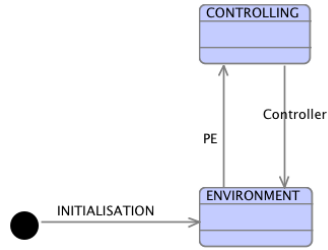


An Event-B Specification of Machine pw0
Creation Date: 13 Dec 2011 @ 11:10:16 PM



MACHINE Machine pw0

VARIABLES

ENVIRONMENT

CONTROLLING

INVARIANTS

typeof_ENVIRONMENT : ENVIRONMENT \in *BOOL*

typeof_CONTROLLING : CONTROLLING \in *BOOL*

distinct_states_in_rrm01 : $partition(\{TRUE\}, \{ENVIRONMENT\} \cap \{TRUE\}, \{CONTROLLING\} \cap \{TRUE\})$

EVENTS

Initialisation

begin

init_CONTROLLING : CONTROLLING := *FALSE*

init_ENVIRONMENT : ENVIRONMENT := *TRUE*

end

Event Controller $\hat{=}$

when

isin_CONTROLLING : CONTROLLING = *TRUE*

then

enter_ENVIRONMENT : ENVIRONMENT := *TRUE*

leave_CONTROLLING : CONTROLLING := *FALSE*

end

Event PE $\hat{=}$

when

isin_ENVIRONMENT : ENVIRONMENT = *TRUE*

then

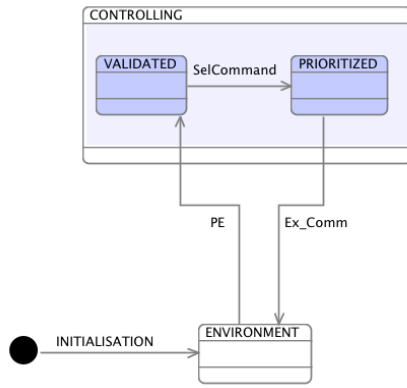
enter_CONTROLLING : CONTROLLING := *TRUE*

leave_ENVIRONMENT : ENVIRONMENT := *FALSE*

end

END

An Event-B Specification of Machine pw1
Creation Date: 13 Dec 2011 @ 11:10:19 PM



MACHINE pw1

REFINES pw0

VARIABLES

ENVIRONMENT

CONTROLLING

VALIDATED

PRIORITIZED

sel_comm Comment

up_out Comment

down_out Comment

INVARIANTS

typeof_VALIDATED : VALIDATED ∈ BOOL

typeof_PRIORITIZED : PRIORITIZED ∈ BOOL

distinct_states_in_rrm01 : (CONTROLLING = TRUE) ⇒ partition({ TRUE }, { VALIDATED } ∩ { TRUE }, { PRIORITIZED } ∩ { TRUE })

VALIDATED_substateof_CONTROLLING : (VALIDATED = TRUE) ⇒ (CONTROLLING = TRUE)

PRIORITIZED_substateof_CONTROLLING : (PRIORITIZED = TRUE) ⇒ (CONTROLLING = TRUE)

sel_comm_type : sel_comm ∈ −1 .. 1
Comment

up_out_type : up_out ∈ 0 .. 1
Comment

down_out_type : down_out ∈ 0 .. 1
Comment

EVENTS

Initialisation

extended

begin

init_CONTROLLING : CONTROLLING := FALSE

init_ENVIRONMENT : ENVIRONMENT := TRUE

init_up_out : up_out := 0

init_down_out : down_out := 0

init_PRIORITIZED : PRIORITIZED := FALSE

init_sel_comm : sel_comm := 0

init_VALIDATED : VALIDATED := FALSE

end

Event PE ≡

extends PE

when

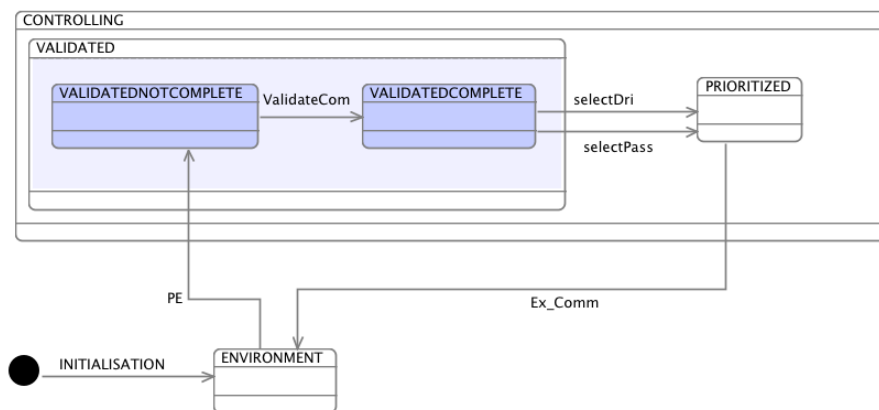
isin_ENVIRONMENT : ENVIRONMENT = TRUE

```

then
  enter_CONTROLLING : CONTROLLING := TRUE
  leave_ENVIRONMENT : ENVIRONMENT := FALSE
  enter_VALIDATED : VALIDATED := TRUE
end
Event Ex.Comm  $\hat{=}$ 
refines Controller
any
  dwn    Comment
  up    Comment
where
  dwn_type : dwn  $\in 0 \dots 1$ 
             Comment
  isin_PRIORITIZED : PRIORITIZED = TRUE
  up_type : up  $\in 0 \dots 1$ 
             Comment
then
  up_out_set : up_out := up
                Comment
  enter_ENVIRONMENT : ENVIRONMENT := TRUE
  down_out_set : down_out := dwn
                  Comment
  leave_PRIORITIZED : PRIORITIZED := FALSE
  leave_CONTROLLING : CONTROLLING := FALSE
end
Event SelCommand  $\hat{=}$ 
any
  cc
where
  isin_VALIDATED : VALIDATED = TRUE
  cc_type : cc  $\in -1 \dots 1$ 
then
  enter_PRIORITIZED : PRIORITIZED := TRUE
  leave_VALIDATED : VALIDATED := FALSE
  sel_comm_set : sel_comm := cc
                  Comment
end
END

```

An Event-B Specification of Machine pw2
Creation Date: 13 Dec 2011 @ 11:10:22 PM



MACHINE pw2

REFINES pw1

VARIABLES

ENVIRONMENT
 CONTROLLING
 VALIDATED
 VALIDATEDNOTCOMPLETE
 VALIDATEDCOMPLETE
 PRIORITIZED
 sel_comm Comment
 up_out Comment
 down_out Comment
 v_dri
 v_pass

INVARIANTS

typeof_VALIDATEDNOTCOMPLETE : $VALIDATEDNOTCOMPLETE \in \text{BOOL}$
 typeof_VALIDATEDCOMPLETE : $VALIDATEDCOMPLETE \in \text{BOOL}$
 distinct_states_in_rrm01 : $(VALIDATED = \text{TRUE}) \Rightarrow \text{partition}(\{\text{TRUE}\}, \{VALIDATEDNOTCOMPLETE, \text{TRUE}\}, \{VALIDATEDCOMPLETE\} \cap \{\text{TRUE}\})$
 VALIDATEDNOTCOMPLETE_substateof_VALIDATED : $(VALIDATEDNOTCOMPLETE = \text{TRUE}) \Rightarrow (VALIDATED = \text{TRUE})$
 VALIDATEDCOMPLETE_substateof_VALIDATED : $(VALIDATEDCOMPLETE = \text{TRUE}) \Rightarrow (VALIDATED = \text{TRUE})$
 v_dri_type : $v_dri \in 0 \dots 1$
 v_pass_type : $v_pass \in 0 \dots 1$

EVENTS

Initialisation

extended

begin

init_CONTROLLING : CONTROLLING := FALSE
 init_ENVIRONMENT : ENVIRONMENT := TRUE
 init_up_out : up_out := 0
 init_down_out : down_out := 0
 init_PRIORITIZED : PRIORITIZED := FALSE
 init_sel_comm : sel_comm := 0
 init_VALIDATED : VALIDATED := FALSE
 init_v_dri : v_dri := 0
 init_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
 init_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
 init_v_pass : v_pass := 0

end

Event $PE \triangleq$

extends PE

when

isin_ENVIRONMENT : ENVIRONMENT = TRUE

then

enter_CONTROLLING : CONTROLLING := TRUE
 leave_ENVIRONMENT : ENVIRONMENT := FALSE
 enter_VALIDATED : VALIDATED := TRUE
 enter_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := TRUE

end

Event $Ex_Comm \triangleq$

extends Ex_Comm

any

```

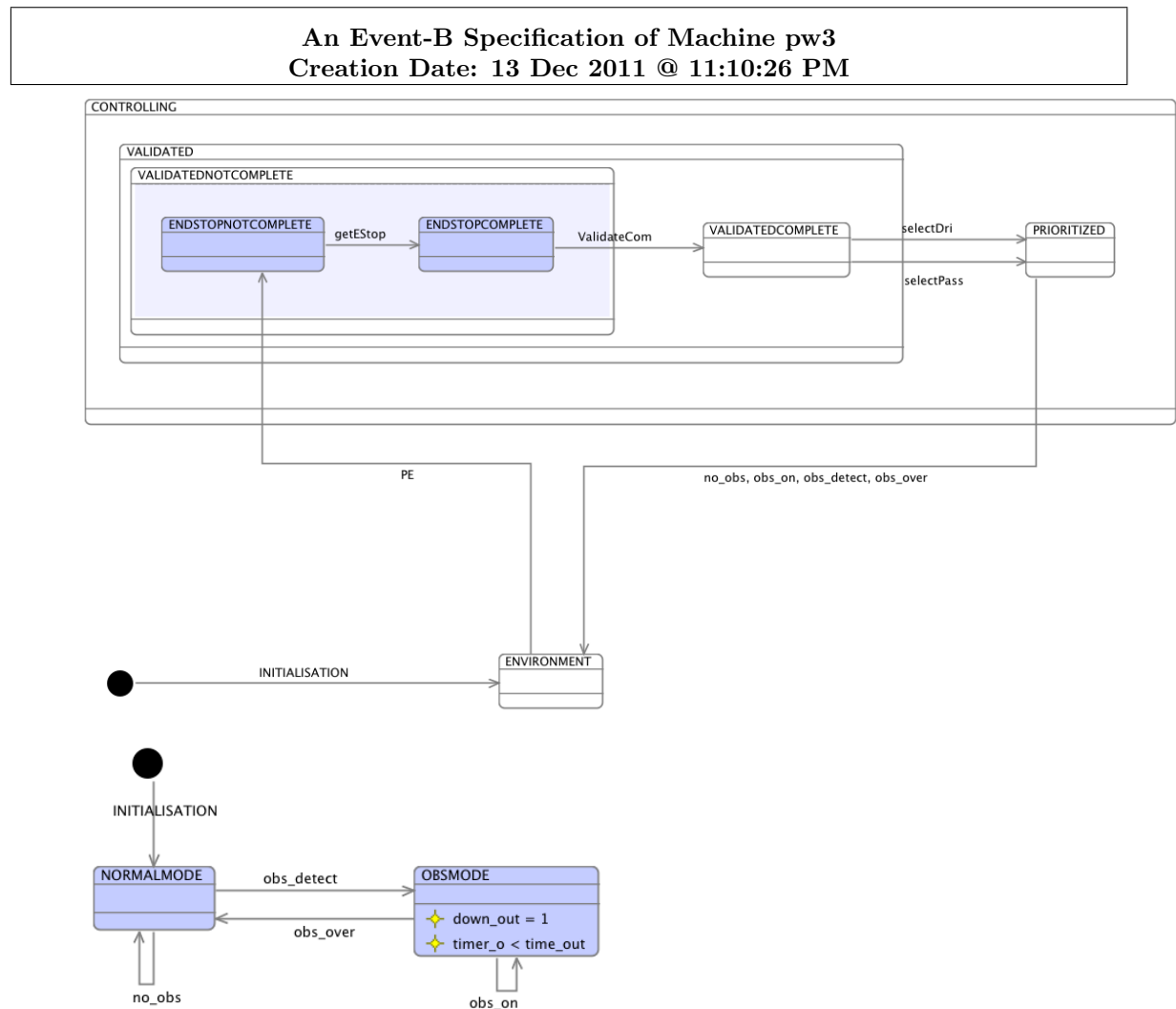
    dwn    Comment
    up     Comment
  where
    dwn_type : dwn ∈ 0 .. 1
    Comment
    isin_PRIORITIZED : PRIORITIZED = TRUE
    up_type : up ∈ 0 .. 1
    Comment
  then
    up_out_set : up_out := up
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    down_out_set : down_out := dwn
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    leave_CONTROLLING : CONTROLLING := FALSE
  end
Event ValidateCom ≡
  any
    pp
    dd
  where
    isin_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE = TRUE
    ppType : pp ∈ 0 .. 1
    ddType : dd ∈ 0 .. 1
  then
    enter_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := TRUE
    set_v_dri : v_dri := dd
    leave_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
    set_v_pass : v_pass := pp
  end
Event selectPass ≡
refines SelCommand
  when
    v_dri_notactive : v_dri = 0
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
  with
    cc : cc = v_pass
  then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    sel_comm_set : sel_comm := v_pass
    Comment
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    leave_VALIDATED : VALIDATED := FALSE
  end
Event selectDri ≡
refines SelCommand
  when
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
    v_dri_active : v_dri ≠ 0
  with
    cc : cc = v_dri
  then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    sel_comm_set : sel_comm := v_dri
    Comment

```

```

    leave_VALIDATED : VALIDATED := FALSE
  end
end
END

```



MACHINE pw3

REFINES pw2

SEES cx3

VARIABLES

ENVIRONMENT

CONTROLLING

VALIDATED

VALIDATEDNOTCOMPLETE

ENDSTOPNOTCOMPLETE

ENDSTOPCOMPLETE

VALIDATEDCOMPLETE

PRIORITIZED

OBSMODE

NORMALMODE

sel_comm Comment

up_out Comment

down_out Comment

```

v_dri
v_pass
obs
timer_o
endstop

```

INVARIANTS

```

typeof_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE ∈ BOOL
typeof_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE ∈ BOOL
distinct_states_in_rrm01 : (VALIDATEDNOTCOMPLETE = TRUE) ⇒ partition({TRUE}, {ENDSTOPNOTCOMPLETE = TRUE}, {ENDSTOPCOMPLETE} ∩ {TRUE})
ENDSTOPNOTCOMPLETE_substateof_VALIDATEDNOTCOMPLETE : (ENDSTOPNOTCOMPLETE = TRUE) ⇒ (VALIDATEDNOTCOMPLETE = TRUE)
ENDSTOPCOMPLETE_substateof_VALIDATEDNOTCOMPLETE : (ENDSTOPCOMPLETE = TRUE) ⇒ (VALIDATEDNOTCOMPLETE = TRUE)
typeof_OBSMODE : OBSMODE ∈ BOOL
typeof_NORMALMODE : NORMALMODE ∈ BOOL
distinct_states_in_ex_comm_sm1 : partition({TRUE}, {OBSMODE} ∩ {TRUE}, {NORMALMODE} ∩ {TRUE})
obs_down_inv : (OBSMODE = TRUE) ⇒ (down_out = 1)
obsModeTimerInv : (OBSMODE = TRUE) ⇒ (timer_o < time_out)
obstype : obs ∈ BOOL
timer_o.type : timer_o ∈ ℕ
endstop.type : endstop ∈ BOOL

```

EVENTS

Initialisation

extended

begin

```

init_CONTROLLING : CONTROLLING := FALSE
init_ENVIRONMENT : ENVIRONMENT := TRUE
init_up_out : up_out := 0
init_down_out : down_out := 0
init_PRIORITIZED : PRIORITIZED := FALSE
init_sel_comm : sel_comm := 0
init_VALIDATED : VALIDATED := FALSE
init_v_dri : v_dri := 0
init_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
init_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
init_v_pass : v_pass := 0
init_OBSMODE : OBSMODE := FALSE
init_endstop : endstop := FALSE
init_NORMALMODE : NORMALMODE := TRUE
init_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE
init_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := FALSE
init_obs : obs := FALSE
init_timer_o : timer_o := 0

```

end

Event *PE* $\hat{=}$

extends *PE*

when

```
isin_ENVIRONMENT : ENVIRONMENT = TRUE
```

then

```

enter_CONTROLLING : CONTROLLING := TRUE
leave_ENVIRONMENT : ENVIRONMENT := FALSE
enter_VALIDATED : VALIDATED := TRUE

```

```

    enter_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := TRUE
    set_obs : obs ∈ BOOL
    Comment
    enter_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := TRUE
end
Event ValidateCom ≐
extends ValidateCom
any
    pp
    dd
where
    isin_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE = TRUE
    ppType : pp ∈ 0..1
    ddType : dd ∈ 0..1
    isin_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE = TRUE
then
    enter_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := TRUE
    set_v_dri : v_dri := dd
    leave_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
    set_v_pass : v_pass := pp
    leave_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE
end
Event selectPass ≐
extends selectPass
when
    v_dri_notactive : v_dri = 0
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    sel_comm_set : sel_comm := v_pass
    Comment
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    leave_VALIDATED : VALIDATED := FALSE
end
Event selectDri ≐
extends selectDri
when
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
    v_dri_active : v_dri ≠ 0
then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    sel_comm_set : sel_comm := v_dri
    Comment
    leave_VALIDATED : VALIDATED := FALSE
end
Event no_obs ≐
refines Ex_Comm
any
    dwn    Comment
    up     Comment
where
    isin_PRIORITIZED : PRIORITIZED = TRUE
    noObstacle : obs = FALSE
    Comment

```



```

    up_type :  $up \in 0 \dots 1$ 
    Comment
    dwn_type :  $dwn \in 0 \dots 1$ 
    Comment
    isin_NORMALMODE :  $NORMALMODE = TRUE$ 
  then
    leave_PRIORITIZED :  $PRIORITIZED := FALSE$ 
    up_out_set :  $up\_out := up$ 
    Comment
    down_out_set :  $down\_out := dwn$ 
    Comment
    enter_ENVIRONMENT :  $ENVIRONMENT := TRUE$ 
    leave_CONTROLLING :  $CONTROLLING := FALSE$ 
  end
Event obs_on  $\hat{=}$ 
refines Ex_Comm
  when
    noTimeout :  $timer\_o + step < time\_out$ 
    user entered
    isin_PRIORITIZED :  $PRIORITIZED = TRUE$ 
    noEndstop :  $endstop = FALSE$ 
    user entered
    isin_OBSMODE :  $OBSMODE = TRUE$ 
  with
    up :  $up = 0$ 
    user entered
    dwn :  $dwn = 1$ 
    user entered
  then
    incTimer_o :  $timer\_o := timer\_o + step$ 
    user entered
    notUp :  $up\_out := 0$ 
    user entered
    leave_CONTROLLING :  $CONTROLLING := FALSE$ 
    enter_ENVIRONMENT :  $ENVIRONMENT := TRUE$ 
    leave_PRIORITIZED :  $PRIORITIZED := FALSE$ 
    goDown :  $down\_out := 1$ 
    user entered
  end
Event obs_detect  $\hat{=}$ 
refines Ex_Comm
  when
    isin_NORMALMODE :  $NORMALMODE = TRUE$ 
    isin_PRIORITIZED :  $PRIORITIZED = TRUE$ 
    obstaclePresent :  $obs = TRUE$ 
    Comment
  with
    up :  $up = 0$ 
    Comment
    dwn :  $dwn = 1$ 
    Comment
  then
    leave_NORMALMODE :  $NORMALMODE := FALSE$ 
    resetTimer_o :  $timer\_o := 0$ 
    Comment
    enter_ENVIRONMENT :  $ENVIRONMENT := TRUE$ 

```

```

    notUp: up_out := 0
    Comment
    leave_PRIORITIZED: PRIORITIZED := FALSE
    goDown: down_out := 1
    Comment
    leave_CONTROLLING: CONTROLLING := FALSE
    enter_OBSMODE: OBSMODE := TRUE
end
Event obs_over ≐
refines Ex_Comm
when
    isin_OBSMODE: OBSMODE = TRUE
    isin_PRIORITIZED: PRIORITIZED = TRUE
    timeoutOrendstop: timer_o + step ≥ time_out ∨ endstop = TRUE
    Comment
with
    up: up = 0
    Comment
    dwn: dwn = 0
    Comment
then
    leave_OBSMODE: OBSMODE := FALSE
    enter_NORMALMODE: NORMALMODE := TRUE
    leave_PRIORITIZED: PRIORITIZED := FALSE
    notUp: up_out := 0
    Comment
    notDown: down_out := 0
    Comment
    enter_ENVIRONMENT: ENVIRONMENT := TRUE
    leave_CONTROLLING: CONTROLLING := FALSE
end
Event getEStop ≐
any
    es    Comment
where
    estype: es ∈ BOOL
    Comment
    isin_ENDSTOPNOTCOMPLETE: ENDSTOPNOTCOMPLETE = TRUE
then
    set_endstop: endstop := es
    Comment
    enter_ENDSTOPCOMPLETE: ENDSTOPCOMPLETE := TRUE
    leave_ENDSTOPNOTCOMPLETE: ENDSTOPNOTCOMPLETE := FALSE
end
END

```

An Event-B Specification of Context cx3
 Creation Date: 13 Dec 2011 @ 11:10:09 PM

CONTEXT cx3

CONSTANTS

time_out

step

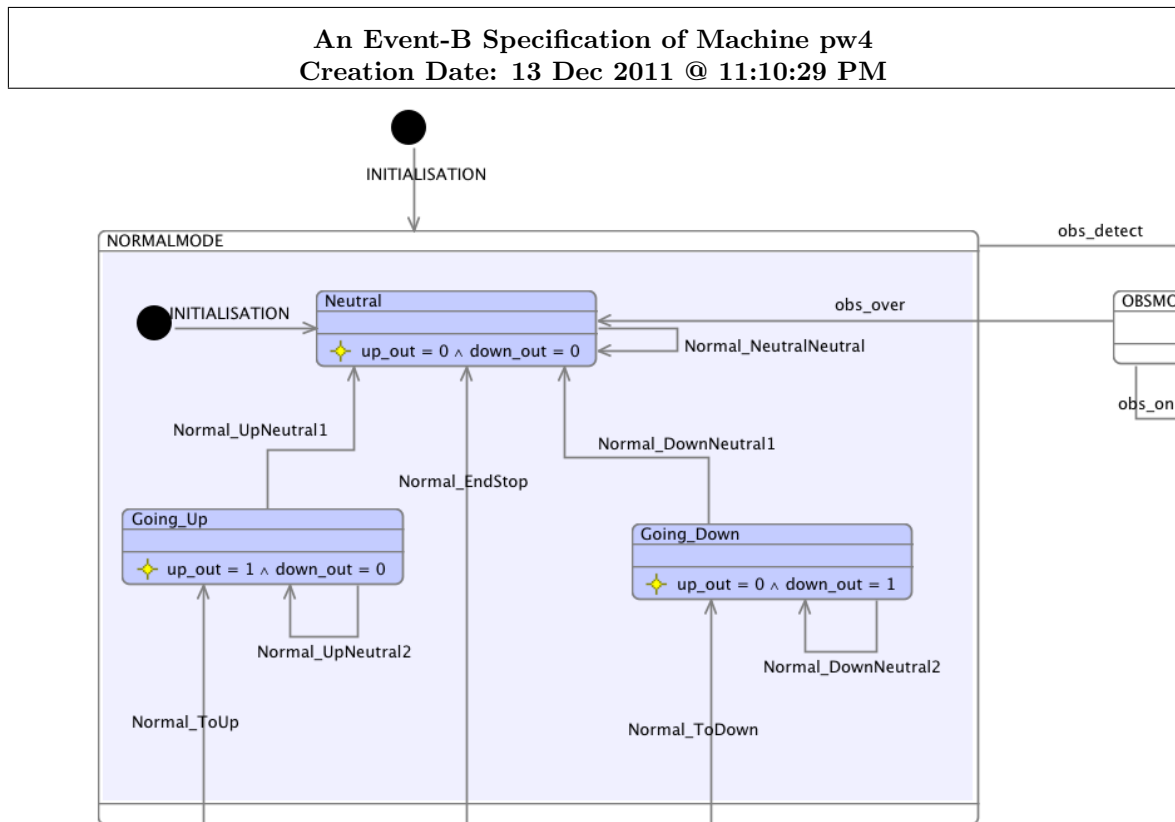
AXIOMS

time_outtype: time_out ∈ ℕ₁

```

steptype :  $step \in \mathbb{N}_1$ 
END

```



```

MACHINE    pw4
REFINES    pw3
SEES       cx3
VARIABLES
    NORMALMODE
    Neutral
    Going_Up
    Going_Down
    OBSMODE
    ENVIRONMENT
    CONTROLLING
    VALIDATED
    VALIDATEDNOTCOMPLETE
    ENDSTOPNOTCOMPLETE
    ENDSTOPCOMPLETE
    VALIDATEDCOMPLETE
    PRIORITIZED
    sel_comm      Comment
    up_out        Comment
    down_out      Comment
    v_dri
    v_pass
    obs

```

```

    timer_o
  endstop
INVARIANTS
  typeof_Neutral : Neutral ∈ BOOL
  typeof_Going_Up : Going_Up ∈ BOOL
  typeof_Going_Down : Going_Down ∈ BOOL
  distinct_states_in_ex_comm_sm1 : (NORMALMODE = TRUE) ⇒ partition({TRUE}, {Neutral} ∩ {TRUE}, {Going_Up} ∩ {TRUE}, {Going_Down} ∩ {TRUE})
  Neutral_substateof_NORMALMODE : (Neutral = TRUE) ⇒ (NORMALMODE = TRUE)
  Going_Up_substateof_NORMALMODE : (Going_Up = TRUE) ⇒ (NORMALMODE = TRUE)
  Going_Down_substateof_NORMALMODE : (Going_Down = TRUE) ⇒ (NORMALMODE = TRUE)
  NeutralInv : (Neutral = TRUE) ⇒ (up_out = 0 ∧ down_out = 0)
  UpInv : (Going_Up = TRUE) ⇒ (up_out = 1 ∧ down_out = 0)
  DownInv : (Going_Down = TRUE) ⇒ (up_out = 0 ∧ down_out = 1)
EVENTS
Initialisation
  extended
  begin
    init_CONTROLLING : CONTROLLING := FALSE
    init_ENVIRONMENT : ENVIRONMENT := TRUE
    init_up_out : up_out := 0
    init_down_out : down_out := 0
    init_PRIORITIZED : PRIORITIZED := FALSE
    init_sel_comm : sel_comm := 0
    init_VALIDATED : VALIDATED := FALSE
    init_v_dri : v_dri := 0
    init_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    init_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
    init_v_pass : v_pass := 0
    init_OBSMODE : OBSMODE := FALSE
    init_endstop : endstop := FALSE
    init_NORMALMODE : NORMALMODE := TRUE
    init_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE
    init_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := FALSE
    init_obs : obs := FALSE
    init_timer_o : timer_o := 0
    init_Going_Down : Going_Down := FALSE
    init_Neutral : Neutral := TRUE
    init_Going_Up : Going_Up := FALSE
  end
Event PE ≜
extends PE
  when
    isin_ENVIRONMENT : ENVIRONMENT = TRUE
  then
    enter_CONTROLLING : CONTROLLING := TRUE
    leave_ENVIRONMENT : ENVIRONMENT := FALSE
    enter_VALIDATED : VALIDATED := TRUE
    enter_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := TRUE
    set_obs : obs ∈ BOOL
    Comment
    enter_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := TRUE
  end
Event ValidateCom ≜

```

extends *ValidateCom*

any

pp
dd

where

isin_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE = TRUE
ppType : pp ∈ 0..1
ddType : dd ∈ 0..1
isin_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE = TRUE

then

enter_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := TRUE
set_v_dri : v_dri := dd
leave_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
set_v_pass : v_pass := pp
leave_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE

end

Event *selectPass* $\hat{=}$

extends *selectPass*

when

v_dri_notactive : v_dri = 0
isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE

then

enter_PRIORITIZED : PRIORITIZED := TRUE
sel_comm_set : sel_comm := v_pass
Comment
leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
leave_VALIDATED : VALIDATED := FALSE

end

Event *selectDri* $\hat{=}$

extends *selectDri*

when

isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
v_dri_active : v_dri ≠ 0

then

enter_PRIORITIZED : PRIORITIZED := TRUE
leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
sel_comm_set : sel_comm := v_dri
Comment
leave_VALIDATED : VALIDATED := FALSE

end

Event *obs_on* $\hat{=}$

extends *obs_on*

when

noTimeout : timer_o + step < time_out
user entered
isin_PRIORITIZED : PRIORITIZED = TRUE
noEndstop : endstop = FALSE
user entered
isin_OBSMODE : OBSMODE = TRUE

then

incTimer_o : timer_o := timer_o + step
user entered
notUp : up_out := 0
user entered
leave_CONTROLLING : CONTROLLING := FALSE
enter_ENVIRONMENT : ENVIRONMENT := TRUE

```

    leave_PRIORITIZED : PRIORITIZED := FALSE
    goDown : down_out := 1
    user entered
end
Event obs_detect ≐
extends obs_detect
when
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    obstaclePresent : obs = TRUE
    Comment
then
    leave_NORMALMODE : NORMALMODE := FALSE
    resetTimer_o : timer_o := 0
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    notUp : up_out := 0
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    goDown : down_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_OBSMODE : OBSMODE := TRUE
    leave_Going_Up : Going_Up := FALSE
    leave_Neutral : Neutral := FALSE
    leave_Going_Down : Going_Down := FALSE
end
Event obs_over ≐
extends obs_over
when
    isin_OBSMODE : OBSMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    timeoutOrendstop : timer_o + step ≥ time_out ∨ endstop = TRUE
    Comment
then
    leave_OBSMODE : OBSMODE := FALSE
    enter_NORMALMODE : NORMALMODE := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    notUp : up_out := 0
    Comment
    notDown : down_out := 0
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Neutral : Neutral := TRUE
end
Event getEStop ≐
extends getEStop
any
    es    Comment
where
    estype : es ∈ BOOL
    Comment
    isin_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE = TRUE
then

```

```

    set_endstop : endstop := es
    Comment
    enter_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := TRUE
    leave_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := FALSE
end
Event Normal_UpNeutral2  $\hat{=}$ 
refines no_obs
  when
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Up : Going_Up = TRUE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
  with
    up : up = 1
    dwn : dwn = 0
  then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    up_out_set : up_out := 1
    Comment
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
  end
Event Normal_ToUp  $\hat{=}$ 
refines no_obs
  when
    grd3 : sel_comm = 1
    grd1 : endstop = FALSE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_NORMALMODE : NORMALMODE = TRUE
  with
    up : up = 1
    dwn : dwn = 0
  then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    enter_Going_Up : Going_Up := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    down_out_set : down_out := 0
    Comment
    up_out_set : up_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_Neutral : Neutral := FALSE
    leave_Going_Down : Going_Down := FALSE
  end
Event Normal_ToDown  $\hat{=}$ 
refines no_obs
  when
    grd3 : sel_comm = -1
    grd1 : endstop = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd2 : obs = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE

```

```

with
  up : up = 0
  dwn : dwn = 1
then
  leave_Going_Up : Going_Up := FALSE
  down_out_set : down_out := 1
    Comment
  leave_CONTROLLING : CONTROLLING := FALSE
  enter_Going_Down : Going_Down := TRUE
  up_out_set : up_out := 0
    Comment
  leave_PRIORITIZED : PRIORITIZED := FALSE
  leave_Neutral : Neutral := FALSE
  enter_ENVIRONMENT : ENVIRONMENT := TRUE
end
Event Normal_DownNeutral2  $\hat{=}$ 
refines no_obs
when
  isin_Going_Down : Going_Down = TRUE
  grd2 : obs = FALSE
  grd1 : endstop = FALSE
  isin_NORMALMODE : NORMALMODE = TRUE
  isin_PRIORITIZED : PRIORITIZED = TRUE
with
  up : up = 0
  dwn : dwn = 1
then
  up_out_set : up_out := 0
    Comment
  leave_CONTROLLING : CONTROLLING := FALSE
  leave_PRIORITIZED : PRIORITIZED := FALSE
  enter_ENVIRONMENT : ENVIRONMENT := TRUE
  down_out_set : down_out := 1
    Comment
end
Event Normal_NeutralNeutral  $\hat{=}$ 
refines no_obs
when
  noObstacle : obs = FALSE
    Comment
  grd1 : sel_comm = 0
  isin_Neutral : Neutral = TRUE
  isin_PRIORITIZED : PRIORITIZED = TRUE
  isin_NORMALMODE : NORMALMODE = TRUE
with
  up : up = 0
  dwn : dwn = 0
then
  leave_PRIORITIZED : PRIORITIZED := FALSE
  up_out_set : up_out := 0
    Comment
  enter_ENVIRONMENT : ENVIRONMENT := TRUE
  leave_CONTROLLING : CONTROLLING := FALSE
  down_out_set : down_out := 0
    Comment
end

```


Event *Normal_UpNeutral1* $\hat{=}$

refines *no_obs*

when

isin_PRIORITIZED : *PRIORITIZED* = *TRUE*
grd2 : *endstop* = *FALSE*
grd1 : *obs* = *FALSE*
isin_NORMALMODE : *NORMALMODE* = *TRUE*
grd3 : *sel_comm* = 0
isin_Going_Up : *Going_Up* = *TRUE*

with

up : *up* = 0
dwn : *dwn* = 0

then

down_out_set : *down_out* := 0
 Comment
leave_PRIORITIZED : *PRIORITIZED* := *FALSE*
enter_ENVIRONMENT : *ENVIRONMENT* := *TRUE*
up_out_set : *up_out* := 0
 Comment
enter_Neutral : *Neutral* := *TRUE*
leave_Going_Up : *Going_Up* := *FALSE*
leave_CONTROLLING : *CONTROLLING* := *FALSE*

end

Event *Normal_EndStop* $\hat{=}$

refines *no_obs*

when

isin_NORMALMODE : *NORMALMODE* = *TRUE*
grd2 : *endstop* = *TRUE*
isin_PRIORITIZED : *PRIORITIZED* = *TRUE*
grd1 : *obs* = *FALSE*

with

up : *up* = 0
dwn : *dwn* = 0

then

up_out_set : *up_out* := 0
 Comment
down_out_set : *down_out* := 0
 Comment
leave_Going_Down : *Going_Down* := *FALSE*
enter_ENVIRONMENT : *ENVIRONMENT* := *TRUE*
leave_CONTROLLING : *CONTROLLING* := *FALSE*
enter_Neutral : *Neutral* := *TRUE*
leave_PRIORITIZED : *PRIORITIZED* := *FALSE*
leave_Going_Up : *Going_Up* := *FALSE*

end

Event *Normal_DownNeutral1* $\hat{=}$

refines *no_obs*

when

grd1 : *obs* = *FALSE*
isin_PRIORITIZED : *PRIORITIZED* = *TRUE*
isin_NORMALMODE : *NORMALMODE* = *TRUE*
grd3 : *sel_comm* = 0
grd2 : *endstop* = *FALSE*
isin_Going_Down : *Going_Down* = *TRUE*

with

up : *up* = 0

```

    dwn : dwn = 0
  then
    leave_PRIORITIZED : PRIORITIZED := FALSE
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Neutral : Neutral := TRUE
    up_out_set : up_out := 0
    Comment
    leave_Going_Down : Going_Down := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
  end
END

```

An Event-B Specification of Context cx4
Creation Date: 13 Dec 2011 @ 11:10:14 PM

CONTEXT cx4

EXTENDS cx3

CONSTANTS

Timeout_dn

Timeout_up

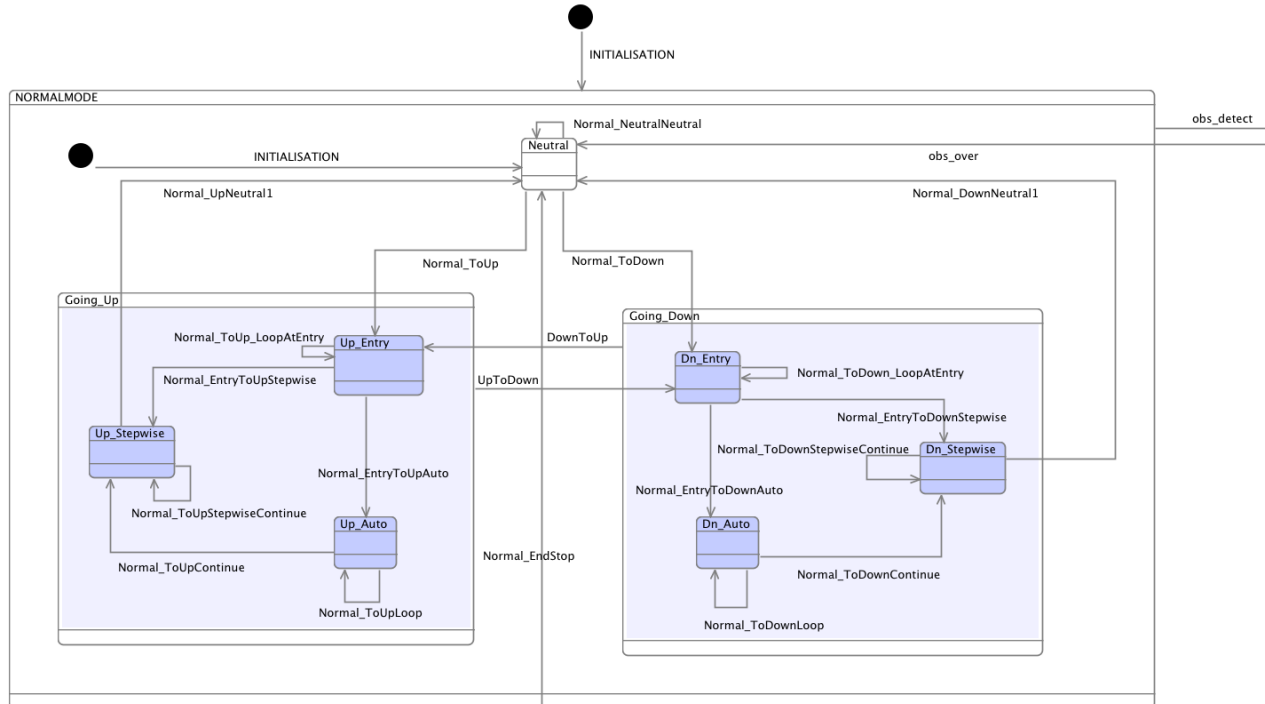
AXIOMS

Timeout_dn_axiom : $Timeout_dn \in \mathbb{N}_1$

Timeout_up_axiom : $Timeout_up \in \mathbb{N}_1$

END

An Event-B Specification of Machine pw5
Creation Date: 13 Dec 2011 @ 11:10:32 PM



MACHINE pw5

REFINES pw4

SEES cx4**VARIABLES**

NORMALMODE
 Neutral
 Going_Up
 Up_Entry
 Up_Auto
 Up_Stepwise
 Going_Down
 Dn_Entry
 Dn_Auto
 Dn_Stepwise
 OBSMODE
 ENVIRONMENT
 CONTROLLING
 VALIDATED
 VALIDATEDNOTCOMPLETE
 ENDSTOPNOTCOMPLETE
 ENDSTOPCOMPLETE
 VALIDATEDCOMPLETE
 PRIORITIZED
 sel_comm Comment
 up_out Comment
 down_out Comment
 v_dri
 v_pass
 obs
 timer_o
 endstop
 timer_auto_dn
 timer_auto_up

INVARIANTS

typeof_Up_Entry : $Up_Entry \in \text{BOOL}$
 typeof_Up_Auto : $Up_Auto \in \text{BOOL}$
 typeof_Up_Stepwise : $Up_Stepwise \in \text{BOOL}$
 typeof_Dn_Entry : $Dn_Entry \in \text{BOOL}$
 typeof_Dn_Auto : $Dn_Auto \in \text{BOOL}$
 typeof_Dn_Stepwise : $Dn_Stepwise \in \text{BOOL}$
 distinct_states_in_ex_comm_sm1 : $(Going_Up = \text{TRUE}) \Rightarrow \text{partition}(\{\text{TRUE}\}, \{Up_Entry\} \cap \{\text{TRUE}\}, \{Up_Auto\} \cap \{\text{TRUE}\}, \{Up_Stepwise\} \cap \{\text{TRUE}\})$
 distinct_states_in_ex_comm_sm2 : $(Going_Down = \text{TRUE}) \Rightarrow \text{partition}(\{\text{TRUE}\}, \{Dn_Entry\} \cap \{\text{TRUE}\}, \{Dn_Auto\} \cap \{\text{TRUE}\}, \{Dn_Stepwise\} \cap \{\text{TRUE}\})$
 Up_Entry_substateof_Going_Up : $(Up_Entry = \text{TRUE}) \Rightarrow (Going_Up = \text{TRUE})$
 Up_Auto_substateof_Going_Up : $(Up_Auto = \text{TRUE}) \Rightarrow (Going_Up = \text{TRUE})$
 Up_Stepwise_substateof_Going_Up : $(Up_Stepwise = \text{TRUE}) \Rightarrow (Going_Up = \text{TRUE})$
 Dn_Entry_substateof_Going_Down : $(Dn_Entry = \text{TRUE}) \Rightarrow (Going_Down = \text{TRUE})$
 Dn_Auto_substateof_Going_Down : $(Dn_Auto = \text{TRUE}) \Rightarrow (Going_Down = \text{TRUE})$
 Dn_Stepwise_substateof_Going_Down : $(Dn_Stepwise = \text{TRUE}) \Rightarrow (Going_Down = \text{TRUE})$
 inv1 : $timer_auto_dn \in \mathbb{N}$
 inv2 : $timer_auto_up \in \mathbb{N}$

EVENTS**Initialisation***extended***begin**

```

init_CONTROLLING : CONTROLLING := FALSE
init_ENVIRONMENT : ENVIRONMENT := TRUE
init_up_out : up_out := 0
init_down_out : down_out := 0
init_PRIORITIZED : PRIORITIZED := FALSE
init_sel_comm : sel_comm := 0
init_VALIDATED : VALIDATED := FALSE
init_v_dri : v_dri := 0
init_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
init_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
init_v_pass : v_pass := 0
init_OBSMODE : OBSMODE := FALSE
init_endstop : endstop := FALSE
init_NORMALMODE : NORMALMODE := TRUE
init_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE
init_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := FALSE
init_obs : obs := FALSE
init_timer_o : timer_o := 0
init_Going_Down : Going_Down := FALSE
init_Neutral : Neutral := TRUE
init_Going_Up : Going_Up := FALSE
init_Up_Stepwise : Up_Stepwise := FALSE
init_Dn_Auto : Dn_Auto := FALSE
init_Up_Entry : Up_Entry := FALSE
act2 : timer_auto_up := 0
init_Dn_Entry : Dn_Entry := FALSE
init_Dn_Stepwise : Dn_Stepwise := FALSE
act1 : timer_auto_dn := 0
init_Up_Auto : Up_Auto := FALSE

```

end**Event** *PE* $\hat{=}$ **extends** *PE***when***isin_ENVIRONMENT* : ENVIRONMENT = TRUE**then**

```

enter_CONTROLLING : CONTROLLING := TRUE
leave_ENVIRONMENT : ENVIRONMENT := FALSE
enter_VALIDATED : VALIDATED := TRUE
enter_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := TRUE
set_obs : obs : $\in$  BOOL
    Comment
enter_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := TRUE

```

end**Event** *ValidateCom* $\hat{=}$ **extends** *ValidateCom***any***pp**dd***where**

```

isin_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE = TRUE
ppType : pp  $\in$  0 .. 1
ddType : dd  $\in$  0 .. 1

```

```

    isin_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE = TRUE
  then
    enter_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := TRUE
    set_v_dri : v_dri := dd
    leave_VALIDATEDNOTCOMPLETE : VALIDATEDNOTCOMPLETE := FALSE
    set_v_pass : v_pass := pp
    leave_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := FALSE
  end
Event selectPass  $\hat{=}$ 
extends selectPass
  when
    v_dri_notactive : v_dri = 0
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
  then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    sel_comm_set : sel_comm := v_pass
    Comment
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    leave_VALIDATED : VALIDATED := FALSE
  end
Event selectDri  $\hat{=}$ 
extends selectDri
  when
    isin_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE = TRUE
    v_dri_active : v_dri  $\neq$  0
  then
    enter_PRIORITIZED : PRIORITIZED := TRUE
    leave_VALIDATEDCOMPLETE : VALIDATEDCOMPLETE := FALSE
    sel_comm_set : sel_comm := v_dri
    Comment
    leave_VALIDATED : VALIDATED := FALSE
  end
Event obs_on  $\hat{=}$ 
extends obs_on
  when
    noTimeout : timer_o + step < time_out
    user entered
    isin_PRIORITIZED : PRIORITIZED = TRUE
    noEndstop : endstop = FALSE
    user entered
    isin_OBSMODE : OBSMODE = TRUE
  then
    incTimer_o : timer_o := timer_o + step
    user entered
    notUp : up_out := 0
    user entered
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    goDown : down_out := 1
    user entered
  end
Event obs_detect  $\hat{=}$ 
extends obs_detect
  when

```

```

    isin_NORMALMODE : NORMALMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    obstaclePresent : obs = TRUE
    Comment
  then
    leave_NORMALMODE : NORMALMODE := FALSE
    resetTimer_o : timer_o := 0
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    notUp : up_out := 0
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    goDown : down_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_OBSMODE : OBSMODE := TRUE
    leave_Going_Up : Going_Up := FALSE
    leave_Neutral : Neutral := FALSE
    leave_Going_Down : Going_Down := FALSE
    leave_Up_Entry : Up_Entry := FALSE
    leave_Up_Auto : Up_Auto := FALSE
    leave_Dn_Auto : Dn_Auto := FALSE
    leave_Dn_Entry : Dn_Entry := FALSE
    leave_Up_Stepwise : Up_Stepwise := FALSE
    leave_Dn_Stepwise : Dn_Stepwise := FALSE
  end
Event obs_over  $\hat{=}$ 
extends obs_over
  when
    isin_OBSMODE : OBSMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    timeoutOrendstop : timer_o + step  $\geq$  time_out  $\vee$  endstop = TRUE
    Comment
  then
    leave_OBSMODE : OBSMODE := FALSE
    enter_NORMALMODE : NORMALMODE := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    notUp : up_out := 0
    Comment
    notDown : down_out := 0
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Neutral : Neutral := TRUE
  end
Event getEStop  $\hat{=}$ 
extends getEStop
  any
    es    Comment
  where
    estype : es  $\in$  BOOL
    Comment
    isin_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE = TRUE
  then
    set_endstop : endstop := es
    Comment

```

```

    enter_ENDSTOPCOMPLETE : ENDSTOPCOMPLETE := TRUE
    leave_ENDSTOPNOTCOMPLETE : ENDSTOPNOTCOMPLETE := FALSE
end
Event Normal_ToUp  $\hat{=}$ 
extends Normal_ToUp
  when
    grd3 : sel_comm = 1
    grd1 : endstop = FALSE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Neutral : Neutral = TRUE
  then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    enter_Going_Up : Going_Up := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    down_out_set : down_out := 0
    Comment
    up_out_set : up_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_Neutral : Neutral := FALSE
    leave_Going_Down : Going_Down := FALSE
    act1 : timer_auto_up := 0
    enter_Up_Entry : Up_Entry := TRUE
  end
Event Normal_ToDown  $\hat{=}$ 
extends Normal_ToDown
  when
    grd3 : sel_comm = -1
    grd1 : endstop = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd2 : obs = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Neutral : Neutral = TRUE
  then
    leave_Going_Up : Going_Up := FALSE
    down_out_set : down_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Going_Down : Going_Down := TRUE
    up_out_set : up_out := 0
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    leave_Neutral : Neutral := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    act7 : timer_auto_dn := 0
    enter_Dn_Entry : Dn_Entry := TRUE
  end
Event Normal_NeutralNeutral  $\hat{=}$ 
extends Normal_NeutralNeutral
  when
    noObstacle : obs = FALSE
    Comment
    grd1 : sel_comm = 0
    isin_Neutral : Neutral = TRUE

```

```

    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_NORMALMODE : NORMALMODE = TRUE
  then
    leave_PRIORITIZED : PRIORITIZED := FALSE
    up_out_set : up_out := 0
    Comment
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_CONTROLLING : CONTROLLING := FALSE
    down_out_set : down_out := 0
    Comment
  end
Event Normal_UpNeutral1  $\hat{=}$ 
extends Normal_UpNeutral1
  when
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd2 : endstop = FALSE
    grd1 : obs = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    grd3 : sel_comm = 0
    isin_Going_Up : Going_Up = TRUE
    isin_Up_Stepwise : Up_Stepwise = TRUE
  then
    down_out_set : down_out := 0
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    up_out_set : up_out := 0
    Comment
    enter_Neutral : Neutral := TRUE
    leave_Going_Up : Going_Up := FALSE
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_Up_Stepwise : Up_Stepwise := FALSE
  end
Event Normal_EndStop  $\hat{=}$ 
extends Normal_EndStop
  when
    isin_NORMALMODE : NORMALMODE = TRUE
    grd2 : endstop = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd1 : obs = FALSE
  then
    up_out_set : up_out := 0
    Comment
    down_out_set : down_out := 0
    Comment
    leave_Going_Down : Going_Down := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Neutral : Neutral := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    leave_Going_Up : Going_Up := FALSE
    leave_Up_Entry : Up_Entry := FALSE
    leave_Up_Auto : Up_Auto := FALSE
    leave_Dn_Stepwise : Dn_Stepwise := FALSE
    leave_Dn_Auto : Dn_Auto := FALSE
    leave_Up_Stepwise : Up_Stepwise := FALSE

```



```

    leave_Dn_Entry : Dn_Entry := FALSE
  end
Event Normal_DownNeutral1  $\hat{=}$ 
extends Normal_DownNeutral1
  when
    grd1 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_NORMALMODE : NORMALMODE = TRUE
    grd3 : sel_comm = 0
    grd2 : endstop = FALSE
    isin_Going_Down : Going_Down = TRUE
    isin_Dn_Stepwise : Dn_Stepwise = TRUE
  then
    leave_PRIORITIZED : PRIORITIZED := FALSE
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Neutral : Neutral := TRUE
    up_out_set : up_out := 0
    Comment
    leave_Going_Down : Going_Down := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_Dn_Stepwise : Dn_Stepwise := FALSE
  end
Event Normal_ToDown_LoopAtEntry  $\hat{=}$ 
extends Normal_DownNeutral2
  when
    isin_Going_Down : Going_Down = TRUE
    grd2 : obs = FALSE
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_Dn_Entry : Dn_Entry = TRUE
    grd4 : timer_auto_dn < Timeout_dn
    grd3 : sel_comm = -1
  then
    up_out_set : up_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    down_out_set : down_out := 1
    Comment
    act1 : timer_auto_up := timer_auto_up + step
  end
Event Normal_ToUp_LoopAtEntry  $\hat{=}$ 
extends Normal_UpNeutral2
  when
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Up : Going_Up = TRUE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_Up_Entry : Up_Entry = TRUE
    grd3 : sel_comm = 1
    grd4 : timer_auto_up < Timeout_up

```

```

    then
        enter_ENVIRONMENT : ENVIRONMENT := TRUE
        up_out_set : up_out := 1
        Comment
        down_out_set : down_out := 0
        Comment
        leave_CONTROLLING : CONTROLLING := FALSE
        leave_PRIORITIZED : PRIORITIZED := FALSE
        act1 : timer_auto_up := timer_auto_up + step
    end
Event Normal_EntryToDownAuto  $\hat{=}$ 
extends Normal_DownNeutral2
when
    isin_Going_Down : Going_Down = TRUE
    grd2 : obs = FALSE
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_Dn_Entry : Dn_Entry = TRUE
    grd4 : sel_comm = 0
then
    up_out_set : up_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    down_out_set : down_out := 1
    Comment
    enter_Dn_Auto : Dn_Auto := TRUE
    leave_Dn_Entry : Dn_Entry := FALSE
end
Event Normal_EntryToUpAuto  $\hat{=}$ 
extends Normal_UpNeutral2
when
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Up : Going_Up = TRUE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_Up_Entry : Up_Entry = TRUE
    grd3 : sel_comm = 0
then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    up_out_set : up_out := 1
    Comment
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    enter_Up_Auto : Up_Auto := TRUE
    leave_Up_Entry : Up_Entry := FALSE
end
Event Normal_EntryToDownStepwise  $\hat{=}$ 
extends Normal_DownNeutral2
when
    isin_Going_Down : Going_Down = TRUE

```

```

    grd2: obs = FALSE
    grd1: endstop = FALSE
    isin_NORMALMODE: NORMALMODE = TRUE
    isin_PRIORITIZED: PRIORITIZED = TRUE
    isin_Dn_Entry: Dn_Entry = TRUE
    grd3: timer_auto_dn  $\geq$  Timeout_dn
    grd4: sel_comm = -1
  then
    up_out_set: up_out := 0
    Comment
    leave_CONTROLLING: CONTROLLING := FALSE
    leave_PRIORITIZED: PRIORITIZED := FALSE
    enter_ENVIRONMENT: ENVIRONMENT := TRUE
    down_out_set: down_out := 1
    Comment
    enter_Dn_Stepwise: Dn_Stepwise := TRUE
    leave_Dn_Entry: Dn_Entry := FALSE
  end
Event Normal_EntryToUpStepwise  $\hat{=}$ 
extends Normal_UpNeutral2
  when
    grd1: endstop = FALSE
    isin_NORMALMODE: NORMALMODE = TRUE
    isin_Going_Up: Going_Up = TRUE
    grd2: obs = FALSE
    isin_PRIORITIZED: PRIORITIZED = TRUE
    grd3: timer_auto_up  $\geq$  Timeout_up
    grd4: sel_comm = 1
    isin_Up_Entry: Up_Entry = TRUE
  then
    enter_ENVIRONMENT: ENVIRONMENT := TRUE
    up_out_set: up_out := 1
    Comment
    down_out_set: down_out := 0
    Comment
    leave_CONTROLLING: CONTROLLING := FALSE
    leave_PRIORITIZED: PRIORITIZED := FALSE
    leave_Up_Entry: Up_Entry := FALSE
    enter_Up_Stepwise: Up_Stepwise := TRUE
  end
Event Normal_ToDownContinue  $\hat{=}$ 
extends Normal_DownNeutral2
  when
    isin_Going_Down: Going_Down = TRUE
    grd2: obs = FALSE
    grd1: endstop = FALSE
    isin_NORMALMODE: NORMALMODE = TRUE
    isin_PRIORITIZED: PRIORITIZED = TRUE
    grd3: sel_comm = -1
    isin_Dn_Auto: Dn_Auto = TRUE
  then
    up_out_set: up_out := 0
    Comment
    leave_CONTROLLING: CONTROLLING := FALSE
    leave_PRIORITIZED: PRIORITIZED := FALSE
    enter_ENVIRONMENT: ENVIRONMENT := TRUE

```

```

        down_out_set : down_out := 1
        Comment
        enter_Dn_Stepwise : Dn_Stepwise := TRUE
        leave_Dn_Auto : Dn_Auto := FALSE
    end
Event Normal_ToUpContinue  $\hat{=}$ 
extends Normal_UpNeutral2
    when
        grd1 : endstop = FALSE
        isin_NORMALMODE : NORMALMODE = TRUE
        isin_Going_Up : Going_Up = TRUE
        grd2 : obs = FALSE
        isin_PRIORITIZED : PRIORITIZED = TRUE
        isin_Up_Auto : Up_Auto = TRUE
        grd3 : sel_comm = 1
    then
        enter_ENVIRONMENT : ENVIRONMENT := TRUE
        up_out_set : up_out := 1
        Comment
        down_out_set : down_out := 0
        Comment
        leave_CONTROLLING : CONTROLLING := FALSE
        leave_PRIORITIZED : PRIORITIZED := FALSE
        leave_Up_Auto : Up_Auto := FALSE
        enter_Up_Stepwise : Up_Stepwise := TRUE
    end
Event Normal_ToDownLoop  $\hat{=}$ 
extends Normal_DownNeutral2
    when
        isin_Going_Down : Going_Down = TRUE
        grd2 : obs = FALSE
        grd1 : endstop = FALSE
        isin_NORMALMODE : NORMALMODE = TRUE
        isin_PRIORITIZED : PRIORITIZED = TRUE
        grd3 : sel_comm = 0
        isin_Dn_Auto : Dn_Auto = TRUE
    then
        up_out_set : up_out := 0
        Comment
        leave_CONTROLLING : CONTROLLING := FALSE
        leave_PRIORITIZED : PRIORITIZED := FALSE
        enter_ENVIRONMENT : ENVIRONMENT := TRUE
        down_out_set : down_out := 1
        Comment
    end
Event Normal_ToUpLoop  $\hat{=}$ 
extends Normal_UpNeutral2
    when
        grd1 : endstop = FALSE
        isin_NORMALMODE : NORMALMODE = TRUE
        isin_Going_Up : Going_Up = TRUE
        grd2 : obs = FALSE
        isin_PRIORITIZED : PRIORITIZED = TRUE
        isin_Up_Auto : Up_Auto = TRUE
        grd3 : sel_comm = 0
    then

```

```

    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    up_out_set : up_out := 1
    Comment
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
end
Event Normal_ToDownStepwiseContinue  $\hat{=}$ 
extends Normal_DownNeutral2
  when
    isin_Going_Down : Going_Down = TRUE
    grd2 : obs = FALSE
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd3 : sel_comm = -1
    isin_Dn_Stepwise : Dn_Stepwise = TRUE
  then
    up_out_set : up_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    down_out_set : down_out := 1
    Comment
  end
Event Normal_ToUpStepwiseContinue  $\hat{=}$ 
extends Normal_UpNeutral2
  when
    grd1 : endstop = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Up : Going_Up = TRUE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_Up_Stepwise : Up_Stepwise = TRUE
    grd3 : sel_comm = 1
  then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    up_out_set : up_out := 1
    Comment
    down_out_set : down_out := 0
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_PRIORITIZED : PRIORITIZED := FALSE
  end
Event UpToDown  $\hat{=}$ 
extends Normal_ToDown
  when
    grd3 : sel_comm = -1
    grd1 : endstop = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    grd2 : obs = FALSE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Up : Going_Up = TRUE
  then

```

```

    leave_Going_Up : Going_Up := FALSE
    down_out_set : down_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    enter_Going_Down : Going_Down := TRUE
    up_out_set : up_out := 0
    Comment
    leave_PRIORITIZED : PRIORITIZED := FALSE
    leave_Neutral : Neutral := FALSE
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    leave_Up_Entry : Up_Entry := FALSE
    leave_Up_Auto : Up_Auto := FALSE
    leave_Up_Stepwise : Up_Stepwise := FALSE
    enter_Dn_Entry : Dn_Entry := TRUE
    act1 : timer_auto_dn := 0
end
Event DownToUp  $\hat{=}$ 
extends Normal_ToUp
when
    grd3 : sel_comm = 1
    grd1 : endstop = FALSE
    grd2 : obs = FALSE
    isin_PRIORITIZED : PRIORITIZED = TRUE
    isin_NORMALMODE : NORMALMODE = TRUE
    isin_Going_Down : Going_Down = TRUE
then
    enter_ENVIRONMENT : ENVIRONMENT := TRUE
    enter_Going_Up : Going_Up := TRUE
    leave_PRIORITIZED : PRIORITIZED := FALSE
    down_out_set : down_out := 0
    Comment
    up_out_set : up_out := 1
    Comment
    leave_CONTROLLING : CONTROLLING := FALSE
    leave_Neutral : Neutral := FALSE
    leave_Going_Down : Going_Down := FALSE
    leave_Dn_Entry : Dn_Entry := FALSE
    leave_Dn_Stepwise : Dn_Stepwise := FALSE
    leave_Dn_Auto : Dn_Auto := FALSE
    enter_Up_Entry : Up_Entry := TRUE
    act1 : timer_auto_up := 0
end
END

```