

---

# Building automation .- Lab assignment - Demo house

---

# Contents

- 1. Introduction ..... 1
- 2. Hardware ..... 4
  - 2.1. Jace..... 4
  - 2.2. Networks..... 5
    - 2.2.1. NDIO ..... 5
    - 2.2.2. Modbus..... 6
    - 2.2.3. LonWorks ..... 7
    - 2.2.4. Z-Wave..... 7
- 3. System functionality..... 9
  - 3.1. Use cases ..... 9
  - 3.2. Control room ..... 10
  - 3.3. Logic ..... 15
- 4. Pre-assignment..... 16
- 5. Exercise ..... 17
  - 5.1. Getting to know the system ..... 18
  - 5.2. Outdoor motion turns on a LED light (0,5 points) ..... 19
    - 5.2.1. Connecting the LED to the Modbus board ..... 19
    - 5.2.2. Adding a new point to modbus ..... 20
    - 5.2.3. Link the point to the logic ..... 21
  - 5.3. Adding logic for the LED ..... 22
  - 5.4. Improve the LED light logic (0,5 points) ..... 22
  - 5.5. Further exercises (1 point) ..... 23

# 1. Introduction

The demo house project is displayed in figure 1 - 1. and figure 1 - 2. The house is equipped with an building automation system from Tridium (Jace 6E) and various networks for communicating with devices, actuators and sensors. The automation system has sensors for measuring temperature, carbon dioxide, humidity and power consumption.

The sensors are communicating over various networks. The demo house supports at this moment four different networks: NDIO (Niagara Direct Input/Output), Modbus, LonWorks and Z-Wave. The NDIO, LonWorks and Modbus networks support simple input/outputs for the system. Z-Wave is a wireless communication protocol and communicates with some on/off switches and temperature and humidity sensor.

The demo house represent a realistic building automation setup, with automatic temperature, humidity and carbon dioxide control. Additional to the controls, it supports an alarm system that will alert in case of intruders.

The software used for configuration and programming logic for the automation system is WorkPlace AX from Tridium. The logic is done with function blocks, and the control panel is created through the WorkPlace AX software.

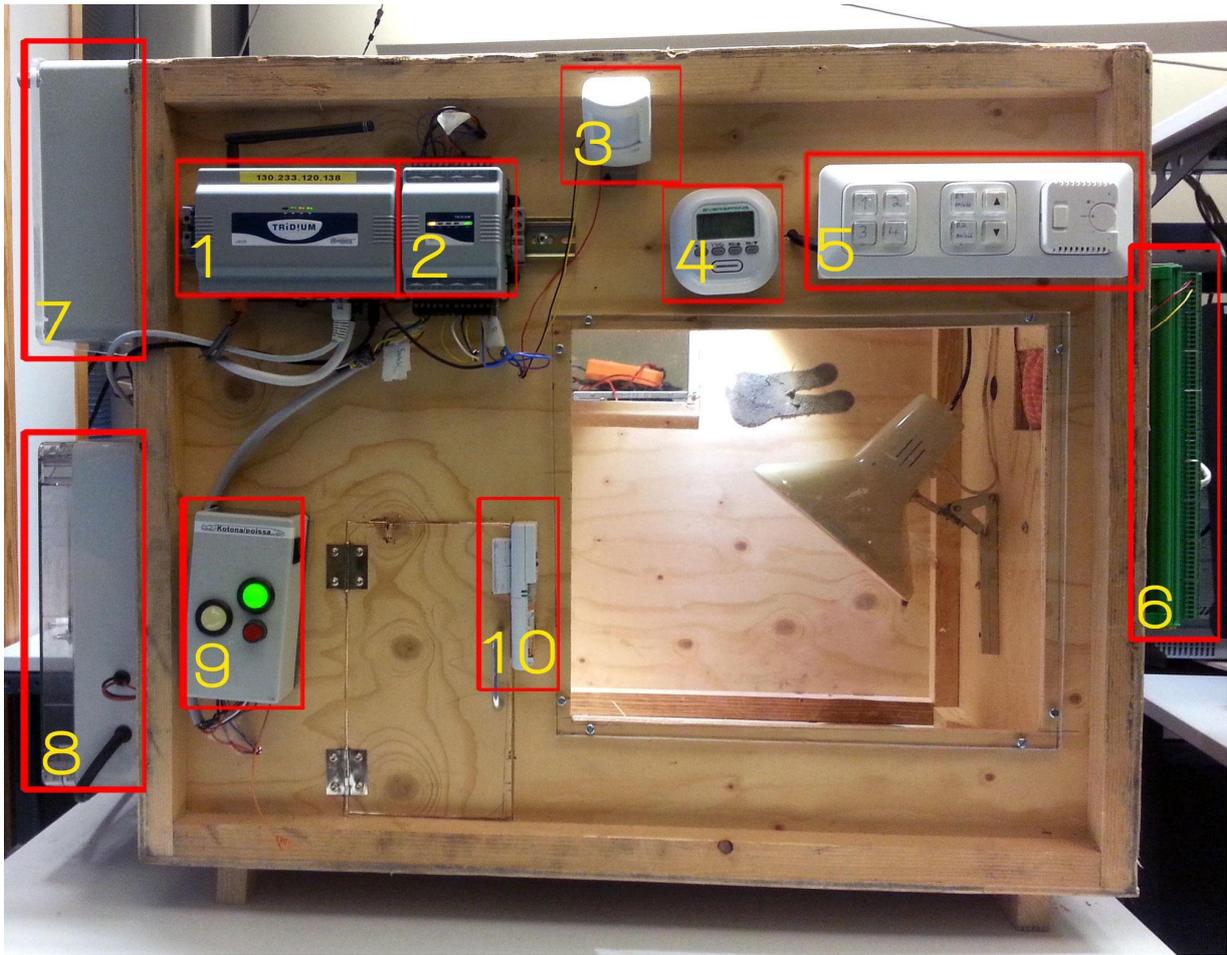


Figure 1 - 1. Front view of the project

1. Jace-6E
2. NDIO-network
3. Outdoor motion sensor (NDIO-network)
4. Temperature and humidity sensor (Z-wave)
5. Control panel (LonWorks)
6. Modbus MIO-52
7. KLon LonWorks module
8. Power supply
9. Home / Away switch (NDIO-network)
10. Magnetic sensor (Z-wave)

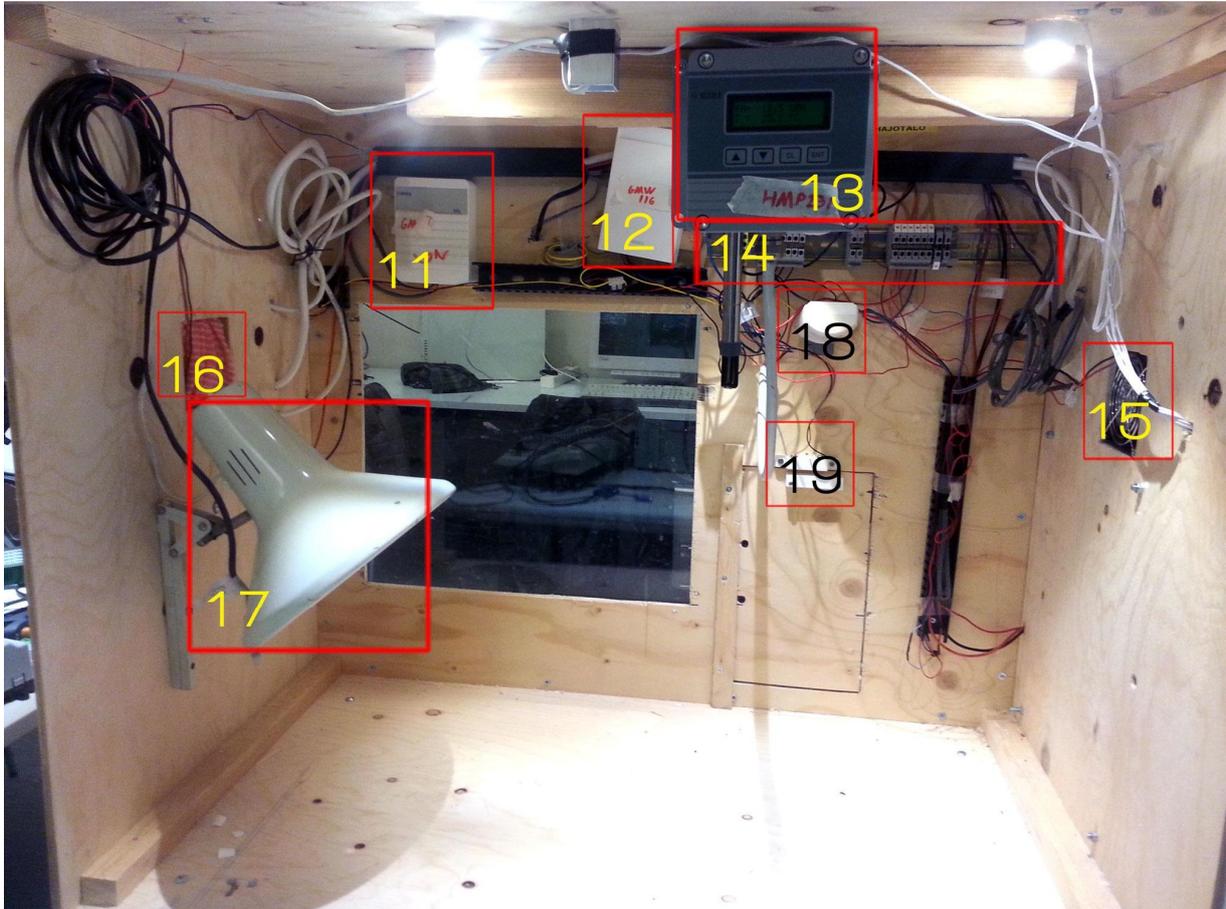


Figure 1 - 2. Back view and inside the project

11. Temperature and carbon dioxide sensor, Vaisala GML20T (LonWorks)
12. Carbon dioxide sensor, Vaisala GMW116 (NDIO-network)
13. Temperature and humidity sensor, Vaisala HMP231 (NDIO-network)
14. Power distributor
15. CO2 fan (NDIO-network)
16. Humidity fan (NDIO-network)
17. Heater lamp (Z-wave)
18. Indoor motion sensor (NDIO-network)
19. Magnetic switch for door sensing (Modbus)

## 2. Hardware

The hardware and various networks for this building automation project is described briefly in the following chapter.

### 2.1. Jace

The control unit used for this project is the Jace 6E (figure 2 - 1.). The Jace 6E features 2 option card slots for adding extra networks, 20 pin I/O connector, RS-485, 2x LAN ports and a serial RS-232 port. The control unit is connected with a power cable that comes straight from the power supply. It runs on 24V DC. The Modbus MIO 52 module is connected to the RS-485 COM port. The NDIO-network is connected to the right of the control unit with the 20 pin I/O connector.

Both option card slots are occupied. One for the Z-Wave network and the other for the LonWorks network.

- Option card slot 1: LonWorks
- Option card slot 2: Z-Wave

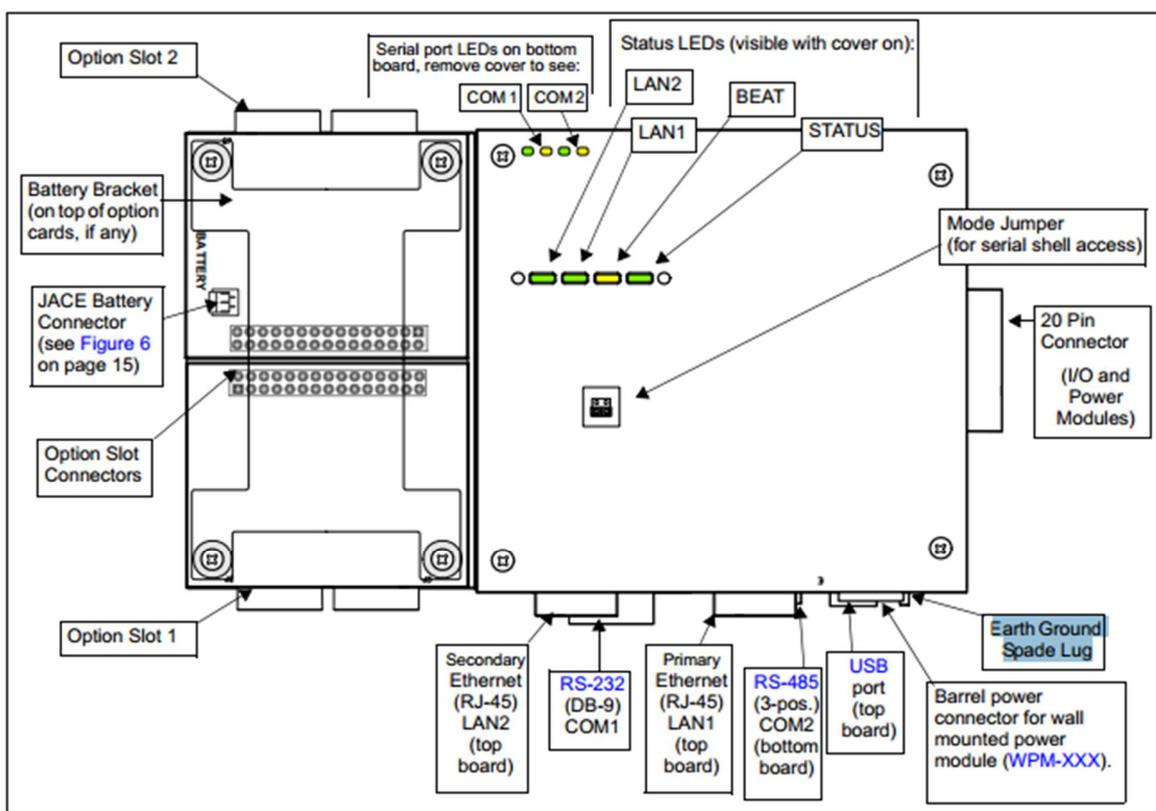


Figure 2.1 - 1. Overview of the control unit and its interfaces

## 2.2. Networks

The current system setup features multiple networks for controlling devices. A short description of the networks is provided in this chapter.

### 2.2.1. NDIO

The project features an IO-16 NDIO module connected to the Jace with the 20 pin I/O connector. The NDIO module is a compact direct I/O module for auxiliary monitoring and control. It gets its power straight from the Jace, and has its own ground connection.



Figure 2.2.1 - 1. NDIO I/O module

Devices connected to the NDIO I/O module are listed below:

#### Inputs:

- Home/away switch
- String switch
- Indoor motion sensor
- Vaisala GMW116 carbon dioxide sensor
- Vaisala HMP231 temperature and humidity sensor

#### Digital Outputs:

- Green LED (Home)
- Yellow LED (Away)
- Humidity fan
- CO2 fan

#### Analog Outputs:

- Led lights

## 2.2.2. Modbus

Modbus is a serial communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). It has become a de facto standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices. Modbus enables communication among many devices connected to the same network. The project features a Modbus MIO-52 module connected via RS-485 to the Jace 6E, figure 2.2.2 - 1. It has 16 digital inputs, 16 analog inputs, 8 analog outputs and 12 digital outputs. The Modbus board is powered by the 24VDC power supply.

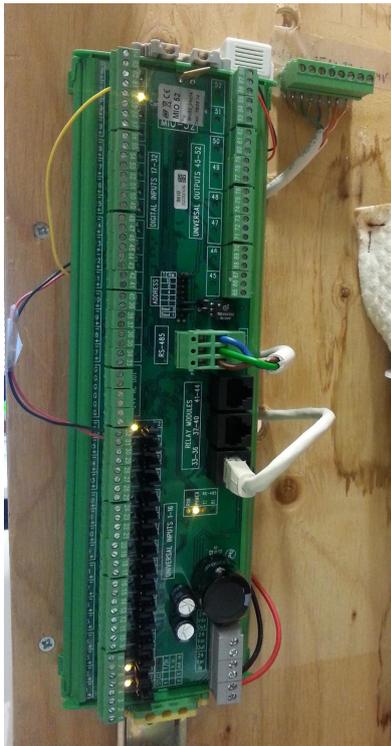


Figure 2.2.2 - 1. Modbus MIO52 installation

Devices connected to the Modbus network are listed below:

### Digital inputs:

- Door sensor (magnetic switch)
- Outdoor motion sensor

### Universal inputs:

- Magnetic sensor for door sensing

### Digital outputs (relay modules):

- nothing

### Universal outputs:

- nothing

### 2.2.3. LonWorks

LonWorks (local operating network) is a networking platform specifically created to address the needs of control applications. The platform is built on a protocol created by Echelon Corporation for networking devices over media such as twisted pair, power lines, fiber optics, and RF. It is used for the automation of various functions within buildings such as lighting and HVAC. The demo house is equipped with a LonWorks network which is connected to the Jace with an option card. The Lon module used here is the Ensto KLON 512, shown in figure 2.2.3 - 1 to the left. The module to the right in the figure is not being used at the moment.



Figure 2.2.3 - 1. The LonWorks installation

The control panel located on the front of the demo house (figure 2.2.3 - 2) is connected to the LonWorks network. The temperature regulator is used to adjust the temperature set point in the control panel.



Figure 2.2.3 - 2. Control panel connected to the LonWorks network

### 2.2.4. Z-Wave

Z-Wave is a wireless communications protocol designed for home automation, specifically for remote control applications in residential and light commercial environments. The technology uses a low-power RF radio embedded or retrofitted into electronic devices and systems, such as lighting, access controls, entertainment systems and household appliances. The Z-wave network is connected to the Jace with an option card. The option card is equipped with an antenna for communication with the wireless sensors, as shown in figure 2.2.4 - 1.



Figure 2.2.4 - 1. The Z-Wave option card installed in the control unit



Figure 2.2.4 - 2. Z-Wave devices used in the project

Figure 2.2.4 - 2 shows some devices connected to the Z-wave network. The devices currently equipped with the project are:

- 2x Wall plug (on/off switch and power consumption meter)
- Temperature and humidity sensor
- Magnetic switch for door sensing

### 3. System functionality

This system is designed for automated control of the conditions in a house. The house is simulated by the wooden box standing on the table. Variables that are controlled are temperature, humidity, CO2 level and lighting. All these variables can be controlled either automatically or manually.

Switching between automatic and manual state is done in the control room by checking or unchecking the *Manual control* checkbox. In the automated state, the user can select setpoints for the desired value of a variable. In manual control the user is able to activate or inactivate an actuator by clicking a toggle button.

The temperature is controlled by switching on the lamp for heating (not in use at the moment), and starting the CO2 fan for cooling. Of course the lamp also produces light, but it is considered as a heater only in this project. CO2 level increases over time when the inhabitants are home and can be decreased by starting the fan on the left side of the house so that fresh air enters the house.

Humidity decreases over time and can be increased by starting the fan on the right side of the house. Note that there has to be water in the cup below the fan so that the wash cloth can absorb the water and blow in moist air. If there is no water in the cup the humidity inside the house will not increase.

Lighting is controlled by setting the voltage level for the three LEDs that are mounted in the ceiling of the box.

#### 3.1. Use cases

The demo house can either be controlled manually or automatically. This feature can be set from the control panel, shown in figure 3.1 - 1. The text right to the navigation will indicate if manual control is on or off.

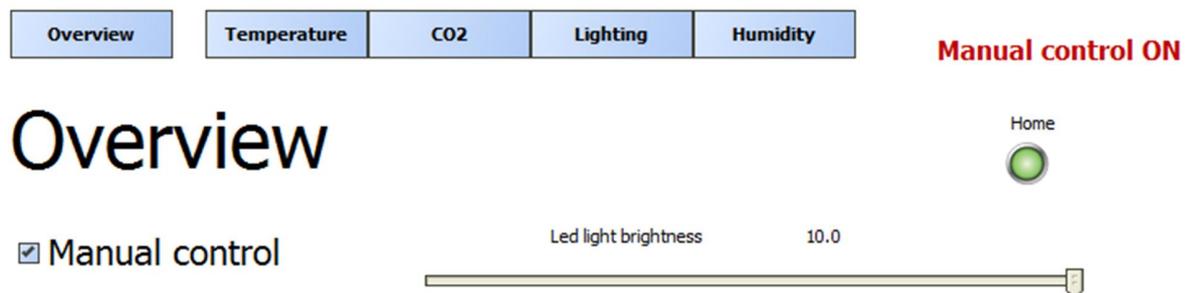


Figure 3.1 - 1. Switching between automatic or manual control

When the house is put into automatic control, it will automatically regulate temperature, carbon dioxide

level, humidity and adjust lighting. Each of the controls can be set to different set points, depending on the state of the house, ie home or away. However, lighting is controlled by motion sensor.

The temperature control will use the heater lamp (not in use at the moment) and the fans for regulating the temperature in the house. If the temperature is too low, then the lamp will be turned on for heating. In the other hand, if the temperature is too high, the fans will turn on to cool the inside air.

Controlling humidity has almost the same functionality as controlling the carbon dioxide level. The humidity control will adjust the humidity level in the house by regulating the fan located on the right side of the house. This fan will insert moist air into the house and this way increase the humidity. In contrast, the carbon dioxide control will regulate the fan located on the left side of the house to adjust the CO2 level inside. If the carbon dioxide level is over the setpoint, then the fan will blow fresh air inside the house to lower the CO2 level.

The demo house features also a home and away functionality, to indicate if inhabitants are present or not. This is controlled with the switch located to the left of the door (9. in figure 1 - 1). The state of the house is indicated by green light when they are home and yellow light when they are away.

An alarm system is installed to the house to prevent intruders when inhabitants are away. The alarm will be triggered when the state of the inhabitants is away and movement is detected inside the house. Triggered alarm will start blinking the lights and sound the siren in sequence.

## 3.2. Control room

The control room is from where the inhabitants of the house would control their automation system. They would have access to it and could change the setpoints and switch between automatic and manual control as they wish. The control room is divided into an overview and the different pages for the house condition variables that can be controlled.

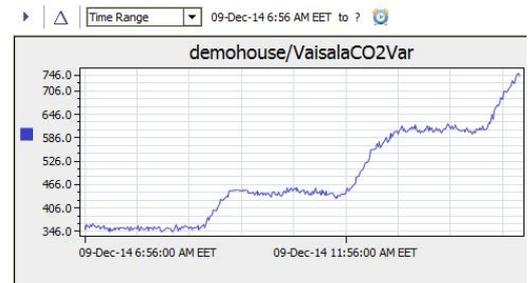
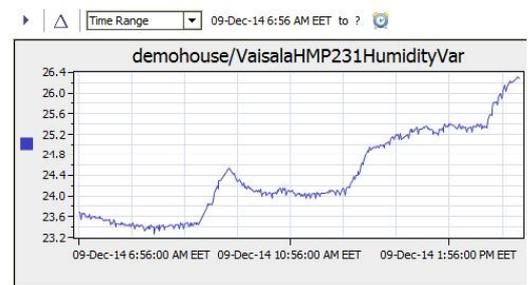
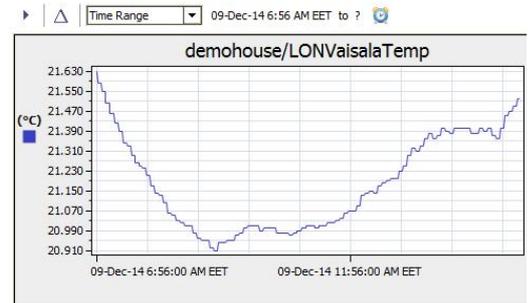
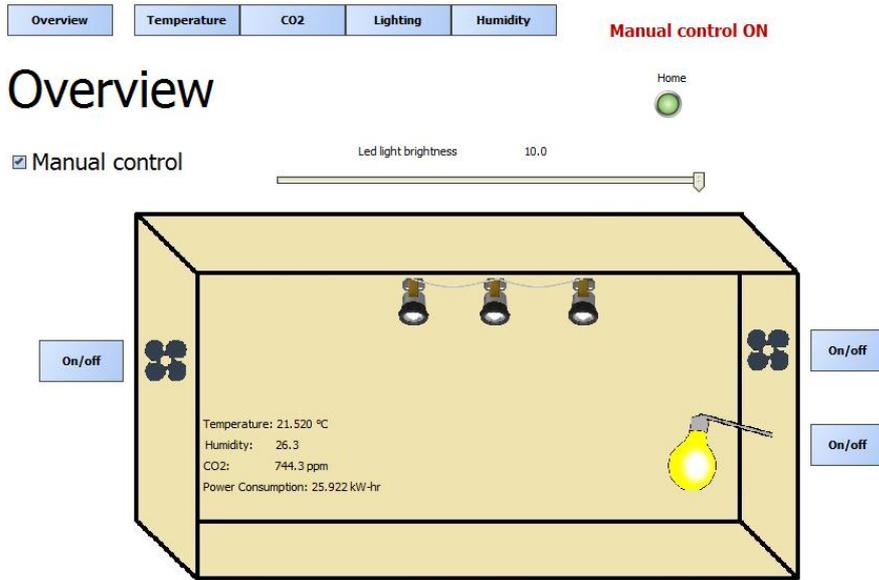


Figure 3.2 - 1. Overview of the control room

The overview shows current values of the variables that can be controlled. It has a checkbox for switching between manual and automatic control, graphs for temperature, humidity and CO2 level. The green light that says *home* will be yellow and say *away* if switch is in the *away* state. The brown box presents the demo house from the front. The fan for lowering CO2 value is on the left side of the box, and the humidity fan is on the right side. Under the humidity fan is the heating lamp and the LEDs are in the ceiling.

When the manual control checkbox is selected, you can toggle the fans and the lamp and set the brightness of the LEDs on the slider. By clicking one of the buttons in the top of the overview you can switch page to see more details about other conditions and set their control values when automatic control is activated.

# Temperature

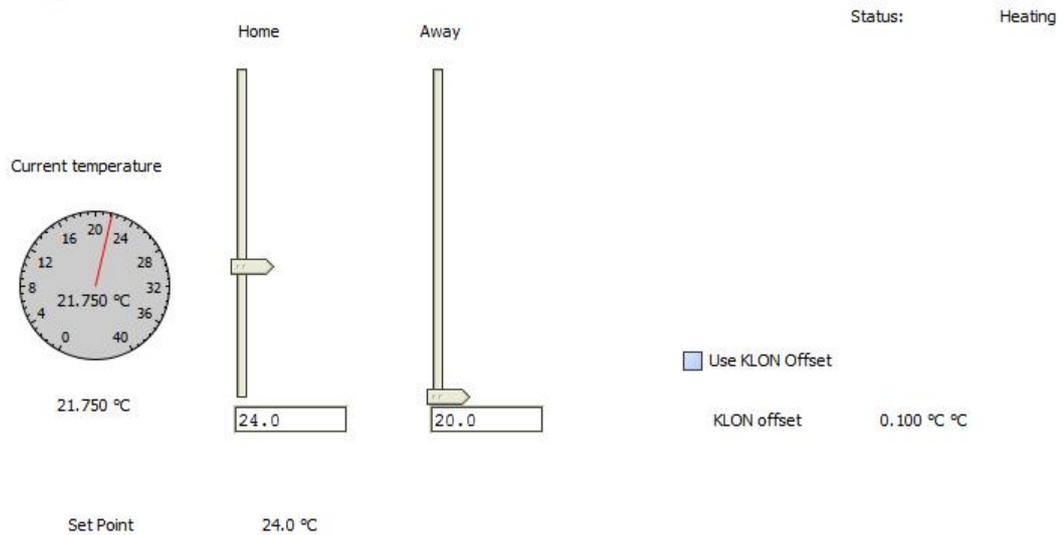


Figure 3.2 - 2. Temperature page in the control room

On the temperature page you can manage the setpoints for when inhabitants are home and away. The away setpoint should generally be lower than the home setpoint because the house does not need to be warm when nobody is home and it also saves energy. The analog meter on the left displays the current temperature. By clicking the checkbox on the right, the setpoint takes into account the offset that is set on the thermostat on the right side in the LonWorks control panel seen in figure 2.2.3. - 2. The status on the right indicates whether the house is currently being heated or cooled.

# CO2

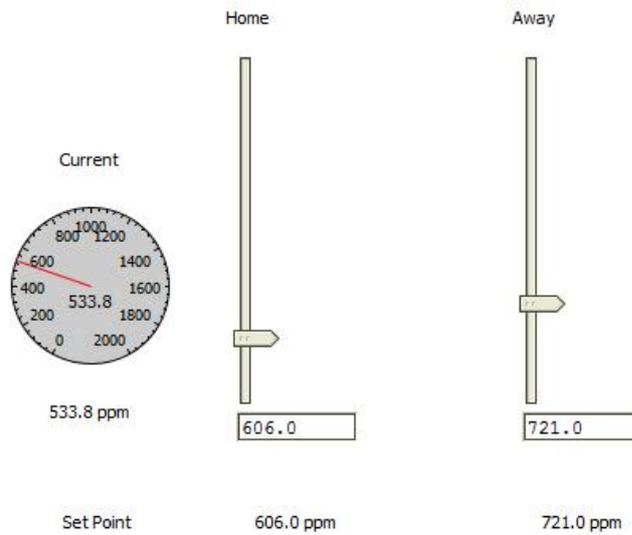


Figure 3.2 - 3. CO2 page in control room

On the CO2 page in the control room the only thing you can do is set the setpoints for CO2 level for home and away states. The analog meter on the left side displays the current CO2 level in ppm. When inhabitants are away the CO2 level can be higher because fresh air is then not so important.

# Lighting

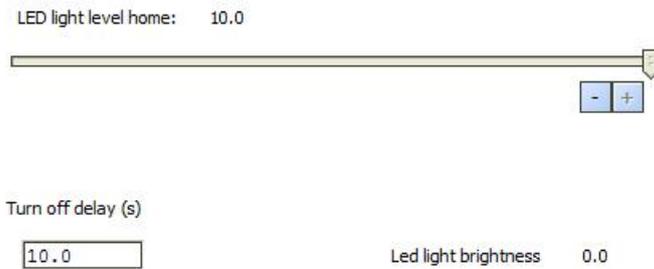


Figure 3.2 - 4. Lighting page in the control room

On the lighting page you can set the LED light brightness for when inhabitants are home. When they are away, the LEDs should not be on and that is why there is no setpoint for the away state. The LEDs will start shining only when the inhabitants are home and the motion sensor inside the house detects movement. On this page you can set the turn off delay for the LEDs, which means how long the LEDs should stay active after the motion sensor no longer senses motion. The current LED brightness is displayed in the lower right corner.

# Humidity

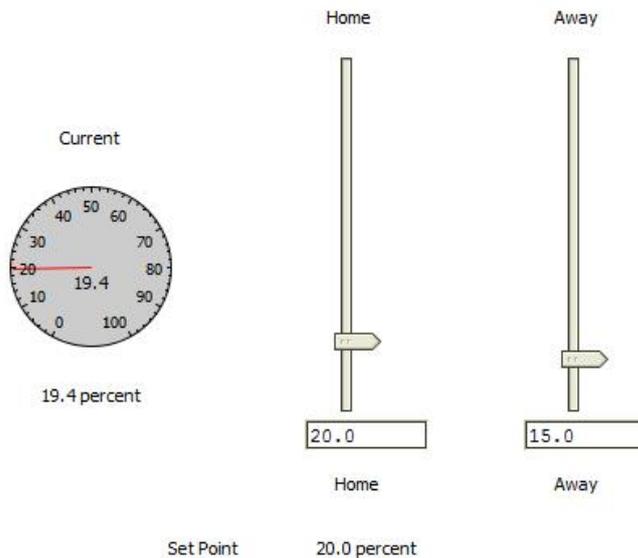


Figure 3.2 - 5. Control room humidity page

On the humidity page you can set setpoints for humidity level for home and away states. The analog meter on the left displays current humidity percentage. The setpoint in the bottom displays which setpoint is currently active. The setpoint for away should be lower than for home, because the air can be dryer when the inhabitants are away.

## 3.3. Logic

The components that you have seen on the different pages of the control room are called widgets. From the control room it is possible to send setpoint values to the logic part of the system. The logics is written using functions block and the view is opened by changing to *wire sheet* in the drop down menu in the upper right corner of WorkPlace AX. Coding with function block is like drawing lines between boxes that does something to the input value. Then usually after a couple of blocks, the output actuator can react to the input value. The logic is separated into different folders for the different types of conditions they are controlling, shown in figure 3.3 -1.

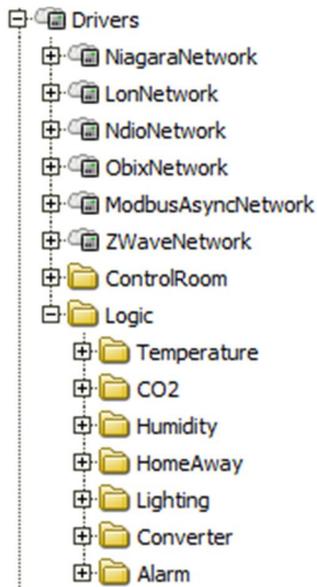


Figure 3.3 - 1. Logic folder structure

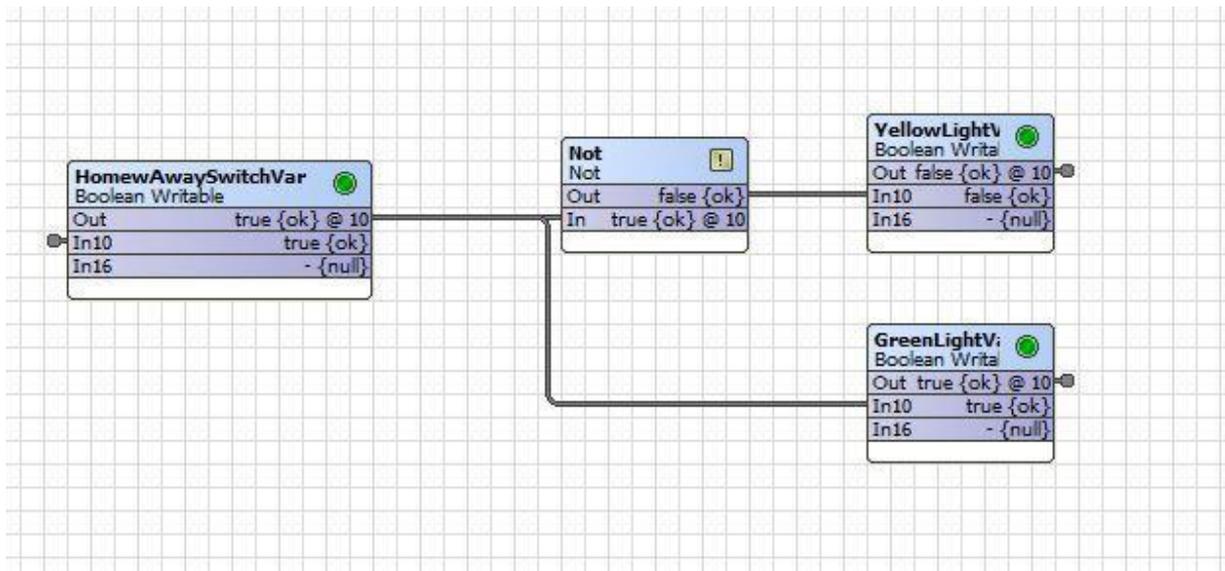
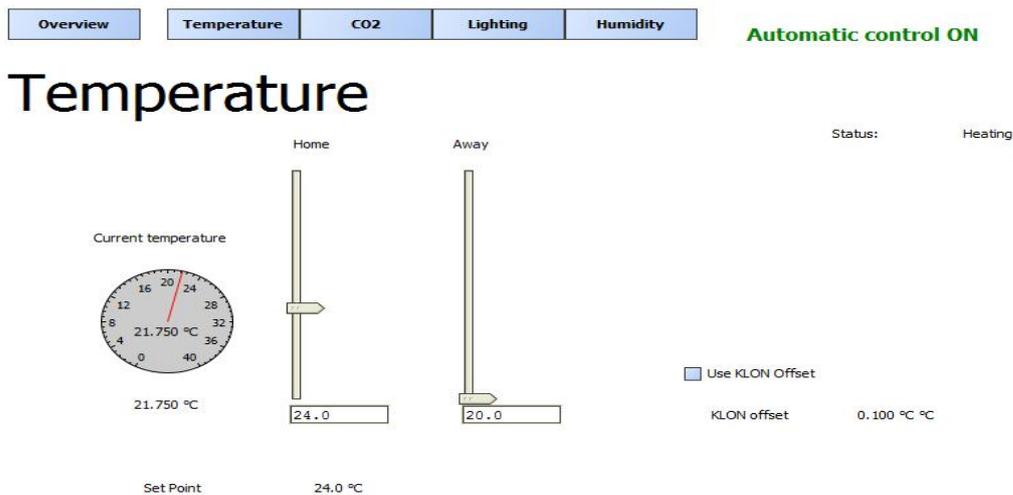


Figure 3.3 - 2. Example code of the home and away logic

## 4. Pre-assignment

This is an individual assignment and everyone participating to the lab should write his/her own answers to the following questions. Take the answers with you to the lab exercise.

Below is a picture of the building automation user interface. Describe what will happen when the checkbox “Use KLON Offset” is activated (0,25 points). How can the user change the KLON offset value (0,25 points)



Your goal is to make a burglar alarm solution that alarms when building owners are away from home but there is movement inside the home. What sensors and actuators of the demohouse would you use in this solution (0,5 points)? How these sensors and actuators are connected to JACE 6E controller (0,5 points)?

Your goal is to make CO2 control to the demohouse. The control should take into account whether there are people inside the house or not. What sensors and actuators of the demohouse would you use in this solution (0,5 points)? Describe the logic of your solution (0,5 points). Discuss also what kind of error situations could happen in the CO2 control and how you have considered these errors in your logic (0,5 points).

After answering the above questions, read the following chapter that describes what will be done during the exercise. It is easier for you to start the exercise when you are familiar with the tasks that need to be done.

You don't need to take this assignment description with you to the lab. There are printed versions available at the lab.

## 5. Exercise

Start the computer and login with the following credentials:

**Username:** automaatti  
**Password:** 84268AUT



realize the changes being done in real time. Change the LED brightness and turn the fans on/off. Remember to have the *'Manual Control'* checked.

After that, uncheck the manual control. This means that the system will control temperature, CO2, humidity and lighting automatically. Navigate in the control room to temperature and change the set points for home and away. See how the house reacts when putting it to the Away-mode, by changing the switch beside the door (9. in figure 1 - 1). Do the same with humidity, CO2. To increase the CO2 value, try exhaling some air into the house from the door.

When this is done, you can try out the alarm system of the house. The alarm is triggered by putting the house into Away-mode and putting your hand inside the door. The indoor motion sensor will detect movement, when the inhabitants are away, which will trigger the alarm. The alarm will end either when the motion ends or the house is put into Home-mode. Note that it takes 10 seconds for the house to go into alarm mode after switching to Away state.

For more detailed information about the control room, see chapter [3.2. Control room](#).

## 5.2. Outdoor motion turns on a LED light (0,5 points)

Your task is to connect a LED light to the Modbus network and add the logic for turning the LED light on when motion is detected outside of the house. You will get 0,5 points for completing this task. The process for achieving this is described in the following chapter. The outside motion sensor is number 3 in figure 1 - 1.

### 5.2.1. Connecting the LED to the Modbus board

The LED light will be added to the Modbus network's universal output number 45. It's labeled with red color, shown in figure 5.2.1. - 1. The LED light has two cables:

- Positive: connected to the pin labeled Y1
- Negative: connected to the ground, G0

This connection will output 0V when off and 10V when it's put on.

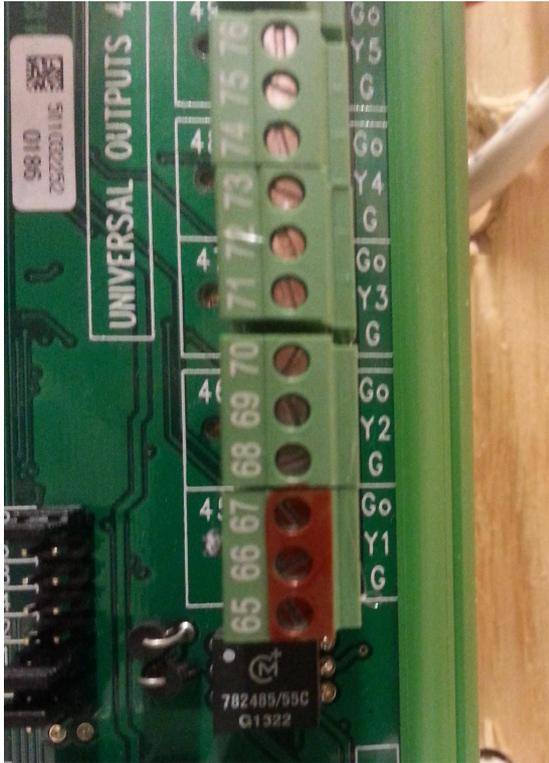


Figure 5.2.1. - 1. Port on the Modbus for connecting the LED light

## 5.2.2. Adding a new point to modbus

The LED light will be connected to the universal output 45 on the Modbus MIO52 board. To control the output in the Workplace AX software, a *point* needs to be added for that universal output. The process on how to add a point is described here.

### **Adding universal output point :**

The universal outputs can have either analog or digital output. Digital outputs are located on Modbus addresses (registers) 13-20 and their data type is Boolean Writable. The value is true or false and correspond to 10V or 0V output on Y1.

Add process:

- Under Station -> Config -> Drivers -> ModbusNetwork -> Modbus Async Device, double-click Points and click New.
- Type to add: Boolean writable
- Address Format: Modbus
- Address: 13 (corresponds to universal output 45)

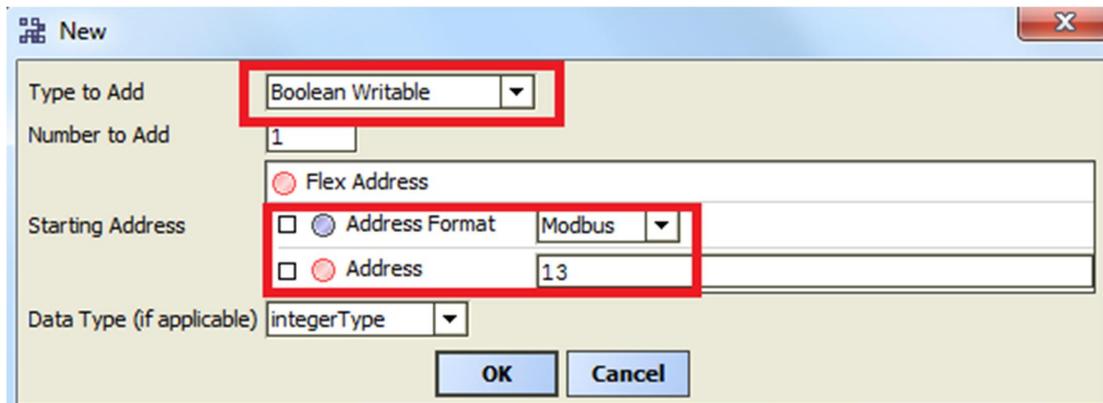


Figure 5.2.2. - 1. Adding digital output on Modbus. In the figure this adds a digital output on universal output 45.

If the point you created shows an error, check that there is not a point added on modbus:13 already. Next, check that the connection is working by right-clicking the point you just created and select Actions -> Set -> True. If everything works, the LED should now switch on. The small LED next to 45 on Modbus should also be bright. Then switch it back off by setting its value to False.

### 5.2.3. Link the point to the logic

Now we have a point in the software connected to the LED output on the Modbus board. The following step is to add the point to the logic, so that the outdoor motion sensor can light up the LED when motion is detected.

Now create a new folder under the Logic folder (Station -> Config -> Drivers -> Logic. This is done by right clicking the 'Logic' folder, and choose New -> Folder. Name it 'Automaatio2'. Double-click the new folder to open the wire sheet view. The logic for the newly added LED light will be added in this location.

Next, create a boolean writable in the wire sheet by right-clicking on the wire sheet, New -> BooleanWritable. The output from this boolean writable is linked to the point you created earlier. This is done by right-clicking the boolean writable and select "Link mark". Then navigate to the point you added to the ModbusNetwork, right-click and select "Link From BooleanWritable". Then link the 'out' from the boolean writable to the point on 'in10'. Try again that the connection is working by activating the boolean writable in the wire sheet (right-click -> Actions -> Set -> True) and see that the LED switches on.

To add more function blocks, take a look at the palette panel (3. in figure 5 - 1). The function blocks are located in the package for *KitControl* (select from dropdown menu in palette). From there, the function blocks can be dragged onto the wire sheet to the right (4. in figure 5- 1). If the wire sheet is not visible, try out the button in the top right corner (5. in figure 5- 1). Whenever you get lost, try out the back button in the top left corner (1. in figure 5- 1).

### 5.3. Adding logic for the LED

The LED will light up when a True value is given to it and shuts down when it gets a False value. The outdoor motion sensor gives these boolean values, so next we will create a boolean writable for the outside motion sensor.

Create a boolean writable in the wire sheet by right-clicking on the wire sheet, New -> BooleanWritable. The input for this boolean point should be linked to the *'OutsideMotionSensorDI31'* point. This is done by right-clicking the boolean writable and select "Link mark". Then navigate to the *'OutsideMotionSensorDI31'* point, located under the Station-> Driver -> Config -> ModbusAsyncNetwork -> Modbus Async Device1 -> Points, right-click and select "Link To BooleanWritable". Then link the *'out'* from the *'OutsideMotionSensor'* to the BooleanWritable on *'in10'*. Now the value of the new boolean writable should be the same as the outdoor motion sensor. Test it out.

Now, combine the output from the OutsideMotionSensor-booleanWritable to the input for the LED-booleanWritable (drag a line from *'out'* to *'in10'* with the mouse). The LED should now light up when there is movement outside the house, and shut off when no motion is detected anymore. Probably it will shut off even though there is movement for a longer period, because the motion sensor only gives true value on certain intervals. This will be improved in the next part of the exercise.

### 5.4. Improve the LED light logic (0,5 points)

You will get 0,5 points for completing this task. To improve the logic for the motion sensor, you can take a look at the logic for the indoor motion sensor. The logic for the demohouse is located under Drivers -> *'Logic'*. Double click the *'Lighting'* folder to view the wire sheet of the lighting logic. DO NOT MAKE CHANGES HERE!

The *'OffDelay'* value is entered from the control panel on the lighting page. This value indicates how long the light should stay on after the value of the motion sensor changes from true to false.

For this exercise, you want to have the value that comes out from the *'Multiply'* function block. This value is the time, in milliseconds, that the logic should wait when turning off the LED after motion has ceased. To get it to your logic, you have to create a link from the *'Multiply'* value to one of your variables in *'Automaatio2'* logic.

Create a new NumericWritable in your own wire sheet. Then link the output of the multiply block to the numeric writable you created.

After linking the *'Multiply'* block as delay in your wire sheet, drag a BooleanDelay-block from the Palette, found in kitControl (dropdown menu) under Timer -> BooleanDelay. The BooleanDelay has a *'Off Delay'* field, but it is not shown in the wire sheet automatically. So you have to add it by right clicking the BooleanDelay block, and choose Pin Slots. Add the *'Off Delay'* as a pinned slot, and press ok.

Now create a connection from 'Out' on NumericWritable to 'Off Delay' on the BooleanDelay block. Then connect the output from the outdoor motion sensor to 'In' on the BooleanDelay. Last, connect the out value from the BooleanDelay block to the BooleanWritable that controls the LED. Now the LED should stay on for an extra 10 seconds after it does not detect movement anymore. If this was for controlling the lighting of a room, it would not turn of the lights immediately after movements stops. 10 seconds would be too short for actual lighting control but is used here for demonstrating the purpose.

## 5.5. Further exercises (1 point)

You can get 1 point for completing this task. Next you can come up with your own ideas that you want to make for the logic. You can use the values from existing points to provide input for your logic. For example you could:

- switch LED on when door is opened
- switch LED on when CO2 level is 600ppm
- switch LED on when humidity is more than 20%

You do not need to implement the above examples, feel free to try out your own ideas! Browse the points on the different networks located under Drivers to see what kind of inputs and outputs you can use. Remember that you can use manual control in the control room overview for controlling the actuators. Most of the function blocks you might need are all found in the kitControl package in Palette.

At the end of the exercise you will demonstrate your implementation for the assistant.