|  |
| --- |
| **HeatSource** |
| real heaterOut = -0.1 {-0.1, +0.1} |
| if (heatControl == On)  heaterOut = +0.1  else  heaterOut = -0.1 |

**SRC Diagram:** << 1 round = 6 seconds; {On, Off} = {True, False} >>

|  |
| --- |
| **Infant** |
| int infantTemp = 98 {97, ... 101} |
| if(round)?  out = infantTemp |

real heaterOut

int out

bool heatControl

event round

|  |
| --- |
| **Thermostat** |
| enum regulatorStatus = init {Init, On, failed},  enum monitorStatus = init {Init, On, failed},  enum thermostatStatus = init {Init, On, failed},  bool alarmControl = Off {On, Off}  bool heatControl = Off {On, Off}  bool internalFailure = False {True, False} |
| if(round)? {  << ThermostatStatus>>  if (lowerDesiredTempStatus or upperDesiredTempStatus == Invalid)  regulatorStatus = failed  if (lowerAlarmTempStatus or upperAlarmTempStatus == Invalid)  monitorStatus = failed  if (regulatorStatus or monitorStatus = failed or sensorStatus = invalid or internalFailure =true)  thermostatStatus = failed  <<Regulator>>  if (regulatorStatus == On)  if (currentTemp < lowerDesiredTemp)  heatControl = On  if (currentTemp > upperDesiredTemp)  heatControl = Off  <<Monitor>>  if (monitorStatus == On)  if (currentTemp < lowerAlarmTemp or currentTemp > upperAlarmTemp)  alarmControl = On  else  alarmControl = Off} |

|  |
| --- |
| **Air** |
| real airTemp = 97 {68, … 105} |
| airTemp = (airTemp + heaterOut + infantTemp)/2 |

real airTemp

|  |
| --- |
| **TemperatureSensor** |
| real currentTemp = 97{68.0, … 105.0}  enum sensorStatus = vaild {vaild, invalid} |
| if (sensorStatus == valid)  currentTemp = int(airTemp) |

int currentTemp

enum sensorStatus

event round

|  |
| --- |
| **OperatorInterface** |
| bool alarmStatus = Off {On, Off}  enum lowerDesiredTempStatus = invalid {Valid, Invalid}  enum upperDesiredTempStatus = invalid {Valid, Invalid}  enum lowerAlarmTempStatus = invalid {Valid, Invalid}  enum upperAlarmTempStatus = invalid {Valid, Invalid}  int displaytemp = 97 {68, … 105} |
| << ValidateInput >>  If (isoletteControl == On & round?}  {  if (lowerAlarmTemp < 93 or >98 or (lowerDesiredTemp - lowerAlarmTemp) > 1)  lowerAlarmTempStatus = Invalid  if (lowerDesiredTemp < 97 or >99 or (upperDesiredTemp - lowerDesiredTemp) > 1 or (lowerDesiredTemp - lowerAlarmTemp) > 1)  lowerDesiredTempStatus = Invalid  if (upperDesiredTemp < 98 or >100 or (upperDesiredTemp - lowerDesiredTemp) > 1 or (upperAlarmTemp - upperDesiredTemp) > 1)  upperDesiredTempStatus = Invalid  if (upperAlarmTemp < 99 or >103 or (upperAlarmTemp - upperDesiredTemp) > 1)  upperAlarmTempStatus = Invalid  displayTemp = currenttemp  out1 = {lowerDesiredTempStatus, upperDesiredTempStatus, lowerAlarmTempStatus, upperAlarmTempStatus}  alarmStatus = alamControl  } |

|  |
| --- |
| Thermostat |
|  |

enum out1[ ]

bool alarmControl

enum thermostatStatus

int currentTemp

event round

|  |
| --- |
| **Nurse** |
| int lowerDesiredTemp = 97 {97, … 99},  int upperDesiredTemp = 98 {98, … 100}  int lowerAlarmTemp = 96 {93, … 98}  int upperAlarmTemp = 99 {99, … 103}  bool isoletteControl = Off {On, Off} |
| << set/ change values according to the operator interface feedback>>  choose: lowerDesiredTemp = {97, … 99}  choose: upperDesiredTemp = {98, … 100}  choose: lowerAlarmTemp = {93, … 98}  choose: upperAlarmTemp = {99, … 103}  choose: isoletteControl = {On, Off}  inTemp = {lowerDesiredTemp, upperDesiredTemp, lowerAlarmTemp, upperAlarmTemp} |

enum thermostatStatus

int displayTemp

bool alarmStatus

int inTemp [ ]

bool isoletteControl

**State Diagram: ISOLETTE**

((Regulator.status = invalid)? || (Monitor.status = invalid)? || (internalFailure = true)?)

int temperatures [ ]

**FAILED**

else

**ON**

**INIT**

((Regulator.status = invalid)? || (Monitor.status = invalid)? || (internalFailure = true)?)