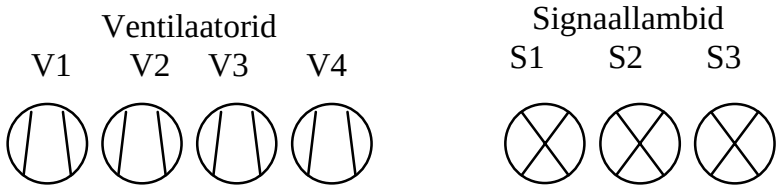
11. Juhtimisprogrammide näiteid

11.1. Ventilatsiooni signalisatsioon

Protsessijuhtimise algoritmi loomist alustatakse tehnoloogilise skeemi koostamisega (joonis 11.1) ja selle kirjeldamisega. Vaadelgem alljärgnevat näidet [13].

Ettevõtte sööklat ventileeritakse nelja ventilaatoriga. Sõltuvalt õhu temperatuurist, niiskusest või töös olevate pliitide arvust reguleeritakse ventilatsiooniõhu hulka. Elektriseadmete hooldusspetsialistil on vaja tagada söökla ventilaatorite korrashoid. Seega huvitab teda töös olevate ventilaatorite arv, milleks kasutatakse signaallampe.



Joonis 11.1. Tehnoloogiaskeem

Pärast tehnoloogiaskeemi koostamist tuleb juhitav protsess operatsioonide kaupa üksikasjalikult kirja panna, et hiljem kasutada seda programmeerimisel ja programmi õigsuse kontrollimisel. Käesolevas näites on protsessi kirjeldus järgmine:

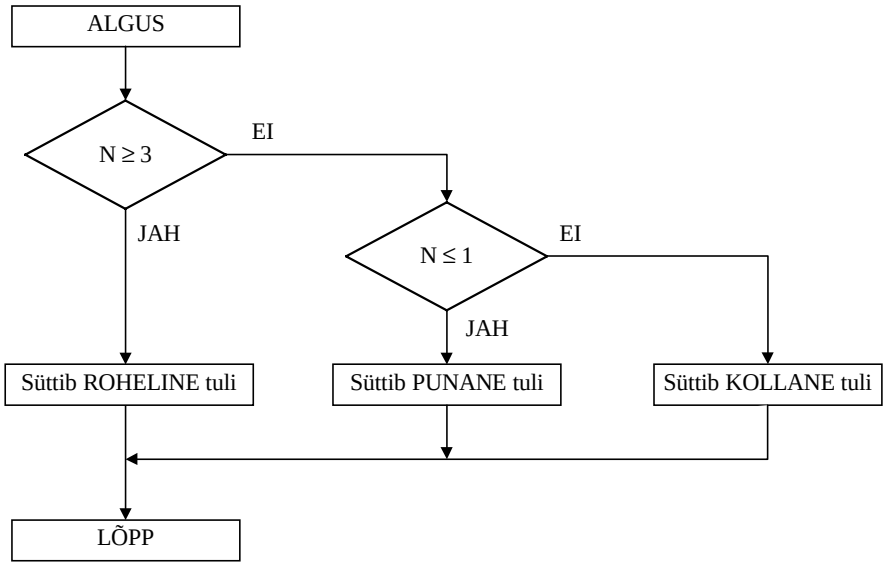
1. Signaallamp 1 (punane) peab olema sisse lülitatud, kui ei tööta ükski ventilaator või töötab ainult üks ventilaator neljast.
2. Signaallamp 2 (kollane) peab olema sisse lülitatud, kui töötab ainult kaks ventilaatorit neljast.
3. Signaallamp 3 (roheline) peab olema sisse lülitatud, kui töötavad kõik ventilaatorid või kolm ventilaatorit neljast.

Pärast tehnoloogiaskeemi ja protsessikirjelduse koostamist tuleb määratleda ja kirjeldada juhtkontrolleri sisend- ja väljundoperande (tabel 11.1)

Tabel 11.1 Juhtkontrolleri sisend- ja väljundoperandid

Operand	Tähis	Kirjeldus
I 0.0	V 1	Ventilaator 1
I 0.1	V 2	Ventilaator 2
I 0.2	V 3	Ventilaator 3
I 0.3	V 4	Ventilaator 4
Q 4.0	S 1	Signaallamp 1 (punane)
Q 4.1	S 2	Signaallamp 2 (kollane)
Q 4.2	S 3	Signaallamp 3 (roheline)

Järgmise sammuna on vaja protsessi tehnoloogiaskeemi ja kirjelduse järgi luua programmeerimist hõlbustav struktuuriskeem või algoritmi plokkskeem. Ülesande esitamisel algoritmina tuleb täpselt arvestada, mis tingimustel operatsioonid toimuvad. Antud näite kohta on joonisel 11.2 toodud algoritmi plokkskeem, kus *N* tähistab mingil hetkel töötavate ventilaatorite arvu.



Joonis 11.2. Signaallampide juhtimise algoritmi plokkskeem

Protsessi juhtalgoritmi plokkskeemi põhjal tuleb koostada loogikaskeem või valem, mida võimaluse korral lihtsustatakse. Antud ülesande lahendamisel on tegemist kombinatsioonloogikaga ning loogikavalemid näevad välja järgmised:

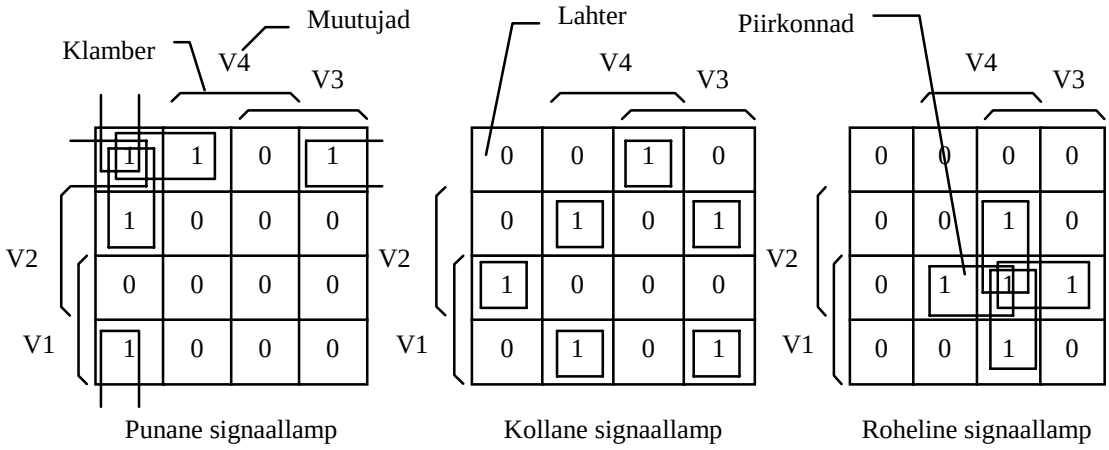
$$S1 = \overline{V1}\cdot\overline{V2}\cdot\overline{V3} + \overline{V1}\cdot\overline{V2}\cdot\overline{V4} + \overline{V1}\cdot\overline{V3}\cdot\overline{V4} + \overline{V2}\cdot\overline{V3}\cdot\overline{V4} + \overline{V1}\cdot\overline{V2}\cdot\overline{V3}\cdot\overline{V4},$$

$$S3 = V1\cdot V2\cdot V3 + V1\cdot V2\cdot V4 + V1\cdot V3\cdot V4 + V2\cdot V3\cdot V4 + V1\cdot V2\cdot V3\cdot V4,$$

$$S2 = \overline{S1}\cdot\overline{S3},$$

kus S1 tähistab punast, S3 rohelist ja S2 kollast signaallampi ning V1…V4 ventilaatoreid.

Kombinatsioonloogikalülituse lihtsustamiseks võib kasutada näiteks Karnaugh' kaarti (joonis 11.3) [7]. Karnaugh' kaardi lahtrite (ruutude) arv sõltub funktsiooni sisendmuutujate arvust *n* ning vastab muutujate kombinatsioonide arvule 2^n. Sisendmuutujate arvuks siin on ventilaatorite arv. Muutujad ja funktsiooni väärtused paigutatakse tabelisse nii, et saaks esitada kõiki muutujate kombinatsioone. See eeldab muutujate erilist paigutust. Klambriga hõivatud alas on muutujal otsene, väljaspool klambrit aga inverteeritud väärtus. Karnaugh' kaardi iseloomulikuks omaduseks on see, et funktsiooni väärtused erinevad kõrvuti asuvates lahtrites vaid ühe muutuja poolest, s. t. naaberlahtrisse minekul muudab oma väärtust vaid üks sisendmuutuja. Naabriteks loetakse ka kaardi äärmised vasak- ja parempoolsed lahtrid omavahel ning ülemised ja alumised lahtrid omavahel. Naaberlahtreid, mis erinevad vaid ühe muutuja poolest, kasutatakse loogikafunktsiooni minimeerimiseks. Lahtritesse, mis vastavad funktsiooni tingimusele, kirjutatakse olek "1". Naaberlahtritest mis omavad sarnast suurust, moodustatakse piirkonnad suurusega 2^n, kus $n = 0, 1, 2, 3, \ldots, m$. Neid piirkondi peab olema nii vähe kui võimalik ja nad peavad olema nii suured kui võimalik. Iga sellise piirkonna jaoks saab kirjutada loogilise korrutamise ja enama kui ühe piirkonna puhul seotakse need piirkonnad loogilise liitmisega. Seega loogikafunktsiooni avaldis kirjutataks *disjunktiivsel normaalkujul*, kus igale piirkonnale vastab elementaarkonjunktsioon muutujatest, mis terve kontuuri jaoks on inverteerimata või inverteeritud.



Joonis 11.3. Karnaugh' kaardiga lihtsustamine

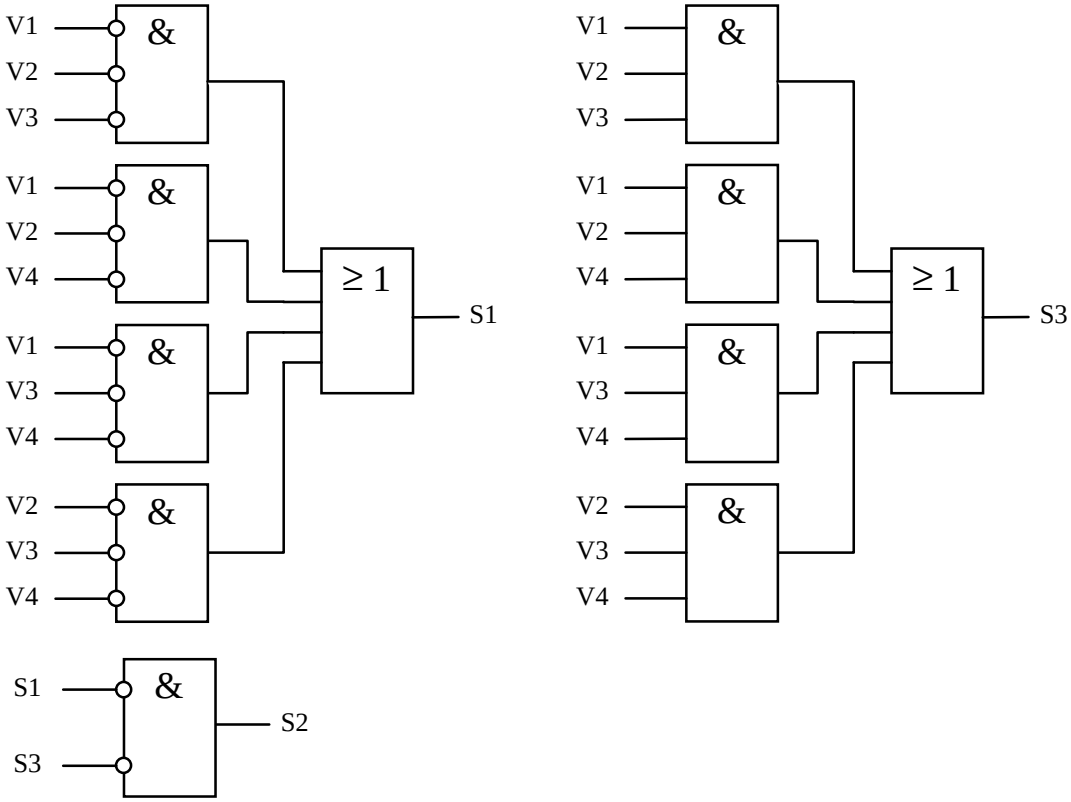
Karnaugh' kaartidelt on näha, et lihtsustada saab punase ja rohelise signaallambi juhtimislülitust. Kõigepealt tuleb kirjutada ruutudesse, mis vastavad funktsiooni tingimusele olek "1". Kuna punase tule süttimise tingimus oli, et töötab üks või ei tööta ühtegi ventilaatorit, tuleb kirjutada olek "1" neisse ruutudesse, mis ei lange kas ühegi klambri piirkonda või langeb ainult ühe klambri piirkonda. Rohelise tule puhul oli tingimus, et lamp põleb, kui töötab 3 ventilaatorit või kõik neli. Seega ruudud, mis langevad korraga kas kolme või nelja klambri (muutuja) piirkonda peavad sisaldama olekut "1". Piirkondade moodustamisel ja loogikafunktsiooni kirjutamisel tuleb jälgida, et piirkonna jaoks väljakirjutatud loogilises korrutises peavad olema nende klambrite ehk sisendmuutujate tähised, mille piirkonda või mille piirkonnast välja nad täielikult langevad. Seega on ülesande lihtsustamisel saadud loogikafunktsioonid järgmised:

$$S1=\overline{V1}\cdot\overline{V2}\cdot\overline{V3}+\overline{V1}\cdot\overline{V2}\cdot\overline{V4}+\overline{V1}\cdot\overline{V3}\cdot\overline{V4}+\overline{V2}\cdot\overline{V3}\cdot\overline{V4}=\overline{V1}\cdot\overline{V2}\cdot(\overline{V3}+\overline{V4})+\overline{V3}\cdot\overline{V4}\cdot(\overline{V1}+\overline{V2})$$

$$S3=V1\cdot V2\cdot V3+V1\cdot V2\cdot V4+V1\cdot V3\cdot V4+V2\cdot V3\cdot V4=V1\cdot V2\cdot(V3+V4)+V3\cdot V4\cdot(V1+V2)$$

$$S2=\overline{S1}\cdot\overline{S3}$$

Saadud loogikavalemite põhjal võime koostada loogikaskeemi (joonis 11.4), mis saab aluseks hiljem loodavale juhtimisprogrammile. Loogilise liitmise puhul, nagu teada, kasutatakse loogikaelementi VÕI ja loogilise korrutamise korral elementi NING.



Joon 11.4. Juhtimisskeemi koostamine

Programmeerimise ettevalmistusse kuulub ka seadmete valik. Kuna antud juhul on tegemist ainult nelja sisendsuuruse ja kolme väljundsuurusega, siis on juhtimiseks vaja kontrollerit, mille sisend/väljundliidesel on vähemalt 4 sisendit ja 3 väljundit (enamasti sisaldab juba protsessoriplokk sisend/väljundliidest). Kuna tegemist on väga lihtsa ja väikesemahulise protsessi ja programmiga, võib vabalt kasutada väikese jõudlusega seadet ning valida selleks SIMATIC S7-200.

Kontrolleri S7-200 puhul kasutatakse programmeerimispaketti *STEP7 MicroWIN*. Programmi esitusviisina valime käsulisti. Kuna programm on väga lühike, pole vaja seda parema ülevaate saamiseks alamplokkideks jaotada, vaid kirjutada tervikuna juhtplokki OB1. Kuna kõik plokid koosnevad segmentidest ja antud juhul koosneb programm kolmest signaallambi juhtimisskeemist, siis võiks selle jaotada kolmeks segmendiks (joonis 11.5).

Signaallampide puhul pole vaja kasutada lisaahelatena hädastopp ja käivituslülitit, sest nad peavad ventilaatorite töö kohta pidevalt infot edastama. Seega pole tarvidust kasutada programmis vastavaid ahelaid.

```
NETWORK 1
//PUNANE SIGNAALLAMP
LDN     I0.0
AN      I0.1
AN      I0.2
LDN     I0.0
AN      I0.1
AN      I0.3
OLD
LDN     I0.0
AN      I0.2
AN      I0.3
OLD
LDN     I0.1
AN      I0.2
AN      I0.3
OLD
=       Q4.0
```

```
NETWORK 2
//ROHELINE SIGNAALLAMP
LD      I0.0
A       I0.1
A       I0.2
LD      I0.0
A       I0.1
A       I0.3
OLD
LD      I0.0
A       I0.2
A       I0.3
OLD
LD      I0.1
A       I0.2
A       I0.3
OLD
=       Q4.1
```

```
NETWORK 3
//KOLLANE SIGNAALLAMP
LDN     Q4.0
AN      Q4.1
=       Q4.2
```

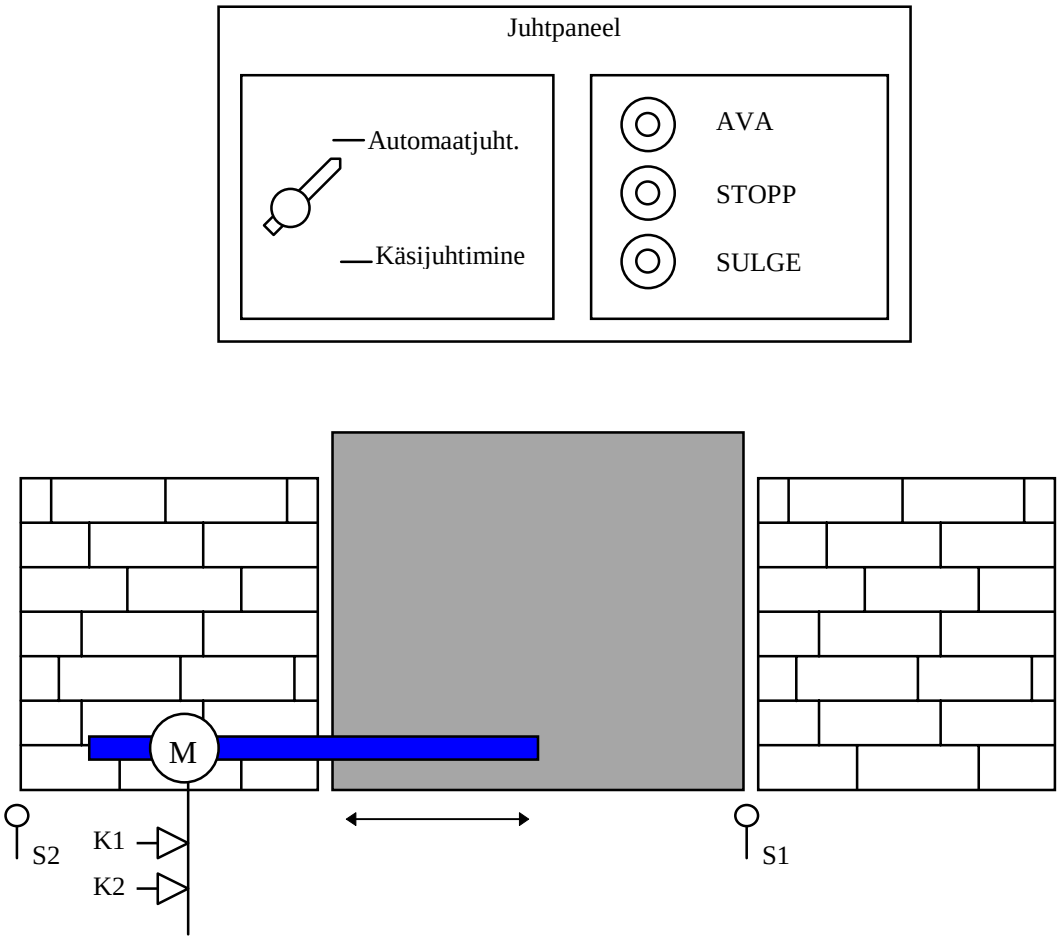
Joonis 11.5. Ventilatsiooni signalisatsioon käsulistina

Antud programmi testimiseks laetakse esmalt juhtplokk OB1 kontrolleri mällu ning seejärel käivitatakse testprogramm.

11.2. Tehase värava juhtimine

Mööblitehases toimub pidev tooraine sissevedu ja valmistoodangu väljavedu. Et tehase territooriumile sissesõit ning sealt väljasõit saaks toimuda, peab väravavaht jälgima, mida ja kuhu veetakse ning vastavalt vajadusele kas avama või sulgema värava. Seda tehakse siiani käsitsi sõltumata sellest, milline ilm parajasti õues on. Et lihtsustada värava avamist ja sulgemist, peeti otstarbekaks see automatiseerida nii, et väravavaht saaks väravat eemalt juhtida, olles ise soojas ruumis, kust monitoridelt saab kaamerate abil värava juures toimuvat jälgida.

Ettevalmistus programmeerimiseks algab analoogiliselt eelmise näitega. Kõigepealt koostatakse juhitava protsessi tehnoloogiaskeem. Tehnoloogiaskeemi koostamisel tuleb kindlasti arvestada liikuvate osade piir- ja vaheasendeid, milles toimub peatumine. See tähendab, et vastavatesse kohtadesse tuleb paigutada andurid. Tehnoloogiaskeemil peavad olema näidatud kõik andurid ja täiturid (joonis 11.6).



Joonis 11.6. Tehnoloogiaskeem

Tehnoloogiaskeemi põhjal tuleb jälle koostada protsessi kirjeldus. Tehase väravat peab olema võimalik elektrimootoriga avada ja sulgeda. Elektrimootorit juhitakse kahe kontaktori K1 ja K2 abil. Kui rakendub kontaktor K1, pöörleb mootor paremale ja värav avaneb. Kontaktori

K2 rakendumisel pöörleb mootor vasakule ja värav sulgub. Mõlemad kontaktorid ei tohi korraga rakenduda. Värava piirasendi olekute kohta annavad infot piirlülitite S1 ja S2 avanevad kontaktid.

Juhtpuldil saab valida käsi- või automaatjuhtimise. Kui on valitud automaatjuhtimine, piisab lühiajalisest vajutamisest lülitile AVA (sulguv kontakt) või SULGE (sulguv kontakt) ning uks sulgub või avaneb, kui seda toimingut ei katkesta piirlüliti (avanevad kontaktid) või lüliti STOPP (avanev kontakt) rakendumine (vajutus). Käsijuhtimisel toimub avanemine või sulgemine nii kaua kui vajutatakse vastavat juhtnuppu või kui uks saavutab piirasendi. Programmeerimisel tuleb jällegi määrata protsessis kasutatavad kontrolleri sisend- ja väljundoperandid koos sümbolvastetega (tabel 11.2).

Tabel 11.2 Sümbolite tabel

Operand	Tähis	Kirjeldus
I 0.0	Automaat-käsijuhtimine	talitlusviisi valik
I 0.1	AVA	Käsklus-ava värav
I 0.2	SULGE	Käsklus-sulge värav
I 0.3	STOPP	Käsklus-värav stopp
I 1.0	S1	Piirlüliti-värav on kinni
I 1.1	S2	Piirlüliti-värav on lahti
Q 4.0	K1	Värava avanemise kontaktor
Q 4.1	K2	Värava sulgemise kontaktor

Protsessi täpsemaks selgitamiseks on järgnevalt soovitav koostada tsüklogramm või olekutabel. Antud juhul on valitud mõlema väljundi jaoks eraldi olekutabel (tabelid 11.3 ja 11.4). Vasakpoolsetes veergudes on toodud antud väljundi kirjeldamise jaoks olulised sisendite olekud, kusjuures sisendina võib kasutada ka mõnda teist väljundit. Tabelites kasutatav sümbol “**x**” tähendab seda, et antud sisendparameetri olek võib olla kas “0” või “1”

Tabel 11.3 Värava avanemise juhtimine

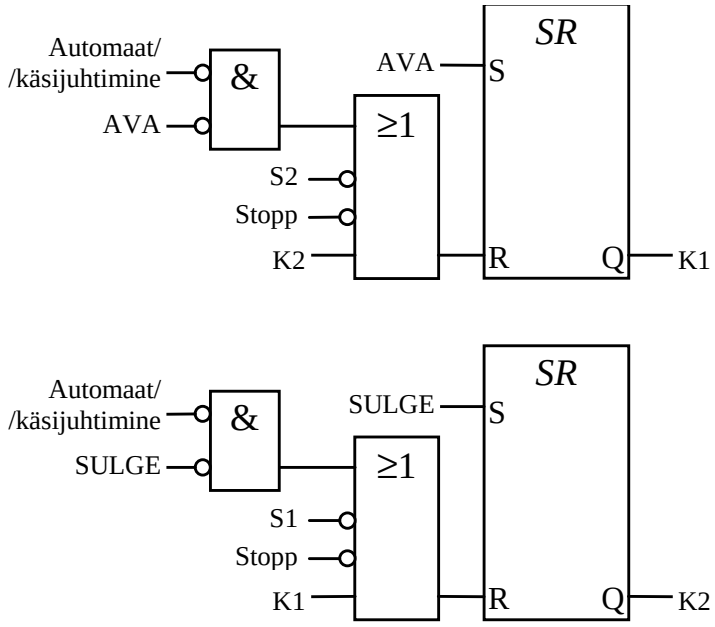
Sisendid					Väljund
Automaat/käsijuhtimine	AVA	STOPP	S2	K2	K1
x	1	1	1	0	1
0	0	x	x	x	0
x	x	0	x	x	0
x	x	x	0	x	0
x	x	x	x	1	0

Tabel 11.4 Värava sulgemise juhtimine

Sisendid					Väljund
Automaat/käsijuhtimine	SULGE	STOPP	S1	K1	K2
x	1	1	1	0	1
0	0	x	x	x	0
x	x	0	x	x	0
x	x	x	0	x	0
x	x	x	x	1	0

Värava sulgemise juhtimise olekutabeli 11.4 esimest rida saab kirjeldada järgmiselt. Sõltumata lüliti *Automaat/käsijuhtimine* olekust, kui lüliti *SULGE* ning lüliti *STOPP* ning lüliti *S1* on olekus “1” ning kontaktor *K1* on olekus “0”, siis kontaktor *K2* läheb olekusse “1” ja jääb sellesse olekusse kuni ei toimi tabeli mõni teine tingimus. Näiteks tabeli teine rida määrab, et sõltumatult lülitite *STOPP* ja *S1* ning kontaktori *K1* olekust, kui lüliti *Automaat/käsijuhtimine* on olekus 0 ja lüliti *SULGE* on olekus “0”, siis lülitatakse kontaktor K2 olekusse “0” ja ta jääb sellesse olekusse kuni ei toimi tabeli mõni teine tingimus.

Eelnevalt koostatud olekutabeli ja protsessi kirjelduse põhjal võib joonistada loogilise juhtskeemi (joonis 11.7), mis toimib vastavalt protsessi kirjeldusele.

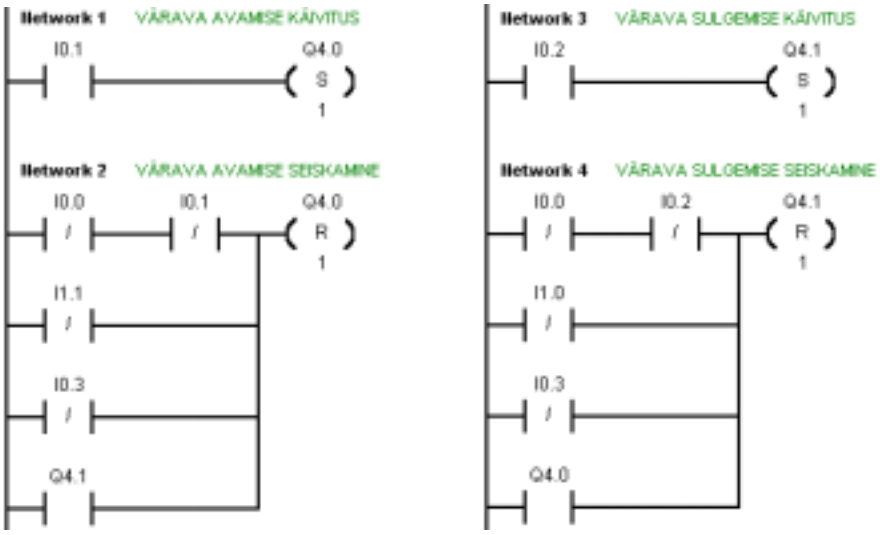


Joonis 11.7. Juhtimisskeem

Kuna antud protsessis on tegemist ainult viie sisendsuuruse ja kahe väljundsuurusega, on antud protsessi juhtimiseks meil vaja toiteplokki, protsessoriplokki ja vähemalt 5 sisendi ja 2 väljundiga sisend/väljundliidest.

Samuti nagu eelmise näite puhul on tegemist väga lihtsa ja väikesemahulise programmiga, seepärast võib juhtimiseks kasutada väikese jõudlusega seadet, näiteks SIMATIC S7-200. Kontrolleri S7-200 programmeerimiseks tuleb jälle kasutada programmeerimispaketti *STEP 7 MicroWIN*. Järgmisena valitakse programmi esitusviis, milleks olgu nt. kontaktaseskeem. Kuna programm on väga lühike, pole seda vaja jaotada alamplokkidesse ning võime kirjutada ta tervikuna juhtplokki OB1 (joonis 11.8). Nagu teada, koosnevad kõik plokid segmentidest ning antud juhtploki segmendis 1 on värava avamise programmi lõik ja segmendis 2 värava sulgemise programmi lõik.

Tehase värava juhtimise korral on tegemist lihtsa protsessiga, mida kontrollib operaator, ning STOPP-lüliti (I 0.3) asendab hädastopplülitust. STOPP-lüliti katkestab koheselt värava liikumise. Siiski ei tohi unustada, et hädastopplüliti peab eksisteerima mitte ainult programmiliselt, vaid ka riistvaraliselt.



Joonis 11.8. Tehase väravate juhtimine kontaktskeemina

Antud programmi testimiseks laetakse esmalt juhtplokk OB1 kontrolleri mällu ning seejärel käivitatakse testprogramm.

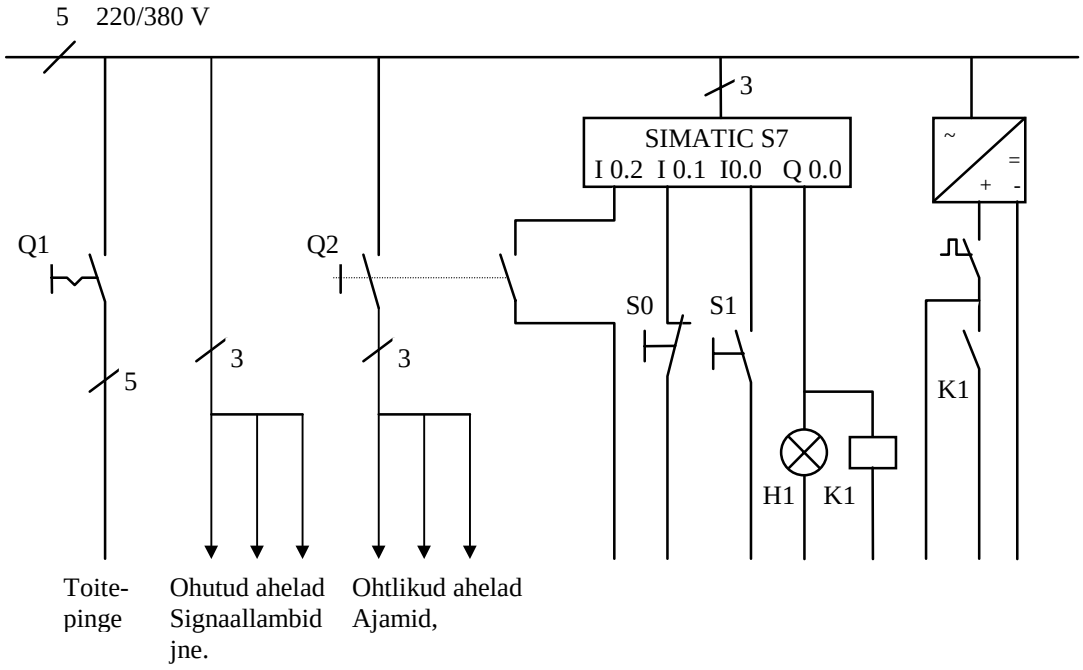
11.3. Hädaseiskamis- ja peakäivituslüliti

Hädaseiskamislüliti peab ohuolukorras peatama ajami, tööpingi või tööpingiosa nii, et ei tekkiks ohtu inimestele ega seadmetele (joonis 11.9). Ohuolukorras tuleb peatada vaid need seadmed, mis on juba põhjustanud ohtlikku olukorra või mis võivad seda lähimal ajal põhjustada.

Joonis 11.9 iseloomustab ühe juhtsüsteemi toiteahelat. Lüliti Q1 (pealüliti) abil lülitatakse sisse kogu juhtimissüsteemi toitepinge.

Hädaseiskamislülitiga Q2 **<u>lülitatakse välja</u> kõik ajamid ja seadmed, mis on põhjustanud või võivad põhjustada ohtliku olukorra inimestele või teistele seadmetele**.

Hädaseiskamislülitiga Q2 **<u>ei lülitata välja</u> seadmeid, mille väljalülitamine võib põhjustada ohtliku olukorra inimestele või teistele seadmetele**.

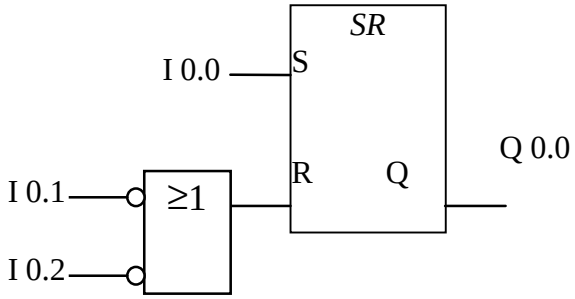


Joonis 11.9. Hädaseiskamislüliti skeem

Hädastopplüliti oleku kohta annab juhtsüsteemile infot abikontakt, mis on ühendatud kontrolleri sisendisse I 0.2 . Seega tuleb tehnoloogilises programmis alati arvestada hädastopplüliti olekut. Hädastopplüliti Q2 võib paigutada ka toiteahelasse, kus paikneb pealüliti Q1, kuid sel juhul peab olema kindel, et ükski ajami ega seadme osa ei vaja toidet hädaolukorras. Tuleb arvestada ka seda, et siis ei saa juhtprogrammis hädastopplüliti olekut arvestada, sest ka juhtsüsteem lülitatakse välja.

Käivituslüliti

Pärast toitesüsteemi sisselülitamist, juhtseadme tsüklilisse talitlusse lülitamist ja pingekatkestust ei tohi kontrolleri väljunditega ühendatud täiturseadmed oma talitlust jätkata (väljundisse ei tohi juhtsignaali anda) enne kui selleks on antud luba, kuna selle läbi võivad saada kannatada inimesed, seadmed ning tootmine. Üks võimalus sellist juhtimist programmiliselt realiseerida on toodud joonisel 11.10.

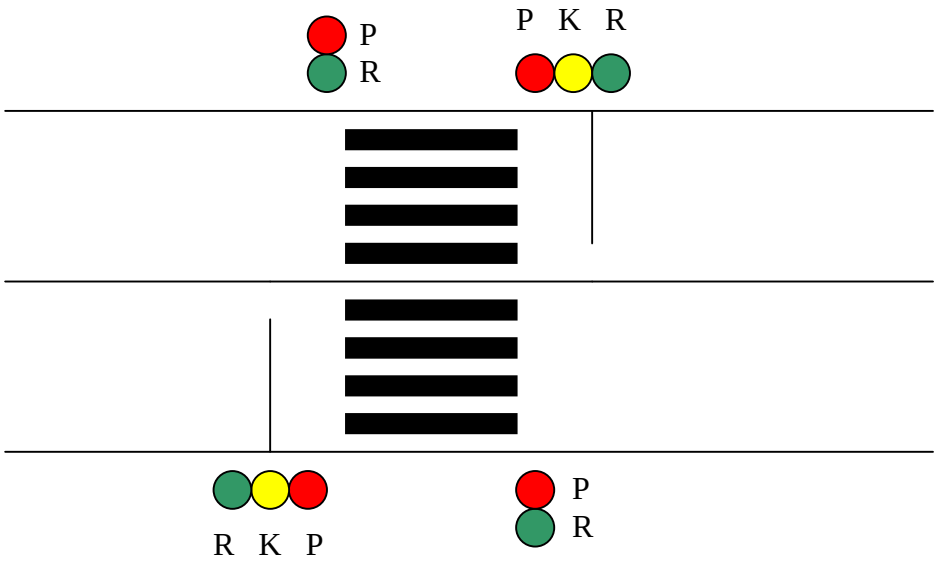


Joonis 11.10. Käivitusahela loogikaskeem

Väljund Q 0.0 lülitatakse lülitiga S1 (I 0.0) sisse. Väljund Q 0.0 lülitakse välja, kui on vajutatud lülitile S0 (I 0.1) või hädastopplülitile Q2 (I 0.2). Samuti toimub väljundi Q 0.0 väljalülitamine andurivea või pingekatkestuse korral vahemälu kustutamise tõttu. Väljundsignaali Q 0.0 väärtust saab mujal programmis kasutada sisse- ja väljalülitustingimusena. Kaitsekontakti K1 abil, mis on ühendatud väljundisse Q 0.0, saab valikuliselt sisse ja välja lülitada ainult teatud täiturite vooluahelad (joonis 11.10).

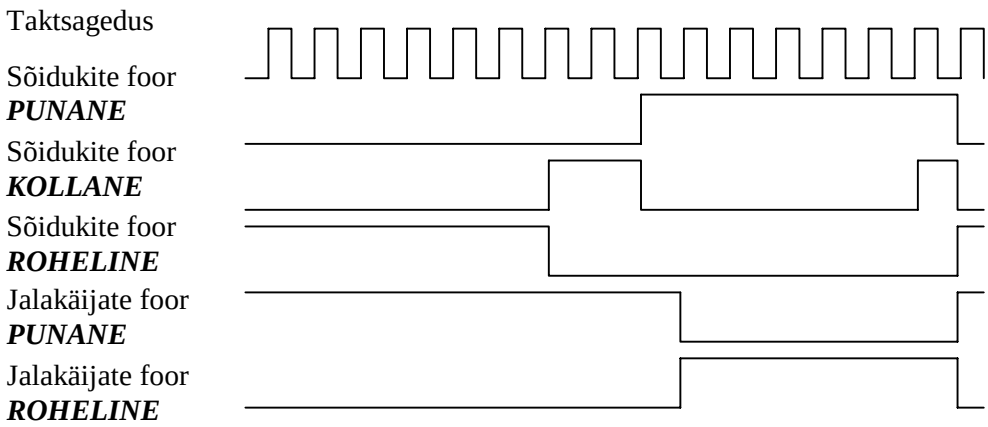
11.4. Valgusfooride juhtimine

Kõige pealt koostame protsessi kirjelduse tehnoloogiaskeemi järgi (joonis 11.11). Ülekäiguraja valgusfooride juhtimiseks kasutatakse kahte tööre¸iimi: päevane ja öine re¸iim. Päevase tööre¸iimi puhul toimub valgusfooride tsükliline juhtimine joonisel 11.12 esitatud ajadiagrammi järgi. Kusjuures taktsageduseks on 1 Hz. Öise tööre¸iimi valiku korral peab kollane valgusfoori lamp vilkuma sagedusega 1 Hz.



Joonis 11.11. Tehnoloogiaskeem

Esitatud ajadiagramm on aluseks edasisel programmeerimisel (joonis 6.13).



Joonis 11.12. Ajadiagramm

Pärast tehnoloogiaskeemi ja protsessikirjelduse koostamist tuleb määratleda ja kirjeldada juhtkontrolleri sisend- ja väljundoperandid (tabel 11.5).

Tabel 11.5 Juhtkontrolleri sisend- ja väljundoperandid

Operand	Tähis	Kirjeldus
I 0.0	LYLITI	Päevase/öise reţiimi lüliti
Q 0.0 Q 0.5	A_PUN_1 A_PUN_2	Sõiduk-PUNANE
Q 0.1 Q 0.6	A_KOL_1 A_KOL_2	Sõiduk-KOLLANE
Q 0.2 Q 0.7	A_ROH_1 A_ROH_2	Sõiduk-ROHELINE
Q 0.4 Q 1.0	J_PUN_1 J_PUN_2	Jalakäija-PUNANE
Q 0.5 Q 1.1	J_ROH_1 J_ROH_2	Jalakäija-ROHELINE

Kuna antud protsessis on tegemist ainult ühe sisendsuuruse ja kümne väljundsuurusega, on antud protsessi juhtimiseks meil vaja toiteplokki, protsessoriplokki ja vähemalt 1 sisendi ja 10 väljundiga sisend-/väljundliidest.

Samuti nagu eelmise näite puhul on tegemist väga lihtsa ja väikesemahulise programmiga, seepärast võib juhtimiseks kasutada väikese jõudlusega seadet, näiteks SIMATIC S7-200. Kontrolleri S7-200 programmeerimiseks tuleb jälle kasutada programmeerimispaketti *STEP 7 MicroWIN*. Järgmisena valitakse programmi esitusviis, milleks olgu nt. käsulist. Kuna programm on väga lühike, pole seda vaja jaotada alamplokkidesse ning võime kirjutada ta tervikuna juhtplokki OB1 (joonis 11.13). Nagu teada, koosnevad kõik plokid segmentidest ning antud juhtplokk koosneb seitsmest segmendist.
Joonisel 11.13 segmentides 1 ja 4 kasutatud süsteemimälu bitt SM0.5 vahetab oma signaali olekut sagedusega 1Hz.

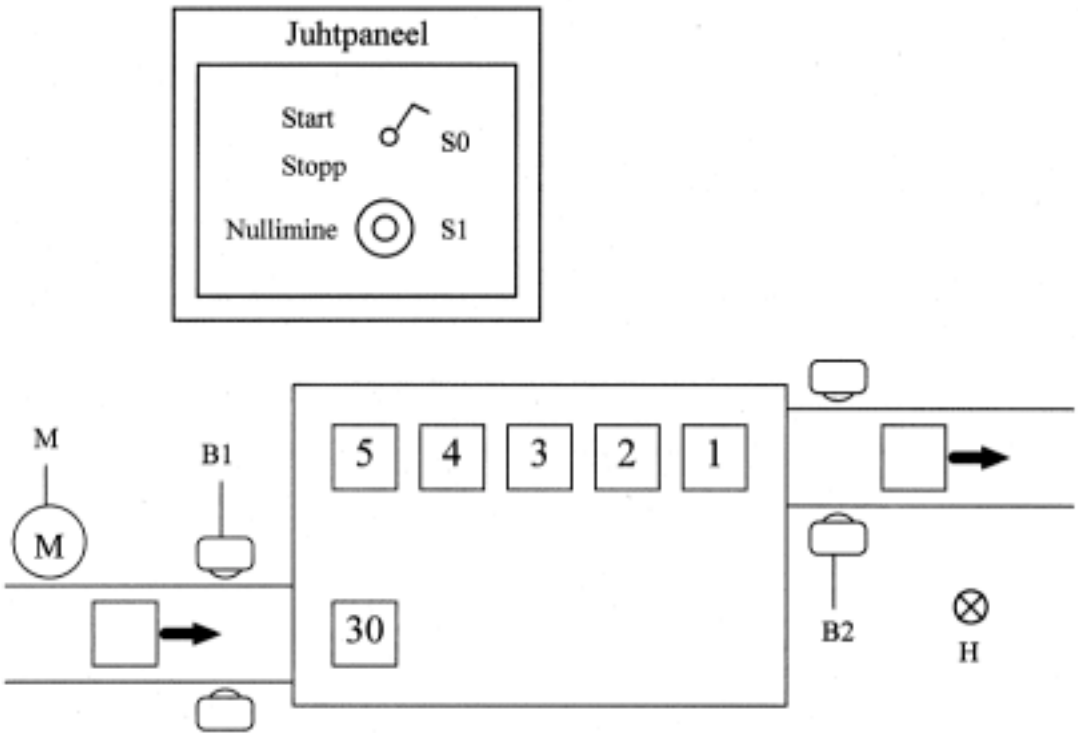
```
NETWORK 1
//LOENDUR
LD      SM0.5
A       "LYL"
LDN     "LYL"
CTU     C1, +1

NETWORK 2
//LOENDURI NULLIMINE
LDW>=   C1, +16
R       C1, 1

NETWORK 3
//AUTODE PUNANE TULI
LDW>=   C1, +9
AW<=    C1, +15
A       "LYL"
=       "A_PUN_1"
=       "A_PUN_2"

NETWORK 4
//AUTODE KOLLANE TULI
LDW>=   C1, +7
AW<=    C1, +8
OW=     C1, +15
A       "LYL"
LDN     "LYL"
A       SM0.5
OLD
=       "A_KOL_1"
=       "A_KOL_2"

NETWORK 5
//AUTODE ROHELINE TULI
LDW>=   C1, +0
AW<=    C1, +6
A       "LYL"
=       "A_ROH_1"
=       "A_ROH_2"

NETWORK 6
//JALAKÄIJATE PUNANE TULI
LDW>=   C1, +0
AW<=    C1, +9
A       "LYL"
=       "J_PUN_1"
=       "J_PUN_2"

NETWORK 7
//JALAKÄIJATE ROHELINE TULI
LDW>=   C1, +10
AW<=    C1, +15
A       "LYL"
=       "J_ROH_1"
=       "J_ROH_2"
```

Joonis 11.13 Valgusfooride juhtprogramm

Antud programmi testimiseks laetakse esmalt juhtplokk OB1 kontrolleri mällu ning seejärel käivitatakse testprogramm.

11.5. Vaheladu

Monteerimisliin (joonis 11.14) koosneb seadmete vahelaost ning sisenevast ja väljuvast transportliinist. Monteerimisliin käivitatakse lüliti S0 viimisega olekusse START. Monteeritavate seadmete sisenemist vahelattu ja sealt väljumist kontrollivad optilised andurid, millede abil seega tuvastatakse laoseis. Kui laos on alla 30 monteeritava seadme töötab mootor M, vastupidisel juhul lülitatakse mootor välja. Kui laos on alla 10 monteeritava seadme süttib signaallamp H. Laoseisu nullimiseks kasutatakse lülitit S1.



Joonis 11.14 Vahelao juhtimise tehnoloogiaskeem

Pärast tehnoloogiaskeemi ja protsessikirjelduse koostamist tuleb määratleda ja kirjeldada juhtkontrolleri sisend- ja väljundoperandid ning kanda nad seejärel STEP7 MicroWIN'i sümboltabelisse (tabel 11.15).

Symbol Table

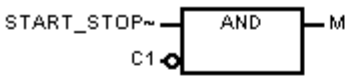
	Name	Address	Comment
1	START_STOPP	I0.0	Kaivitab voi seiskab protsessi
2	NULLIMINE	I0.1	Nullib laoseisu
3	B1	I0.2	Lattu sisenevaid detaile tuvastav andur
4	B2	I0.3	Laost valjuvaid detaile tuvastav andur
5	M	Q0.0	Monteerimisliini mootor
6	H	Q0.1	Signaallamp, kui detailide arv laos alla 10

Joonis 11.15 Sümboltabel

Kuna antud protsessis on tegemist ainult nelja sisendsuuruse ja kahe väljundsuurusega, on antud protsessi juhtimiseks meil vaja toiteplokki, protsessoriplokki ja vähemalt 4 sisendi ja 2 väljundiga sisend-/väljundliidest.

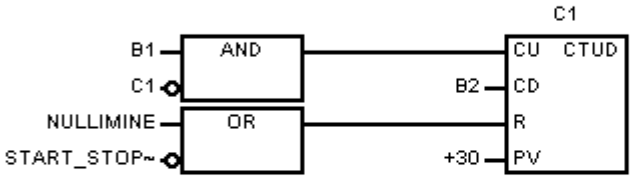
Samuti nagu eelmise näite puhul on tegemist väga lihtsa ja väikesemahulise programmiga, seepärast võib juhtimiseks kasutada väikese jõudlusega seadet, näiteks SIMATIC S7-200. Kontrolleri S7-200 programmeerimiseks tuleb jälle kasutada programmeerimispaketti *STEP 7 MicroWIN*. Järgmisena valitakse programmi esitusviis, milleks olgu nt. loogikaskeem. Kuna programm on väga lühike, pole seda vaja jaotada alamplokkidesse ning võime kirjutada ta tervikuna juhtplokki OB1 (joonis 11.16). Nagu teada, koosnevad kõik plokid segmentidest ning antud juhtplokk koosneb seitsmest segmendist.

Network 1



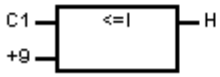
M	Q0.0	Monteerimisliini mootor
START_STOPP	I0.0	Kaivitab voi seiskab protsessi

Network 2 NETWORK TITLE (single line)



B1	I0.2	Lattu sisenevaid detaile tuvastav andur
B2	I0.3	Laost valjuvaid detaile tuvastav andur
NULLIMINE	I0.1	Nullib laoseisu
START_STOPP	I0.0	Kaivitab voi seiskab protsessi

Network 3

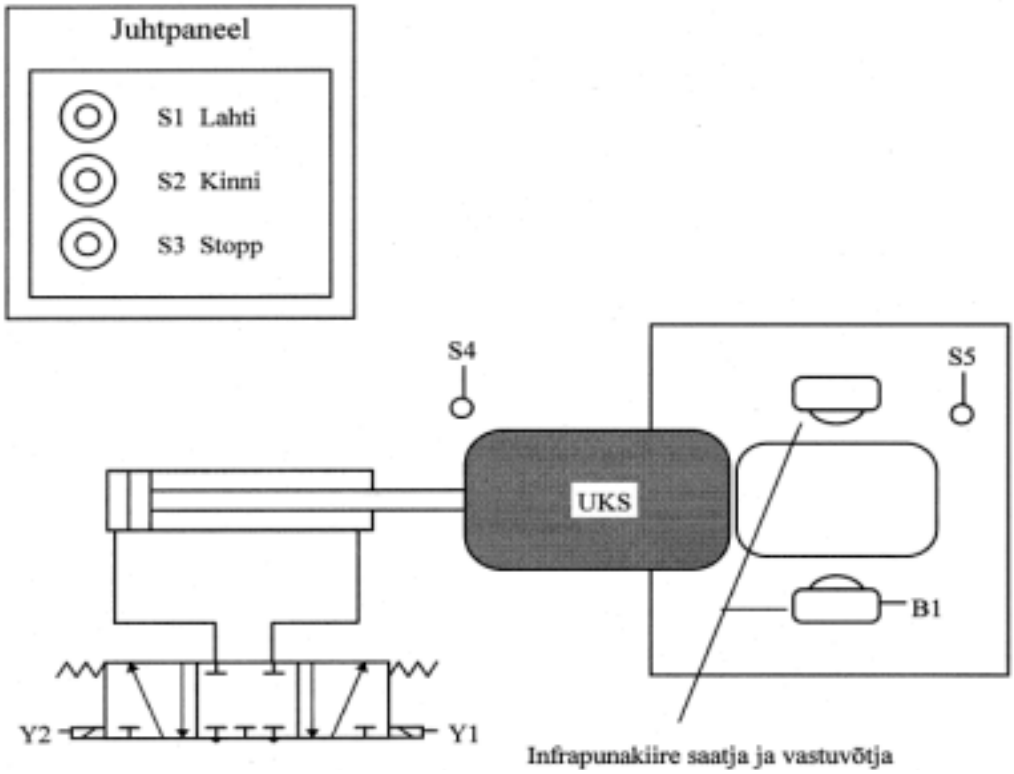


H	Q0.1	Signaallamp, kui detailide arv laos alla 10

Joonis 11.16 Vahelao juhtprogramm

11.6. Sulatusahju ukse juhtimine

Sulatusahju ust (joonis 11.17) juhitakse hüdro- või pneumosilindri abil. Algasendis on ahju uks suletud. Lülitile S1 vajutamisel uks avaneb, ning avanemine katkestatakse lülitile S3 vajutamisel või piirlüliti S4 rakendumisel. Kui uks on avanenud, hakkab ta automaatselt peale 6 sekundi möödumist või lülitit S2 vajutades sulguma. Ukse sulgemine lõpetatakse kas lülitile S3 vajutades või kui piirlüliti S5 on rakendunud. Lisaks katkestatakse ukse sulgemine juhul kui infrapunaandur tuvastab ukse vahel detaili. Kui ukse vahelt on detail eemaldatud jätkub ukse sulgumine.



Joonis 11.17 Sulatusahju ukse juhtimise tehnoloogiaskeem

Pärast tehnoloogiaskeemi ja protsessikirjelduse koostamist tuleb määratleda ja kirjeldada juhtkontrolleri sisend- ja väljundoperandid ning kanda nad seejärel STEP7 MicroWIN'i sümboltabelisse (tabel 11.18).

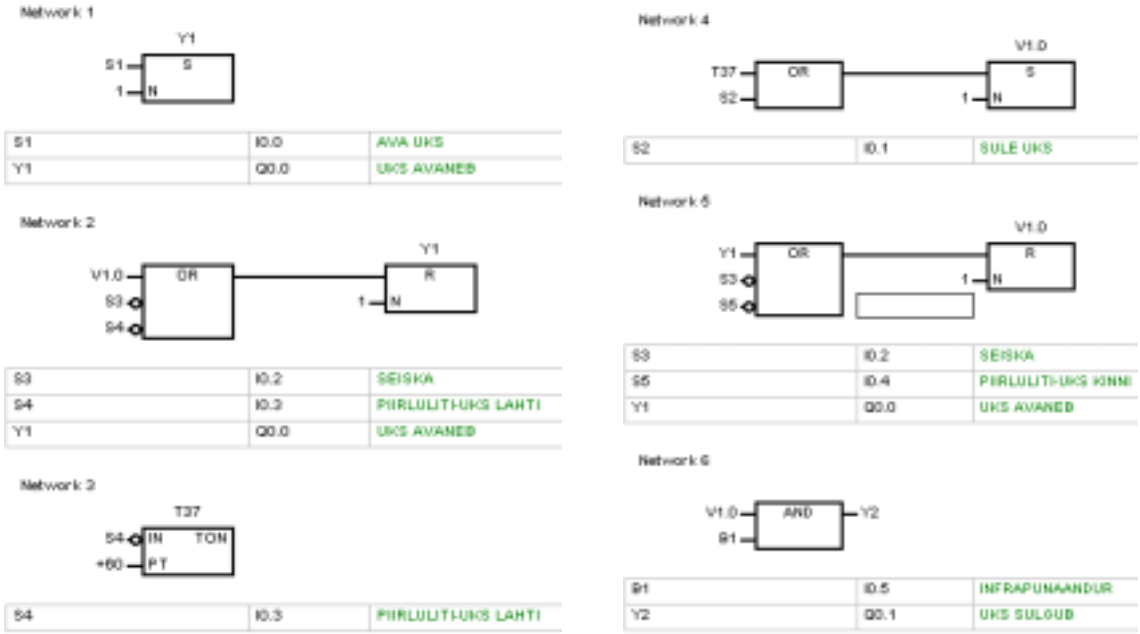
Symbol Table

	Name	Address	Comment
1	S1	I0.0	AVA UKS
2	S2	I0.1	SULE UKS
3	S3	I0.2	SEISKA
4	S4	I0.3	PIIRLULITI-UKS LAHTI
5	S5	I0.4	PIIRLULITI-UKS KINNI
6	B1	I0.5	INFRAPUNAANDUR
7	Y1	Q0.0	UKS AVANEB
8	Y2	Q0.1	UKS SULGUB

Joonis 11.18 Sümboltabel

Kuna antud protsessis on tegemist ainult nelja sisendsuuruse ja kahe väljundsuurusega, on antud protsessi juhtimiseks meil vaja toiteplokki, protsessoriplokki ja vähemalt 4 sisendi ja 2 väljundiga sisend-/väljundliidest.

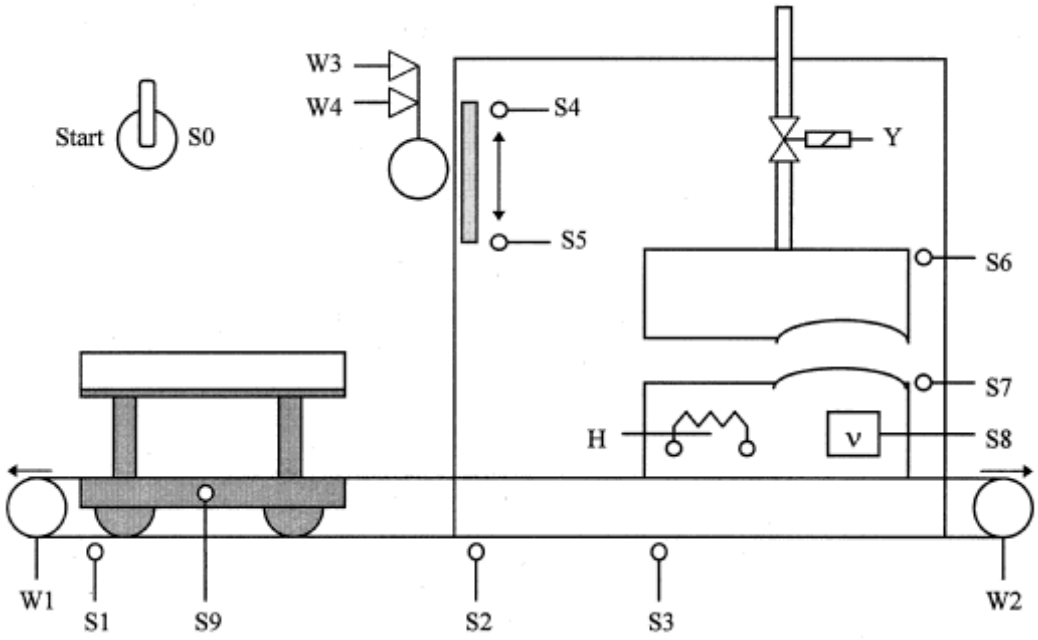
Samuti nagu eelmise näite puhul on tegemist väga lihtsa ja väikesemahulise programmiga, seepärast võib juhtimiseks kasutada väikese jõudlusega seadet, näiteks SIMATIC S7-200. Kontrolleri S7-200 programmeerimiseks tuleb jälle kasutada programmeerimispaketti *STEP 7 MicroWIN*. Järgmisena valitakse programmi esitusviis, milleks olgu nt. loogikaskeem. Kuna programm on väga lühike, pole seda vaja jaotada alamplokkidesse ning võime kirjutada ta tervikuna juhtplokki OB1 (joonis 11.19). Nagu teada, koosnevad kõik plokid segmentidest ning antud juhtplokk koosneb seitsmest segmendist.



Joonis 11.19 Sulatusahju ukse juhtprogramm

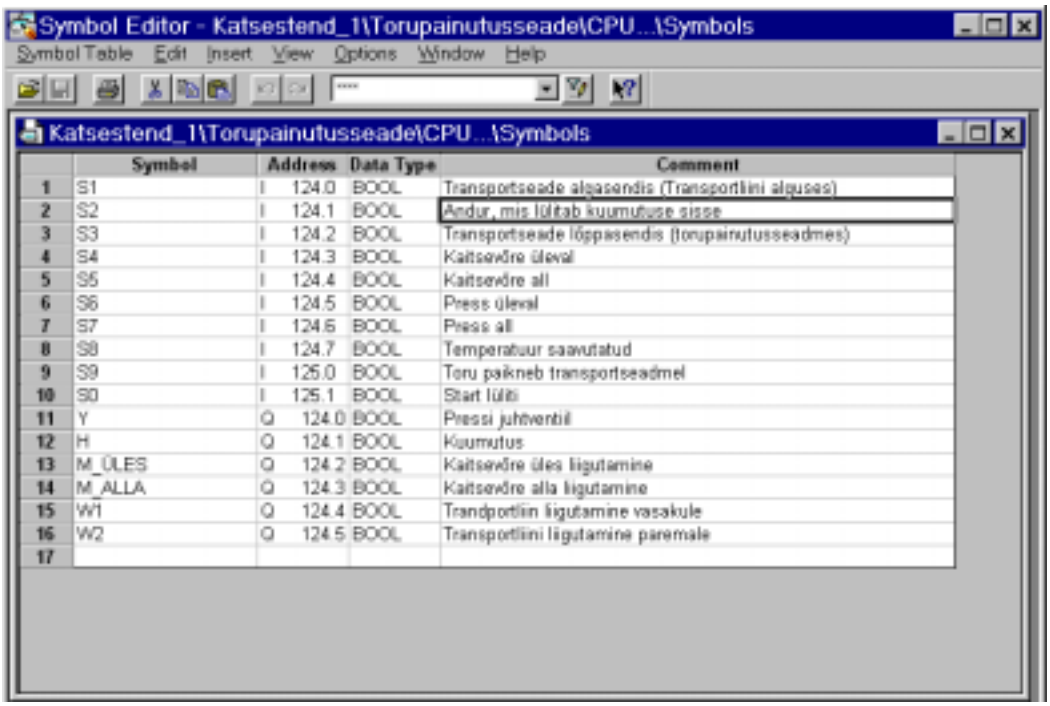
7.1 Torupainutusseadme juhtimine

Torupainutuprotsess (joonis 11.20) käivitub kui transportseade on algpositsioonis (S1 rakendunud), seadmel asetseb detail (S9 rakendunud), press on ülemises asendis (S6 rakendunud) ja kaitsevõre on üleval (S4 rakendunud).
Transportseade hakkab mootori W2 mõjul liikuma paremale. Kui transportseade on jõudnud kohakuti anduriga S2 (S2 rakendunud), lülitatakse kuumutus H sisse. Transportseadme liikumisel piirlülitini S3 (S3 rakendunud), lülitatakse mootor W2 välja ja kaitsevõre alustab liikumist alla (W3 rakendunud). Kui termostaat S8 rakendub ja kaitsevõre on alumises piirasendis (S5 rakendunud), siis algab pressimine (Y rakendunud). Kui press on alumises piirasendis (S7 rakendunud), siis lülitatakse küttekeha H välja ning toimub 10 sekundiline seisak. Peale seisakut lõpetatakse pressimine (Y väljalülitatud) ja kaitsevõre liigub ülesse (W4 rakendunud). Kui laitsevõre on üleval (S4 rakendunud), käivitatakse transportseadme mootor W1. Kui transportseade on algasendis (S1 rakendunud), seiskub mootor W1 ja kogu protsess võib otsast alata.



Joonis 11.20 Torupainutusseadme juhtimise tehnoloogiaskeem

Pärast tehnoloogiaskeemi ja protsessikirjelduse koostamist tuleb määratleda ja kirjeldada juhtkontrolleri sisend- ja väljundoperandid ning kanda nad seejärel STEP7 Manager'i sümboltabelisse (tabel 11.21).



	Symbol	Address	Data Type	Comment
1	S1	I 124.0	BOOL	Transportseade algasendis (Transportlini alguses)
2	S2	I 124.1	BOOL	Andur, mis lülitab kuumutuse sisse
3	S3	I 124.2	BOOL	Transportseade lõppasendis (torupainutusseadmes)
4	S4	I 124.3	BOOL	Kaitsevõre üleval
5	S5	I 124.4	BOOL	Kaitsevõre all
6	S6	I 124.5	BOOL	Press üleval
7	S7	I 124.6	BOOL	Press all
8	S8	I 124.7	BOOL	Temperatuur saavutatud
9	S9	I 125.0	BOOL	Toru paikneb transportseadmel
10	S0	I 125.1	BOOL	Start lüliti
11	Y	Q 124.0	BOOL	Pressi juhtventiil
12	H	Q 124.1	BOOL	Kuumutus
13	M_ÜLES	Q 124.2	BOOL	Kaitsevõre üles liigutamine
14	M_ALLA	Q 124.3	BOOL	Kaitsevõre alla liigutamine
15	W1	Q 124.4	BOOL	Trandportlin liigutamine vasakule
16	W2	Q 124.5	BOOL	Transportlini liigutamine paremale
17				

Joonis 11.21 Sümboltabel

Programmeerimise ettevalmistusse kuulub ka seadmete valik. Kuna antud juhul on tegemist ainult kümne sisendsuuruse ja kuue väljundsuurusega, siis on juhtimiseks vaja kontrollerit, mille sisend/väljundliidesel on vähemalt 10 sisendit ja 6 väljundit (enamasti sisaldab juba protsessoriplokk sisend/väljundliidest).

Address	Decl.	Name	Type	Initial Value	Comment
0.0	in	S1	BOOL	FALSE	
0.1	in	S2	BOOL	FALSE	
0.2	in	S3	BOOL	FALSE	
0.3	in	S4	BOOL	FALSE	
0.4	in	S5	BOOL	FALSE	
0.5	in	S6	BOOL	FALSE	
0.6	in	S7	BOOL	FALSE	
0.7	in	S8	BOOL	FALSE	
1.0	in	S9	BOOL	FALSE	
1.1	in	S0	BOOL	FALSE	
2.0	in	TAIMER	TIMER		
4.0	in	AEG	S5TIME	S5T#0MS	
6.0	out	Y	BOOL	FALSE	
6.1	out	H	BOOL	FALSE	
6.2	out	M_ÜLES	BOOL	FALSE	
6.3	out	M_ALLA	BOOL	FALSE	
6.4	out	W1	BOOL	FALSE	
6.5	out	W2	BOOL	FALSE	
	in_out				
	stat				
0.0	temp	MUUT_1	BOOL		
0.1	temp	MUUT_2	BOOL		
0.2	temp	MUUT_3	BOOL		
0.3	temp	MUUT_4	BOOL		

Joonis 11.22 Defineeritud muutujatega funktsioonploki päis

Kuna tegemist on väga lihtsa ja väikesemahulise protsessi ja programmiga, võib vabalt kasutada väikese või keskmise võimsusega seadet ning valida selleks SIMATIC S7-200 või 300. Valime S7-300, mis on ettenähtud väikese ja keskmise võimsusega ülesannete lahendamiseks.

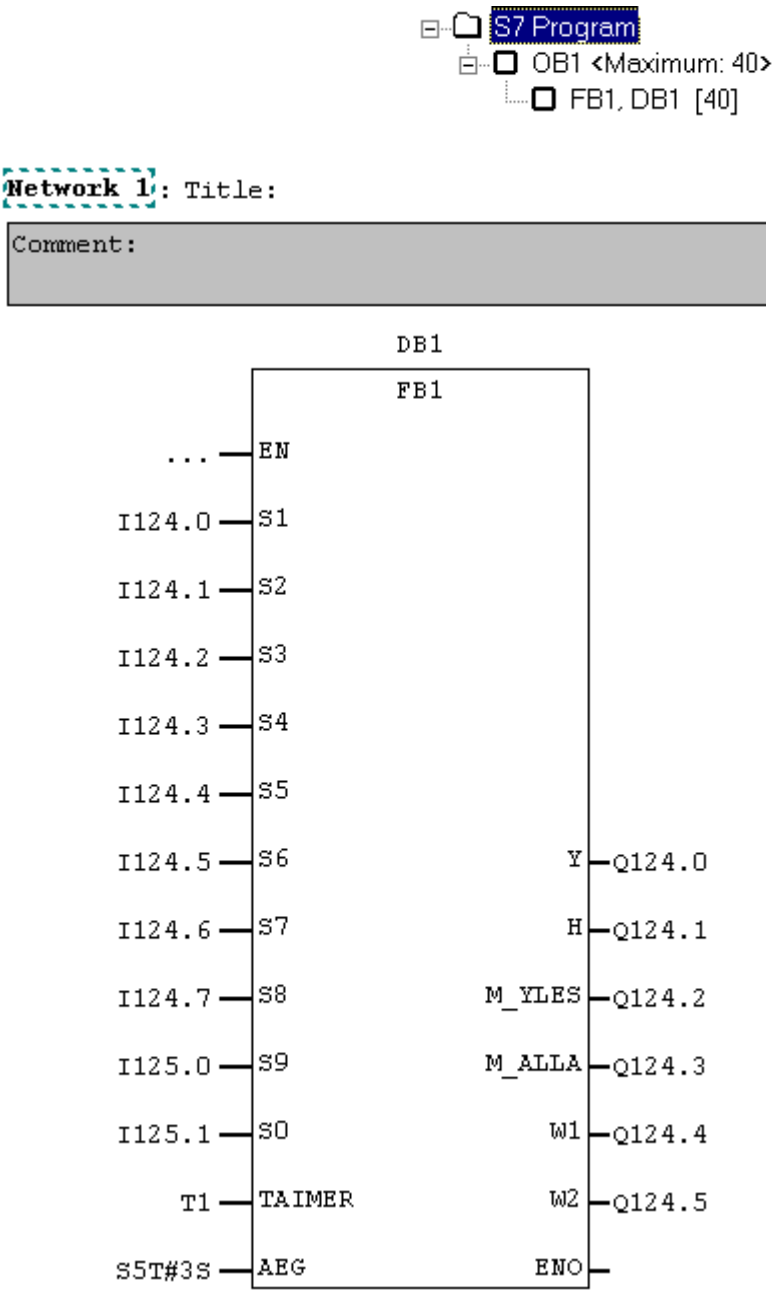
Kontrolleri S7-300 puhul kasutatakse programmeerimispaketti *SIMATIC Manager*. Programmi esitusviisina valime käsulisti. Programmi kirjutame funktsioonplokki FB1 ja operandidena kasutame muutujaid, et meil oleks võimalik tulevikus aega kokku hoida analoogse torupainutuseadme juhtimise programmeerimisel. Selleks deklareerimime funktsioonploki päises kasutatavad muutujad (joonis 11.22), mida kasutame hiljem programmi kirjutamisel (joonis 11.23).

```
Network 1: Transportliini mootori
           W2 juhtimine
      A(
      A     #S1
      A     #W1
      R     #MUUT_1
      A     #S1
      A     #S9
      A     #S0
      S     #MUUT_1
      A     #MUUT_1
      )
      A     #S4
      A     #S6
      AN    #S3
      AN    #S8
      AN    #MUUT_4
      =     #W2

Network 2: Kütte juhtimine
      A     #S2
      A     #W2
      S     #M
      A     #S7
      R     #M
      NOP   0

Network 3: Vahemuutuja
      A     #S8
      S     #MUUT_2
      AN    #S3
      R     #MUUT_2
      NOP   0

Network 4: Kaitsevõre alla
      A     #S3
      AN    #S5
      AN    #MUUT_2
      =     #M_ALLA

Network 5: Pressi juhtimine
      O(
      A     #S5
      AN    #S7
      A     #Y
      A     #MUUT_2
      L     #AEG
      SF    #TAIMER
      NOP   0
      NOP   0
      NOP   0
      A     #TAIMER
      )
      O     #S8
      =     #Y

Network 6: Kaitsevõre üles
      A(
      A     #S7
      A     #Y
      S     #MUUT_3
      A     #S1
      R     #MUUT_3
      A     #MUUT_3
      )
      A     #S6
      AN    #S4
      =     #M_YLES

Network 7: Transportliini mootori
           W1 juhtimine
      A(
      A     #S1
      R     #MUUT_4
      A     #M_YLES
      S     #MUUT_4
      A     #MUUT_4
      )
      AN    #S1
      A     #S4
      A     #S6
      AN    #S8
      =     #W1
```

Joonis 11.23 Funktsioonplokki kirjutatud juhtprogramm

Seejärel kutsume OB1-st välja FB1 kasutades selleks CALL käsku (joonis 11.24). Lõplik programmi struktuur näeb välja alljärgnev:

Seejärel kutsume OB1-st välja FB1 kasutades selleks CALL käsku (joonis 11.24). Lõplik programmi struktuur näeb välja alljärgnev:



Joonis 11.24 Juhtplokki OB1 laetud muutujatega funktsioonplokk

Seejärel kui programm on koostatud, laetakse see kontrollerisse ja testitakse