

---

# Lab4 Answer Sheet

## Report on Configuration VxWorks

### Overridden Properties

#### Subjects:

CG

#### Metaclasses:

CGGeneral

#### Properties:

GeneratedCodeInBrowser: False

# PACKAGES

## Default

### EVENTS:

---

#### evAlarm

*An Alarm has occurred somewhere in the system.  
The alarm type is in its parameter.*

Args:

OMString alarmType

*Pointer to the text representing the type of alarm*

---

#### evAlarmover

*This event is generated when the operator acknowledges an alarm via a 'z' at the keyboard.*

---

#### evAlarmReset

*A sensor has gone out of alarm somewhere in the system. The alarm type is in its parameter.*

Args:

OMString alarmType

*Pointer to the text representing the type of alarm.*

### GLOBALS:

Relations:

**itsMyDisplay**

Composition of myDisplay, Multiplicity of 1, Uni-directional

**itsDoor**

Composition of door, Multiplicity of 1, Uni-directional

**itsWindow**

Composition of window, Multiplicity of 1, Uni-directional

**itsTemperature**

Composition of temperature, Multiplicity of 1, Uni-directional

**itsMyPIC**

Composition of myPIC, Multiplicity of 1, Uni-directional

**itsMyKeyboard**

Composition of myKeyboard, Multiplicity of 1, Uni-directional

Instantiated Relations:

**itsTemperature**

of itsMyDisplay with itsTemperature

**itsMyDisplay**

of itsTemperature with itsMyDisplay

**itsWindow**

of itsMyDisplay with itsWindow

**itsMyDisplay**

of itsWindow with itsMyDisplay

**itsDoor**

of itsMyDisplay with itsDoor

**itsMyDisplay**

of itsDoor with itsMyDisplay

**itsMyKeyboard**

of itsMyDisplay with itsMyKeyboard

Functions:

**CompleteRelations**

*This completes all necessary relations that are not automatically set by the system.*

Primitive-operation , Public, Return type is void

## Body

```
itsWindow->setItsMyPIC(itsMyPIC);  
itsDoor->setItsMyPIC(itsMyPIC);  
itsTemperature->setItsMyPIC(itsMyPIC);
```

# CLASSES:

---

## door

*This is the door sensor class, monitoring button 6.*

### Relations:

#### **itsMyDisplay**

Association with myDisplay, Multiplicity of 1, Bi-directional

### Superclasses:

sensor

Public

### Operations:

#### **alarmReset**

*This operation compares the value read from the input device using readValue and returns a boolean if it is in range.*

Virtual, Primitive-operation, Public, Return type is OMBoolean

### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is =< this value the it returns true.*

### Body

```
return itsMyPIC->keyJustReleased(6) != desired;
```

#### **alarmValue**

*This operation compares the value read from the input device using readValue and returns a boolean if it is out of range.*

Primitive-operation, Public, Return type is OMBoolean

### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is not =< this value the it returns true.*

### Body

```
return itsMyPIC->keyJustPressed(6) != desired;
```

## door

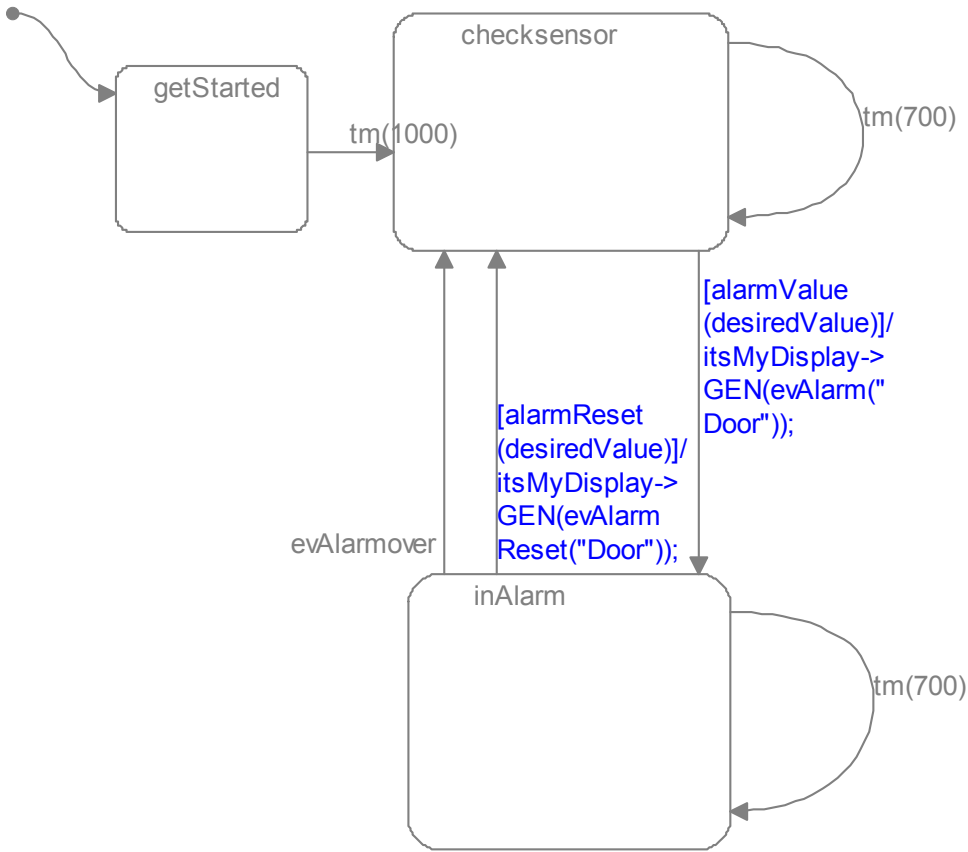
*initializes the desired value of the door sensor.*

Constructor, Public

### Body

```
desiredValue = ((int) false);
```

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.20.2003::18:11:31

**ROOT**

Or-state

Inherited

Substates:

checksensor

getStarted

inAlarm

Default Transition

Inherited

Target:

getStarted

**checksensor**

Or-state

Inherited

Out Transition

Inherited

Modified

[alarmValue (desiredValue)]/ itsMyDisplay->GEN(evAlarm("Door"));

Target:

inAlarm

Out Transition

Inherited

tm(700)

Target:  
checksensor  
**getStarted**  
Or-state  
Inherited  
Out Transition  
Inherited  
tm(1000)

Target:  
checksensor  
**inAlarm**  
Or-state  
Inherited  
Out Transition  
Inherited  
evAlarmover

Target:  
checksensor  
Out Transition  
Inherited  
tm(700)

Target:  
inAlarm  
Out Transition  
Inherited  
Modified  
[alarmReset (desiredValue)]/ itsMyDisplay->GEN(evAlarmReset("Door"));

Target:  
checksensor

---

## myDisplay

*This is the display class that outputs all communication to the terminal.*

### Overridden Properties

#### Subjects:

CG

#### Metaclasses:

Class

#### Properties:

Concurrency: active

### Relations:

#### **itsTemperature**

Association with temperature, Multiplicity of 1, Bi-directional

#### **itsWindow**

Association with window, Multiplicity of 1, Bi-directional

#### **itsDoor**

Association with door, Multiplicity of 1, Bi-directional

#### **itsMyKeyboard**

Association with myKeyboard, Multiplicity of 1, Uni-directional

### Operations:

#### **evAlarm**

*An Alarm has occurred somewhere in the system.*

*The alarm type is in its parameter.*

Event

#### **evAlarmover**

*This event is generated when the operator acknowledges an alarm via a 'z' at the keyboard.*

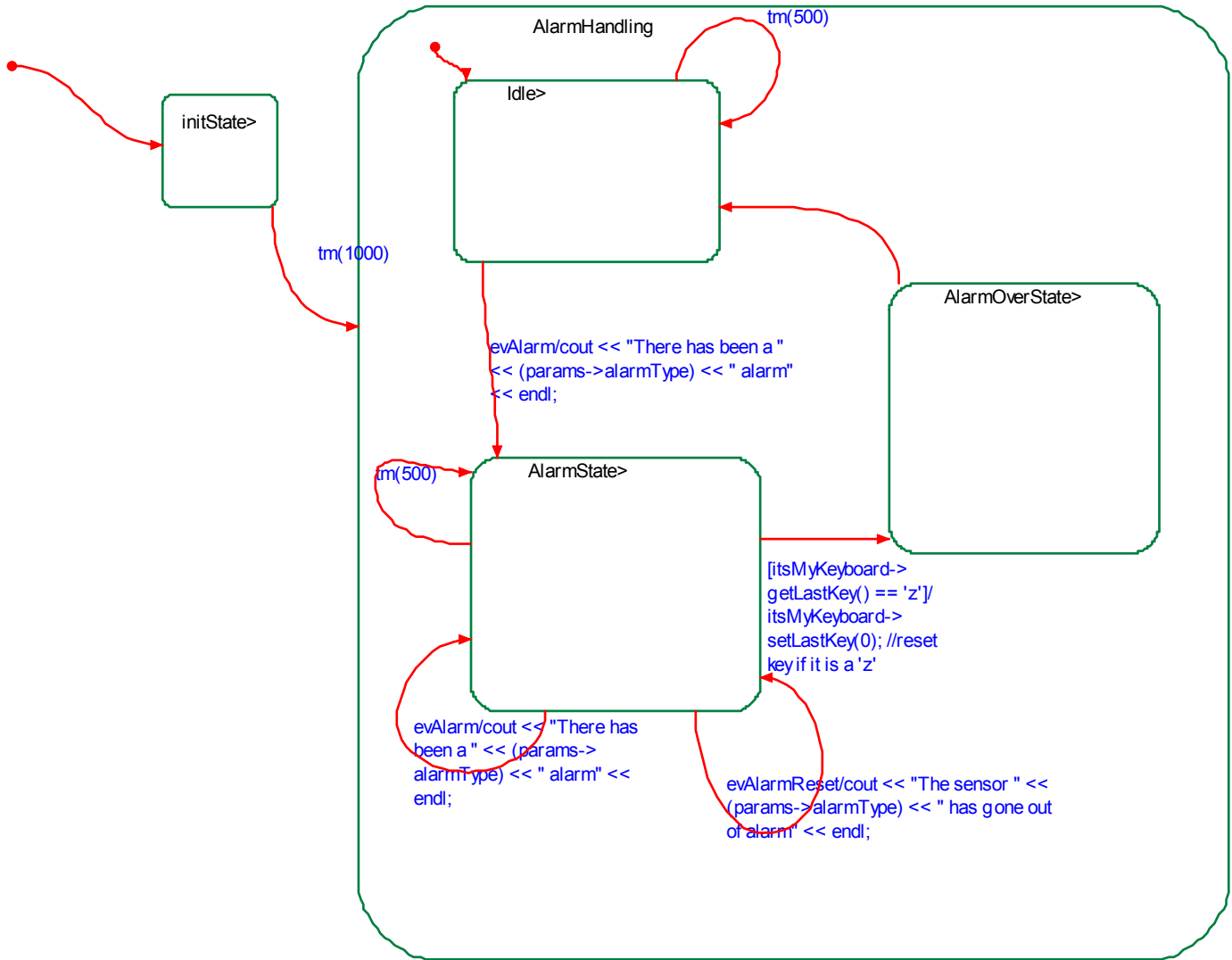
Event

#### **evAlarmReset**

*A sensor has gone out of alarm somewhere in the system. The alarm type is in its parameter.*

Event

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.20.2003::17:58:45

**ROOT**

Or-state

Substates:

AlarmHandling

initState

Default Transition

Target:

initState

**AlarmHandling**

*Primary alarm monitoring state.*

Or-state

Substates:

AlarmOverState

AlarmState

Idle

Default Transition

Target:  
Idle

### AlarmOverState

*The alarm has been reset by the operator.*

Or-state

#### EntryAction

```
itsDoor->GEN(evAlarmover);  
itsWindow->GEN(evAlarmover);  
itsTemperature->GEN(evAlarmover);
```

```
cout << "Alarm reset " << endl;
```

Out Transition

Target:  
Idle

### AlarmState

*We are in alarm! we stay here until the alarm is reset by the operator typing a 'z' at the keyboard.*

Or-state

#### EntryAction

```
cout << "alarm Alarm ALARM!!! " << endl;
```

Out Transition

```
[itsMyKeyboard->getLastKey() == 'z']/ itsMyKeyboard->setLastKey(0); //reset key if it is a  
'z'
```

#### Target:

AlarmOverState

Out Transition

```
tm(500)
```

#### Target:

AlarmState

Out Transition

```
evAlarm/cout << "There has been a " << (params->alarmType) << " alarm" << endl;
```

#### Target:

AlarmState

Out Transition

```
evAlarmReset/cout << "The sensor " << (params->alarmType) << " has gone out of alarm" <<  
endl;
```

#### Target:

AlarmState

### Idle

*Sits here idly waiting for something to happen.*

Or-state

#### EntryAction

```
cout << "Still idle " << endl;
```

Out Transition

```
evAlarm/cout << "There has been a " << (params->alarmType) << " alarm" << endl;
```

#### Target:

AlarmState

Out Transition

```
tm(500)
```

#### Target:

Idle

### initState

*This state allows the system to perform any required initializations.*

Or-state

#### EntryAction

```
CompleteRelations(); //complete the relations for the sensor classes.
```

Out Transition

```
tm(1000)
```

#### Target:

AlarmHandling

# myKeyboard

*Keyboard handler, reading the last key pressed on the keyboard.*

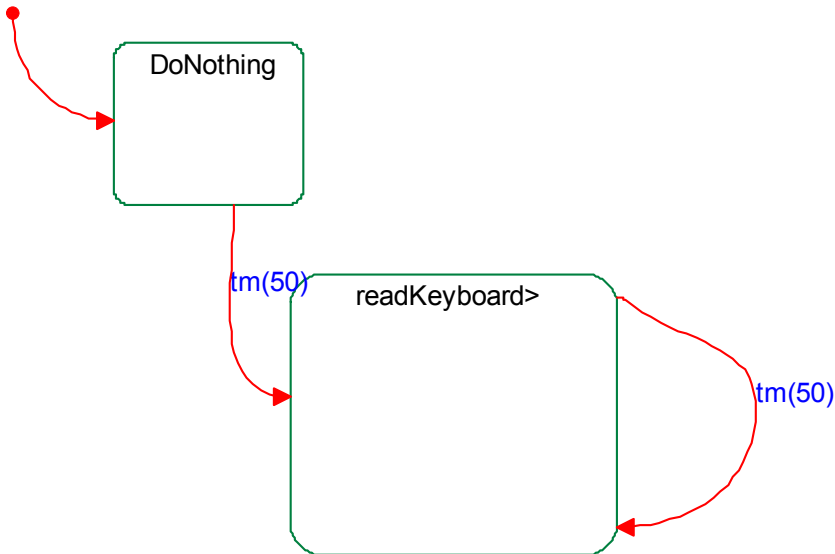
## Attributes:

### **lastKey**

*This is the last key read in from the keyboard.*

Type of char, Public, Initial Value: 0

## Statechart



## Overridden Properties

### Subjects:

#### **General**

#### Metaclasses:

#### **Diagram**

#### Properties:

LastModifiedTime: 10.18.2003::23:11:53

## ROOT

Or-state

### Substates:

DoNothing

readKeyboard

Default Transition

### Target:

DoNothing

## DoNothing

*allows for starting all of the state machines prior to executing a CIN;*

Or-state

Out Transition

tm (50)

### Target:

readKeyboard

## readKeyboard

*reads in the keys from the keyboard.*

Or-state

### EntryAction

cin>>lastKey;

Out Transition

tm (50)

### Target:

readKeyboard

## myPIC

*A subclass of the PICserial class containing additional functionality.*

### Superclasses:

PICserial

Public

### Operations:

#### keyJustPressed

*Returns true only one time indicating that the silkkey has just been pressed.*

Primitive-operation , Public, Return type is OMBoolean

#### Args:

int silkkey

*This is the value of the silkscreened key to interegate.*

#### Body

```
return ((getAllKeyData(keymap[silkkey]) & (long) (keymap[silkkey] | (keymap[silkkey] << 16))) ==  
((long) (keymap[silkkey] | (keymap[silkkey] << 16))));
```

#### keyJustReleased

*Returns true only one time indicating that the silkkey has just been released.*

Primitive-operation , Public, Return type is OMBoolean

#### Args:

int silkkey

*This is the value of the silkscreened key to interegate.*

#### Body

```
return ((getAllKeyData(keymap[silkkey]) & (long) (keymap[silkkey] | (keymap[silkkey] << 16))) ==  
(long) (keymap[silkkey] << 16));
```

#### keyPressed

*This function returns true as long as the silkkey is pressed.*

*It does not reset the COS bits.*

Primitive-operation , Public, Return type is OMBoolean

#### Args:

int silkkey

*This is the value of the silkscreened key to interegate.*

#### Body

```
return ((getAllKeyData(0) & keymap[silkkey]) != 0);
```

## myPIC

Constructor , Public

#### Body

```
keymap[0] = 0x0002;  
keymap[1] = 0x4000;  
keymap[2] = 0x2000;  
keymap[3] = 0x1000;  
keymap[4] = 0x0400;  
keymap[5] = 0x0200;  
keymap[6] = 0x0100;  
keymap[7] = 0x0040;  
keymap[8] = 0x0020;  
keymap[9] = 0x0010;  
keymap[10] = 0x0800; //enter  
keymap[11] = 0x0004; // .  
keymap[12] = 0x0001; //CHS
```

#### scaledAnalog

*Returns a scaled analog value fromt the pot reading.*

Primitive-operation , Public, Return type is int

#### Args:

int hi = 255

*Value corresponding to a 255 count reading from the sensor.*

int lo = 0

*Value corresponding to a 0 count reading from the sensor.*

int n = 0

*Indicates which analog pot to read. (0-2)*

#### Body

```
return (int) ((float) getAnalogValues(n) * ((float) (hi-lo) / (float) 255) + (float) lo);
```

Attributes:

**keymap**

*Maps the logical silkscreened key numbers into bit positions.*

**Overridden Properties**

Subjects:

CPP\_CG

Metaclasses:

**Attribute**

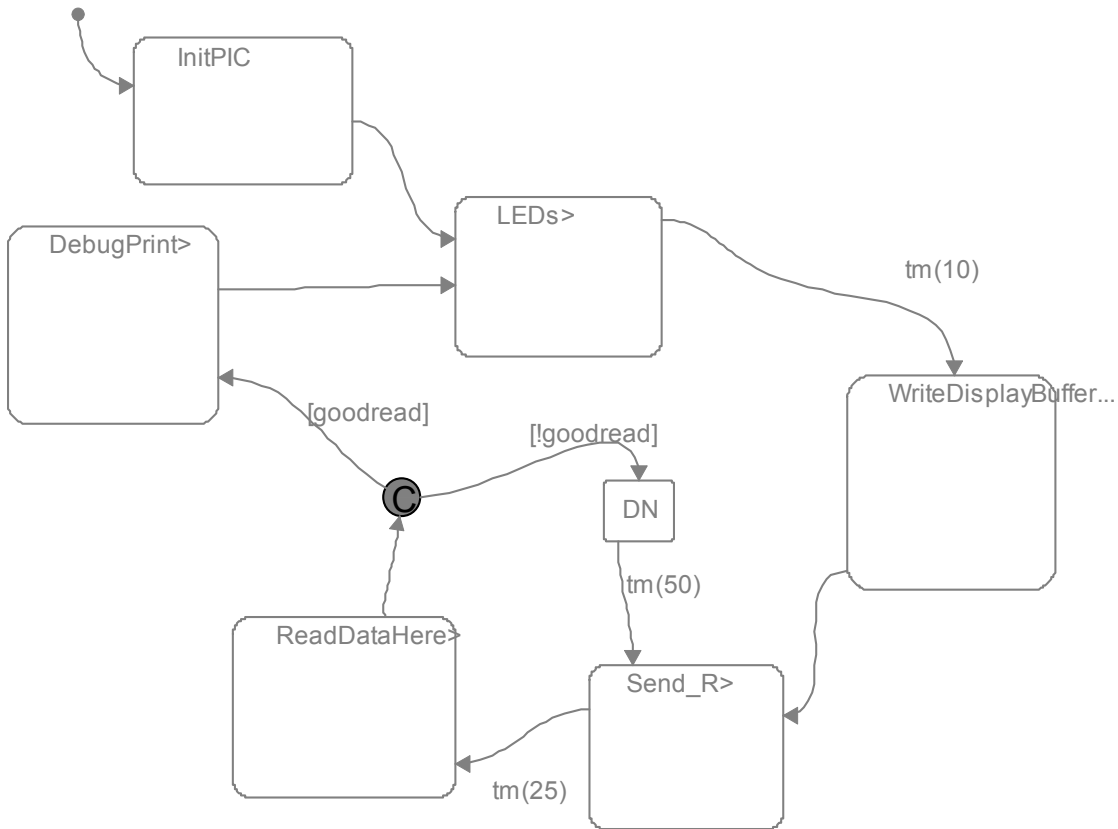
Properties:

MutatorGenerate: False

AccessorGenerate: True

Type of 'unsigned int %s[13];', Public

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.18.2003::19:57:57

**ROOT**

Or-state

Inherited

Substates:

DebugPrint

DN

InitPIC

LEDs

ReadDataHere

Send\_R

WriteDisplayBuffer

Default Transition

Inherited

Target:  
InitPIC

**DebugPrint**

Or-state

Inherited

EntryAction

```
//cout << " keys " << setbase(16) << setfill('0') << setw(8) << getAllKeyData() ;  
//cout << " a0 " << (int)getAnalogValues(0);  
//cout << " a1 " << (int)getAnalogValues(1);  
//cout << " a2 " << (int)getAnalogValues(2);  
//cout << endl;  
  
//cout << setbase(16) << setfill('0') << setw(8) << getAllKeyData() << " " <<  
(int)getAnalogValues(0) << " " << (int)getAnalogValues(1) << " " << (int)getAnalogValues(2)  
<< endl;  
  
//for (int i=0;i<5;i++)  
//{ cout << " " << setbase(16) << (int) readBuffer[i]; }  
//cout << endl;
```

Out Transition

Inherited

Target:  
LEDs

**DN**

Or-state

Inherited

Out Transition

Inherited

tm(50)

Target:  
Send\_R

**InitPIC**

Or-state

Inherited

Out Transition

Inherited

Target:  
LEDs

**LEDs**

Or-state

Inherited

EntryAction

writeLeds();

ExitAction

//leds++;

Out Transition

Inherited

tm(10)

Target:  
WriteDisplayBuffer

**ReadDataHere**

Or-state

Inherited

EntryAction

goodread = ReadData();

Out Transition

Inherited

Condition Connector

Branches:

[goodread]

Target:  
DebugPrint  
[!goodread]

Target:  
DN

**Send\_R**

Or-state

Inherited

EntryAction

```
ControlChar[0] = 'R';  
write(fd, ControlChar, 1); //write an 'R' to the PIC board to read the buffers.
```

Out Transition

Inherited

tm(25)

Target:

ReadDataHere

**WriteDisplayBuffer**

Nested Statechart

Or-state

Inherited

EntryAction

```
//for(int i=0;i<32;i++) Display[i]=leds;
```

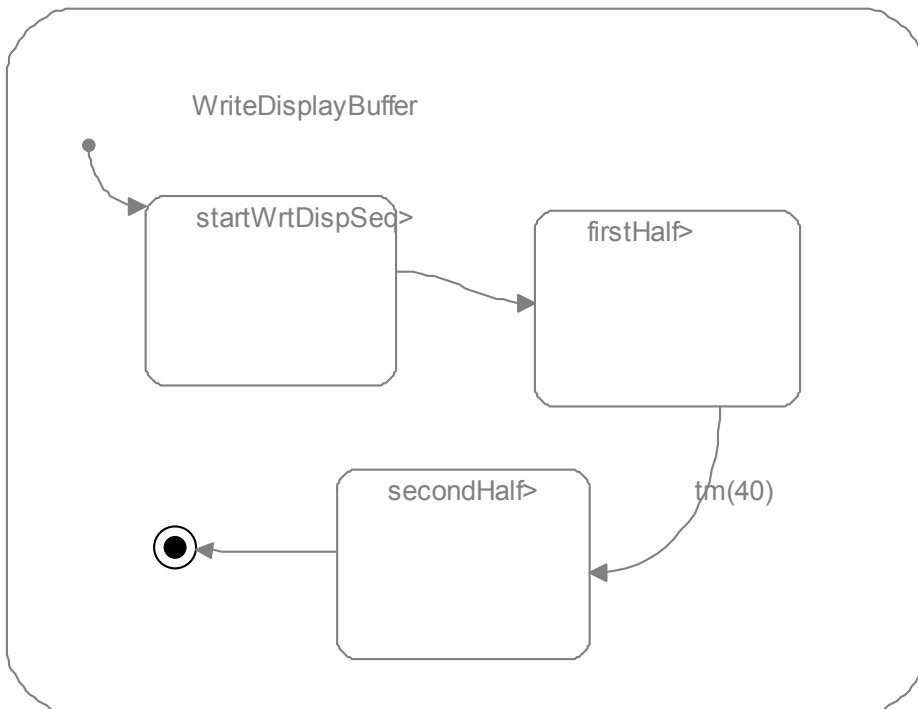
Out Transition

Inherited

Target:

Send\_R

Statechart of WriteDisplayBuffer



**Overridden Properties**

Subjects:

General

Metaclasses:

Diagram

Properties:

LastModifiedTime: 10.18.2003::19:57:57

**ROOT**

Or-state

Inherited

Substates:

WriteDisplayBuffer

**WriteDisplayBuffer**

Or-state

Inherited

Substates:

firstHalf

secondHalf

startWrtDispSeq

state\_0

Default Transition

Inherited

Target:

startWrtDispSeq

**firstHalf**

Or-state

Inherited

EntryAction

```
write(fd, Display, 16); //write out 16 characters.
```

Out Transition

Inherited

tm(40)

Target:

secondHalf

**secondHalf**

Or-state

Inherited

EntryAction

```
write(fd, &Display[16], 16); //write out 16 characters.
```

Out Transition

Inherited

Target:

state\_0

**startWrtDispSeq**

Or-state

Inherited

EntryAction

```
ControlChar[0] = 'B';  
write(fd, ControlChar, 1); //write an 'B' to the PIC board to read the buffers.
```

Out Transition

Inherited

Target:

firstHalf

**state\_0**

Local Termination State

Inherited

---

## sensor

*This is the sensor superclass used by all sensor subclasses.*

### Overridden Properties

#### Subjects:

CG

#### Metaclasses:

Class

#### Properties:

Concurrency: active

#### Relations:

##### **itsMyPIC**

Association with myPIC, Multiplicity of 1, Uni-directional

#### Subclasses:

door

temperature

window

#### Operations:

##### **alarmReset**

*This operation compares the value read from the input device using readValue and returns a boolean if it is in range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is =< this value the it returns true.*

##### **alarmValue**

*This is a virtual operation that compares the value read from the input device using readValue and returns a boolean if it is out of range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is not =< this value the it returns true.*

##### **evAlarmover**

*This event is generated when the operator acknowledges an alarm via a 'z' at the keyboard.*

Event

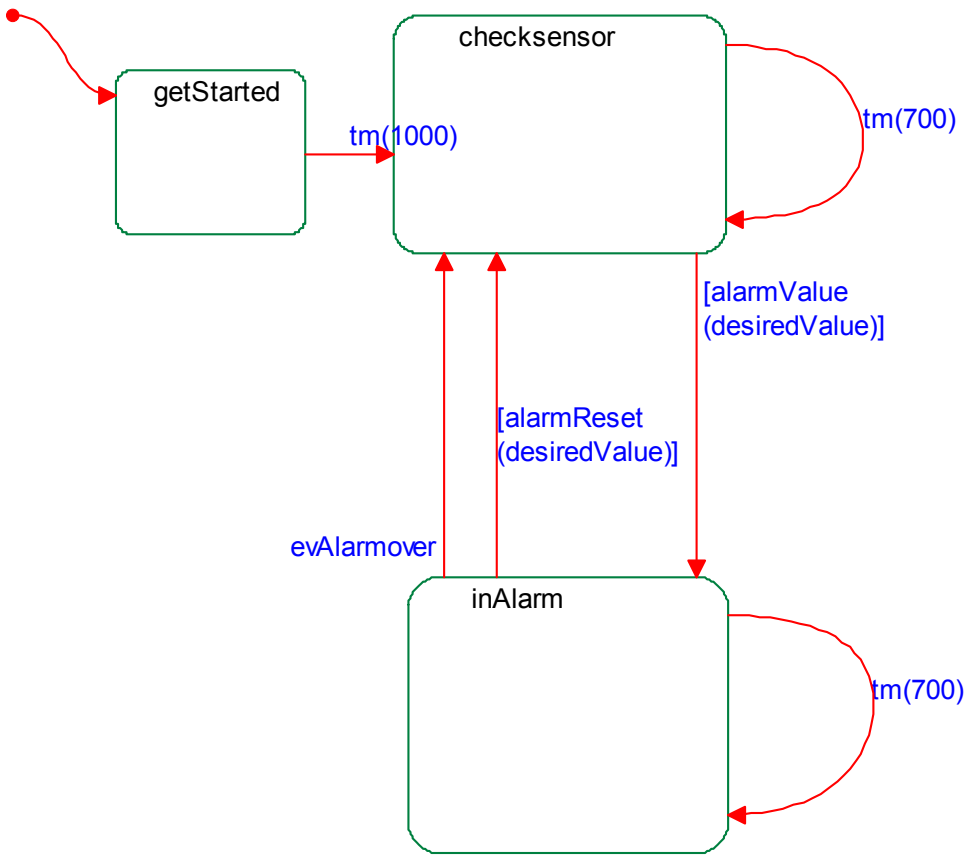
#### Attributes:

##### **desiredValue**

*This is the desired value for the input device.*

Type of int, Public, Initial Value: 0

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.20.2003::18:1:5

**ROOT**

Or-state

Substates:

checksensor

getStarted

inAlarm

Default Transition

Target:

getStarted

**checksensor**

*This checks to see if the desired point is in alarm. If so, it sends an event to the display class.*

Or-state

Out Transition

[alarmValue (desiredValue)]

Target:

inAlarm

Out Transition

tm(700)

Target:

checksensor

**getStarted**

Or-state

Out Transition

tm(1000)

Target:  
    checksensor  
**inAlarm**  
Or-state  
Out Transition  
    evAlarmover

Target:  
    checksensor  
Out Transition  
    tm(700)

Target:  
    inAlarm  
Out Transition  
    [alarmReset (desiredValue)]

Target:  
    checksensor

---

## temperature

*This is the temperature sensor class, monitoring analog pot 1 and alarming at 100 degrees. it has a hi of 120 and a lo of 32.*

### Relations:

#### **itsMyDisplay**

Association with myDisplay, Multiplicity of 1, Bi-directional

### Superclasses:

sensor

Public

### Operations:

#### **alarmReset**

*This operation compares the value read from the input device using readValue and returns a boolean if it is in range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is =< this value the it returns true.*

#### Body

```
return itsMyPIC->scaledAnalog(120,32,1) < desired;
```

#### **alarmValue**

*This is a virtual operation that compares the value read from the input device using readValue and returns a boolean if it is out of range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is not =< this value the it returns true.*

#### Body

```
return itsMyPIC->scaledAnalog(120,32,1) >= desired;
```

#### **temperature**

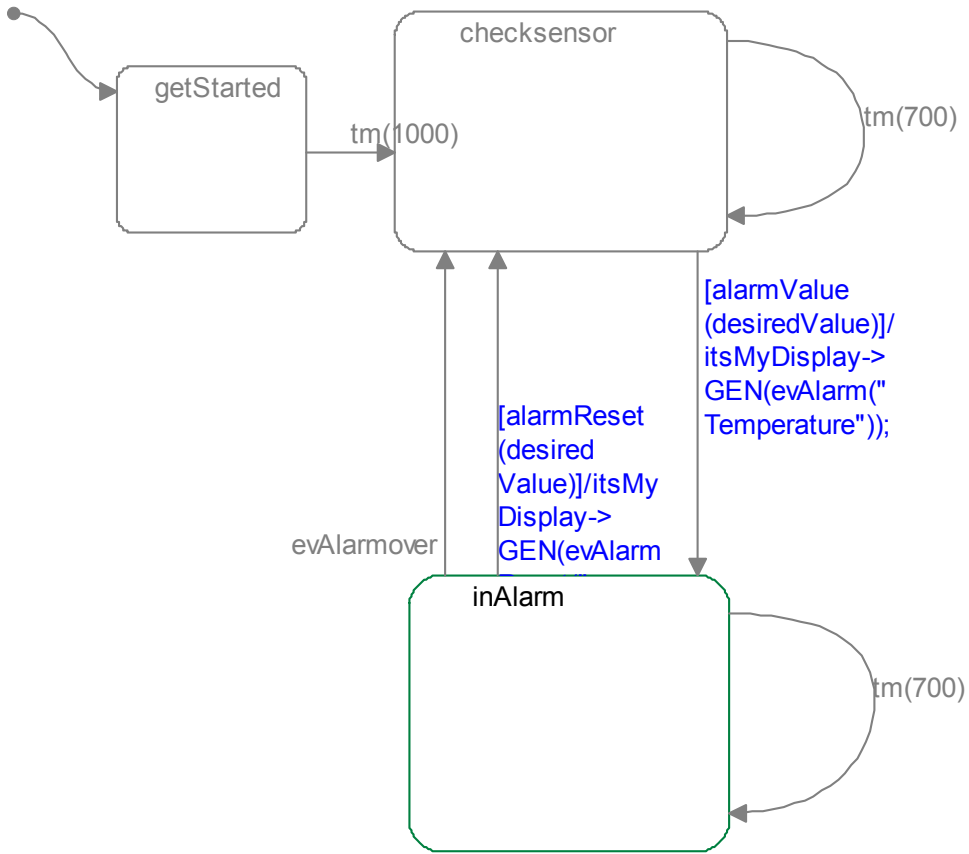
*initializes the desired value of the temperature sensor.*

Constructor , Public

#### Body

```
desiredValue = 100;
```

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.20.2003::18:12:6

**ROOT**

Or-state

Inherited

Substates:

checksensor

getStarted

inAlarm

Default Transition

Inherited

Target:

getStarted

**checksensor**

Or-state

Inherited

Out Transition

Inherited

Modified

[alarmValue (desiredValue)]/ itsMyDisplay->GEN(evAlarm("Temperature"));

Target:

inAlarm

Out Transition

Inherited

tm(700)

Target:  
checksensor

**getStarted**

Or-state  
Inherited  
Out Transition  
Inherited  
tm(1000)

Target:  
checksensor

**inAlarm**

Or-state  
Inherited  
Out Transition  
Inherited  
evAlarmover

Target:  
checksensor

Out Transition  
Inherited  
tm(700)

Target:  
inAlarm

Out Transition  
Inherited  
Modified

[alarmReset (desiredValue)]/itsMyDisplay->GEN(evAlarmReset("Temperature"));

Target:  
checksensor

---

## window

*This is the window sensor class, monitoring button 5.*

### Relations:

#### **itsMyDisplay**

Association with myDisplay, Multiplicity of 1, Bi-directional

### Superclasses:

sensor

Public

### Operations:

#### **alarmReset**

*This operation compares the value read from the input device using readValue and returns a boolean if it is in range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is =< this value the it returns true.*

#### Body

```
return itsMyPIC->keyJustReleased(5) != desired;
```

#### **alarmValue**

*This is a virtual operation that compares the value read from the input device using readValue and returns a boolean if it is out of range.*

Virtual, Primitive-operation , Public, Return type is OMBoolean

#### Args:

int desired = 0

*this represents the normal value of the input device. If the input value is not =< this value the it returns true.*

#### Body

```
return itsMyPIC->keyJustPressed(5) != desired;
```

#### **window**

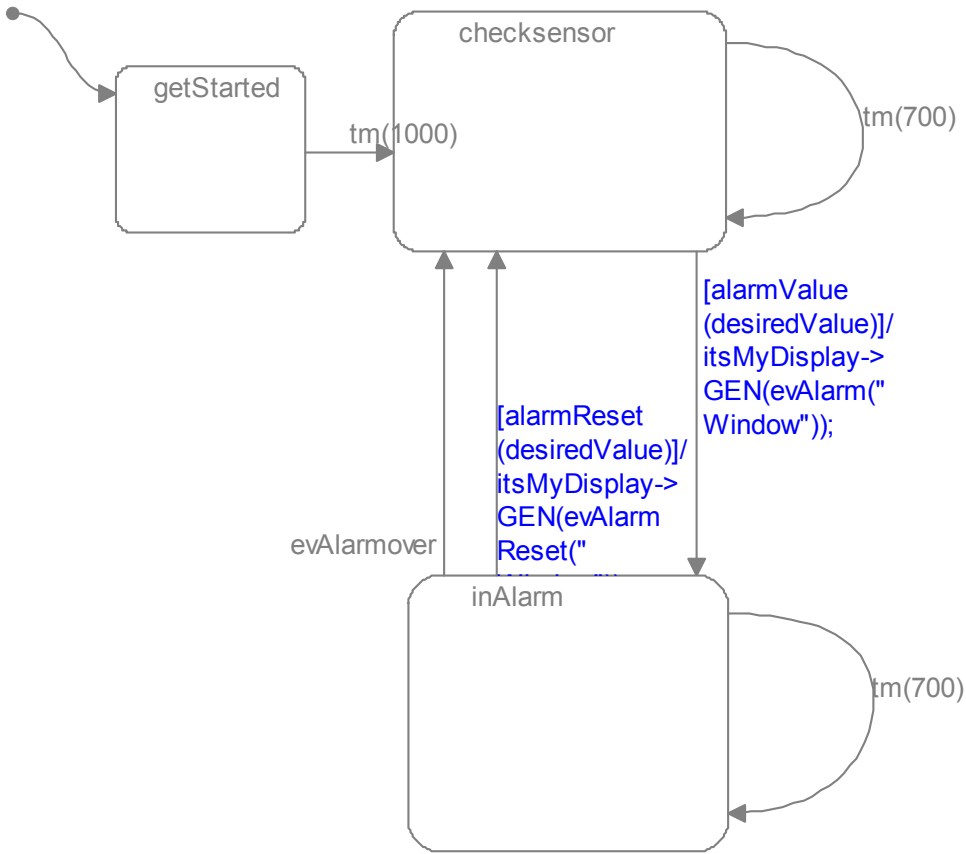
*initializes the desired value of the window sensor.*

Constructor , Public

#### Body

```
desiredValue = ((int>false);
```

Statechart



**Overridden Properties**

Subjects:

**General**

Metaclasses:

**Diagram**

Properties:

LastModifiedTime: 10.20.2003::18:7:19

**ROOT**

Or-state

Inherited

Substates:

checksensor

getStarted

inAlarm

Default Transition

Inherited

Target:

getStarted

**checksensor**

Or-state

Inherited

Out Transition

Inherited

Modified

[alarmValue (desiredValue)]/ itsMyDisplay->GEN(evAlarm("Window"));

Target:

inAlarm

Out Transition

Inherited

tm(700)

Target:  
checksensor  
**getStarted**  
Or-state  
Inherited  
Out Transition  
Inherited  
tm(1000)

Target:  
checksensor  
**inAlarm**  
Or-state  
Inherited  
Out Transition  
Inherited  
evAlarmover

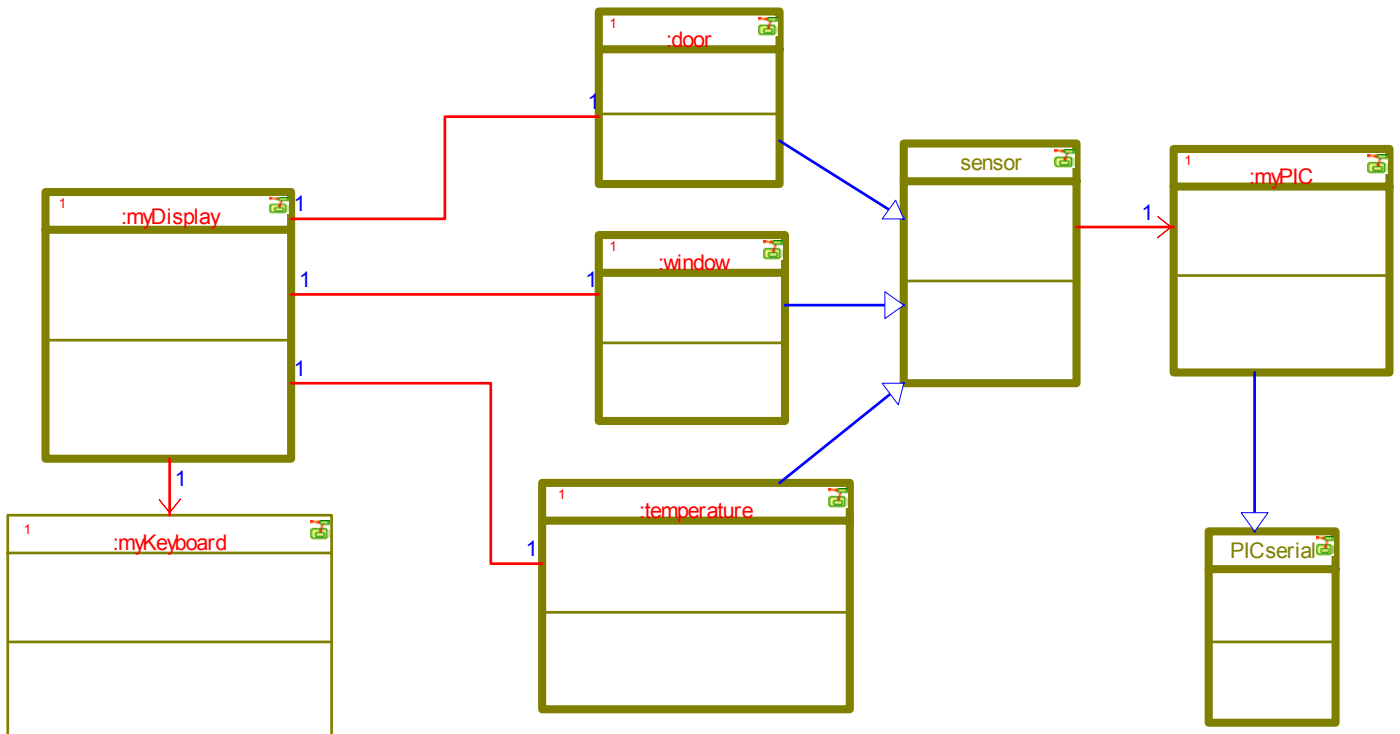
Target:  
checksensor  
Out Transition  
Inherited  
tm(700)

Target:  
inAlarm  
Out Transition  
Inherited  
Modified  
[alarmReset (desiredValue)] / itsMyDisplay->GEN(evAlarmReset("Window"));

Target:  
checksensor

# OBJECT MODEL DIAGRAMS

## Model1



### Overridden Properties

#### Subjects:

#### General

#### Metaclasses:

#### Diagram

#### Properties:

LastModifiedTime: 10.18.2003::23:39:55

# COMPONENTS

## DefaultComponent

### COMPONENT SETTINGS:

Build type: Executable

### CONFIGURATIONS:

---

#### VxWorks

*This configuration creates an executable in the VxWorks environment.*

##### Overridden Properties

##### Subjects:

CPP\_CG

##### Metaclasses:

##### Configuration

##### Properties:

Environment: VxWorks

Scope type: Explicit

Instrumentation type: None

Time-model type: Real-time

Statechart generation type: Flat

Standard headers: iostream.h,ioLib.h,tyLib.h,iomanip.h

Include path: c:\tornado\target\h

---

#### Windows

Scope type: Explicit

Instrumentation type: None

Time-model type: Real-time

Statechart generation type: Flat

### FILES AND FOLDERS: