseista käyttäen apuna Siemens AG:n automaatiovälineistöä. Tätä dataa tulisi prosessoida edelleen ja luoda siistejä toimeksiantajan määrittelemiä raportteja Exceliä hyväksikäyttäen. Datankeräysominaisuutta voitaisiin tarjota prosessiteollisuuden

asiakkaiden käyttöön, esimerkiksi luomaan panosprosesseista erilaisia eräkohtaisia ainemääräraportteja. Lopullisen tuotteen tulisi antaa asiakkaalle lisäarvoa tarkemman prosessin seuraamisen kautta sekä olla helposti myytävä ja kaupallinen.

Alkuperäinen tehtävänanto sisälsi datan keräämisen 20 yhtäaikaiselle mittaukselle sekunnin syklillä. Toivottavaa myös oli, että datan keräys jatkuisi ensimmäisenä luodun tiedoston päälle tallennusvälineen täytyessä. Lisäksi tehtävänä oli luoda Exceliin ohjelma, jolla kerätystä datasta pystyttäisiin luomaan toimeksiantajan raporttipohjien mukaisia raportteja. Sovelluksien luontiin annettiin muuten hyvin vapaat kädet.

1.3 Henkilökohtaiset tavoitteet

Henkilökohtaisiksi tavoitteiksi opinnäytetyöprosessille asetettiin WinCC:n opiskelu syvemmin, varsinkin Visual Basic skriptien kehitys. Osaaminen Visual Basic ohjelmoinnissa tulisi varmasti olemaan hyödyllistä tulevaisuuden kannalta. Sillä vaikkakin Visual Basic-ohjelmointikieli on vanha, sitä käytetään erilaisten ohjelmien kehityksessä vielä nykypäivänä laajastikin. Taito kirjoittaa skriptejä tuo hyödyllisen lisän moniin ohjelmiin, sillä niillä pystytään toteuttamaan monimutkaisiakin asioita suhteellisen helposti. Yleinen ohjelmointikokemus auttaa sitä paitsi kehittämään sovelluksia myös millä tahansa muulla ohjelmointikielillä.

Työssä vastaan tulevat erilaiset massamuistien, tietojärjestelmien, ohjelmistojen, tiedonkeruulaitteistojen sekä -järjestelmien tekniset ominaisuudet herättävät mielenkiintoa. Yksi tavoitteista onkin tutustua kyseisiin tuotteisiin tietoperustan parissa työskennellessä. Myös opinnäytetyöprosessille tyypillinen tiedonhaku sekä -jäsenitys ovat asioita, joissa kehittymiseltä ei voi opinnäytetyötä tehdessä vältyä. Teknisen raportin kirjoittaminen onkin osa-alue, jolla soisi tapahtuvan kehitystä.

Työn aihe sisältää paljon automaatiotekniikasta tuttuja ominaisuuksia, mutta lisäksi myös esimerkiksi elementtejä perinteisestä ohjelmisto- ja tietotekniikasta. Tämän voi nähdä etuna, sillä tällä tavalla työstä muodustuu monipuolinen, ja tarvittavaa poikkitieteellistä tietoa tulee myös hankittua sähkö-automaatiotekniikan lisäksi muiltakin insinöörialoilta. Lisäksi toivottavaa olisi tutustuminen teollisuusasiakkaiden käytäntöihin liittyen sovelluksien käyttöliittymiin sekä raporttien ulkoasuun.

2 Tietoperusta

2.1 Tiedonkeruu teollisuudessa

Kipon ja Tikan (2008, 7) mukaan lähtökohtana automaatiolle voidaan pitää mittaantureille tehtyjen havaintojen seuraamista. Luonnollisesti tästä seuraa, että automaatioprosessista saadaan runsaasti tuotantoon, panoksiin ja mittauksiin liittyvää dataa, joka voidaan viedä sitä varten luotuun historiatietokantaan. Näistä tiedoista voidaan luoda kokoamalla tuotantoyhteenvetoja, panosten toteutumatietoja, ajoreseptejä sekä prosessin tilatietoja. Tälläisen datahistoriatietokannan kannalta olennaista on, että siellä on sopivaa tietoa sopiva määrä ja se on sopivasti lajiteltu esimerkiksi ajan tai tuotantolinjan mukaan. Turhaa tietoa ei siis tulisi tallentaa, vaan hyödyntää älykästä laskentaa tai keskiarvoistusta ja tiedon kompressointia. Lisäksi käyttäjän manuaalisesti syöttämän muuta informaatiota tukevan lisäinformaation tallentaminen datan tallennuksen yhteydessä voi olla välttämätöntä. (Tommila 1993, 96-97.)

Teollisuuden prosessissa tiedonkeruu palvelee ainakin seuraavia käyttötarkoituksia (Tommila 1993, 97).

- Tuotanto- ja panostietojen uudelleen haku esimerkiksi reklamaatioiden selvittämiseksi

- Pitkän aikavälin tuotantoyhteenvetojen tarkasteleminen
- Tuotteiden laatuun, häiriöihin ja taloudellisuuteen liittyvä tilastollinen analyysi esimerkiksi:
 - reseptin parametrien virittäminen
 - osaprosessien tehokkuuden tarkkailu

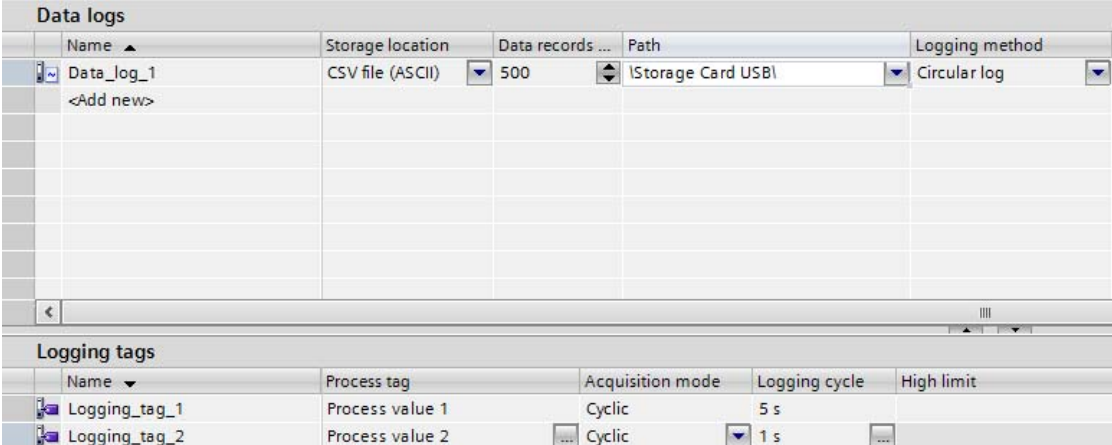
Lokitiedoista on myös perusteltua luoda esitettäväksi raportteja, joita ihminen voi soveltaa omaan ympäristöönsä. Valmiit raportit lisäävät ymmärrystä, jolloin niitä voidaan hyödyntää näitä valinnoissa ja päätöksissä. (Suominen & Suominen 2015, 19.) Tarjoamalla oikeaa tietoa, oikeassa muodossa, oikeille henkilöille oikeaan aikaan voidaan raporteilla parhaimmillaan kuvata laajoja kokonaisuuksia ja huomata syy-seuraussuhteita, sekä havaita poikkeuksia ja kehityssuuntia (Mts. 7). Raportti onkin ennen kaikkea dokumentti, joka sisältää dataa analysointiin ja tarkasteluun. Raporteista tyypillisesti tehtävät johtopäätökset ovatkin tulkintaa (Mts. 13).

2.2 Siemens WinCC

WinCC on Siemens AG:n SCADA-sovellus. WinCC:llä voidaan luoda näyttöjen käyttöliittymiä sekä luoda näille dynaamisia toimintoja. Siemens WinCC sisältää toimintoja muun muassa datan keruuseen, hälytyksiin, ajastettuihin tehtäviin, simulointiin sekä skripteihin liittyen. WinCC:tä on mahdollista käyttää niin sanotussa Runtimessa, jolloin tietokoneesta tulee HMI eli valvomonäyttö. Siemens AG tarjoaa WinCC:stä eritasoisia räätälöityjä paketteja. Paketit jakaantuvat kommunikointitagien sekä toimintojen määrän mukaan. Ohjelmistosta saa myös pelkkää Runtime-lisenssiä, josta puuttuu kokonaan kehitysohjelmat. (WinCC-nettisivu, Packages.) WinCC:n markkinoinnissa painotetaan tulevaisuuden yhteensopivuutta. Siemens AG kertoo WinCC:n olevan avoin kehityksen ja yhteensopivuuden kannalta, innovatiivinen, koska se sisältää uusia toimintoja, tehokas, sillä WinCC sisältää turvallisuutta ja suunnittelun tehokkuutta edistäviä toimintoja ja järjestelmän olevan skaalattavissa ylöspäin asiakkaan laitteiston laajentuessa. (Mts. Added Value.)

2.3 WinCC historical data tool

Siemensin WinCC ohjelmisto sisältää historical data -työkalun (kuvio 1), joka mahdollistaa data- sekä hälytyslokin luomisen. Datalokiin voidaan kerätä näyttöpaneelin ulkoisien ja sisäisien tagien arvoja talteen. Ulkoisella tagilla tarkoitetaan tässä tapauksessa tagia, joka on linkitetty johonkin toiseen laitteeseen näytön ulkopuolelle. Sisäinen taas tarkoittaa pelkästään saman näyttöpaneelin sisällä olevaa muuttujaa.



The screenshot shows the 'Data logs' and 'Logging tags' configuration windows in WinCC. The 'Data logs' window has a table with columns: Name, Storage location, Data records, Path, and Logging method. The 'Logging tags' window has a table with columns: Name, Process tag, Acquisition mode, Logging cycle, and High limit.

Data logs				
Name ▲	Storage location	Data records ...	Path	Logging method
Data_log_1	CSV file (ASCII)	500	Storage Card USB1	Circular log
<Add new>				

Logging tags				
Name ▼	Process tag	Acquisition mode	Logging cycle	High limit
Logging_tag_1	Process value 1	Cyclic	5 s	
Logging_tag_2	Process value 2	Cyclic	1 s	

Kuvio 1. Historical data -työkalu

Muuttujina on mahdollista käyttää tageja paneelin tagilistasta tai suoraan viitata CPU:n tageihin (WinCC Advanced V13 SP1-järjestelmämanuaali 2014, 4253). Tagille on myös mahdollista asettaa ylä- sekä alaraja. Täten tagin tallentaminen voidaan rajata tilanteeseen, kun sen arvo on näiden rajojen rajaaman alueen sisä- tai ulkopuolella käyttäjän tekemän valinnan mukaan.

Muuttujan tallentaminen lokiin voidaan liipaista kolmella tapaa: (Mts. 4252.)

- Syklimuuttujaan kytkemällä
- LogTag -järjestelmäfunktiolla

- Oman arvonsa muutoksesta

Siemensin historical data -työkaluun voidaan asettaa seuraavia parametreja.

Asetettavien parametrien runsaus vaihtelee kyseessä olevan paneelin mukaan. (Mts. 4250):

- Datalokin nimi (Maks. 60 merkkiä)
- Datalokin tiedostomuoto (CSV, RDB tai TXT)
- Datalokin tallennuspolku (Maks. 250 merkkiä)
- Data-arvojen lukumäärä (Maks. 500 000 kpl)
- Segmenttien lukumäärä (Maks. 500 kpl)(Liittyy segmentoituun lokien kiertoon)
- Hälytyksen aiheuttava täytösaste prosentteina (Liittyy hälytyksen antoon lokin täyttyessä)
- Käyntilupa, kun näyttösovellus käynnistetään
- Toiminta lokin täyttyessä (Mts. s4255)
 - Yhden lokin kierto. Kun data-arvoja on asetetun lukumäärän verran, aloitetaan lokin uudelleen kirjoitus vanhimman data-arvon päälle.
 - Segmentoitu lokien kierto. Kun loki-arvoja on asetetun lukumäärän verran, luodaan uusi loki. Kun lokeja on asetetun lukumäärän verran, aloitetaan lokien uudelleen kirjoitus vanhimman lokin päälle.
 - Hälytyksen anto. Kun asetettu täytösaste on saavutettu, annetaan järjestelmähälytys. Uusia arvoja ei oteta enää vastaan, jos massamuisti saavuttaa 100 % täytösasteen.
 - Tapahtuma. Lokin saavuttaessa yli 100 % täytösasteen, käynnistetään järjestelmässä jokin käyttäjän määräämä tapahtuma.
- Toiminta, kun näyttösovellus käynnistetään uudelleen
 - Lokien tyhjennys
 - Lokien jatkaminen

Historical data -lokin nimi täytyy päättää etukäteen, eikä dynaamista, esimerkiksi päivämäärän mukaan vaihtuvaa nimeä voi käyttää. Useamman datalokin samanaikainen avaaminen aiheuttaa myös muistin loppumisen suorittavasta näyttöpaneelistä. Lisäksi eri muuttujat pakataan taulukkoon päällekkäin (Taulukko 1)(Mts. 3326).

Taulukko 1. Historical data -työkalun loki

VarName	TimeString	VarValue	Validity	Time_ms
Value1	23.9.2015 15:35	0	1	42270649671
Value2	23.9.2015 15:35	20	1	42270649671
Value1	23.9.2015 15:35	0	1	42270649683
Value2	23.9.2015 15:35	20	1	42270649683
Value1	23.9.2015 15:35	0	1	42270649695
Value2	23.9.2015 15:35	20	1	42270649695

2.3.1 WinCC Skriptit

WinCC V13 sisältää skriptien kirjoitukseen tarkoitetun integroidun ohjelmistonkehitysympäristön nimeltään scripts editor (kuvio 2). Scripts editorilla voidaan luoda käyttäjän omia skriptejä lähinnä Visual Basic Script -ohjelmointikielellä, mutta myös C-kielellä. C-kielellä luodut skriptit vaativat WinCC:n Professional version sekä WinCC Runtimen oikealla PC:llä. (Vinkkejä WinCC-skripteihin 2015.)

```

1 Sub Format()
2 'Tip:
3 ' 1. Use the <CTRL+SPACE> or <CTRL+I> shortcut to open a list of all objects and functions
4 ' 2. Write the code using the HMI Runtime object.
5 '   Example: HmiRuntime.Screens("Screen_1").
6 ' 3. Use the <CTRL+J> shortcut to create an object reference.
7 'Write the code as of this position:
8
9 Err.Clear 'Clear any possible stored error messages
10
11 Dim fso 'Define object variable
12
13 Set fso = CreateObject("filect1.filesystem") 'Use this object to handle files
14
15 fso.kill "\\Storage Card USB\*" 'Delete files and folders from USB-card
16
17 ShowSystemAlarm "USB stick formatted" 'Show message to user
18
19 End Sub

```

Kuvio 2. WinCC scripts editor IDE

WinCC:n sisältämä scripts editor sisältää lukuisia ohjelmoijaa hyödyttäviä toimintoja. Ohjelmistoympäristö muun muassa tarkastaa käyttäjän syöttämää koodia reaaliaikaisesti ja alleviivaa virheelliset komennot punaisella. IDE sisältää myös erillisen syntaksintarkastustoiminnon, joka käy koko koodin yksityiskohtaisesti läpi syntaksivirheiden löytämistä varten. (WinCC Advanced V13 SP1-järjestelmämanuaali 2014, 4587.) Työkalun sisältämä syntaksin tyyppien mukaan tapahtuva tekstin värjäys helpottaa niin ikään ohjelmankehittämistä. Värit ovat lisäksi täysin käyttäjän asetuksista vaihdettavissa. Myös kirjanmerkit, usean rivin kerrallaan kommentointi sekä sisennyksen muuttaminen ovat ominaisuuksia, joita scripts editor sisältää. Tämän lisäksi se tarjoaa listat käytettävistä komennoista, vinkkilaatikoita, automaattisen täydennyksen sekä muutaman yleisimmin koodirakenteen valmiiksi tehdyt pohjat. (Mts. 4758-4780.)

Scripts editorilla luotavat skriptit tarvitsevat ennen kaikkea uniikin nimen, jolla skriptiin viitataan. Käyttäjän luomat skriptit ilmestyvät paneelin funktiolistaan yhdessä järjestelmän omien funktioiden kanssa. (Mts. 4572-4573.) Skriptit voivat olla joko funktioita tai aliohjelmia. Funktio eroaa aliohjelmasta siten, että se palauttaa arvon, joka voidaan kytkeä tagiin (Mts. 4586). Skriptille voidaan viedä muuttujia parametreina. Parametrit ovat joko ByRef- tai ByVal-tyyppisiä. Ero näiden kahden välillä on se, että ByVal-parametrin arvoa ei voida muuttaa sen ohjelman sisällä, johon se viedään. (Mts. 4584.) Parametreihin voidaan kytkeä joko vakio merkkijono, kokonais-, tai liukuluku. Myös tagin liittäminen parametriin onnistuu. Kuten Visual

Basicille tyypillistä on, voidaan WinCC:n VBS-skripteissä viitata myös toisiin VBS-skripteihin tai järjestelmän valmisfunktioihin. (Mts. 4573.)

Runtimessa käyttäjän tekemiä skriptejä suoritetaan ikään kuin jonossa yksi kerrallaan. Järjestelmän omat funktiot osaavat myös vaihtaa paikkaa tässä jonossa, jos siitä on suorituksen kannalta etua. Jonon järjestys riippuu käyttäjän tekemästä kokoonpanosta. Tapahtumassa sijaitseva funktioliista käydään ylhäältä alas. (Mts. 4600.) Huomattavaa myös on, että silmukkarakenteet suoritettavassa ohjelmistossa kuitenkin varaavat prosessoriaikaa itselleen ja täten estävät muiden funktioiden ja skriptien suorituksen. Suoritettavilla funktioilla ja skripteillä on kuitenkin matalampi prioriteetti kuin esimerkiksi paneelin ruudunpäivityksellä, millä pyritään takaamaan paneelin sulava toiminta myös skriptejä käytettäessä. (Mts. 4601.)

Skriptejä sekä järjestelmän funktioita voidaan WinCC:ssä käynnistää seuraavin menetelmin (Mts. 4576, 4605 ,5977):

- Ajoitettuna tehtävänä
- Tagin arvon vaihtumisesta
- Ohjausobjektin tapahtumana

WinCC scripting editor tukee vain vähäistä määrää tiedostojen käsittelyyn liittyviä rakenteita (kuvio 3), kun ohjelmia kirjoitetaan Windows CE 6.0 käyttöjärjestelmälle. (Siemens TIA Portal V13 SP 1 sisäinen manuaali n.d).

VBScript for Windows CE

This chapter contains the following information:

- [Attr \(Panels\)](#)
- [Close \(Panels\)](#)
- [CreateObject \(Panels\)](#)
- [Dir \(Panels\)](#)
- [EOF \(Panels\)](#)
- [FileCopy \(Panels\)](#)
- [FileLen \(Panels\)](#)
- [FileDateTime \(Panels\)](#)
- [Get \(Panels\)](#)
- [GetAttr \(Panels\)](#)
- [Input \(Panels\)](#)
- [InputFields \(Panels\)](#)
- [InputB \(Panels\)](#)
- [Kill \(Panels\)](#)
- [LineInputString \(Panels\)](#)
- [LinePrint \(Panels\)](#)
- [Loc \(Panels\)](#)
- [LOF \(Panels\)](#)
- [MkDir \(Panels\)](#)
- [MoveFile \(Panels\)](#)
- [Open \(Panels\)](#)
- [Put \(Panels\)](#)
- [Rmdir \(Panels\)](#)
- [Seek \(Panels\)](#)
- [SetAttr \(Panels\)](#)
- [WriteFields \(Panels\)](#)

Kuvio 3. Windows CE paneeleissa tuetut tiedostonhallintamenetelmät

2.4 Visual Basic

Visual Basic on korkeatasoinen tekstipohjainen ohjelmointikieli, joka perustuu Ruby-komentotulkkiin (*shell*) sekä vanhaan Quick Basic-ohjelmointikieleen (Halvorson 2010, XVII). Visual Basicin voidaan sanoa olevan niin sanottu skriptiohjelmointikieli, joka tarkoittaa sitä, että kielellä yhdistellään muilla ohjelmointikielillä luotuja komponentteja toisiinsa (Harsu 2005, 279). Visual Basicilla on mahdollista toteuttaa tapahtumavetoista ohjelmointia. Tämä tarkoittaa sitä, että ohjelmaa suoritetaan vain, kun se saa ulkoisen syötteen. Esimerkkinä kellonajan vaihtuminen määrättyyn arvoon. (Halvorson 2010, 160). Siemensin näyttöpaneelleille Visual Basic -skriptejä voidaan suorittaa vain järjestelmän tapahtumien perusteella, esimerkiksi painonapin painalluksella (WinCC Advanced V13 SP1-järjestelmämanuaali 2014, 4042).

Visual Basicin versio 1.0 julkaistiin 1991 Microsoftin toimesta (Mack 2002). Vuodesta 1991 vuoteen 1998 julkaistiin yhteensä kuusi eri versionumeron sisältänyttä evoluutiovaihetta Visual Basicista. Vuonna 2002 Microsoft julkaisi Visual Basic .NET:n, joka sisältää Microsoftin .NET framework -kirjastot. (Halvorson 2010, XVII.) .NET-framework -kirjastot sisältävät valmiiksi ohjelmoituja komponentteja, joita voidaan käyttää Windows-käyttöjärjestelmien kera. .NET-framework -kirjastoa käytetään niin ikään myös Visual C++ sekä Visual C# kanssa. (Halvorson 2010, 153.) Visual Basiciin perustuva murre Visual Basic for Applications julkaistiin vuonna 1993 Microsoft Excel 5:n yhteydessä. Toinen murre, Visual Basic Scripting edition, julkaistiin Visual Basic 4:n kanssa samana vuonna 1995 (Mack 2002).

Tyypillistä Visual Basicille on, että sitä kehitetään integroiduilla ohjelmointisovelluksilla (*IDE*), kuten Microsoft Visual Studiolla. IDE sisältää koodieditorin, debuggerin sekä työkalut ohjelmistoprojektin hallintaan (Halvorson 2010, 30). Visual Basicin syntaksi koostuu englanninkielisistä sanoista. (Kuvio 4) Ohjelmointikielessä on mahdollista hyödyntää valmiita komponentteja käyttäjäliittymän luontiin (Mack 2002). Ohjelmointikieli on rivipohjainen, eikä siinä käytetä puolipisteitä käskyjen välillä kuten esimerkiksi C++:ssa. Visual Basicin syntaksissa voidaan käyttää hyppyrakenteita ja virheenohitustoimintoja. Visual Basic sisältää myös tuen useille silmukkarakenteille. Visual Basicin yksi ominaisuuksista on hybridirakenne tulkattavuuden ja käännettävyyden välillä. Tulkattava ohjelmointikieli tarkoittaa sitä, että sitä käännetään korkeamman tason kielestä konekielelle sitä mukaan, kun koodia suoritetaan. Käännettävä ohjelmointikieli terminä taas tarkoittaa, että koko ohjelma käännetään kerralla ennen suoritusta. Visual Basic käännetään ennen suoritusta, mutta stand-alone -sovellukset luottavat usein myös laajoihin ajonaikaisiin kirjastoihin. (Kingsley-Hughes, Kingsley-Hughes & Read 2008, 38.) Tämä mahdollistaa paremman virheiden eristämisen ja virhetilanteista toipumisen suorituksen raskauden kustannuksella (Halvorson 2010, 30).

```
Sub Painike1_Napsauta()  
On Error Resume Next  
Dim textstring  
Application.ScreenUpdating = False  
textstring = "Hello world!"  
For i = 1 To 12 Step 1  
    Application.Worksheets("tauli1").Cells(1, i).Value = Mid(textstring, i, 1)  
Next  
Application.ScreenUpdating = True  
End Sub
```

Kuvio 4. Visual Basic syntaksia

2.4.1 Visual Basic for Applications

Visual Basic for Applications (*VBA*) on sovellusten päällä toimiva sulautettu Visual Basic komponentti. Suurin ero natiiviin Visual Basiciin on se, että VBA:n pohjana toimii toinen, valmis ohjelma, kun taas natiivi Visual Basic voi toimia itsenäisesti. Esimerkiksi Microsoft Excel taulukkolaskentasovelluksen käyttäjän tallentamat makrot toimivat VBA-ohjelmointikielellä. Myös muut Microsoft Officen ohjelmat kuten Word, Power point, Access, Outlook, Publisher, sekä Project tukevat VBA-kieltä. (Merensalmi 2007, 4-5.) Myös jotkut muut ohjelmat tukevat lisenssillä VBA-ominaisuuksia. (Mack 2002) Eräs esimerkki tämmöisestä ohjelmasta on vektorigrafiikkaohjelma AutoCad, johon on mahdollista ladata VBA-laajennus (Artikkeli VBA-tuesta Autocadissa 2014). Excelissä VBA-makroja voi muokata siihen integroidulla IDE:llä nimeltään Visual Basic editor. VBA mahdollistaa sovelluksissa vaikkapa Windows käyttöjärjestelmän ominaisuuksien laajan hyväksikäytön. VBA sisältää yleensä myös isäntäsovelluskohtaisia komponentteja (Kingsley-Hughes, Kingsley-Hughes & Read 2008, 39).

2.4.2 Visual Basic Scripting edition

Visual Basic Scripting Edition (joskus myös Visual Basic Script) on niin ikään isäntäsovelluksen päällä toimiva, ei itsenäiseen suoritukseen pystyvä, Visual Basic-murre. Eli isäntäohjelmana voi toimia esimerkiksi Internet Explorer -selain tai Windows käyttöjärjestelmä itsessään käyttämällä Windows script host - ominaisuutta. (Kingsley-Hughes, Kingsley-Hughes & Read 2008, 39.) VBS on tulkettava ohjelmointikieli eli se tarvitsee hostin, joka pystyy kääntämään sitä niin sanotusti lennosta. VBS kehitettiin jotta Visual Basicin osaajat voisivat luoda dynaamisia internetsivuja, aivan kuten Javascriptillä on mahdollista. (Mts. 39.) Lisäksi muun muassa Siemensin HMI-ohjelmointiin tarkoitettu WinCC V13 mahdollistaa käyttäjän omien skriptien luonnin käyttämällä Visual Basic Script-ohjelmointikieltä (WinCC Advanced V13 SP1-järjestelmämanuaali 2014, 4950).

2.5 Windows Embedded CE

Windows Embedded CE on Microsoftin sulautettuja järjestelmiä varten luotu kevyt käyttöjärjestelmä (Phung 2008, Windows Embedded CE). Windows CE ei ole riisuttu versio mistään Microsoftin työpöytäkäyttöön tarkoitettusta käyttöjärjestelmästä, vaan sen komentotulkki on erilainen järeämpiin systeemeihin verrattuna (Mts. The Windows Embedded Family). Ensimmäinen versio Windows CE-perheestä julkaistiin 1996. Viimeisin versio Windows CE:stä on CE 6.0, joka tukee yhteensä 32 000 prosessia ja 2 gigatavua virtuaalista muistia. 15.04.2008 jälkeen Microsoft on jatkanut tuotteiden kehitystä Microsoft Embedded Compact -nimikkeeseen alla. (Mts. Windows Embedded CE 6.0.)

Windows CE sisältää vähätehoisella laitteella tarvittavat edistykselliset resurssienjako-ominaisuudet. Tästä johtuen laitteistoa voidaankin hyödyntää monissa aikakriittisissä sovelluksissa, johon työpöytä Windowsit eivät sovellu ilman tarkoituksenmukaista ohjelmistoa. Käyttöjärjestelmään on lisätty virranhallintakirjastoja sekä sitä on mahdollista pyörittää niukalla ja edullisella

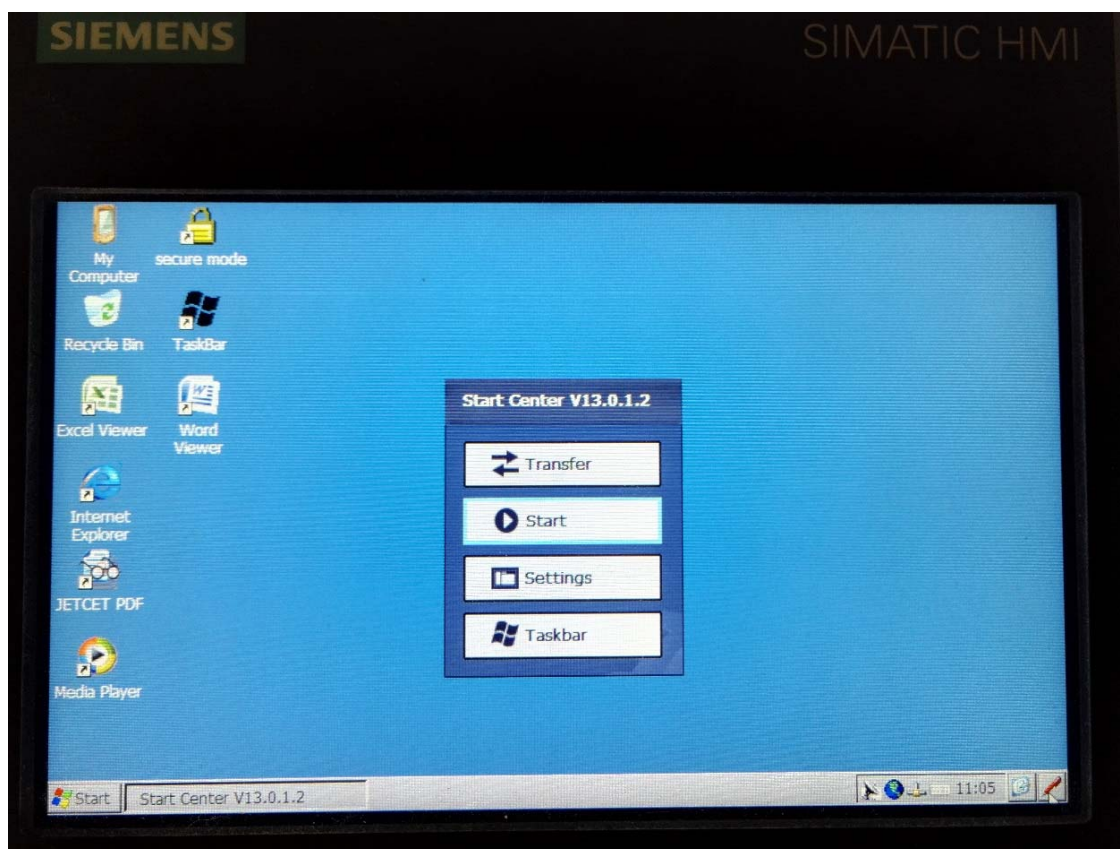
laitteistolla. Windows CE tukeekin neljää erilaista prosessoriarkkitehtuuria, jotka ovat ARM, MIPS, x86 sekä SH-4. (Mts. Windows Embedded CE 6.0.)

Yksi Windows CE:n pääpiirteistä on modulaarisuus. Microsoft tarjoaakin asiakkailleen valmiita pohjia erilaisten laiteratkaisujen toteuttamiseen. Tämän lisäksi käyttäjällä on mahdollisuus valita käyttöjärjestelmään lisättävät lisälaitte- ja laitteistoajurit, sovellukset sekä kirjastot. Windows CE 6.0 tukeekin seuraavia frameworkkeja: ATL, ActiveX, COM, MFC, .NET (Mts. Developing CE Applications). Myös käyttöliittymä on muokattavissa. Monesti Windows CE:tä hyödynnetään vain laitteiston taustalla, ja varsinainen käyttöjärjestelmän päälle luotu käyttöliittymäsovellus käynnistyy heti käynnistyksen jälkeen. Windows CE:tä löytyykin muun muassa tietoliikenteen yhdyskäytävien, kassakoneiden ja automaatiotekniikan näyttöpaneelien takaa. (Mts. Windows Embedded CE 6.0) Automaatilaitevalmistaja Siemens hyödyntää tietyissä paneeleissa Windows CE-käyttöjärjestelmää, joissa Siemensin oma Runtime-näyttösovellusta pyöritetään Windows CE työpöydän päällä (kuvio 5). Siemensin Comfort -paneelit sisältävät esiasennettuja ohjelmia, kuten Word Viewer, Excel Viewer, Windows Media Player, JETCET PDF Viewer ja Internet Explorer. Comfort-paneelit tukevat myös skriptejä (Siemens Comfort -paneelien käyttöohjeet 2012, 14).

Siemensin tällä hetkellä valmistamat HMI:t, jotka perustuvat Windows CE-käyttöjärjestelmään:

- MP 177, MP 270B, MP 277, M P370, MP 377 (Siemens Multi Panel 170 -paneelin käyttöohjeet 2008, 15)
- KP 400 Comfort, KTP 400 Comfort, KP 700 Comfort, KP 900 Comfort, KP 1200 Comfort, KP 1500 Comfort (Siemens Comfort-paneelien käyttöohjeet 2012, 65)
- TP 900 Comfort, TP 1200 Comfort, TP 1500 Comfort, TP 1900 Comfort, TP 2200 Comfort (Mts. 65)

- Mobile Panel 170 (Siemens Mobile Panel 170 -paneelin käyttöohjeet 2002, 1-1)
- Mobile Panel 170 (Siemens Mobile Panel 177 -paneelin käyttöohjeet 2005, 6-2)
- Mobile Panel 277 (Siemens Mobile Panel 277 -paneelin käyttöohjeet 2008, 115)
- Mobile Panel 277 IWLAN (Siemens Mobile Panel 170 -paneelin käyttöohjeet 2008, 105)



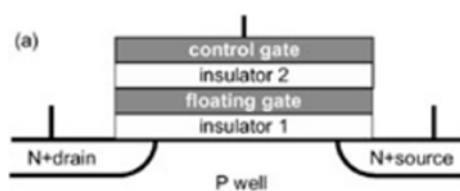
Kuvio 5. Windows CE 6.0 -työpöytä Siemensin KP700 Comfort -paneelissa

2.6 Flash-muisti

Flash-muisti on haihtumatonta puolijohdemuistia, joka on ohjelmoitavissa ja tyhjennettävissä sähköisesti. Haihtumattomuus tässä tapauksessa tarkoittaa sitä, että muistiin tallennetut tiedot eivät katoa, vaikka tallennusväline ei olisikaan yhteydessä virtalähteeseen. Kyseinen ominaisuus on saavutettu käyttämällä

transistoreja, joissa on eristetty elektronisäilö. Flash-muisti eroaa sen pohjana toimineesta EEPROM-muistista, niin että Flash-muisti tyhjenetään koko piiri tai piirin lohko kerrallaan. EEPROM-muistia taas voidaan tyhjentää tavutasolla. (Brewer & Gill 2008, Elementary memory concepts.) Flash-muistin on kehittänyt Toshiba vuonna 1985. Flash-muisti on saanut nimensä siitä, että koko muistipiirin kerrallaan tyhjentäminen kävi verrattain nopeasti verrattuna aikaisempiin teknologioihin. (Mts. Foreford overview.)

Flash-muistin perustana toimivat FET-muistitransistorit (kuvio 6), joita yksittäisessä muistipiireissä on useita. Transistoreissa on normaalin FET-transistorin tapaan nielu (*drain*), lähde (*source*) ja ohjaushila (*control gate*). Lisäksi ne sisältävät muistitoiminnan mahdollistavan kelluvan hilan (*floating gate*). Tähän kelluvaan hilaan on mahdollista säilöä elektroneita, jotka varaavat hilan sähköisesti. Hilan varauksella voidaan vaikuttaa transistorin johtavuuteen ja tätä kautta lähteeltä nielulle kulkevaan virtaan. (Mts. Elementary memory concepts.) Kun muistihilalla on läsnä elektroneja ja tästä johtuen virta ei pääse läpi, on tämän muistitransistorin tila tulkittavissa loogiseksi nollaksi. Tieto esitetäänkin flash-muisteissa digitaalisina ykkösten ja nollien yhdistelminä eli bitteinä. (Mts. Floating-gate principle.)



Kuvio 6. Flash-muistitransistorin rakenne (Brewer & Gill 2008, Elementary memory concepts)

Flash-muistiin kirjoittaminen tapahtuu joko Fowler-Nordheim -tunneloinnin tai Channel hot electron injection -ilmiön avulla. Fowler-Nordheim -tunnelointi perustuu siihen, että elektronit saavat sähkökentältä energiaa, jotta ne voivat tunneloitua johdekanavasta hyvin ohuen eristekerroksen läpi kelluvalle muistihilalle. (Mts.

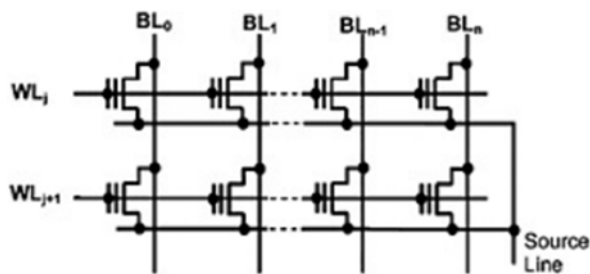
Fowler-Nordheim tunneling.) CHEI on vastaava ilmiö, mutta siinä voimakas sähkökenttä antaa osalle elektroneista tarpeeksi liike-energiaa läpäistä eristekerros. CHEI-ilmiöllä ei voida tyhjentää muistitransistoreja. (Mts. Channel hot electron injection.) FN-tunnelointi tarvitsee toimiakseen siis ohuet eristekerrokset ja CHEI-tunnelointi taas korkean jännitteen. FN-tunnelointia käytetään yleensä NAND-arkkitehtuurin kirjoittamisessa ja CHEI:tä NOR-arkkitehtuurissa (Mts. Floating-gate principle). Voimakkaisiin sähkökenttiin tarvittavat korkeammat jännitteet luodaan piirin sisällä varauspumpulla matalammasta käyttöjännitteestä (Mts. Elementary memory concepts). Muistin kirjoittaminen ja tyhjentäminen näiden ilmiöiden avulla aiheuttaa johdekanavan ja muistihilan välissä olevan eristemateriaalian rakenteellista kulumisesta. Muistitilan lukeminen taas ei tätä aiheuta. Tästä johtuen flash-muisteilla on rajallinen määrä kirjoitus- ja tyhjentämiskertoja. (Mts. Oxide degradation.)

Muistitransistoreihin tallennettavilla jännitetasojen määrällä voidaan vaikuttaa sinne mahtuvaan datamäärään. Kahdella eri jännitetasolla, jolloin puhutaan single-level cell media (SLC) arkkitehtuurista, voidaan esittää yksi bitti. Multi-level cell (MLC) media arkkitehtuurissa yhteen muistitransistoriin on mahdollista tallentaa useampia jännitetasoja, mikä mahdollistaa luonnollisesti useampien bittien esittämisen. Haittapuolena tästä voi olla lukunopeuden hidastuminen, sillä mitä lähempänä jännitetasojen rajat ovat toisistaan, sen vaikeampaa on selvittää eksaktisti transistorin tila. Vertical/3D Nand -arkkitehtuurissa otetaan käyttöön myös pystysuunta bittien säilytykseen pelkän tasomaisen arkkitehtuurin sijaan. (Rouse 2015, NAND flash memory storage types; Brewer & Gill 2008, elementary memory concepts.) NROM-arkkitehtuurissa liitetään kaksi SLC-transistoria yhteen, jolloin kelluvan hilan kumpaankin päähän on mahdollista säilöä elektroneja. Luettava hila valitaan lukusuunnan perusteella. (Brewer & Gill 2008, Flash memory and flash cell variations.) Käytännössä siis yhteen transistoriin tallennettavien eri jännitetasojen määrällä voidaan kasvattaa muistipiirin pinta-alalle mahtuvaa datamäärää.

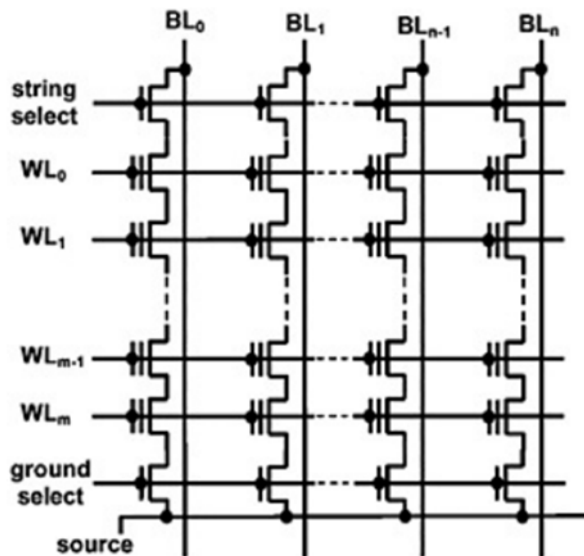
Koska yhteen transistoriin mahtuu vain vähäinen määrä tietoa, on välttämätöntä, että näitä transistoreja kytetään yhteen, jotta isompi datamäärä saadaan mahdutettua kortille. Transistorien kytkennässä on kaksi päätapaa: sarjamuotoinen NAND ja rinnakkainen NOR –arkkitehtuuri (kuviot 7 ja 8). NOR-arkkitehtuuri käyttää

omia datalinjoja joka transistorille, joten periaatteessa kustakin tietosolusta voidaan lukea tietoa yksitellen. NAND-arkkitehtuurissa on käytössä transistorien yhteisiä linjoja, jolloin yhden bitin lukemiseen täytyy usein lukea koko lohkon tiedot. NAND-muistia on hieman hitaampi lukea kuin NOR-muistia, mutta kirjoittaminen ja pyyhkiminen on huomattavasti nopeampaa. Tästä syystä voidaankin yleistää, että NAND-arkkitehtuuri on parempi datan lukemiseen ja kirjoittamiseen ja NOR esimerkiksi ohjelmakoodin suorittamiseen. (Mts. Elementary memory concepts)

Huomattavaa on myös, että NAND-arkkitehtuuri on yleensä hieman yksinkertaisempi valmistaa ja täten edullisempi. (Rouse 2015, NOR vs. NAND flash memory)



Kuvio 7. NOR-arkkitehtuuri (Brewer & Gill 2008, Elementary memory concepts)



Kuvio 8. NAND-arkkitehtuuri (Brewer & Gill 2008, Elementary memory concepts)

Flash-muistiin täytyy sisältyä ohjauspiiri, joka toimii yhteysrajapintana isäntälaitteisiin kuten esimerkiksi tietokoneisiin. Ohjainpiiri sisältää ohjelmiston, jolla flash-muistin datan hallintaa hoidetaan. Tähän ohjelmistoon on mahdollista sisällyttää erilaisia flash-muistien huonoja ominaisuuksia kompensoivia toimintoja. Eräs tämänkaltainen toiminto on kulumisentasaustoiminto (*wear leveling*). Tässä järjestelmässä ohjainpiiri pitää yllä taulukkoa, joka sisältää kunkin tietoyksikön erilliset fyysiset ja loogiset osoitteet, joiden avulla missä vaan fyysisesti sijaitseva tieto on mahdollista löytää loogisen osoitteen avulla. Tämä mahdollistaa sen, että tietueita voidaan siirrellä vähemmän kuluneisiin transistoreihin muistin sisällä. Vanha datan sisältänyt solu merkitään tällöin pois käytöstä aina tyhjäksi pyyhkimiseen asti. Koska flash-muisteja tyhjenetään lohkoissa, ei näitä käytöstä poistettuja soluja saada välttämättä kovin nopeasti käyttöön. Tästä johtuen ohjainpiirin ohjelmistossa on mahdollista olla roskienkeräystoiminto (*garbage collecting*). Tässä toiminnossa validi data siirretään tyhjään lohkoon, jolloin vanha, sekä epävalidia että validia dataa, sisältävä lohko päästään tyhjentämään kokonaan. Ohjainpiiri on myös vastuussa kuluneita transistoreja sisältävien lohkojen korvaamisesta muistipiirissä tätä varten sijaitsevilla varalohkoilla. Toimintoa kutsutaan nimellä huonojen lohkojen hallinta (*bad block handling*). Flash-muistien kulumistaipumuksien takia dataa tulisi valvoa virheenkorjauskoodilla (*error correction code*). Kirjoitettaessa muistiin virheenkorjauskoodi muodostetaan ja tallennetaan datan mukana. Luettaessa dataa virheenkorjauskoodi muodostetaan uudestaan luetusta datasta ja näitä kahta virheenkorjauskoodia verrataan toisiinsa, jolloin on mahdollista huomata virhetilanne. Joissain tapauksissa ohjainpiiri osaa myös korjata virheitä datassa käyttämällä hyväkseen epäjohdonmukaisuuksia virheenkorjauskoodissa. (Lowe 2015, Controller.)

2.7 SD-muistikortti

SD-muistikortti on noin postimerkin kokoinen flash-muistikortti. SD eli Secure Digital -muistikortin ensimmäisen version kehitti vuonna 1999 Panasonic, Toshiba ja SanDisk, jotka lisäksi loivat SD Associationin tammikuussa 2000. SD Association on yrityskoalitio, joka ylläpitää ja kehittää SD-muistikortteihin liittyviä standardeja. SD

Associationiin voi liittyä jäseneksi maksua vastaan, jolloin jäsen saa käyttöoikeuden täysiin standardeihin. (SD association nettisivut 2015, FAQ.) Muistikortin nimi Secure Digital tulee kortin kyvystä valvoa kopiosuojattua materiaalia (Mts. Copyright protection). SD-muistikortti perustuu vanhemman Multi Media Card -kortin pohjalle. MMC-kortin mitat ovat 32 x 24 x 1,4mm (MMC-korttien mekaaninen standardi 2007, 4) ja normaalin SD-kortin taas 32 x 24 x 2,1mm (SD-korttien fyysinen tekninen spesifikaatio 2013, 11). Kuitenkin huomioitavaa on, että on standardi määrittelee myös ohuen, thin SD-kortin, joka on kooltaan täysin identtinen MMC-kortin kanssa (Mts. 4). Tämä kokojen samankaltaisuus mahdollistaa tietyissä erikoistapauksissa korttien käytön toistensa paikoissa. Pinnijärjestys on samanlainen, vaikkakin kehitysversiona riippuen kortit sisältävät erilaisen lukumäärän datankuljetuslinjoja. SD-kortti, toisin kuin MMC, sisältää mekaanisen kytkimen tietojen lukittamiselle. (SD-korttien fyysinen tekninen spesifikaatio, 11; MMC-korttien mekaaninen standardi, 2.)

SD-kortteja sekä sen mikro- ja miniversioita käytetään tätä nykyä laajasti mobiililaitteissa, esimerkiksi videokameroissa, tietokoneissa sekä matkapuhelimissa. Myös monissa automaattilaitteissa, muun muassa Siemensin Comfort-paneeleissa on paikka SD/MMC-kortille (Siemens Comfort-paneelien käyttöohjeet 2012, 192-195).

2.7.1 Mini SD

Mini SD -muistikortti on normaalia SD-korttia pienempi flash-muistikortti. Sen fyysiset mitat ovat 21,5 x 20 x 1,4 mm. Mini SD-kortin painaa n. 1 g. Kortista puuttuu täysikokoiselle versiolle tyypillinen kirjoituslukituskytkin. Mini SD -muistikortin kontaktipinnien lukumäärä on 11. (SD association nettisivut 2015, Capacity (SD/SDHC/SDXC).)

2.7.2 Micro SD

Micro SD -muistikortti on nimensä mukaisesti SD-muistikorttiperheen pienin kortti. Micro SD palvelee käyttöä varsinkin matkapuhelimissa datavarastona sen erityisen

pienen koon ansiosta. Micro SD -muistikortin dimensiot ovat 15 x 11 x 1 mm ja se painaa noin 0,5 g. Micro SD -kortissa on täysikokoiseen versioon verrattuna vain 8 pinniä eikä siinäkään ole mekaanista kirjoituksenestokytkintä. (Mts. Capacity (SD/SDHC/SDXC).)

2.7.3 SD-korttien kapasiteettiluokitukset

Secure Digital -muistikortteja luokitellaan niille mahtuvan maksimitietomäärän mukaan seuraaviin kapasiteettiluokkiin (SD-korttien fyysinen tekninen spesifikaatio 2013, 5).

SDSC	Standard capacity I. normaali kapasiteetti, koko alle 2 Gb.
SDHC	High capacity I. korkea kapasiteetti, koko 2 – 32 Gb.
SDXC	Extended capacity I. laajennettu kapasiteetti, koko 32 Gb – 2 Tb.

2.7.4 SD-korttien nopeusluokitukset

SD-muistikorttien tiedonsiirto voidaan jakaa normaalien väylien nopeuksien osalta viiteen ja ultranopeiden väylien osalta kahteen eri nopeusluokkaan. (SD-korttien fyysinen tekninen spesifikaatio 2013, 5) Ultranopeat ja normaalit nopeusluokat käyttävät erilaista symbolia luokkansa esittämiseen (kuviot 9 ja 10) (SD association nettisivut 2015, Speed Class). Huomattavaa on, että nopeusluokituksissa käytettävät tiedonsiirtonopeuksien monikerrat esitetään kertoimen 1000 välein datamäärän kertoimen 1024 sijaan. Tämä johtuu siitä, että väylänopeuksien maksimit lasketaan perustuen väylän taajuuteen. (SD-korttien fyysinen tekninen spesifikaatio 2013, 6.) Todelliseen väylänopeuteen vaikuttavat vielä mm. laitteiston kokoonpano sekä SD-kortin täyttömäärä ja tietojen pirstoutumisaste. (Vrt. Mts. 94)

- Class 0** Tämä luokka ei määritä mitään konkreettista nopeusarvoa, vaan sisältää kaikki vanhat ennen standardin versioita 2.00 (julkaistu 25.09.2006) tehdyt kortit niiden oikeasta tiedonsiirtonopeudesta huolimatta. (Mts. 5)
- Class 2** Minimissään 2 MB/s kirjoitusnopeus. Vaatii default speed mode väyläprotokollan.
- Class 4** Minimissään 4 MB/s kirjoitusnopeus. Vaatii default speed mode väyläprotokollan.
- Class 6** Minimissään 6 MB/s kirjoitusnopeus. Vaatii default speed mode väyläprotokollan.
- Class 10** Minimissään 10 MB/s kirjoitusnopeus. Vaatii high speed mode väyläprotokollan.
- UHS Class 1** Minimissään 10MB/s kirjoitusnopeus. Vaatii vähintään UHS-I tai UHS-II väyläprotokollan.
- UHS Class 3** Minimissään 30MB/s kirjoitusnopeus. Vaatii vähintään UHS-I tai UHS-II väyläprotokollan.



Kuvio 9. Class 6-nopeusluokkamerkki (SD association nettisivut 2015, Speed Class)



Kuvio 10. UHS Class 3-nopeusluokkamerkki (SD association nettisivut 2015, Speed Class)

Vaihtoehtoinen tapa esittää kortin minimi tiedonsiirtonopeus on käyttää niin sanottua x-luokitusta. Tässä tavassa siirtonopeus esitetään alkuperäisen CD-ROM standardin siirtonopeuden (150 kB/s) monikertana. Tällöin kortin todellinen tiedonsiirtonopeus lasketaan käyttäen kaavaa: (Lexarin raportti SD-korttien nopeusluokituksista 2010, 1.)

$$\text{Tiedonsiirtonopeus} = X \text{ luokitus} \times 150 \text{ kB/s}$$

Esimerkiksi: Tiedonsiirron X-nopeusluokitus 40X

$$\text{Tiedonsiirtonopeus} = 40 \times 150 \text{ KB/s}$$

$$\text{Tiedonsiirtonopeus} = 6000 \text{ kB/s}$$

$$\text{Tiedonsiirtonopeus} = 6 \text{ MB/s}$$

X-nopeusluokitus 40X vastaa siis Class 6 nopeusluokitusta.

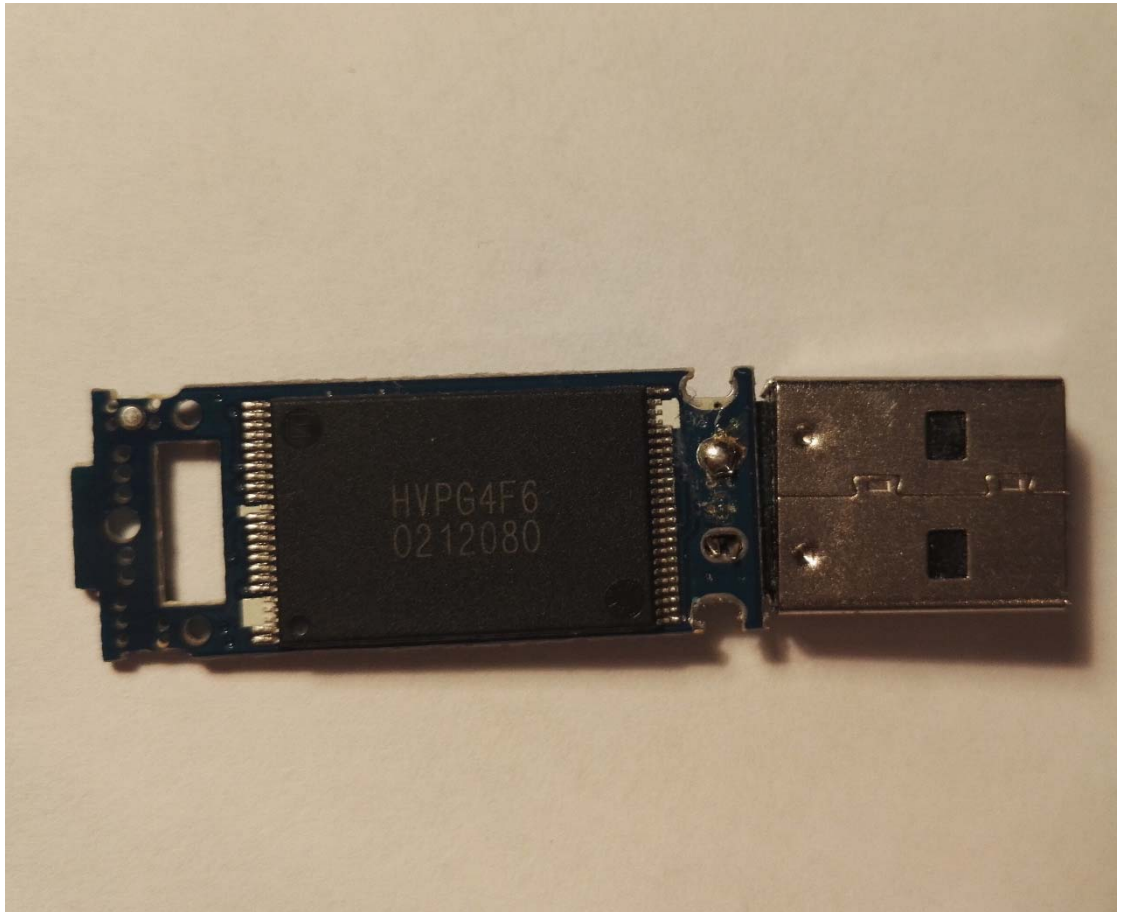
2.7.5 SPI-Väylä

SPI-väylä eroaa muista SD-mustikorteissa käytettävistä, niin sanotuista SD-väylistä (DS, HS ja UHS-I) niin, että SPI-väylällä yhdellä kellojaksolla siirretään yhtä linjaa pitkin bitti ja SD-väylillä käytetään neljää linjaa, joita pitkin siirretään neljä bittiä per kellojakso. (SD-korttien fyysinen tekninen spesifikaatio 2013, 9; Mts. 153) SPI-väylä ei ole erilaisuutensa takia yhteensopiva HS-, DS- ja UHS-väylien kanssa (Mts. 11). SPI-väylä on enemmänkin vaihtoehtoinen kommunikointitapa SD-kortille, joka voidaan ottaa käyttöön ohjelmallisesti. SPI-väylä on käyttökelpoinen varsinkin SD-kortin liittämiseksi mikrokontrollereihin. SPI-väylän nopeuden määrää isäntälaitteen SPI-väylän kellonastan kirjoittama kellotaajuus, jolla tiedonsiirtoa ajoitetaan. (Mts. 153.)

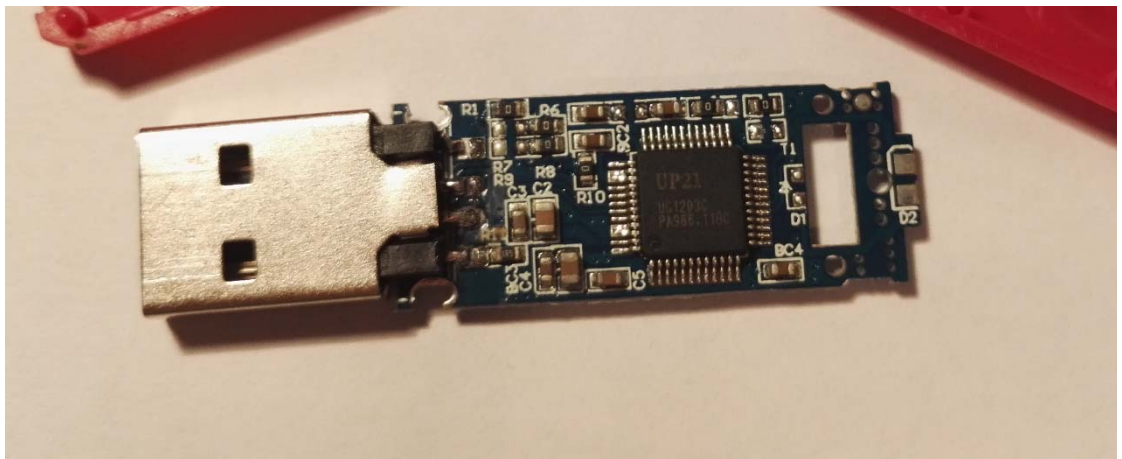
2.8 USB-muistitikku

Flash-muistipiiri voidaan istuttaa myös käytettäväksi USB-väylän päähän, jolloin puhutaan USB-muistitikusta. USB-tikut ovat yleensä pitkulaisia, sillä monesti laitteissa olevat USB-portit sijaitsevat ryhmissä lähellä toisiaan, jolloin leveät tai korkeat laitteet peittäisivät myös muita portteja. Valtaosa USB-tikuista käyttää A-tyyppin USB-porttia, mutta muihin porttityyppeihin nojautuvat muistitikut eivät ole mahdottomia. Esimerkiksi elektroniikkavalmistaja Sony valmistaa USB-tikkuja, joissa on sekä USB-A-että USB-mikro-portti (Sony USMSA2-sarjan USB Flash-muistin tekniset tiedot 2015).

USB-muistitikun voidaan ajatella koostuvan itse muistipiiristä (kuvio 11), muistipiirin ohjainprosessorista, koko muistitikun ohjainprosessorista sekä USB-yhteyspiiristä. USB-yhteyden hallinta voidaan jättää myös muistipiirin ohjaimen haltuun (kuvio 12). (Axelson 2006, 4.) Kaikki USB-massamuistilaitteet tukevat sekä USB-rajapintaa isäntälaitteen, yleensä tietokoneen suuntaan, että tikun sisäistä väylää muistikortille. Yleisiä muistipiirin ja USB-ohjaimen välisiä väyliä ovat muun muassa SPI- ja SD-väylä. (Axelson 2006, XIV). Erikoistapaus USB-massamuistilaitteesta on USB-yhteydellä toimiva SD-kortinlukija, josta SD-kortti voidaan irrottaa ja käyttää näin myös SD-muistikorttipaikassa.



Kuvio 11. HVPG4F6 flash-muistiipiiri USB-muistitikussa



Kuvio 12. UP 21 -ohjainpiiri sekä muita komponentteja USB-muistitikussa

USB-väylätekniikan standardit on kehittänyt kansainvälinen yrityscoalatio nimeltään Universal Serial Bus Impleters Forum eli lyhennettynä USB-IF. USB-IF:ää johtaa tällä

hetkellä HP, Microsoft, Intel, Renesas Electronics sekä STMicroelectronics. USB-IF on voittoa tavoittelematon korporatio, joka tarjoaa työkaluja USB-laitteiden kehitykseen. (Axelson 2009, 7; USB-IF:n nettisivut 2015) USB-IF on luonut yhteensopivuuden testausohjelman nimeltään Compliance Program. Tällä USB-laitteiden valmistajat voivat varmistaa valmistamiensa USB-laitteiden täyttävän USB IF:n kehittämät laatustandardit. Laitteet, jotka täyttävät standardien vaatimukset, saavat oikeuden käyttää USB-IF -logoa (Axelson 2006, 7.) Eri USB-laitteet standardien versiosta 3.0 versioon 1.0 pystyvätkin kommunikoimaan toistensa kanssa (Mts. 12-13). Huomattavaa kuitenkin on, että standardin eri versioiden tuki on erilainen liittyen maksimiin tiedonsiirtonopeuteen sekä virran kuljettamiseen. Tämä voi aiheuttaa yhteensopivuusongelmia, jos laitepari ei osukaan määritysten sisään.

2.9 Tiedostojärjestelmä

Tallennusvälineissä täytyy olla tiedostojärjestelmä, jolla tiedonhallintaa hoidetaan. Tiedostojärjestelmä hoitaa muun muassa tiedostopolkujen hierarkian kansioden ja yksittäisen tiedostojen välillä. Lisäksi tiedostojärjestelmä vastaa tiedoston nimen, alkukohdan, koon sekä käytettävien allokaatioyksiköiden säilytyksestä. (Bunting 2012, FileSystem.) Tallennusvälineet säilyttävät tietoa allokaatioyksiköissä, jotka koostuvat useista sektoreista. Sektori on pienin fyysinen paikka levyllä, johon tietoa voi tallentaa (Mts. Computer Hardware). Allokaatioyksikkö taas on tallennusvälineen muistitilan kiinteä varausyksikkö, jota tiedostojärjestelmä käyttää. Kiinteän kokonsa takia tiedosto vie aina minimissään yhden allokaatioyksikön verran tilaa. Tiedoston koon ollessa yli yhden allokaatioyksikön, jaetaan sen säilytys useampiin allokaatioyksikköihin. Tiedostojärjestelmän vastuulla onkin pitää kirjaa tyhjästä ja täysistä allokaatioyksiköistä muistissa, jotta tietoa ei vahingossa ylikirjoiteta. Tiedostojärjestelmät ovat standardisoituja, jotta tallennusvälinettä voidaan käsitellä eri laitteilla ja sovelluksilla. (Mts. FileSystem; Mts. NTFS Basics.) Huomattavaa myös on, että tiedostojärjestelmän komponentit sijaitsevat tallennusvälineessä ja täten vievät tilaa sieltä.

2.9.1 FAT

FAT eli File Allocation Table on vuonna 1977 esitelty tiedostojärjestelmä. FAT-tiedostojärjestelmiä voidaan käyttää useilla käyttöjärjestelmillä. FAT:ia hyödynnetään vielä nykyäänkin paljon flash-muisteissa. FAT-tiedostojärjestelmä perustuu kahdennettuun tiedostoalokaatiotaulukkoon, joka sisältää tallennusvälineen kaikkien tiedostojen nimet, koot, klusterit sekä tiedostoon liittyvään muun datan. Taulukoita on kaksi, jottei yhden taulukon vahingoittuminen sekoita koko muistia. FAT-tiedostojärjestelmästä on julkaistu vuosien aikana useita eri versioita, kuten FAT12, FAT16, sekä FAT32. Nimen perässä oleva numero kertoo tiedostoalokaatiotaulukon yhden tiedoston tiedoille varatun kentän koon tavuina. Versiot eroavat myös jonkin verran käynnistysohjelmien osalta. (Mts. CHP2 FAT Basics.)

FAT-tiedostojärjestelmän varsinaiset tiedostot sijaitsevat muistissa erillään taulukosta ja ne etsitään tiedostoalokaatiotaulukon perusteella. FAT-tiedostojärjestelmästä tiedostoa poistettaessa, tiedostoalokaatiotaulukossa tiedoston varaamat klusteripaikat kirjoitetaan vapaiksi käyttöä varten. Itse data jätetään muistiin koskemattomaksi, jolloin se on mahdollista saada palautetuksi. (Mts. CHP2 FAT Basics; Mts. Overview.) Kuitenkin jotkut formatointiohjelmat, kuten Windows 7:n hitaampi formatointityökalu, poistavat myös itse datan taulukon modifioimisen sijaan. (Mts. NTFS Basics).

2.9.2 exFAT

ExFAT eli Extended File Allocation Table on Windows CE 6.0:n yhteydessä 2006 julkaistu tiedostojärjestelmä. ExFAT on kehitetty juuri flash-muisteja varten. ExFAT:a kutsutaan joissakin yhteyksissä myös FAT64:ksi, vaikkakin sen tiedostoalokaatiotaulukko sisältää 32 tavuiset kentät. ExFAT eroaa muista FAT-perheen tiedostojärjestelmistä siinä suhteessa, että vapaata tilaa seurataan bittikartalla NTFS:n tapaan. (Mts. exFAT Basics.)

2.9.3 NTFS

NTFS eli New Technology File System on esitelty vuonna 1993 korvaamaan FAT:in puutteet. FAT:iin verrattuna NTFS mahdollistaa paremmat tiedostonhakuominaisuudet, kohennetut turvallisuusominaisuudet, pidemmät tiedostonimet sekä yksittäisen tiedoston pakkaamisen. Windows käyttöjärjestelmien oletustiedostojärjestelmä on ollut NTFS Windows XP:stä lähtien. NTFS:tä on niin ikään tehty useita eri versioita, ja Windows 7:ssä onkin versio numeroltaan NTFS 3.1, vaikkakin joskus käytetään sekaisin ohjelmistokomponentin versiota NTFS.sys 6.1. NTFS-tiedostojärjestelmä sisältää FAT:n tiedostoalokaatiotaulukkoa vastaavan päätiedostotaulukon *\$mft* sekä bittikartan *\$bitmap*. Päätiedostotaulukolla seurataan esimerkiksi tiedostojen nimiä, kokoa sekä luontiaikaa. Bittikartalla taas pidetään yhtä bittiä kääntämällä kirjaa vapaista ja varatuista alokaatiopaikoista muistissa. NTFS:n erikoisominaisuus on, että päätiedostotaulukkoon voidaan tallentaa pienikokoisia tiedostoja. Isommat tiedostot jaetaan klusteripaikkoihin ja taulukkoon lisätään paikkojen numerot. NTFS-tiedostojärjestelmässä poistaminen yleensä myös muokkaa vain tiedostoon johtavia taulukkotietoja, ei itse tiedostoa. (Mts. NTFS Basics.)

2.10 CSV-tiedostomuoto

CSV-tiedosto on taulukkomuotoinen tekstitiedosto. Nimi CSV tulee englanninkielisistä sanoista Comma Separated Values. Tiedostolle on tyypillistä se, että tietueet ovat erotettu toisistaan ennaltasovitun merkin avulla. Erotusmerkki on yleensä pilkku, mutta se voidaan ohjelmallisesti vaihtaa. Kukin tietue tai pystyrivi erotetaan toisistaan tällä erotusmerkillä. Kukin vaakarivi lopetetaan rivinlopetusmerkkijonolla, joka on ASCII-merkistössä joko "10" tai "13" Windowsilla, "LF" Unixissa ja "CR" Macintosh-tietokoneissa. (CSVReader.com artikkeli .CSV-tiedostomuodosta n.d) CSV-tiedostoja voidaan hyödyntää viemään taulukkomuotoista dataa ohjelmasta toiseen. CSV-tiedoston formaatti esitellään SolidMatrix Technologies Incorporatedin muistiossa RFC 4180. (Shafranovich 2005.)

CSV-tiedostoja käsiteltäessä on tärkeää se, että lainaus-, rivinvaihto- tai erotusmerkkejä käytetään vain niille tarkoitetuissa paikoissa. Esimerkiksi desimaalilukujen erottimena tietyissä maissa toimiva pilkku tulisikin sulkea lainausmerkkien sisään, jotta voidaan erottaa desimaali- ja pystyrivierotin toisistaan. Lainausmerkit CSV-tiedostoissa osoittavat tekstin alkamista ja loppumista, joten esitettäväksi haluttavat lainausmerkit tulee olla myös lainausmerkkien sisällä. Vaihtoehtoinen tapa on käyttää erotusmerkinä esimerkiksi puolipistettä. Microsoft Excel 2013 taulukkolaskentaohjelmassa käytettävää listaerotinta voidaan vaihtaa asetuksista. Jotkut sovellukset eivät myöskään huomioi yksittäisiä välilyöntejä, jos niitä ei ole laitettu lainausmerkkien sisään. (CSVReader.com artikkeli .CSV-tiedostomuodosta n.d.)

3 Toteutus

3.1 Suunnittelu

Automaatiossa laatu muodostuu tuotteen kyvystä täyttää asiakkaan suorat tai epäsuorat tarpeet. Asiakkaan määrittelemien laatuominaisuuksien puuttuminen aiheuttavat välitöntä tyytymättömyyttä tuotteeseen. Osa ylimääräisistä laatuominaisuuksista, vaikka asiakas ei näitä olekaan määritellyt, taas lisää tyytyväisyyttä ja laadukasta vaikutelmaa tuotteesta. (Ajo, Hakonen, Harju, Järvi, Kaskes, Lenardic, Niukkanen, Nurminen, Ritala, Tolppanen & Tommila 2001, 5.) Lopullinen tuotteen laatutaso määräytyy määritettyjen toimintojen, resurssien, ja laitteiston mukautuvuuden perusteella.

Siemens historical data -työkalu tarjoaa perustasoisen ratkaisun datankeruuseen. Siinä on kuitenkin rajoituksia, joita ilman käyttäjän kokema laatutaso voisi olla huomattavasti parempi. Keskusteluissa toimeksiantajan kanssa nousi esille muassa mahdollisuus järjestää datahistoriatiedostot viikkokohtaisiin kansioihin, tiedostonimen seuraaminen sen hetkistä päivämäärää sekä datan esittämisen parantaminen. Näiden huomioiden takia syntyi päätös korvata Siemensin oman historical data -työkalu luomalla korvaava skripti WinCC scripts editorilla. Historical data -työkalun rajoitusten takia yhdeksi työtä ohjaavista periaatteista muodostui mahdollisimman suuri modulaarisuus, jotta valmis tuote olisi muokattavissa useaan käyttökohteeseen.

Työn toteuttaminen aloitettaisiin skriptin luomisella WinCC:hen. Skripti tulisi testata ja hyväksyä ennen siirtymistä seuraavaan vaiheeseen. Excel VBA skriptillä käsitellään WinCC skriptin luomia historialokeja. Tämän takia on perusteltua, että pohjana toimiva WinCC-skripti on lopullisessa muodossa, jotta tämän pohjalle syntyvä Excel VBA-skripti ei muuttuisi aina WinCC-skriptin muuttuessa. Kummatkin skriptit testattaisiin erikseen sekä yhdessä. Lopulta ohjelmista luotaisiin helppokäyttöisiä paketteja, jotta niiden käyttöönotto ja liittäminen projekteihin olisi yksinkertaista.

Työn toteutusvaihe tehtäisiin englannin kielellä sen ollessa universaali kommunikointikieli ympäri maailman.

3.2 WinCC-skriptit

WinCC-skriptit luotiin WinCC:n omalla scripts editor -koodinkäsittelyohjelmalla. Koodi kommentoitiin kauttaaltaan englannin kielellä, jotta myös muut kuin itse kehittäjä pystyvät sitä tarvittaessa ymmärtämään. WinCC-skriptejä tehtiin kaksi, jotta varsin erilaiset toiminnot ovat selkeästi omissa skripteissään. Skriptien nimet ovat DataLogging sekä FileMonitoring.

3.2.1 Datalogging-skripti

Varsinainen datankeräys tapahtuu WinCC-skriptillä nimeltään DataLogging. Se sisältää noin 220 riviä hyvin jäseneltyä koodia. Skripti hakee DataLogging Values taulukosta arvoja ja vie nämä CSV-tiedostoon käyttäjän valitsemaan tallennuspaikkaan. Käyttäjällä on mahdollisuus vaikuttaa taulukkoon tuleviin tietoihin vaihtamalla asetuksia DLSettings-tagitaulukosta.

DataLoggingValues-tilukon arvoihin viitataan vakioilla nimellä, jossa on kuitenkin juokseva järjestysnumero. Taulukko sisältää myös tallennettavien arvojen lukumäärää esittävän kokonaislukumuuttujan. Kyseisellä kokonaislukumuuttujalla määritetään, että montako arvoa taulukosta lisätään CSV-tiedostoon ensimmäisestä lähtien. Tämä mahdollistaa sen, että käyttäjä voi itse kasvattaa tallennettavien arvojen määrää lisäämällä DataLoggingValues-tilukkoon samalla logiikalla nimettyjä arvoja ja kasvattamalla arvojen lukumäärämuuttujaa.

Skripti yhdistää tiedostonimet käyttäjän syöttämästä alkuosasta sekä sen hetkisestä käyttöjärjestelmän päivämäärästä. Yksittäiset tiedostot järjestellään viikkokohtaisiin kansioihin, joiden nimi kehitetään vakio-osan sekä viikkonumeron perusteella.

Lokitiedostoihin lisättävä ajan arvo lasketaan käyttäen hyväksi VBS-funktiota, joka palauttaa millisekuntiarvon keskiyöstä laskien. Ajan arvo muotoillaan nolliä tarvittaessa lisäämällä aina vakiomittaiseksi lisäselkeyden saavuttamiseksi.

DataLogging-skripti sisältää lukuisia suorituksen aikana tehtäviä tarkistuksia, joilla valvotaan skriptin toimintaa. Skripti sisältää lisäksi VBS-rakenteen, jolla on mahdollista hylkiä virhetilanteita. Jos skriptissä tehtävä tarkastus ei mene läpi, annetaan käyttäjälle kustakin virhetilanteesta yksilöllinen hälytys sekä jätetään skriptin suoritus kesken. Suorituksen loppumisesta johtuen skripti antaa hälytyksen vain ensimmäisenä löytyneestä virheestä, josta johtuen virhetarkastukset on pyritty järjestämään niin, että perustavanlaatuisin tarkastetaan ensin.

Virheentarkastusominaisuus on luonnollisesti kytkettävissä pois käytöstä.

DataLogging-skripti tarkastaa suorituksen aikana seuraavat asiat:

- Tiedostonimi on syötetty parametreihin
- Tallennusvälineen tiedostopolku on syötetty parametreihin
- Tallennusväline on saatavilla
- Viikkokohtaisen kansion luonti onnistui
- Seurattavien arvojen lukumäärä on syötetty oikein
- Lokitiedoston luonti onnistui ensimmäisen kirjoituksen yhteydessä
- Lokitiedosto on saatavilla kirjoittamista varten myöhemmillä kerroilla
- Lokitiedostoon kirjoittaminen onnistui
- Yksittäisen lokitiedoston koon määrittäminen onnistui
- Yleinen Visual Basic Script -toimintavirhe

3.2.2 FileMonitoring-skripti

FileMonitoring-skripti on lokitiedostojen ominaisuuksien valvontaan kehitetty sovellus. Se sisältää noin 210 riviä kommentoitua ja jäsenneiltyä Visual Basic Script -koodia. FileMonitoring-skriptiä suoritetaan aliohjelman DataLogging-skriptissä. Suorittamisen saa kuitenkin kytkettyä pois päältä.

Skripti luo ensimmäisellä suorituskerrallaan tiedoston nimeltään FileMonitoring.csv. Tällöin tiedostoon tallennetaan muun muassa sen hetkinen ajan arvo sekä muita toiminnan kannalta oleellisia muuttujia. Seuraavilla suorituskertoilla jo luodusta tiedostosta haetaan rivejä ohjelman muistiin. Näistä riveistä kaivetaan esille hyödyllinen data, jota käytetään uusien arvojen laskennassa. Esimerkiksi kortilla olevan kokonaistietomäärän kasvun laskemiseen tarvitaan nykyisen sekä edellisen suorituskerran tiedoston koko.

FileMonitoring-tiedosto sisältää tietoa seuraavista asioista:

- Skriptin ensimmäisen suorituskerran päivämäärä
- Skriptin ensimmäisen suorituskerran kellonaika
- Skriptin viimeisimmän suorituskerran päivämäärä
- Skriptin viimeisimmän suorituskerran kellonaika
- Viimeisimpänä käsitellyn tiedoston vaakarivimäärä
- Viimeisimpänä käsitellyn tiedoston pystyrivimäärä
- Viimeisimpänä käsitellyn tiedoston solumäärä
- Viimeisimpänä käsitellyn tiedoston koko tavuina
- Kaikkien tiedostojen yhteinen vaakarivimäärä
- Kaikkien tiedostojen yhteinen pystyrivimäärän painotettu keskiarvo
- Kaikkien tiedostojen yhteinen solumäärä
- Tallennusvälineen kokonaistietomäärä, yksikkö vaihtuu dynaamisesti arvon suuruuden mukaan

- Alku- ja loppuajan erotus muodossa hh:mm:ss

FileMonitoring-skripti tarkkailee DataLogging-skriptin tavoin toimintansa virheettömyyttä. FileMonitoring-skriptin virhetarkkailu on kytkettävissä pois päältä samalla asetuksella kuin DataLogging-skriptinkin. Hälytysviesteissä on mainittu, että kummassa skriptissä virheellinen tilanne tapahtui.

FileMonitoring-skriptin seuraavista virhetilanteista annetaan käyttäjälle hälytysviesti:

- Tiedostojärjestelmän varausyksikön koko on syötetty väärin
- Tiedostoa ei voitu luoda
- Tiedostoon ei saatu kirjoitettua
- Yleiset järjestelmän virheet

3.2.3 DLSettings-taulukko

DLSettings tag table –taulukko sisältää seuraavanlaisia muuttujia (Taulukko 2), joilla voidaan muuttaa skriptien toimintaa käyttäjän haluamaan suuntaan. Taulukossa säilytetään myös skriptien toimintaan liittyviä apumuuttujia.

Taulukko 2. DLSettings-tagitaulukon rakenne

Nimi	Tyyppi	Tarkoitus
AlarmMasking	Bool	Määrittää, että näytetäänkö sama hälytys yhden vai useamman kerran
AlarmState	USInt	Apumuuttuja hälytyksen identifioivalle numerolle
AllocationUnit	Dint	Tallennusvälineessä käytettävä varaussyksikön koko tavuina
Cells	Dint	Viimeisimmän tiedoston solumäärä
Columns	Dint	Viimeisimmän tiedoston vaakarivimäärä
Counter	Dint	Apumuuttuja FileMonitoring skriptin suorituskerroille
Disable	Bool	Datalogging-skriptin poiskytkentä
DisableAlarming	Bool	Hälytysviestien poiskytkentä
DisableDate	Bool	Taulukkoon lisättävän päivämäärän poiskytkentä
DisableMillisecs	Bool	Millisekuntien poiskytkentä taulukkoon lisättävästä ajan arvosta
DisableSecs	Bool	Sekuntien poiskytkentä taulukkoon lisättävästä ajan arvosta
DisableTime	Bool	Taulukkoon lisättävän ajan arvon poiskytkentä
FileMonitoringCycle	Bool	FileMonitoring-aliohjelman suoritusväli ohjelman suoritusmäärinä
Rows	Dint	Viimeisimmän tiedoston pystyrivimäärä
ScriptRun	Bool	Skriptin suoritukseen liittyvä vahtikoirabitti
TotalFileLength	Wstring	Tallennusvälineen käytetty muisti

3.3 Excel VBA skripti

Excel VBA-skripti nimeltään DataloggingReportHandling luotiin käyttämällä Excel 2013:sta olevaa VBA-koodinkehitysympäristöä. Excelin VBA osuus sisältää kaikkiaan noin 1000 riviä koodia. Koodi on jäsennelty ja sisennetty syntaksirakenteiden mukaan loogisesti. Se on myös kommentoitu kauttaaltaan. Koko Excel osuus on täysin englanninkielinen. Ohjelman käyttöliittymä on Excel taulukko, johon on lisätty ActiveX-painikkeita (kuvio 13) sekä soluja, joilla tietoa saadaan visualisoitua.

Kuvio 13. JEEC Datalogging report handling -käyttöliittymä

Koko ohjelma perustuu siihen, että käyttäjä hakee haluamastaan sijainnista Windowsin tiedostodialokitoiminnolla tiedoston. Tiedostodialokitoiminto sisältää suodattimen, joka näyttää pelkät CSV-datalokitiedostot. Tämän tiedoston sisältämä data kopioidaan erilliseen taulukkoon. Tiedostosta esitetään käyttäjälle sen koko sekä datataulukon vietävä solujen määrä sekä pysty- ja vaakarivien lukumäärä. Näitä vaaka- ja pystyrivien lukumääriä hyödynnetään muissa toiminnoissa datataulukon koon määrittämisessä. Ohjelma pystyy käsittelemään sekä datalogging-skriptillä että Siemensin historical data -toiminnolla tehtyjä tiedonkeräyslokeja. Datan sisältävään taulukon taustaan voidaan painikkeella valita joko korkeakontrastinen tai täysin valkoinen värimaailma. Solujen väliset reunarajat ovat niin ikään kytkettävissä päälle ja pois yhdellä painikkeella.

Excel osiolla voidaan luonnollisesti luoda raportteja datataulukon tiedoista. Raportit viedään niin ikään omalle taulukkolehdelleen, joka nimetään joko pelkän päivämäärän tai panosnumeroiden sekä päivämäärän perusteella riippuen kyseessä olevasta raportista. Käyttäjän syöttämä prosessin nimi lisätään raportin otsikoksi. Ohjelma sisältää tarkistuksen raporttitaulukoiden nimille, ja kahden samannimisen raportin tapauksessa myöhemmin luotavaan lisätään järjestysnumero perään. Raportinluontitoiminto olettaa, että dataloki on tehty tietyn mallin mukaan, koska viittaukset ovat kirjoitettu koodiin solujen paikan perusteella. Raporttien luonti tapahtuu kullekin raporttityypille omalla ActiveX-painikkeella.

DataloggingReportHandling-sovelluksella luotavat raportit ovat:

- Panosraportti (Liite 1)
- Panosperusteinen tuotantoraportti (Liite 2)
- Päiväperusteinen tuotantoraportti (Liite 3)
- Kulutusraportti (Liite 4)

Panosraportti sekä panosperusteinen raportti tarvitsevat tiedon, että mistä panoksista raportit luodaan. Panosnumeroiden syöttö on toteutettu hyödyntämällä painiketta painettaessa esille pomppaavaa lomaketta, joka hakee datan sisältävästä taulukosta kyseisen päivän panoksien numerot. Käyttäjä voi täten valita lomakkeesta haluamansa panosnumerot. Panosperusteisen tuotantoraportin luonnin yhteydessä olevassa lomakkeessa valitaan kaksi eri panosta, ja skripti on luotu niin, että käyttäjä ei voi valita ensimmäistä panosnumeroa suuremmaksi kuin toista ja päinvastoin.

DataloggingReportHandling sisältää toiminnon, jolla taulukoista voidaan viedä raportteja tallennettavaksi Windowsin polunvalintaikkunalla valittavaan paikkaan. Ominaisuus on toteutettu käyttämällä painikkeesta esiin tulevaa lomaketta, joka listaa kaikki luodut raporttitaulukot ja kysyy käyttäjää valitsemaan tallennettavat raportit. Lomake sisältää asetuksia, kuten raporttien yhdistäminen yhteen tiedostoon sekä automaattinen avaaminen tallentamisen jälkeen. Raportit voidaan viedä valinnan mukaan .XLSX, .XLS tai .PDF-tiedostomuotoon.

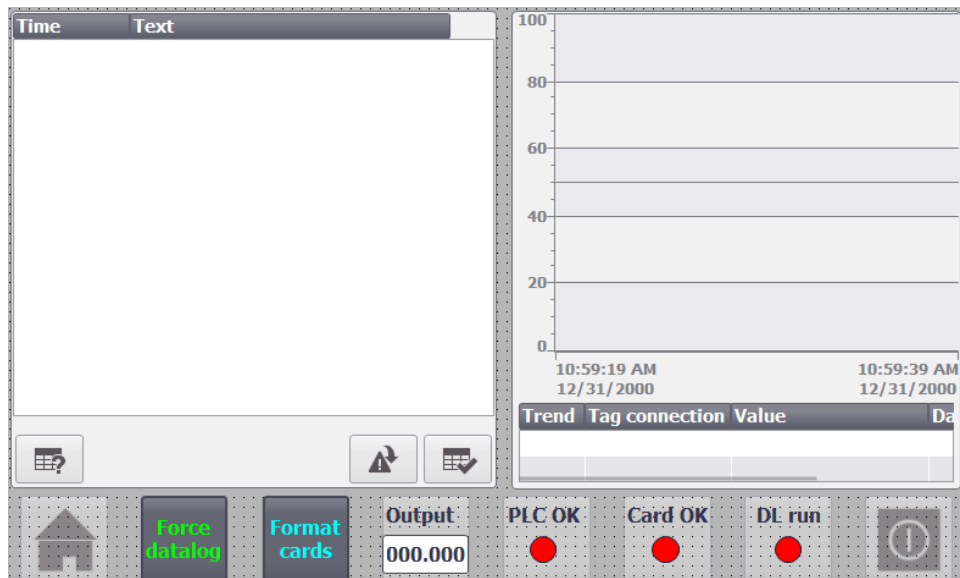
3.4 Testaaminen

Skriptien testausta tapahtui luomisen yhteydessä sekä sen jälkeen. Testausta toteutettiin pääasiassa siten, yhden valmiin osion toiminta testattiin välittömästi sen luonnin jälkeen. Tällä tavoin sovellus pysyi koko ajan virheettömänä, eikä uusia ominaisuuksia rakennettu vanhojen, mahdollisesti viallisten rakenteiden päälle. Lisäksi isompia kokonaisuuksia sekä skriptejä käytettiin valmistumisen jälkeen samalla tavoin, kuin käyttäjä ajaisi valmista sovellusta. Näin oli mahdollista löytää epäloogisuuksia, ajatusvirheitä ja sellaisia ongelmia, joihin asiakas mahdollisesti sovellusta hyödyntäessään törmäisi. Tällä tavoin oli mahdollista saada myös arvokasta käyttäjäkokemusta valmiista tuotteesta.

WinCC:n sekä Excelin Visual Basic -koodieditorit sisältävät automaattisen reaaliaikaisen tarkastuksen käyttäjän kirjoittamalle koodille. Tämä eliminoi jo

suurimman osan syntaksivirheistä, vaikkakin koneellinen tarkastelu ei välttämättä huomaa kaiken tyyppisiä virheitä. Esimerkiksi WinCC:ssä tiettyjen ActiveX-objektien käyttö meni kääntäjästä läpi, vaikkakaan WinCC Scripting editor ei kyseisiä rakenteita tue. Excelin Visual Basic IDE sisältää lisäksi askellustoiminnon, jolla koodia on mahdollisuus suorittaa rivi kerrallaan sekä tarkkailla samalla muuttujien arvoja. Tästä tarkan seuraamisen mahdollistavasta toiminnosta oli hyvin paljon hyötyä koodin epäjohdonmukaisuuksia selvittäessä. Kuitenkin testausta ja vikojen etsimistä häiritsi jollain tapaa Visual Basicin ominaisuus, että muuttujia ei välttämättä tarvitse esitellä ennen käyttöä. Jos muuttujan nimessä esiintyi vaikkapa jossain kohtaa koodia kirjoitusvirhe, ohjelma toimi, vaikka tässä kohtaa arvot eivät luonnollisestikaan siirtyneet erinimisestä muuttujasta.

WinCC skriptien testausta suoritettiin käyttäen Siemensin CPU 315-2PN/DP ohjelmoitavaa logiikkaa, johon oli kytketty Ethernet-kaapelilla KP700 Comfort -näyttöpaneeli. CPU suoritti ohjelmaa, joka arpoi satunnaisia lukuja perustuen yhden ohjelmakierron suoritusaikaan sekä sinifunktioon. Nämä luvut vietiin CPU:lta näyttöpaneelin muuttujiksi ja siitä edelleen tallennusvälineelle skriptin avulla. Toimintaa valvottiin varta vasten luoduilla näyttökuvilla (kuvio 14), jossa oli tarpeellisia toimintoja testausta varten. Skripteistä on mahdollista viedä virhetilanteen tietoja järjestelmän hälytyksiksi, ja nämä hälytykset voidaan esittää kehittäjälle WinCC:n oman hälytysobjektin avulla. Tämän kaltaisella järjestelmällä voitiin kokeilla sovellusta osana automaatioprosessia sekä etsiä mahdollisia ohjelmointivirheitä. Tallennusvälineen tallennettuja datalokeja tarkasteltiin testausseSSIoiden jälkeen Excel taulukkolaskentaohjelmalla. Näin pystyttiin toteamaan, että datan kerääminen toimii ja historialokin muotoilu on halutun kaltainen. Laitteisto jätettiin myös muutamia kertoja keräämään dataa yön yli, jotta nähtäisiin sen toiminta suurien datamäärien käsittelyssä.



Kuvio 14. WinCC testausnäyttö

4 Tulokset

4.1 Tulosten arviointi

Opinnäytetyöprosessin tuloksena syntyi ohjelmistokokonaisuus, joka kerää sekä prosessoi dataa teollisuuden automaatiolaitteista. Datasta voidaan luoda toimeksiantajan määrittelemiä raportteja ja näitä raportteja voidaan viedä ja tallentaa käyttäjän määrittelemään tiedostosijaintiin. Itse sovellukselle annetut päätavoitteet siis täyttyivät. Sovellukseen lisättiin monimutkaisia virhe- ja poikkeustilannemekanismeja, joilla turvataan oikea toiminta kaikissa tilanteissa. Sovellus sisältää erilaisia valintoja asetuksille, joiden ansiosta se voidaan sovittaa mitä erilaisimpiin tilanteisiin.

Koska WinCC ei tarjoa valmista järjestelmää, jolla voitaisiin helposti tarkastella tiedostojen kokoja, päädyttiin luomaan WinCC-skripti, joka valvoo tiedostojen kokoja tallentamalla niiden ominaisuuksia tekstitiedostoon. WinCC scripts editor sisältää FileLen-funktion, jolla pystytään määrittämään avoinna olevan tiedoston koko. Hyödyntämättä jätettiin kuitenkin rakenne, jossa tallennusvälineen täyttöaste lasketaan siten, että jokainen tiedosto avataan ja käytetään summaamiseen FileLen-funktiota. Tämän ajateltiin olevan liian hidas suurten tiedostomäärien kanssa vähäresurssisella laitteistolla. FileLen-funktiota hyödynnetään kuitenkin, kun FileMonitoring-skriptissä määritetään jo valmiiksi avoinna olevan tiedoston koko. FileMonitoring-skriptin ominaisuuksien avulla käyttäjälle on mahdollista esittää tallennusvälineen sisältämä datamäärä. Tätä ominaisuutta ajateltiin käytettäväksi myös automaattisen tilanraivauksen yhteydessä, mutta kehitetty kokeellisen skriptin suoritus aika osoittautui aivan liian hitaaksi. Yhden viikkokansion tyhjentäminen tällä skriptillä kesti noin viidestä kymmeneen sekuntia. Puuttuva tilanraivausominaisuus korvattiin Excel osioon lisätyllä tuella Siemensin omille datalokeille, joihin kyseinen ominaisuus on mahdollista sisällyttää.

Sovellusta on testattu laajasti koko sen kehityskaaren ajan. Testauksesta saaduista huomioista voidaankin päätellä skriptin toiminnan virhevapautta. Datankeräys paneelilta toimii ja taulukon muotoilu on halutun kaltainen. FileMonitoring laskee tiedostojen koon oikein ja osaa ottaa huomioon tiedostojärjestelmän varausyksikköjen vaikutuksen. Siemensin paneelien toimintojen priorisoinnin takia paneelin toiminnot eivät häiriinny skriptin suorituksesta. Excel VBA-osuus pystyy hakemaan tiedot kortilta ja luomaan määrätyt raportit. Kuitenkin kovakoodattujen viittausten ja vain yhdenlaisen sallitun datalokimuodon takia tietyt erilaiset ratkaisut voivat vaatia itse Excelin VBA-koodiin kajoamista. Toiminnan voidaan sanoa kuitenkin pääpiirteittäin olevan virhevapaa ja halutunlaista.

Datalogging-ohjelmistokokonaisuus sekä Siemens historical data ovat hyvin pitkälti vastaavia tiedonkeräystyökaluja. Joitakin erilaisuuksia kuitenkin löytyy. Ehkäpä perustavanlaatuisin eroavaisuus lienee se, että Siemens historical data -työkalu käsittelee kerättäviä tietoja yksi kerrallaan. Esimerkiksi tiedonkeräysaika tallennetaan kullekin muuttujalle erikseen. Datalogging-skripti taas hakee yhden tiedonkeräysajan ja tallentaa koko muuttujalistan läpi kerrallaan. Tästä johtuen Siemensin lokitaulukko on aina yhtä leveä ja kasvaa vain pystysuuntaan. Datalogging-skriptin dataloki leviää muuttujien lisääntyessä myös vaakasuuntaan. Siemensin historian data -työkalu sisältää edistyneitä toimintoja, esimerkiksi vanhojen lokien päälle tallentamisen tallennusvälineen täyttyessä. Tällainen toiminto on erittäin vaikea toteuttaa WinCC scripting editorin rajoittuneilla työkaluilla, joten se puuttuu datalogging-skriptistä. FileMonitoring-skripti tarjoaa kuitenkin näytölle vietävään tagiin kortin nykyisen täyttöasteen tekstimuodossa. DataLogging-skriptin etuja ovat selkeämmät taulukot, selkeä kansiorakenne tallennusvälineellä ja aavistuksen laajemmat asetukset. Siemens historian data -työkalu tarjoaa taas ison brändin valmiiksi testatun ratkaisun. Yhtä kaikki mahdollisimman suuren modulaarisuuden ajatusta tukee se, että kummallakin työkalulla tehtyjä lokeja voidaan lukea Excel pohjaisella DataloggingReportHandling-työkalulla.

Yksi toimeksiantajan tavoitteista oli saada valmista tiedonkeruuratkaisua tarjottua myyntiin. Tästä tietenkin seuraa, että ohjelmistopakettien tulisi olla käyttöliittymältään ja ilmeeltään kaupallisen oloinen. Käyttöliittymän operointia on testattu runsaasti ja hiottu sitä myöten käyttäjäystävällisempään suuntaan. Objektien sijoittelussa on pyritty hyödyntämään koulussa opittuja oppeja selkeydestä, käytettävyydestä sekä neutraaleista väreistä. Myös operoinnin helpottaminen esimerkiksi yksikäsitteisillä painiketeksteillä ja tarpeenmukaisilla oletusvalinoilla tukee tuotteen valmiutta kaupalliseen käyttöön. Käyttöliittymään lisättiin myös toimeksiantajayrityksen logo ikään kuin mainoksena. Kuitenkin huomioitavaa on, että lopullisen arvion ohjelmistopakettien kaupallisen aspektin onnistumisesta näkee vasta asiakkaiden yleisestä vastaanotosta sekä palautteesta oikeassa käytössä.

4.2 Jatkokehitys

Opinnäytetyöprosessin aikana syntyi erilaisia ideoita ohjelmistokokonaisuuden jalostamiseksi parempaan suuntaan. Lisäksi sovelluksen vaatimukset ja toimintojen käyttökelpoisuus tarkentuu, kunhan sitä vain on käytetty jonkun aikaa osana teollisuusprosessia. Koska sovelluspaketti on kaupallinen tuote, tullaan ominaisuuksia varmasti lisäämään ja puutteita korjaamaan sitä mukaa, kun palautetta saadaan ja aikataulut antavat myöten.

Suurin opinnäytetyötä tehdessä esiintynyt huolenaihe oli ohjelmiston tietoturvallisuus sekä siihen liittyvät riskit. Windows CE:n sanotaan olevan paremmin turvassa viruksilta kuin työpöytä Windowsit riisutun ja mobiilikäyttöjärjestelmämäisen olemuksensa takia (Tilley, 2006). Kuitenkin ajoittain vaihdettavat tiedot automaatiolaitteiston ja PC:n välillä mahdollistavat uuden rajapinnan hyökkäyksille automaatiolaitteistoa kohtaan. Tätä voitaisiin torjua esimerkiksi lisäämällä älykkäitä tarkastuksia tiedonsiirron yhteyteen tai varmistamalla, että tiedot ovat kontaktissa vain varmasti virusvapaisiin laitteisiin.

Itse ohjelmistoon liittyen sovelluksen toimintaa voisi parantaa optimoimalla WinCC skriptin suoritusnopeutta HMI:n puolella. Vähäresurssisella laitteistolla koodin keveys sekä sulava suoritus on elintärkeää, ettei käyttäjä havaitse laatuvaikutelmaa heikentävää käyttöliittymän tökkimistä tai hidastelua. Skriptin suoritusta voisi varmastikin nopeuttaa entisestään tutkimalla erilaisten koodirakenteiden suoritusnopeutta ja tämän perusteella järkeistää sovellusta. Lisäksi Excel-skriptiin kovakoodatut soluviittauksien vaihtaminen algoritmeilla määritettyihin viittauksiin mahdollistaisi skriptin käytön myös erilaisten datalokipohjien kanssa.

Tavoitteissa oli ollut rakenne, jolla tallennusvälineen totaalinen täyttyminen estetään. Tämä saattaisi olla mahdollista toteuttaa, vaikkakin asia vaatii syvempää perehtymistä aiheeseen. Yksi ideoitu ratkaisu olisi käyttää datankeruuvälineistönä näyttöjen sijaan itse automaatio-ohjaimia. Siemensin uuden sukupolven ohjelmoitavat logiikat mahdollistavat datan keräämisen PLC:n omaan muistiin tai siihen asetetulle muistikortille. Nämä datalokit ovat haettavissa internetselaimella PROFINET:in kautta PLC:ltä, jolloin erillistä ajottain tapahtuvaa muistikortin vaihtoa ei tarvitsisi suorittaa. (Siemens WinCC Advanced V13 SP1-järjestelmämanuaali, 2806-2808.) Tämä mahdollistaisi järjestelmän, jossa data haetaan automaattisesti PLC:ltä ja arkistoidaan suoraan raportteina tietokoneelle. PLC:n muistista datalokit tyhjennettäisiin aina, kunhan vanhat datalokit ovat arkistoitu. Järjestelmän etuna olisi suljettu rakenne ja nykyisten PC-tietokoneiden muistien koko rajoittaisi raporttien määrää vasta kymmenien vuosien jälkeen. Lisäksi raporttien luonti vaatisi hyvin vähän tai ei ollenkaan työpanosta operaattorilta.

Toimeksiantajan liiketoiminta käsittää kotimaan lisäksi kansainvälisetkin projektit. Tästä johtuen ohjelmistopakedin kielen kääntäminen muutamalle eri variaatiolle mahdollistaisi toimeksiantajalle mahdollisuuden tarjota vielä asteen enemmän räätälöityjä ratkaisuja asiakkaidensa tarpeisiin. Operaattorille tuttu kieli auttaisi keskittymään itse sovelluksen käyttöön eikä aikaa kuluisi epäolennaiseen toimintaan, jolloin asiakkaan kokema hyöty lisääntyisi.

5 Pohdinta

Viime aikoina kasvavana trendinä tuottavassa teollisuudessa on ollut raaka-aineiden sekä energian säästö. Tästä luonnollisesti seuraa entistä suurempi tarve seurata prosesseja tarkemmin, mikä tekee työstä varsin ajankohtaisen. Aiheesta on luonnollisesti kirjoitettu runsaasti, mutta huomioon otettavaa on, että varsinkin massamuistilaitteisto uudistuu tällä hetkellä valtavaa vauhtia. Lisäksi moderni, täysin suomenkielinen kuvaus ohjelmointikielistä, tiedonkeruumenetelmistä ja tallennusvälineistä lisää tietoa varmasti automaatioteollisuudessa, mutta myös esimerkiksi SD-korteista kiinnostuneiden kameraharrastajien parissa.

Osaltaan työtä voi ajatella kokeena siitä, että mitä Visual Basic -skriptityökalulla voi luoda. Työn aikana tulikin selvästi esille, että miten erilaista on kehittää ohjelmistoja WinCC VBS IDE:llä verrattuna Excelin VBA-editoriin. WinCC scripting editor on pelkistetty ja rajoitettu ympäristö. Sovelluksen kirjoittaminen WinCC:llä vaatikin välillä oivalluksia, joilla kierretään näitä rajoituksia. Excelin koodikirjoitustyökalu taas sisältää Microsoftin tuotteena laajan ja käyttökelpoisen kirjaston täynnä monikäyttöisiä ja valmiita ohjelmistokomponentteja. Vaikka Visual Basic on jo vanhahko ohjelmointikieli, tekevät VBA-skriptit vielä tänäkin päivänä Excelistä tehokkaasti taipuvaisen mitä monimutkaisimpiin tehtäviin.

Tärkein työssä luotu asia oli konkreettinen sovelluspaketti toimeksiantajaryitykselle. Kuitenkaan ei ole syytä unohtaa työn aikana tapahtunutta henkistä kehitystä myös työn tekijälle itselleen. Työssä luodut noin 1500 riviä Visual Basic koodia olivat omiaan kasvattamaan ohjelmointitaitoa monella tapaa. Testauksen itse läpivieminen auttoi niin ikään ymmärtämään oikeiden ohjelmarakenteiden valintaa sekä näihin vaikuttavia ratkaisuja. Kehitystä tapahtui myös projektien läpivientiteknikassa ja aikataulutuksessa. Kontakti asiakasyrityksiin jäi vielä tässä vaiheessa vähäiseksi, joten tavoite oppia asiakasyritysten tiedonkeruu- ja raportointikäytännöistä ei aivan täysin toteutunut. Kuitenkin sovellus on tarkoitus sovittaa vielä vuoden 2015 puolella oikeaan asiakastyöhön.

Opinnäyteprosessi oli ajateltua kivisempi tie. Varsinkin työn laajuus ja tiettyjen asioiden jääminen toteutuskelpoisuuden ulkopuolelle rajoitusten takia nosti vaikeustasoa. Ottamalla huomioon opinnäytetyön poikkitieteellisyyden ja valmiin tuotteen toimintojen määrän ja toiminallisuuden, voi opinnäytetyön läpivientiin olla kuitenkin hyvinkin tyytyväinen.

Lähteet

Ajo, R., Hakonen, S., Harju, H., Järvi, J., Kaskes, K., Lenardic, E., Niukkanen, E., Nurminen, T., Ritala, P., Tolppanen, M. & Tommila, T. 2001. Laatu automaatioissa. Saarijärvi: Suomen automaatioseura.

Artikkeli VBA tuesta AutoCadissa. 2014. Viitattu 29.09.2015.

<http://knowledge.autodesk.com/support/autocad/troubleshooting/caas/sfdcarticles/sfdcarticles/VBA-support-in-AutoCAD.html>

Axelson, J. 2006. USB Mass Storage : Designing and Programming Devices and Embedded Hosts. USA: Lakeview Research.

Axelson, J. 2009. USB Complete : The Developer's Guide. 4th Edition. USA: Lakeview Research.

Brewer, J. & Gill, M. 2008. Nonvolatile Memory Technologies with Emphasis on Flash—A Comprehensive Guide to Understanding and Using NVM Devices. Hoboken, NJ: John Wiley & Sons.

Bunting, S. 2012. EnCase Computer Forensics: The Official EnCE: EnCase Certified Examiner Study Guide. Third Edition. John Wiley & Sons.

CSVReader.com artikkeli CSV-tiedostomuodosta. N.d. Viitattu 17.09.2015.
http://www.csvreader.com/csv_format.php

Halvorson, M. 2010. Visual Basic 2010 Step by Step. Redmond, WA: Microsoft Press

Harsu, M. 2005. Ohjelmointikielet: Periaatteet, käsitteet, valintaperusteet. Tampere: Talentum

JEEC Oy nettisivut. 2015. Viitattu 14.09.2015. <http://www.jeec.fi/index.htm>

Kingshey-Hughes, A., Kingshey-Hughes, K. & Read, D. VB Script programmer's reference. Kolmas painos.

<http://site.ebrary.com.ezproxy.jamk.fi:2048/lib/jypoly/reader.action?docID=102964>
11

Kippo, A. & Tikka, A. 2008. Automaatiotekniikan perusteet. Helsinki: Edita Publishing.

Lexarin raportti SD-korttien nopeusluokituksista. 2010. 21.09.2015.

http://www.lexar.com/files/product/datasheet/Understanding_SD_Association_Speed_Ratings_122010.pdf

Lowe, S. 2015. A Flash Storage Technical and Economic Primer. Artikkelin flash-

muisteista netissä. Viitattu 15.09.2015. <http://www.flashstorage.com/flash-storage-technical-economic-primer/>

Mack, G. 2002. Artikkelin Visual Basicin historiasta. Kolmas versio. Viitattu 28.09.2015.

http://www.ojodepez-fanzine.net/network/qbdl/history_of_visual_basic.html

Merensalmi, J. 2007. Excel VBA yrityskäytössä. Porvoo: WSOY.

MMC-korttien mekaaninen standardi. 2007. Viitattu 18.09.2015. Vaatii

rekisteröitymisen. <https://www.jedec.org/standards-documents/results/JESD84-C01>

Phung, S. 2009. Professional Microsoft Windows CE 6.0. Indianapolis, IN: Wiley.

Rouse, M. 2015. Flash Memory Definition. Artikkelin flash-muisteista. Viitattu

15.09.2015. <http://searchstorage.techtarget.com/definition/flash-memory>

SD association nettisivut. 2015. Viitattu 18.09.2015. <https://www.sdcard.org/>

SD-korttien fyysinen tekninen spesifikaatio. 2013. Versio 4.10. Yksinkertaistettu.

Viitattu 18.09.2015 https://www.sdcard.org/downloads/pls/part1_410.pdf

Shafranovich, Y. 2005. Solid Matrix Inc:n muistio RFC 4180 .CSV-tiedostomuodosta.

Viitattu 17.09.2015. <http://www.ietf.org/rfc/rfc4180.txt>

Siemens Comfort-paneelien käyttöohjeet. 2012. Viitattu 05.10.2015.

https://cache.industry.siemens.com/dl/files/233/49313233/att_59649/v1/hmi_comfort_panels_operating_instructions_en-US_en-US.pdf

Siemens Mobile Panel 170 -paneelin käyttöohjeet. 2002. Viitattu 05.10.2015.

<https://support.industry.siemens.com/cs/document/13506223/simatic-hmi-mobile-panel-170?dti=0&lc=en-WW>

Siemens Mobile Panel 177 -paneelin käyttöohjeet. 2005. Viitattu 05.10.2015.

<https://support.industry.siemens.com/cs/document/21896316/mobile-panel-177-%28wincc-flexible%29?dti=0&lc=en-WW>

Siemens Mobile Panel 277 IWLAN -paneelin käyttöohjeet. 2008. Viitattu 05.10.2015.

<https://support.industry.siemens.com/cs/document/29036579/simatic-hmi-hmi-device-mobile-panel-277-iwlan?dti=0&lc=en-WW>

Siemens Mobile Panel 277 -paneelin käyttöohjeet. 2008. Viitattu 05.10.2015.

<https://support.industry.siemens.com/cs/document/23342762/simatic-hmi-hmi-device-mobile-panel-277-%28wincc-flexible%29?dti=0&lc=en-WW>

Siemens Multi Panel 170 -paneelin käyttöohjeet. 2008. Viitattu 05.10.2015.

<https://support.industry.siemens.com/cs/document/31247475/simatic-hmi-hmi-device-mp-177-%28wincc-flexible%29?dti=0&lc=en-WW>

Siemens TIA Portal V13 SP 1 sisäinen manuaali. N.d. Viitattu 19.10.2015.

Siemens WinCC Advanced V13 SP1 -järjestelmämanuaali. 2014. Viitattu 23.09.2015.

<https://support.industry.siemens.com/cs/document/109091876/wincc-advanced-v13.0-sp1?dti=0&lc=en-CZ>

Siemens WinCC -esite. 2014. Viitattu 08.10.2015.

<https://c4b.gss.siemens.com/resources/images/articles/e20001-a820-p810-x-7600.pdf>

Siemens WinCC -nettisivu. 2015. Viitattu 01.10.2015.

<http://w3.siemens.com/mcems/human-machine-interface/en/visualization-software/scada/Pages/system-overview.aspx>

Sony USMSA2 -sarjan USB-flash -muistin tekniset tiedot. 2015. 22.09.2015.

<http://www.sony.fi/electronics/usb-flash-muistit/usmsa2-series/specifications>

Suominen S. & Suominen J. 2015. Laatu raportointiin Excelillä. Saarijärvi: Docendo.

Tilley, C. 2006. Artikkelin Windows CE:n antivirus ominaisuuksista. Viitattu 29.10.2015.

<http://www.smartphonemag.com/cms/archives/Aug06/antivirus.aspx?page=0%2C0>

Tommila, T. 1993, VTT Tiedotteita - panosprosessien automaatio. Espoo: VTT.

USB-IF:n nettisivut. Esittely USB-IF:stä. 2015. Viitattu 22.09.2015.

<http://www.usb.org/about>

Vinkkejä WinCC skripteihin. 2015. Viitattu 29.09.2015.

<https://support.industry.siemens.com/cs/document/57132412/tips-and-tricks-for-creating-scripts-in-wincc-%28tia-portal%29?dti=0&lc=en-WW>

Liitteet

Liite 1. panosraportti

EXAMPLE PROCESS BATCH REPORT			
Batch Number	1		
Date	30.12.1999		
Starting time	00:00:00		
Stopping time	23:59:59		
	Setpoint value	Actual value	
Substance 1	3	4 %	
Substance 2	5	6 kg	
Substance 3	7	8 kg	
Substance 4	9	10 kg	
Substance 5	1	2 min	
Total	21	24 kg	
Comments	Ok		

Liite 2. panosperusteinen tuotantoraportti

EXAMPLE PROCESS BATCH PRODUCTION REPORT			
Date	30.12.1999		
Batch numbers	1	2	
Amount of batches	2		
	Setpoint value	Actual value	
Substance 1	10	12 kg	
Substance 2	14	16 kg	
Substance 3	18	20 kg	

Liite 3. päiväkohtainen tuotantoraportti

EXAMPLE PROCESS DAILY PRODUCTION REPORT				
Date	30.12.1999			
	Setpoint value		Actual value	
Substance 1	10		12 kg	
Substance 2	14		16 kg	
Substance 3	18		20 kg	
Substance 4	22		24 kg	

Liite 4. kulutusraportti

EXAMPLE PROCESS CONSUMPTION REPORT				
Date	30.12.1999			
	Setpoint value	Actual value		
Total substance 1	13	14	kg	
Total substance 2	15	16	kg	
Total substance 3	17	18	kg	