```
Machine Envir1Impl

  TaskType: periodic(100)
  Priority: 5

  taskbody is
    e1: IF Envir1Impl.ENPressIncrease_Target_Temperature  END IF
        ELSE IF Envir1Impl.ENReset_Increase_Flag END IF;
    e2: IF Envir1Impl.ENPressDecrease_Target_Temperature  END IF
        ELSE IF Envir1Impl.ENReset_decrease_Flag END IF;
    e3: Envir1Impl.ENAlter_Heater_Status;
    e4: Envir1Impl.ENAlter_Temperature_Sensor1;
    Output("current temperature: ", ctd)

VARIABLES
  inc_flag
  dec_flag
  ts1
  ts2
  hss
  anha
  ttd
  hsa
  ctd
  aota

INVARIANTS
  aota ∈ BOOL
  ttd ∈ ℤ
  ts2 ∈ ℤ
  hss ∈ BOOL
  ts1 ∈ ℤ
  inc_flag ∈ BOOL
  dec_flag ∈ BOOL
  hsa ∈ BOOL
  anha ∈ BOOL
  ctd ∈ ℤ

EVENTS
INITIALISATION ≙
 BEGIN
  THEN
   hss ≔ FALSE
   ts1 ≔ 0
   ts2 ≔ 0
   inc_flag ≔  FALSE
   dec_flag ≔  FALSE
   hsa ≔ FALSE
   anha ≔ FALSE
```

```
  aota ≔ FALSE
  ctd ≔ 20
  ttd ≔ 25
 END

ENPressIncrease_Target_Temperature ≙
 Branch
 BEGIN
  WHEN
   inc_flag = FALSE
  THEN
   inc_flag ≔ TRUE
 END

ENSense_PressIncrease_Target_Temperature ≙
 Sensing
 BEGIN ANY
   formalOut state_inc
  WHEN
   state_inc ∈ BOOL
   state_inc = inc_flag
  THEN
    skip
 END

ENReset_Increase_Flag ≙
 Branch
 BEGIN
  WHEN
   inc_flag = TRUE
  THEN
   inc_flag ≔ FALSE
 END

ENPressDecrease_Target_Temperature ≙
 Branch
 BEGIN
  WHEN
   dec_flag = FALSE
  THEN
   dec_flag ≔ TRUE
 END

ENSense_PressDecrease_Target_Temperature ≙
 Sensing
 BEGIN ANY
   formalOut state_dec
  WHEN
   state_dec ∈ BOOL
   state_dec = dec_flag
  THEN
```

```
   skip
 END

ENReset_decrease_Flag ≙
 Branch
 BEGIN
  WHEN
   dec_flag = TRUE
  THEN
   dec_flag ≔ FALSE
 END

ENDisplay_Target_Temperature ≙
 Actuating
 BEGIN ANY
   formalIn tm_tt
  WHEN
   tm_tt ∈ ℤ
  THEN
   ttd ≔tm_tt
 END

ENAlter_Temperature_Sensor1 ≙
 ProcedureDef
 BEGIN
  THEN
   ts1 ≔ ts1 + 1
 END

ENAlter_Temperature_Sensor2 ≙
 ProcedureDef
 BEGIN
  THEN
   ts2 ≔ ts2 + 1
 END

ENSense_Temperatures ≙
 Sensing
 BEGIN ANY
   formalOut t1
   formalOut t2
  WHEN
   t1 ∈ ℤ
   t2 ∈ ℤ
   t1 = ts1
   t2 = ts2
  THEN
    skip
 END

ENDisplay_Current_Temperature ≙
```

```
Actuating
BEGIN ANY
  formalIn tm_avt
 WHEN
  tm_avt ∈ ℤ
 THEN
  ctd ≔ tm_avt
END
```

ENActuate_Heat_Source ≜

```
Actuating
BEGIN ANY
  formalIn state_hsc
 WHEN
  state_hsc ∈ BOOL
 THEN
  hsa ≔ state_hsc
END
```

ENActuate_OverHeat_Alram ≜

```
Actuating
BEGIN ANY
  formalIn state_ota
 WHEN
  state_ota ∈ BOOL
 THEN
  aota ≔ state_ota
END
```

ENSense_Heater_Status ≜

```
Sensing
BEGIN ANY
  formalOut state_hss
 WHEN
  state_hss ∈ BOOL
  state_hss = hss
 THEN
   skip
END
```

ENActuate_NoHeat_Alarm ≜

```
Actuating
BEGIN ANY
  formalIn state_nha
 WHEN
  state_nha ∈ BOOL
 THEN
  anha ≔ state_nha
END
```

ENAlter_Heater_Status ≜

```
ProcedureDef
BEGIN
 THEN
  hss := FALSE
END
```