School of Electrical, Electronic & Computer Engineering

Smart Homes Modelling and Implementation

Michael Seaman

Technical Report Series

NCL-EECE-MSD-TR-2009-148

August 2009

Contact: **mikeseaman@mslive.co.uk**

michael.seaman@alumni2009.newcastle.ac.uk

I would like to thank Dr. Fei Xia for his continued support throughout the length of this project.

NCL-EECE-MSD-TR-2009-148

School of Electrical, Electronic and Computer Engineering,

Merz Court,

Newcastle University,

Newcastle Upon Tyne,

NE1 7RU

UK

http://www.async.org.uk

Abstract

This Project is the creation and implementation of a methodical process to create a Smart Home. Initially a Functional specification is written, and then once this has been completed a model is created to verify this functional specification. When the model is created a control strategy is written and implemented using the model, and relationships between the two are given. Finally, a method is given to show a relationship between the three systems and indicate their usefulness.

The Project will also demonstrate this method in use with the implementation of full model environment and control system.

Acknowledgements

I would like to thank my Project Supervisor Dr. Fei Xia for being extremely supportive and very helpful throughout my individual project; I could not have had a better supervisor.

I would also like send my gratitude to Andrey Mokhov for helping me with the initial stages of understanding the Petri Net Development Program, WorkCraft. As well as that, I would like to acknowledge the free educational Ladder Logic software, TriLOGI by Tri-PLC as it has been extremely reliable, easy to understand and free for educational use.

Most importantly of all I would like thank my Parents for helping support me through this Automation & Control Engineering Masters Degree.

Contents

1.0 Introduction

Domotics, Smart Homes, Intelligent Building or Building Automation, have the following classification:

"Automation is the control of a process, machinery or objects without the need or reduced need of human interaction."

As described above, automation is the control of something without the need for a human manually operating it. Within a building, automation could include the control of lights, doors, windows and security systems without a human controlling or manually moving them.

Building automation is a technology that has been implemented for many years. Many large commercialised buildings use automation to manage their environmental and security systems.

One of the most popular building automation systems is the HVAC (Heating, Ventilation and Air Conditioning). The HVAC controls the climatic conditions of a buildings environment and tends to be implemented in most modern day buildings.

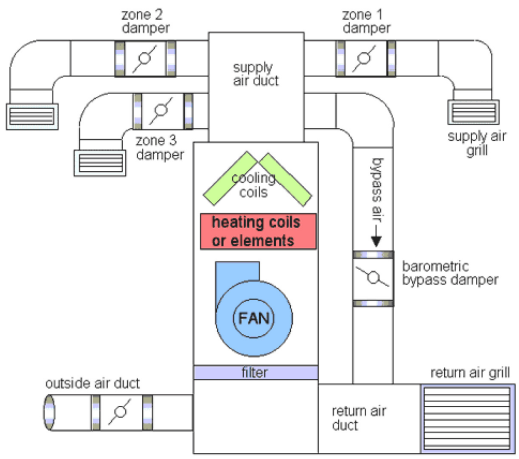


Figure 1 - HVAC by Smart Home Ideas [12]

The heating part of the HVAC is built up from a network of water pipes to radiators or air ducts for force air heating. This network system is built from a centralised system that we call the "central heating system", therefore the need for separate systems in each room is unnecessary and wasteful energy.

Ventilation is a very important part of the HVAC, it is the process of changing or replacing air within a room to either control the temperature, remove moisture, odours and most important anything dangerous or hazardous to the room occupants health, smoke, airborne pathogens. Circulation fans can be used to move air throughout the building and if necessary remove harmful things from within a particular room.

The air conditioning is provided for the removal of heat, not as a necessity as the previous two systems, however it is very easily integrated to make a complete environmental control solution. The refrigeration is mixed in with the ventilation and therefore cold air is passed down an air duct into the desired rooms.

This project is going to look at a very similar type of system. However, the system that will be developed and examined is designed to work in a small home environment.

The smart home environment is done on a much smaller scale, however most systems should be considered.

Next we will take a look at the implications of a Smart Home, and what should be considered as a smart home environment.

1.1 The Implications of a Smart Home

"It's No Place Like Home" [1] gives a brief view into what is predicted for the future of our homes and how technology will play a major role in this. It suggests that all of our household items will be fully automated systems that include security, home entertainment, white goods and environmental controls.

However, the general modern family and their home will not be the only type of market area to benefit from this. It is suggested in some of the articles that home automation is going to significantly improve elderly and disabled people's lives.

One paper discusses the problems facing a global health system with the currently registered disabled people and the factors preceding that with the ever expanding elderly population. This expanding elderly and disabled population can have a strain on a health care service and the paper suggests that more can be done within a home or care environment with the use of automation systems for lighting, doors, blinds or even monitoring systems for health vitals and chemical analysis [2].

From the Office of National Statistics in the UK a report was produced on the national population and it says:

"The percentage of the total population who are over 65 is predicted to rise from 16% to nearly 20% in 2031 and 26.6% in 2071. By 2011, the mean age of the UK population will exceed 40 for the first time and by about 2017/18, there will be more people over 40 than below 40." [9]

Another source from US census [10] is taking recorded population number from the years 1900 – 2006, making a probability model projected for the years 2010 – 2050.

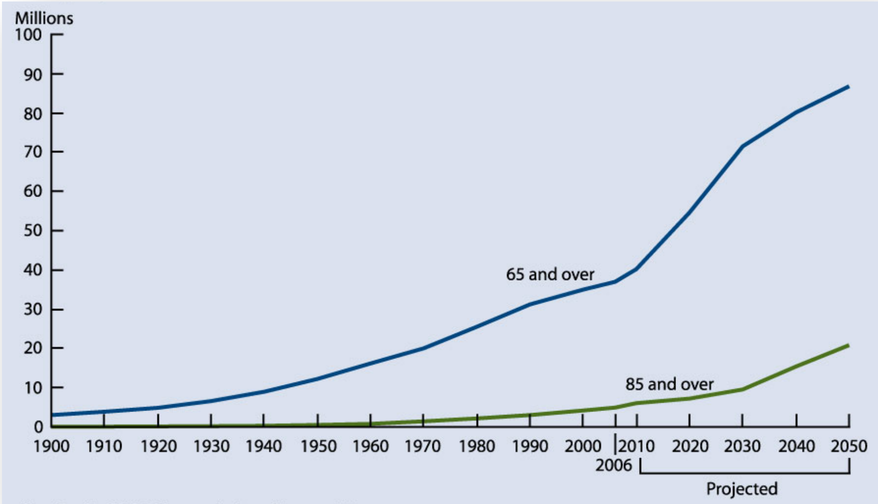


Figure 2 US Census Report Results [10]

These are some significant figures from both sources, papers and journals suggest that this will have an enormous strain on any health service around the globe; this is exactly where automation comes in, especially for elderly people. The elderly are usually required to go into private care later in life because of a systematic breakdown in their health that can leave them unable to do the usual errands around the house. Many of the elderly do not want to enter care because of the loss of their independence.

There are many different types of possible scenarios that automation could play in the household or commercial environment, especially when the current welfare model is simply not able to cope with such a large elderly population.

Many of the journals have said many times that building automation is currently unsustainable as an emerging market simply because of its cost. Within today's global economic crisis and in the near future, we see that cost plays a large part on everything. On the other hand, home automation can save time and a lot of money. With successful integration it can shave a lot of money off energy bills simply because it is a more efficient way of operating things, whether that is turning off a light or computer when you exit the room or altering the temperature effectively using a balance of air-conditioning and correct ventilation.

All of those boils down to the benefits of having a Smart Home, there is a need.

1.2 Motivation for Project

The literature review details exactly the needs for a smart home, current trends available today concentrate solely on the Multimedia end of home automation. The operational side of current commercial smart home technology tends to include the following:

- Lighting
- Environmental Temperature
- Security
- Multimedia Distribution

The problem being with these automatic systems is that they are rarely integrated together.

The task within this project is to integrate a lot of varying systems into one global control system, allowing master states to control the overall running via an operational state system that indicates whether the individual is at home, absent or on holiday.

Also, there is a large need to find a **design process** that can implement a control system for a smart home effectively.

For example, within industry the traditional method of control implementation is to first have a functional specification that details in words exactly how the plant should operate. Once the functional specification is written and fully checked a control system can be developed, primarily using Ladder Logic as is the case with most commonly industrial projects.

Many of the systems that are on the market are based on varying types of programming and use a software shell, like Windows CE or Windows XP. Introducing such a complicated system into a control problem that is simple is rather unnecessary. The task for the project is to use relatively basic control systems to develop a similar end result. One of the tasks that will be omitted by this project is multimedia design. Multimedia systems are a separate entity altogether and therefore is considered beyond the scope of this project.

1.3 The Design Process

The design process plays a fundamental part in this project as a method of identification and validation before a control system is designed is beneficial.

The project will use a 3 tier method broken down into the following processes:

- Functional Specification
- Modelling
- Control Implementation

The functional specification stage allows a verbal representation of the system and therefore pinpoints its exact function. This approach happens in industry to allow this verbal understanding of a control system.

The use of an additional stage to the design process introduces a model environment that can be drawn up and allow an easy representation of a complex control system. The use of the model is primarily to prove what is written in the functional specification works and to test it.

The testing phase is complete and therefore the development can be moved onto actual implementation of the control system. The control system will be developed from the working model in step 2 and allow for easy interpretation of an operational environment from a visual plan. If a problem occurs with the logic in the control system then a problem solving approach can be taken on the model and then the solution can be re-introduced into the control system.

One of the most important parts in this project is the method and links between these three 3 tiers.

2.0 The Smart Home Environment

The smart home environment is one of a complex network of components incorporated in a control system.

The environment that will be modelled within this project is a very basic single level house with the following rooms:

- Bedroom
- Bathroom
- Living Room
- Kitchen
- Hallway
- Exterior

Figure 3 shows the generic layout for this Smart Home Environment

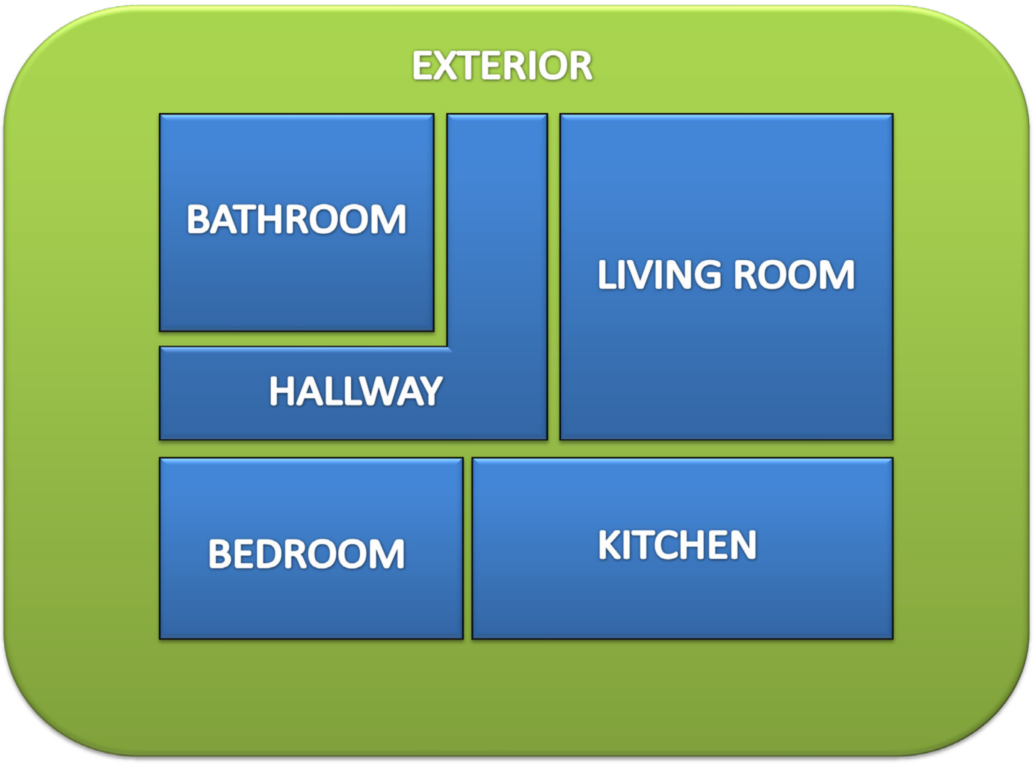


Figure 3 Layout of Generic Building Environment

The following sub sections discuss each of the rooms within the environment and their contents.

2.1 Bedroom

2.1.1 Inputs

Heat Sensor	Active	Blind Manual Open/Close Switch	Active	Blind Position Closed Switch	Active
	Inactive		Inactive		Inactive
Movement Sensor	Active	Door Manual Open/Close Switch	Active	Blind Position Open Switch	Active
	Inactive		Inactive		Inactive
Bed Light Switch	Active	Window Manual Open/Close Switch	Active	Door Position Closed Switch	Active
	Inactive		Inactive		Inactive
Main Light Switch	Active	Temperature State	Hot	Door Position Open Switch	Active
	Inactive		Ok		Inactive
			Cold	Window Position Closed Switch	Active
					Inactive
				Window Position Open Switch	Active
					Inactive

2.1.2 Outputs

Main Light	Active	Blind Opening	Active	Door Opening	Active
	Inactive		Inactive		Inactive
Bed Light	Active	Blind Closing	Active	Door Closing	Active
	Inactive		Inactive		Inactive
Radiator	Active	Window Opening	Active	Window Closing	Active
	Inactive		Inactive		Inactive

2.2 Bathroom

2.2.1 Inputs

Heat Sensor	Active	Blind Manual Open/Close Switch	Active	Blind Position Closed Switch	Active
	Inactive		Inactive		Inactive
Movement Sensor	Active	Door Manual Open/Close Switch	Active	Blind Position Open Switch	Active
	Inactive		Inactive		Inactive
Main Light Switch	Active	Window Manual Open/Close Switch	Active	Door Position Closed Switch	Active
	Inactive		Inactive		Inactive
		Temperature State	Hot	Door Position Open Switch	Active
			Ok		Inactive
			Cold	Window Position Closed Switch	Active
					Inactive
				Window Position Open Switch	Active
					Inactive

2.2.2 Outputs

Main Light	Active	Blind Opening	Active	Door Opening	Active
	Inactive		Inactive		Inactive
Radiator	Active	Blind Closing	Active	Door Closing	Active
	Inactive		Inactive		Inactive
		Window Opening	Active	Window Closing	Active
			Inactive		Inactive

2.3 Living Room

2.3.1 Inputs

Heat Sensor	Active	Blind Manual Open/Close Switch	Active	Blind Position Closed Switch	Active
	Inactive		Inactive		Inactive
Movement Sensor	Active	Door Manual Open/Close Switch	Active	Blind Position Open Switch	Active
	Inactive		Inactive		Inactive
Table Light Switch	Active	Window Manual Open/Close Switch	Active	Door Position Closed Switch	Active
	Inactive		Inactive		Inactive
Main Light Switch	Active	Temperature State	Hot	Door Position Open Switch	Active
	Inactive		Ok		Inactive
			Cold	Window Position Closed Switch	Active
					Inactive
				Window Position Open Switch	Active
					Inactive

2.3.2 Outputs

Main Light	Active	Blind Opening	Active	Door Opening	Active
	Inactive		Inactive		Inactive
Table Light	Active	Blind Closing	Active	Door Closing	Active
	Inactive		Inactive		Inactive
Radiator	Active	Window Opening	Active	Window Closing	Active
	Inactive		Inactive		Inactive

2.4 Kitchen

2.4.1 Inputs

Heat Sensor	Active	Blind Manual Open/Close Switch	Active	Blind Position Closed Switch	Active
	Inactive		Inactive		Inactive
Movement Sensor	Active	Internal Door Manual Open/Close Switch	Active	Blind Position Open Switch	Active
	Inactive		Inactive		Inactive
Main Light Switch	Active	Window Manual Open/Close Switch	Active	Internal Door Position Closed Switch	Active
	Inactive		Inactive		Inactive
Temperature State	Hot	Back Door Manual Open/Close Switch	Active	Internal Door Position Open Switch	Active
	Ok		Inactive		Inactive
	Cold	Back Door Position Closed Switch	Active	Window Position Closed Switch	Active
			Inactive		Inactive
		Back Door Position Open Switch	Active	Window Position Open Switch	Active
			Inactive		Inactive

2.4.2 Outputs

Main Light	Active Inactive	Blind Opening	Active Inactive	Door Opening	Active Inactive
Bed Light	Active Inactive	Blind Closing	Active Inactive	Door Closing	Active Inactive
Radiator	Active Inactive	Window Opening	Active Inactive	Back Door Opening	Active Inactive
Fridge Normal Operation	Active Inactive	Window Closing	Active Inactive	Back Door Closing	Active Inactive
Fridge Lower Refrigeration	Active Inactive				

2.5 Hallway

2.5.1 Inputs

Heat Sensor	Active Inactive	Blind Manual Open/Close Switch	Active Inactive	Blind Position Closed Switch	Active Inactive
Movement Sensor	Active Inactive	Front Door Manual Open/Close Switch	Active Inactive	Blind Position Open Switch	Active Inactive
Main Light Switch	Active Inactive	Temperature State	Hot Ok	Front Door Position Closed Switch	Active Inactive
		Letter Box Position Closed Switch	Active Inactive	Front Door Position Open Switch	Active Inactive

2.5.2 Outputs

Main Light	Active	Blind Opening	Active	Door Opening	Active
	Inactive		Inactive		Inactive
Letter Box Lock	Active	Blind Closing	Active	Door Closing	Active
	Inactive		Inactive		Inactive
Radiator	Active				
	Inactive				

2.6 Security & Core Systems

2.6.1 Inputs

Boiler Water Manual Shutoff	Active	Gas Utility Manual Shutoff	Active	Water Utility Manual Shutoff	Active
	Inactive		Inactive		Inactive
Boiler Central Heating Manual Shutoff	Active	Electricity Utility Manual Shutoff	Active		
	Inactive		Inactive		

2.6.2 Outputs

Full Alarm	Active	Boiler Water	Active	Gas Utility	Active
	Inactive		Inactive		Inactive
Pulse Warning Alarm	Active	Boiler Central Heating	Active	Electricity Utility	Active
	Inactive		Inactive		Inactive
				Water Utility	Active
					Inactive

2.7 Exterior

2.7.1 Inputs

Exterior Light Manual	Active	Movement Sensor Front	Active	Movement Sensor Right	Active
	Inactive		Inactive		Inactive
Movement Sensor Back	Active	Movement Sensor Left	Active	Front Access Switch	Active
	Inactive		Inactive		Inactive
				Gate Position Closed Switch	Active
					Inactive

2.7.2 Outputs

Gate Lock	Active	Front Light	Active	Right Light	Active
	Inactive		Inactive		Inactive
Back Light	Active	Left Light	Active		
	Inactive		Inactive		

3.0 Functional Specification

The functional specification denotes exactly how the Smart Home Automation system should operate. The functional specification will be split into areas from within the building, these areas will generally interact with each, however shall be governed by the overall master state.

3.1 Master States

The Master State is what governs the overall environment of the smart home and denotes whether the environment should be locked down and secure when the individual is absent or on holiday. The following states are classified as Master States:

- **Home** (Denotes that there is an active person within the Home environment)
- **Absent** (Indicates that the individual has left the building, but only for a small amount of time)
- **Holiday** (Indicates that the individual has left the building for a long period of time)

Within these master states, there can be multiple combinations of them together. For example if the individual is on holiday and another individual (neighbour) wants to come in and check the house however does not want windows to open or initiate gas & electricity systems, i.e. is only in the house for a small amount of time then the system should allow both **Home** and **Holiday** to be selected. The gives arise to the following logical combinations for the Master State System:

- Home
- Home & Absent
- Absent
- Home & Holiday
- Holiday

Figure 4 shows that the system must always have to go through **Home** and the system can never be left without a state.

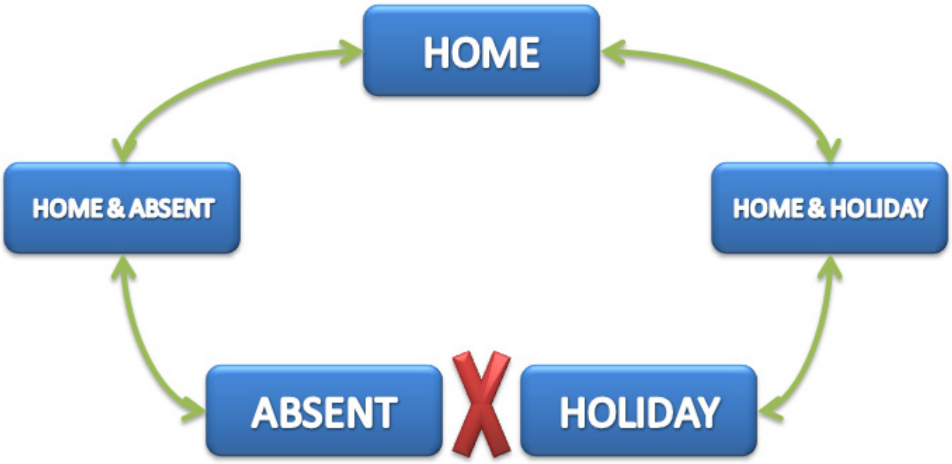


Figure 4 Master State Routine

Therefore if the person is to set the system to **Absent**, you must first have **Absent** and **Home** enabled before **Home** can be disabled to leave only **Absent** Enabled. This ruling is exactly the same for the **Holiday** state, however the **Holiday** state cannot allow **Absent** to be enabled at the same time, therefore you can never have the state **Absent** & **Holiday**. Shown in Figure 4 is this factor that **Absent** & **Holiday** cannot have a shared state and therefore must travel the route through home before either of them may be an active state.

3.2 Bedroom

The bedroom is to be governed by the **Masters States** and their routine is described in section 3.1, this should influence this systems operation.

This room is down into many individual controllable systems and have relevant inputs and outputs to control, these are listed in section 2.1 on page 15.

3.2.1 Occupied

This is to show that the room has an individual inside it. The micro system consists of a **heat sensor** and **movement sensor** and can be activated by either of these sensors being active. To deactivate this room-occupied micro system both the **heat sensor** and **movement sensor** have to be inactive for the room to show as un-occupied.

For the system to be active, i.e. the room is **Occupied**.

- Heat Sensor **Active**

OR

- Movement **Active**

For the room to show that it is **Un-occupied**, you require the following in an AND situation.

- Heat Sensor **Inactive**
- Movement **Inactive**

3.2.2 Bed Light

The bed light is turned **ON** by an AND expression that consists of the following being active to work.

- Bed light switch **Active**
- Home **Active**
- Outside Light Sensor **Dark**
- Room **Occupied**

The Bed light can be turned **OFF** by an OR expression that consists of the following:

- Bed light switch **Inactive**

OR

- Home **Inactive**

OR

- Outside Light Sensor **Light**

OR

- Room **Un-occupied**

3.2.3 Main Light

The main light is turned **ON** by an AND expression that consists of the following being active to work:

- Main light switch **Active**
- Home **Active**
- Outside Light Sensor **Dark**
- Room **Occupied**

The Main light can be turned **OFF** by an OR expression that consists of the following:

- Main light switch **Inactive**

OR

- Home **Inactive**

OR

- Outside Light Sensor **Light**

OR

- Room **Un-occupied**

3.2.4 Radiator

The radiator is turned **ON** by an AND expression that consists of the following being active to work:

- Room Temperature Sensor **Cold**
- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Room **Occupied**

The radiator can be turned **OFF** by an OR expression that consists of the following:

- Home **Inactive**

OR

- Absent **Active**

OR

- Holiday **Active**

OR

- Room Temperature Sensor **Ok**

OR

- Room Temperature Sensor **Hot**

OR

- Room **Unoccupied**

3.2.5 Blind

The Blind System is a much more complex than the previous. With this system there are opening and closing outputs; one output turns the motor one way to open the blind and the other output turns the motor the other way to close the blind.

The Blind runs on the following loop effect as denoted in Figure 5.

Figure 5 Opening Closing Routine

Figure 5 shows that once the blind is opened or even closed then the routine must go back to Halt. The next few stages should explain what logical systems need to be active to allow the transitions in Figure 5.

<u>Manual Override</u>

The manual override allows the blind system to know that the blind is being controlled manually and should not operate automatically.

For the manual override to be active it requires the following to be active.

- Blind Manual Open/Close Switch **Active**

For the manual override to be inactive it requires the following to be inactive.

- Room **Unoccupied**

<u>Halt to Opening</u>

- Home **Active**

- Absent **Inactive**
- Holiday **Inactive**
- Outside Light Sensor **Light**
- Blind Position Closed Switch **Active**
- Blind Manual Override **Inactive**

OR

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Blind Manual Open/Close Switch **Active**
- Blind Position Closed Switch **Active**
- Blind Manual Override **Active**

<u>Opening to Halt</u>

- Blind Position Open Switch **Active**

<u>Halt to Closing</u>

- Blind Position Open Switch **Active**
- Outside Light Sensor **Dark**
- Blind Manual Override **Inactive**

OR

- Blind Manual Open/Close Switch **Active**
- Blind Manual Override **Active**
- Blind Position Open Switch **Active**
- Home **Active**

OR

- Blind Position Open Switch **Active**

- Absent **Active**

OR

- Blind Position Open Switch **Active**
- Holiday **Active**

Closing to Halt

- Blind Position Closed Switch **Active**

3.2.6 Window

The Window System is a very similar to that of the blind system, except with a couple of modifications. With this system there are opening and closing outputs, one output turns the motor one way to open the Window and the other output turns the motor the other way to close the Window.

The Window runs as shown previously in Figure 5.

The next few stages should explain what logical systems need to be active to allow the transitions in Figure 5.

Manual Override

The manual override allows the window system to know that the window is being controlled manually and should not operate automatically.

For the manual override to be active it requires the following to be active:

- Window Manual Open/Close Switch **Active**

For the manual override to be inactive it requires the following to be inactive:

- Room **Unoccupied**

Halt to Opening

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Room Temperature Sensor **Hot**
- Window Position Closed Switch **Active**

- Window Manual Override **Inactive**

OR

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Window Manual Open/Close Switch **Active**
- Window Position Closed Switch **Active**
- Window Manual Override **Active**

<u>Opening to Halt</u>

- Window Position Open Switch **Active**

<u>Halt to Closing</u>

- Window Position Open Switch **Active**
- Room Temperature Sensor **Cold**
- Window Manual Override **Inactive**

OR

- Window Manual Open/Close Switch **Active**
- Window Position Open Switch **Active**
- Window Manual Override **Active**
- Home **Active**

OR

- Window Position Open Switch **Active**
- Absent **Active**

OR

- Window Position Open Switch **Active**
- Holiday **Active**

Closing to Halt

- Window Position Closed Switch **Active**

3.2.7 Door

The internal Door System is almost identical to that of the window system however; the only major difference is there is a closing to opening routine, which allows the door to skip the Halt stage. With this system there are opening and closing outputs, one output turns the motor one way to open the door and the other output turns the motor the other way to close the door.

The door runs on the following loop effect as denoted in Figure 6.

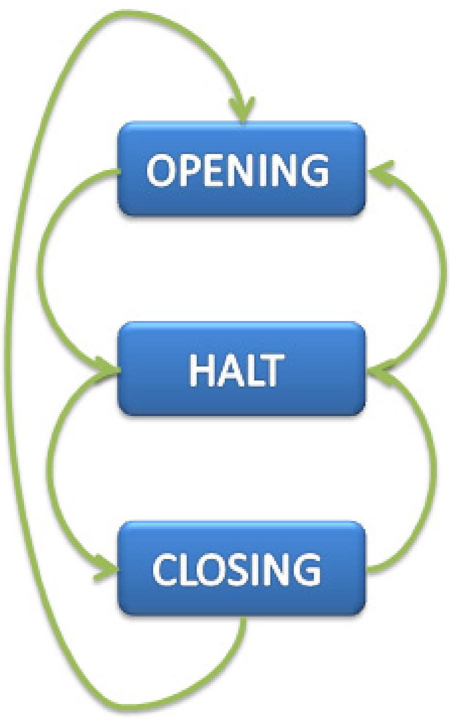


Figure 6 Door Opening Closing Routine

Figure 6 shows that once the door is opened or even closed then the routine must go back to Halt. However when the door is closing it is able to skip the halt phase, for example if the door is closing then it can jump automatically to opening without the need to pass the Halt point and also does not require to check position sensor for whether the door is shut.

The next few stages should explain what logical systems need to be active to allow the transitions in Figure 6.

Manual Override

The manual override allows the door system to know that the blind is being controlled manually and should not operate automatically.

For the manual override to be active it requires the following to be active.

- Door Manual Open/Close Switch **Active**

For the manual override to be inactive it requires the following to be inactive.

- Room **Unoccupied**

Halt to Opening

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Room Temperature Sensor **Hot**
- Door Position Closed Switch **Active**
- Door Manual Override **Inactive**

OR

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Door Manual Open/Close Switch **Active**
- Door Position Closed Switch **Active**
- Door Manual Override **Active**

Opening to Halt

- Door Position Open Switch **Active**

Halt to Closing

- Door Position Open Switch **Active**
- Room Temperature Sensor **Cold**

- Door Manual Override **Inactive**

OR

- Door Manual Open/Close Switch **Active**
- Door Position Open Switch **Active**
- Door Manual Override **Active**
- Home **Active**

OR

- Door Position Open Switch **Active**
- Absent **Active**

OR

- Door Position Open Switch **Active**
- Holiday **Active**

<u>Closing to Halt</u>

- Door Position Closed Switch **Active**

<u>Closing to Opening</u>

- Home **Active**
- Absent **Inactive**
- Holiday **Inactive**
- Door Manual Open/Close Switch **Active**
- Door Manual Override **Active**

3.3 Bathroom

The bathroom is controlled by the Masters States as previously explained in section 3.2 and should therefore influence the operation of this room.

This room is down into many individual controllable systems and have relevant inputs and outputs to control, these are listed in section 2.2 on page 16.

3.3.1 Occupied

Please see 3.2.1 on Page 24

3.3.2 Main Light

Please see 3.2.3 on Page 25

3.3.3 Radiator

Please see 3.2.4 on Page 26

3.3.4 Blind

Please see 3.2.5 on Page 27

3.3.5 Window

Please see 3.2.6 on Page 29

3.3.6 Door

Please see 3.2.7 on Page 31

3.4 Living Room

The living room is controlled by the Masters States as previously explained in section 3.2 and should therefore influence the operation of this room.

This room is down into many individual controllable systems and have relevant inputs and outputs to control, these are listed in section 2.3 on page 17.

3.4.1 Occupied

Please see 3.2.1 on Page 24

3.4.2 Table Light

The table light is turned **ON** by an AND expression that consists of the following being active to work:

- Table light switch **Active**
- Home **Active**
- Outside Light Sensor **Dark**
- Room **Occupied**

The table light can be turned **OFF** by an OR expression that consists of the following:

- Table light switch **Inactive**

OR

- Home **Inactive**

OR

- Outside Light Sensor **Light**

OR

- Room **Un-occupied**

3.4.3 Main Light

Please see 3.2.3 on Page 25

3.4.4 Radiator

Please see 3.2.4 on Page 26

3.4.5 Blind

Please see 3.2.5 on Page 27

3.4.6 Window

Please see 3.2.6 on Page 29

3.4.7 Door

Please see 3.2.7 on Page 31

3.5 Kitchen

The kitchen is controlled by the Masters States as previously explained in section 3.2 and should therefore influence the operation of this room.

This room is down into many individual controllable systems and have relevant inputs and outputs to control, these are listed in section 2.4 on page 19.

3.5.1 Occupied

Please see 3.2.1 on Page 24

3.5.2 Main Light

Please see 3.2.3 on Page 25

3.5.3 Radiator

Please see 3.2.4 on Page 26

3.5.4 Blind

Please see 3.2.5 on Page 27

3.5.5 Window

Please see 3.2.6 on Page 29

3.5.6 Door

Please see 3.2.7 on Page 31

3.5.7 Back Door

The back door system consists of a locking mechanism, position sensor and movement actuators.

The system is always locked by having the following AND configuration.

- Security **Active**
- Back Door Position Closed Switch **Active**

The lock is unlocked by the following logical combinations

- Security **Inactive**

Now that the door system is **unlocked** it should allow pressing a manual entry button or key lock entry which should give the following logical combinations.

Halt to Opening

- Back Door **Unlocked**
- Back Door Manual Open/Close Switch **Active**
- Back Door Position Closed Switch **Active**

Opening to Halt

- Door Position Open Switch **Active**

Halt to Closing

- Door Manual Open/Close Switch **Active**
- Door Position Open Switch **Active**
- Home **Active**

OR

- Door Position Open Switch **Active**
- Security **Active**

Closing to Halt

- Door Position Closed Switch **Active**

Closing to Opening

- Back Door **Unlocked**
- Door Manual Open/Close Switch **Active**

3.5.8 Fridge

The fridge can be classified as a utility system; however it will not be inactive when the individual in the house is on holiday or absent. Instead it drops the temperature levels to allow food to last longer.

Normal operating conditions of the fridge require the following logical combinations.

- Home **Active**

OR

- Away **Active**

For the normal operating conditions to be deactivated it requires the following:

- Holiday **Active**

To activate the lower refrigeration levels and drop the temperature in the fridge, it requires the following to be active:

- Holiday **Active**

For this lower refrigeration level system to be deactivated then:

- Holiday **Inactive**

3.6 Hallway

3.6.1 Occupied

Please see 3.2.1 on Page 24

3.6.2 Main Light

Please see 3.2.3 on Page 25

3.6.3 Radiator

Please see 3.2.4 on Page 26

3.6.4 Blind

Please see 3.2.5 on Page 27

3.6.5 Front Door

Please see 3.5.7 on Page 37

3.6.6 Letter Box

The letter box system consists of a lock and position sensor.

The system is always **locked** by having the following AND configuration:

- Security **Active**
- Front Access **Inactive**
- Letter box Position Closed Switch **Active**

The letter box is **unlocked** by the following logical combinations:

- Security **Inactive**

OR

- Security **Active**
- Front Access **Active**

3.7 Security & Core Systems

3.7.1 Security Activation

The Security Active logic is used by other systems to understand when someone is at home with correct authorisation.

It is activated by the following:

- Home **Inactive**
- Bedroom Window Position Closed Switch **Active**
- Bathroom Window Position Closed Switch **Active**
- Living room Window Position Closed Switch **Active**
- Kitchen Window Position Closed Switch **Active**
- Front Door Position Closed Switch **Active**
- Back Door Position Closed Switch **Active**
- Letter box Position Closed Switch **Active**

It is de-activated by the following:

- Home **Active**

As can be seen, the logic for security is a further expanse of the Home Active / Inactive logic, it denotes to the rest of the system that someone is at home with correct authorisation. However, it checks other systems are secure before it can be fully active.

3.7.2 Full Alarm

The Full Alarm is a further extension of the Warning Alarm, however this denotes to everyone that there is an intruder in the house.

It can be activated by the following OR situations:

- Bedroom **Occupied**
- Security **Active**

OR

- Bathroom **Occupied**

- Security **Active**

OR

- Living Room **Occupied**
- Security **Active**

OR

- Kitchen **Occupied**
- Security **Active**

OR

- Hallway **Occupied**
- Security **Active**

It can be also activated via the following window open switch combinations too.

- Bedroom Window Position Closed Switch **Inactive**
- Security **Active**

OR

- Bathroom Window Position Closed Switch **Inactive**
- Security **Active**

OR

- Living Room Window Position Closed Switch **Inactive**
- Security **Active**

OR

- Kitchen Window Position Closed Switch **Inactive**
- Security **Active**

OR

- Back Door Position Closed Switch **Inactive**
- Security **Active**

OR

- Front Door Position Closed Switch **Inactive**
- Security **Active**

3.7.3 Warning Alarm

The warning alarm is an alarm that gives warning to intruder; it is pulse which will deter the person entering the vicinity of the building.

The warning alarm is activated via the following logical combinations.

- Security **Active**
- Security Movement Sensor Back **Active**

OR

- Security **Active**
- Security Movement Sensor Front **Active**
- Front Access **Inactive**

OR

- Security **Active**
- Security Movement Sensor Left **Active**

OR

- Security **Active**
- Security Movement Sensor Right **Active**

There are two methods to switching the alarm off and they are:

- Security Movement Sensor Back **Inactive**
- Security Movement Sensor Front **Inactive**
- Security Movement Sensor Left **Inactive**
- Security Movement Sensor Right **Inactive**

OR

- Security Movement Sensor Back **Inactive**
- Security Movement Sensor Front **Active**

- Security Movement Sensor Left **Inactive**
- Security Movement Sensor Right **Inactive**
- Front Access **Active**

3.7.4 Water Boiler

The water boiler is made up of two separate systems. One of which is for heating the water you see in the tap and storing in a tank, and the other is for turning the central heating on.

Hot Water Supply System

The hot water needs to be active when there is someone at home and absent for short periods of time. However if the individual goes on holiday, then it needs to be completely inactive.

So therefore we want the following to make this system active:

- Gas **Active**
- Water **Active**
- Holiday **Inactive**
- Boiler Water Manual Shutoff **Inactive**

For the hot water supply system to be deactivated it requires the following logical combination:

- Holiday **Active**

OR

- Gas **Inactive**

OR

- Water **Inactive**

OR

- Boiler Water Manual Shutoff **Active**

Central Heating Supply System

The central heating system should only be active when the individual is at home and that there is not an absent or holiday option selected. Also the central heating system should only be activated when a radiator calls it into action. There does not want to be a central heating system operating and no radiator on to distribute the heat.

Therefore, it can be said that the system should be active when:

- Bedroom Radiator **Active**
- Gas **Active**
- Water **Active**
- Boiler Central Heating Manual Shutoff **Inactive**

OR

- Hallway Radiator **Active**
- Gas **Active**
- Water **Active**
- Boiler Central Heating Manual Shutoff **Inactive**

OR

- Living Room Radiator **Active**
- Gas **Active**
- Water **Active**
- Boiler Central Heating Manual Shutoff **Inactive**

OR

- Kitchen Radiator **Active**
- Gas **Active**
- Water **Active**
- Boiler Central Heating Manual Shutoff **Inactive**

OR

- Bathroom Radiator **Active**
- Gas **Active**
- Water **Active**

- Boiler Central Heating Manual Shutoff **Inactive**

The Central heating system will be inactive when:

- Bedroom Radiator **Inactive**
- Hallway Radiator **Inactive**
- Living Room Radiator **Inactive**
- Kitchen Radiator **Inactive**
- Bathroom Radiator **Inactive**

OR

- Gas **Inactive**

OR

- Water **Inactive**

OR

- Boiler Central Heating Manual Shutoff **Active**

3.7.5 Utility Systems

This is a safety shut off for all of the systems in the house. The will only operate when the individual goes on Holiday. It is a secure shut off for all the house non-essential systems. It prevents gas and water leaks and minimises the risk of electrical fires when there is not anyone in the building. Also, the electricity cut off is very useful to prevent standby systems uselessly costing electricity. For example when the system is active the utility is flowing, and when the system is inactive then the utility is not flowing.

<u>Gas</u>

For the gas to be **flowing** then:

- Holiday **Inactive**
- Gas Manual Shutoff **Inactive**

For the gas to be **not** flowing and **disconnected** then:

- Holiday **Active**
- Home **Inactive**

OR

- Gas Manual Shutoff **Active**

<u>Electricity</u>

For the electricity to be **flowing** then:

- Holiday **Inactive**
- Electricity Manual Shutoff **Inactive**

For the electricity to be **not** flowing and **disconnected** then:

- Holiday **Active**
- Home **Inactive**

OR

- Electricity Manual Shutoff **Active**

<u>Water</u>

For the electricity to be **flowing** then:

- Holiday **Inactive**
- Water Manual Shutoff **Inactive**

For the electricity to be **not** flowing and **disconnected** then:

- Holiday **Active**
- Home **Inactive**

OR

- Water Manual Shutoff **Active**

3.8 Exterior

3.8.1 Back Lighting

The Back Lighting is activated using a movement sensor that covers that area of the building. The following should cause it to be activated.

- Security Movement Sensor Back **Active**
- Outside **Dark**
- Security **Active**

OR

- Home **Active**
- Security Lighting Manual Switch **Active**
- Outside **Dark**

For the outside back lighting to switch off the system must have the following logical combinations:

- Security Movement Sensor Back **Inactive**

OR

- Security **Inactive**
- Security Lighting Manual Switch **Inactive**

OR

- Outside **Light**

3.8.2 Front Lighting

The Front lighting system is identical to that described in the back lighting functional standard, however the only difference being is that movement sensor is on the front of the building.

Please see section 3.8.1 on Page 48.

3.8.3 Left Lighting

The Left lighting system is identical to that described in the back lighting functional standard, however the only difference being is that movement sensor is on the left of the building.

Please see section 3.8.1 on Page 48.

3.8.4 Right Lighting

The Right lighting system is identical to that described in the back lighting functional standard, however the only difference being is that movement sensor is on the right of the building.

Please see section 3.8.1 on Page 48.

3.8.5 Gate

The gate is a different system as it has no moving parts. It consists of a lock and position sensor.

The system is always locked by having the following AND configuration:

- Security **Active**
- Front Access **Inactive**

The lock is unlocked by the following logical combinations:

- Security **Inactive**

OR

- Front Access **Active**

4.0 Modelling

The modelling stage, as previously explained in the introduction, is the process which validates the functional specification. Investigations into different modelling techniques led to the use of Petri Nets.

4.1 Petri Nets

Petri Nets are also known as Place / Transitions Networks and is one that is shown using a type of bipartite graph from which it is built up from place holders, transitions and directed arcs.

Below Figure 7 explains how the Petri Net technique works.

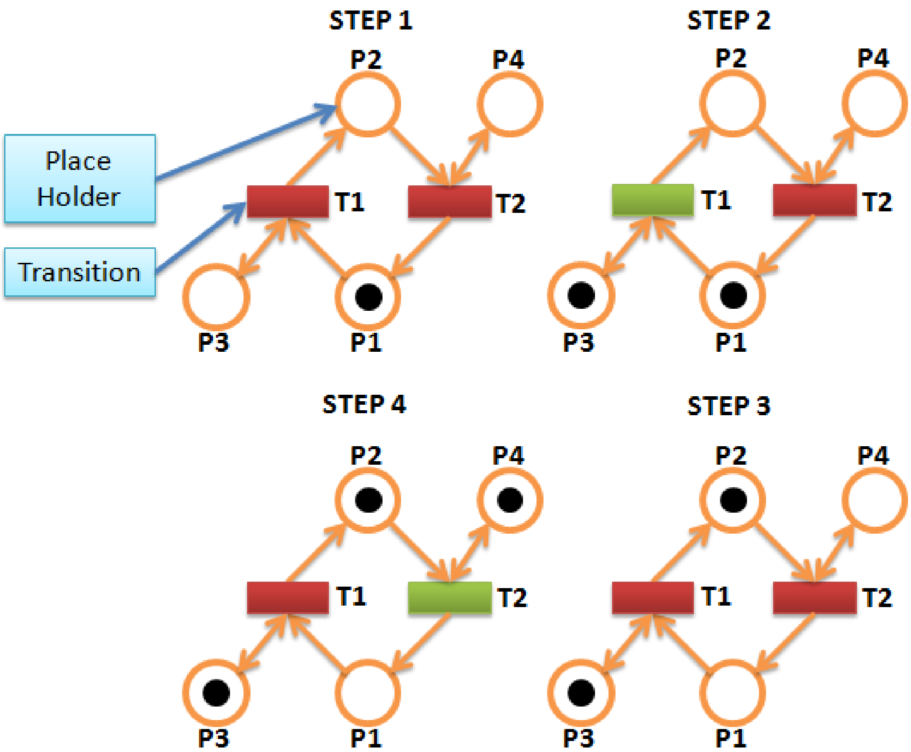


Figure 7 Petri Net Examples

The place holders contain a black marker which is classified as a **token**. When the **token** is within a place holder, it indicates that the place holder is active.

So for example in **Step 1** the place holder **P1** is active however transition **T1** is inactive and will not let a token pass through. The transition **T1** requires that both **P3** and **P1** are containing tokens before it can become active.

Step 2 illustrates what happens when Place holders **P1** and **P2** contain tokens and therefore shows that the transition is **active.** Thus a token is allowed to pass through to the following place holder, which in this case is **P2**.

Step 3 shows that **P2** now contains a token and that this has come from **P1** and **P3**. One of the most important things to explain is to why **P3** has not lost its token through the transition as is the case with **P1**, the reason for this being is the directed arc. There are two different types of arc, they are:

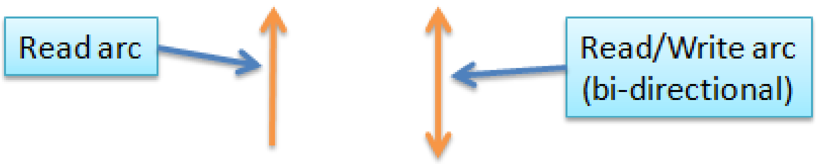


Figure 8 Petri Net Arcs

When a transition is active, it will take whatever comes into it and fire it back out the other side to the new place holder. When this occurs, there is a single read arc that only pulls the token from one side and does not put it back from where it came from. The bi-directional read/write arc is very similar only that the Petri Net is effectively passing the token back along the same arc it came. This two-headed arrow is in fact two arcs going in opposite directions. In the case of **P3** the token is passed from this place holder into the transition **T1** and then few back to **P3** again. This bi-directional allows other systems to not lose their token; in the case of this control system this is imperative as you cannot have a model of a switch that has an inactive and active place holder that actually is neither inactive nor active because the token has been lost to another sub-system.

Step 4 again reiterates the demonstration made in **Step 2** in that now both **P2** and **P4** are active so therefore the transition **T2** is active and will allow the token to pass to **P1**. Note that the transition **T1** is still inactive because it is a logical **AND** situation and requires **P1** to be active before the transition can be made active.

4.2 Modelling Program

Now that Petri Nets have been chosen as a valid method of modelling in this project, we must use a package that is able to develop the Petri Nets.

A software package called WorkCraft is going to be used in this project to develop a working model.

The package drawing wise is similar to what was explained in the previous section however the only difference is the allowance for a simulation protocol. The simulation is very useful to test the model that is developed and find exactly if there are deadlocks in a system or even if a token is misplaced into another subsystem as previously explained.

As can be seen in Figure 9, once in simulation mode active transitions are highlighted blue to show that they are now in fact active and pressing the left mouse button initiates them to fire.

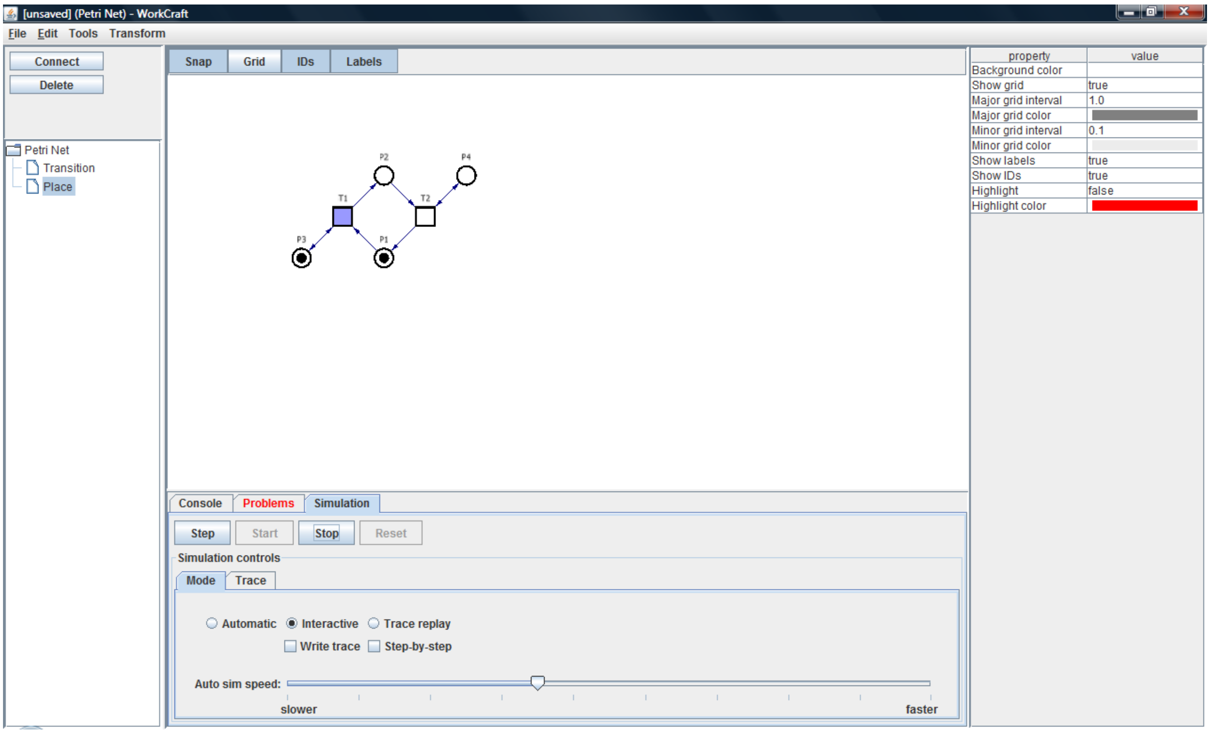


Figure 9 WorkCraft Simulation Example

The benefit of using this package is the factor that transitions that are ready to fire (high-lighted blue) do not actually fire until you click the left mouse button. This interaction with the mouse allows the system to be effectively paused until they have been fully verified that they actually are working and the right conditions are present for the transitions to actually fire.

4.3 Model Designs

The following (4.3.1 Model States, 4.3.2 Occupied and 4.3.3 Bedroom Main Light) are small fractions of the full modelled system. Please see the Appendices B starting on page 73 to see a full model.

4.3.1 Master States

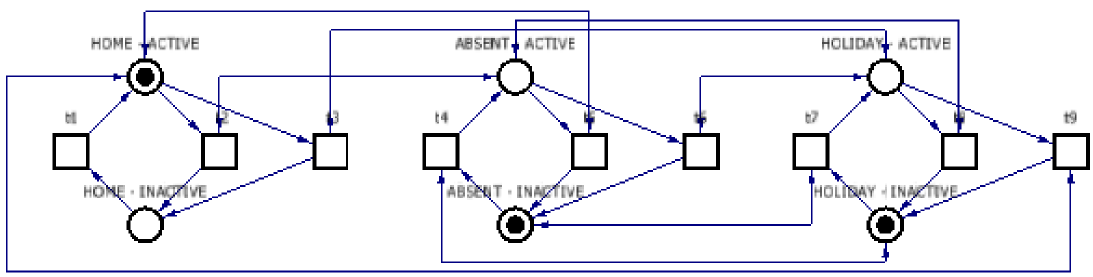


Figure 10 Masters States

This is a very complex model as it is a built sequence that allows interlocks to occur. Initially, there is a token in the **HOME_ACTIVE**. To deactivate **HOME_ACTIVE** and pass the token to **HOME_INACTIVE** requires either **ABSENT_ACTIVE** OR **HOLIDAY_ACTIVE**. Therefore when **ABSENT_ACTIVE** OR **HOLIDAY_ACTIVE** have the token, the transition between **HOME_ACTIVE** and **HOME_INACTIVE** becomes true. There are no pre-requisites for the token to be passed from **HOME_INACTIVE** to **HOME_ACTIVE**, this can therefore happen at any time when the token is in the **HOME_INACTIVE** position. When this is the case and **HOME_ACTIVE** and either **ABSENT_ACTIVE** or **HOLIDAY_ACITVE** is true it can then allow the **HOLIDAY** and **ABSENT** states to go inactive. As can be seen only when **ABSENT_INACTIVE** has the token can the transition between **HOLIDAY_INACTIVE** to **HOLIDAY_ACTIVE** be allowed to fire, this is the same for **HOLIDAY** and the impact it has on **ABSENT**. This interlocking effect stops them being active at the same time. The token must therefore pass through **HOME** to be then activated individually as shown in the functional specification in section 3.1 on page 23.

4.3.2 Bedroom Occupied

The room-occupied system is an internal action and does not reflect to any outside outputs. As can be seen in the model, there are two transitions with wiring from **B_MOVEMENT_SENSOR_ACTIVE** to the first transition and

B_HEAT_SENSOR_ACTIVE to the second transition. This is as per the functional specification and thus means that if any of these states become active i.e. contains the token then the **B_OCCUPIED** becomes active since the transition in either of the two transitions will be allowed to fire. To deactivate the **B_OCCUPIED** and go to the place **B_UNOCCUPIED** then both **B_MOVEMENT_SENSOR_INACTIVE** AND **B_HEAT_SENSOR_ACTIVE** must contain the token.

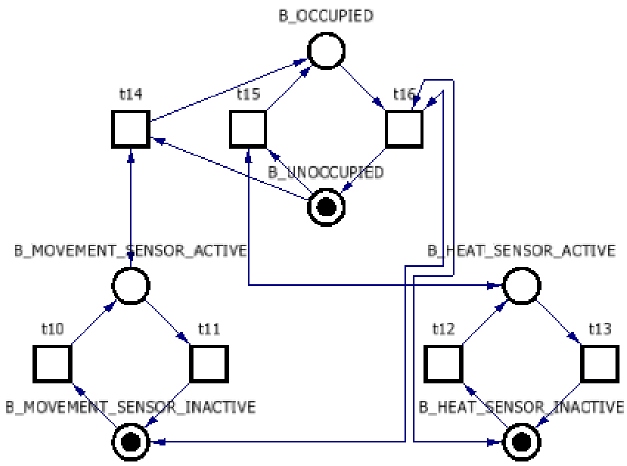


Figure 11 Bedroom Occupied

4.3.3 Bedroom Main Light

Figure 12 Bedroom Main Lightillustrates a single AND to active the light and 4 OR's to deactivate the main light. When **B_OCCUPIED** AND **HOME_ACTIVE** AND **OUTSIDE_DARK** AND **B_ML_SW_ACTIVE** then the transition will fire and allow the token to pass from **B_ML_OFF** to **B_ML_ON**. For the any of the OR transitions and allow the token to pass from **B_ML_ON** to **B_ML_OFF** requires **B_UNOCCUPIED** OR **HOME_INACTIVE** OR **OUTSIDE_LIGHT** OR **B_ML_SW_INACTIVE**.

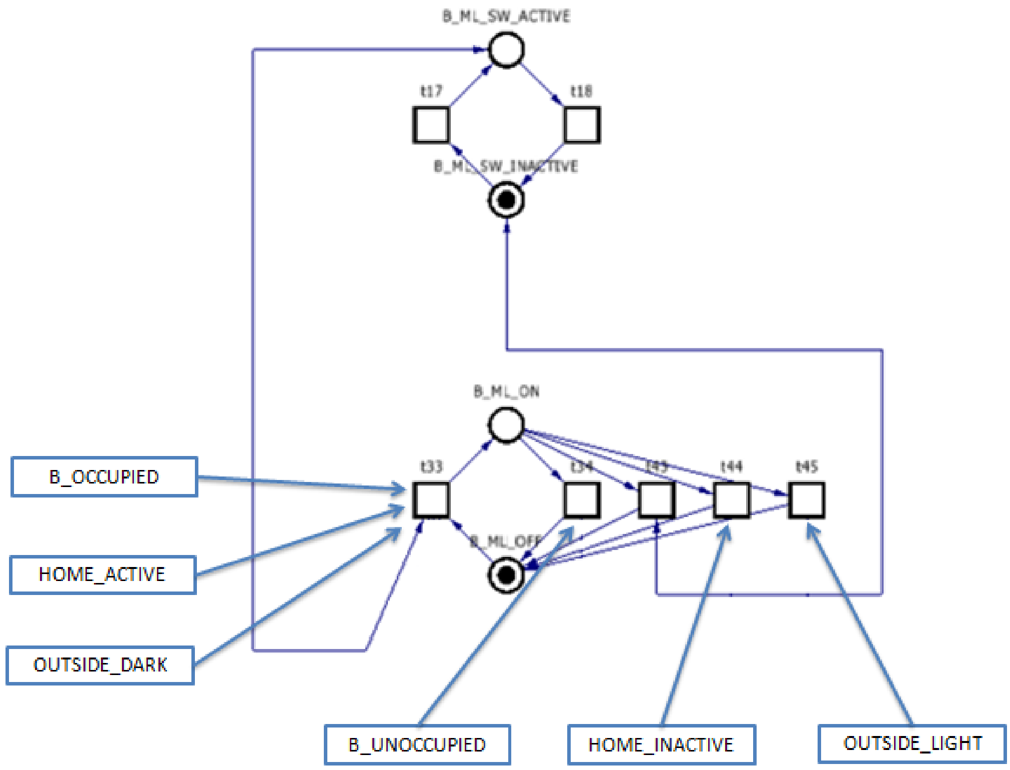


Figure 12 Bedroom Main Light

5.0 Control Implementation

5.1 Ladder Logic

Ladder Logic is a programming language that is represented by a graphical diagram which is originally based on an older method of control called Relay Logic. Relay Logic is automation that is based solely on hardware and electromagnetic relays turning on and off.

The Ladder Logic language is primarily used in PLC's (Programmable Logic Controller) which are extensively in industrial applications. The way ladder logic functions is very simple and has been around for many years.

A PLC has vast amounts of input and output terminals which allows it to interpret or write to via high and low signals. The logic allows for the control of devices that lend themselves to on/off control. Within the world of modern industry it always comes to the point, that if it works why make changes to it; relay logic works so again why change it, from which Ladder Logic was born.

Ladder Logic works by scanning lines called rungs from top to bottom and left to right.

For example in Figure 13 it uses a Normally-Open Contact **IN1** to switch an output coil **OUT1** on. Therefore when **IN1** is active then **OUT1** is active and when **IN1** is inactive then **OUT1** is inactive.

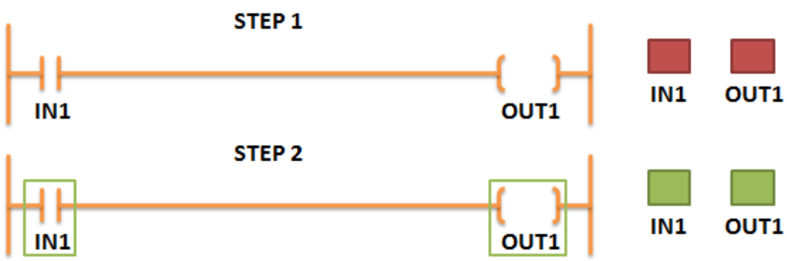


Figure 13 Ladder Logic On/Off (Normally-Open Contact)

Figure 14 is same idea as Figure 13 however this time it uses at Normally-Closed Contact **IN1** to switch the output coil **OUT1** on. Therefore when **IN1** is active then **OUT1** is inactive and when **IN1** is inactive then **OUT1** is active.

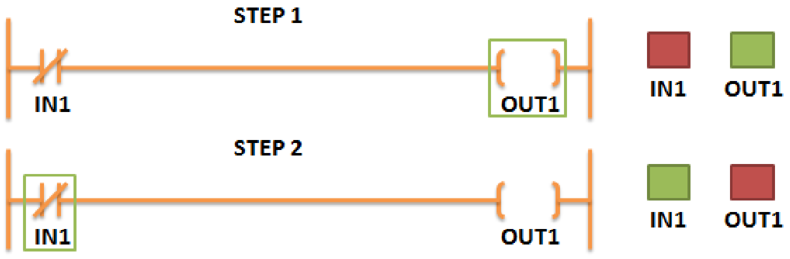


Figure 14 Ladder Logic On/Off (Normally-Closed Contact)

Figure 15 is now implementing some logical functions. In this system there is an OR logic to turn on the **OUT1** coil. It can be said that if either **IN1** OR **IN2** are active then **OUT1** is active. However when **OUT1** needs to be deactived then it requires both **IN1** AND **IN2** to be inactive. So therefore for this we can say having inputs wired in parallel gives rise to an OR statement to turn **OUT1** on and an AND statement to turn **OUT1** off.

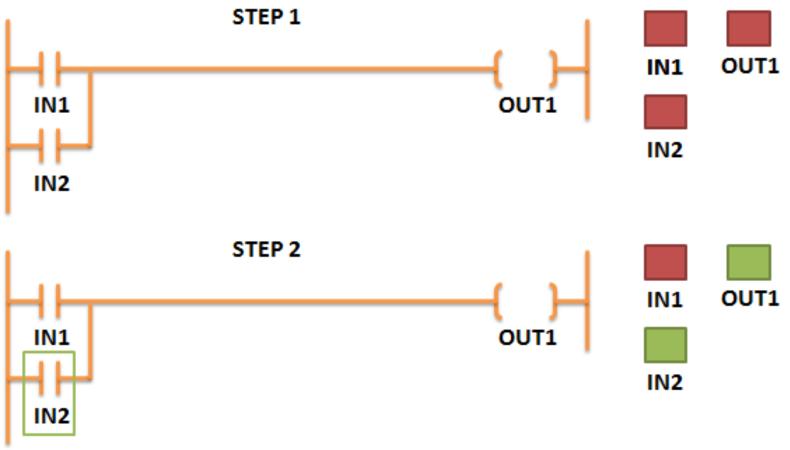


Figure 15 Ladder Logic OR On & AND Off (Normally-Open Contact)

Figure 16 is now implementing and AND logical function to turn on the **OUT1** coil. It can be said that if both **IN1** AND **IN2** are active then **OUT1** is active. However when **OUT1** needs to be deactived then it requires either **IN1** OR **IN2** to be inactive. So therefore having inputs wired in series gives rise to an AND statement to turn **OUT1** on and an OR statement to turn **OUT1** off.

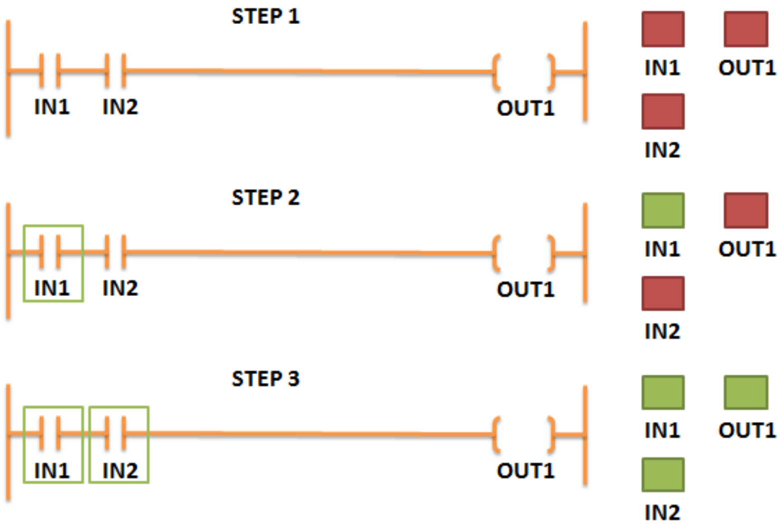


Figure 16 Ladder Logic AND On & OR Off (Normally-Open Contact)

5.2 Control Program

The ladder logic program called TriLOGI and is a free educational ladder logic program produced by Tri-PLC ltd. It has very simple features and allows for easy simulation of inputs and outputs within the personal computing environment.

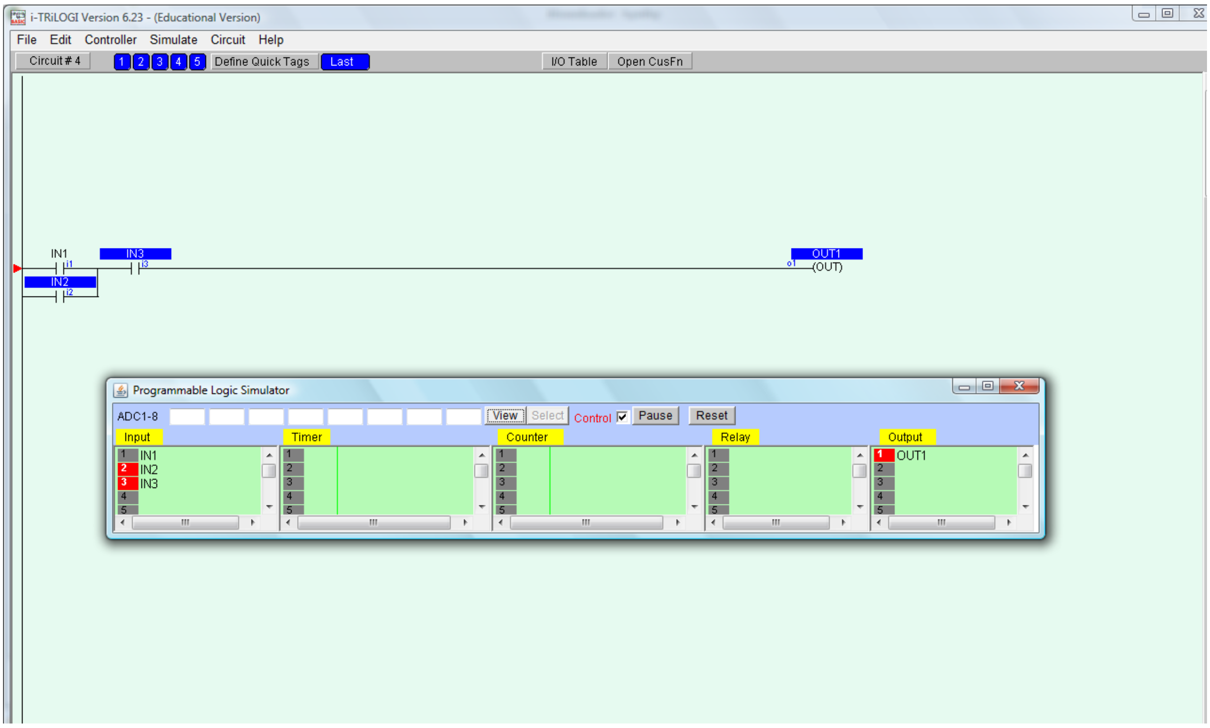


Figure 17 TRiLOGI Simulation Screenshot

As can be seen in Figure 17 the simulation is showing three inputs and one output. The logic reads if either **IN1** OR **IN2** are active AND **IN3** is active then **OUT1** is active. The simulation shows that the **IN2** AND **IN3** are active by the red indication and therefore **OUT1** is showing red illustrating that it is active also.

5.3 Control Programs

The following (5.3.1 Model States, 5.3.2 Occupied and 5.3.3 Bedroom Main Light) are a small fraction of the full control implementation. Please see the

Appendices B starting on page 74 to see the full control system.

5.3.1 Master States

Implementing the Master States using Ladder Logic is more complex and requires the use of a latching input and counters to operate. Since the input switches for the Master States are not constant (Not permanently on when pressed), there need to be a method of latching them on so that when pressed they stay on. Looking in Figure 18, on rung 2 it can be seen that there is an OR situation and an AND situation to activate **HOME** as a state. **INITIALISE** is pressed to start with as there needs to be **HOME** active to have at the start of the system. At the start **HOMEABS** AND **HOMEHOL** will be inactive, so when **INITIALISE** is activated the output to **HOME** is active. Immediately when **HOME** is active it is fed back into the same rung to keep it active, so therefore it stays on permanently until the **HOMEABS** OR **HOMEHOL** become active, this is called a latched output (latching itself on).

The following interim tags depict which direction the Sequence is going as shown in section 3.1 on page 23. Direction is important as it makes sure the correct side of the logic is turned off whether that be going from **HOME** to **ABSENT** and **HOME** needs to be switched off or **ABSENT** to **HOME** and **ABSENT** is needed to be switched off.

Ladder Logic Counters

Counters in this ladder logic program are important and implemented using the up-counter regime. When the up-counter is tripped there is an increment noted and the number on the counter is increased. A max value must be set for the counter, for example in this master state program there is a max value for the counter of 1. All of the counters will initially start at 1, when the counter is at the start the output is inactive, so therefore if all the counters start at 1 then the next time they are activated they are automatically go to 0 and thus the logical output from the counter is active. If the counter value was set to 2 then it would take 2 pulses from an input to make logical output from the counter active.

HOMEABS (Home to Absent)

When **ABSENT** AND **SW_HOME** is active then the circuit is complete and the counter is tripped, therefore **HOMEABS** is now active and the rung for the **HOME** output (rung 2) is disconnected because **HOMEABS** is a normally closed contact therefore when it is active it breaks the circuit and turns off **HOME**.

When **SW_HOME** is pressed for a second time the counter is tripped (i.e. counts up from 0 to 1) and thus the logical output from this Counter is inactive, within this same cycle because **SW_HOME** is active the **HOME** becomes active in rung 2.

<u>HOMEHOL (Home to Holiday</u>

When **HOLIDAY** AND **SW_HOME** is active then the circuit is complete and the counter is tripped, therefore **HOMEHOL** is now active and the rung for the **HOME** output (rung 2) is disconnected. This is because **HOMEHOL** is a normally closed contact, therefore when it is active it breaks the circuit and turns off **HOME**.

When **SW_HOME** is pressed for a second time the counter is tripped (i.e. counts up from 0 to 1) and thus the logical output from this Counter is now inactive, within this same cycle because **SW_HOME** is active the **HOME** becomes active in rung 2.

<u>ABSHOME (Absent to Home)</u>

When **HOME** is active AND **SW_ABSENT** is active AND **HOLIDAY** is inactive then the circuit is complete and the counter is tripped. Therefore, **ABSHOME** is now active and the rung for the **ABSENT** output (rung 5) is disconnected because **ABSHOME** is a normally closed contact therefore when it is active it breaks the circuit and turns off **ABSENT**.

When **SW_ABSENT** is pressed for a second time the counter is tripped (i.e. counts up from 0 to 1) and thus the logical output from this Counter is inactive, within this same cycle because **SW_ABSENT** is active the **ABSENT** becomes active in rung 5.

<u>HOLHOME (Holiday to Home)</u>

When **HOME** is active AND **SW_HOLIDAY** is active AND **ABSENT** is inactive then the circuit is complete and the counter is tripped therefore **HOLHOME** is now active and the rung for the **HOLIDAY** output (rung 7) is disconnected because **HOLHOME** is a normally closed contact therefore when it is active it breaks the circuit and turns off **HOLIDAY**.

When **SW_HOLIDAY** is pressed for a second time the counter is tripped (i.e. counts up from 0 to 1) and thus the logical output from this Counter is inactive, within this same cycle because **SW_HOLIDAY** is active the **HOLIDAY** becomes active in rung 7.

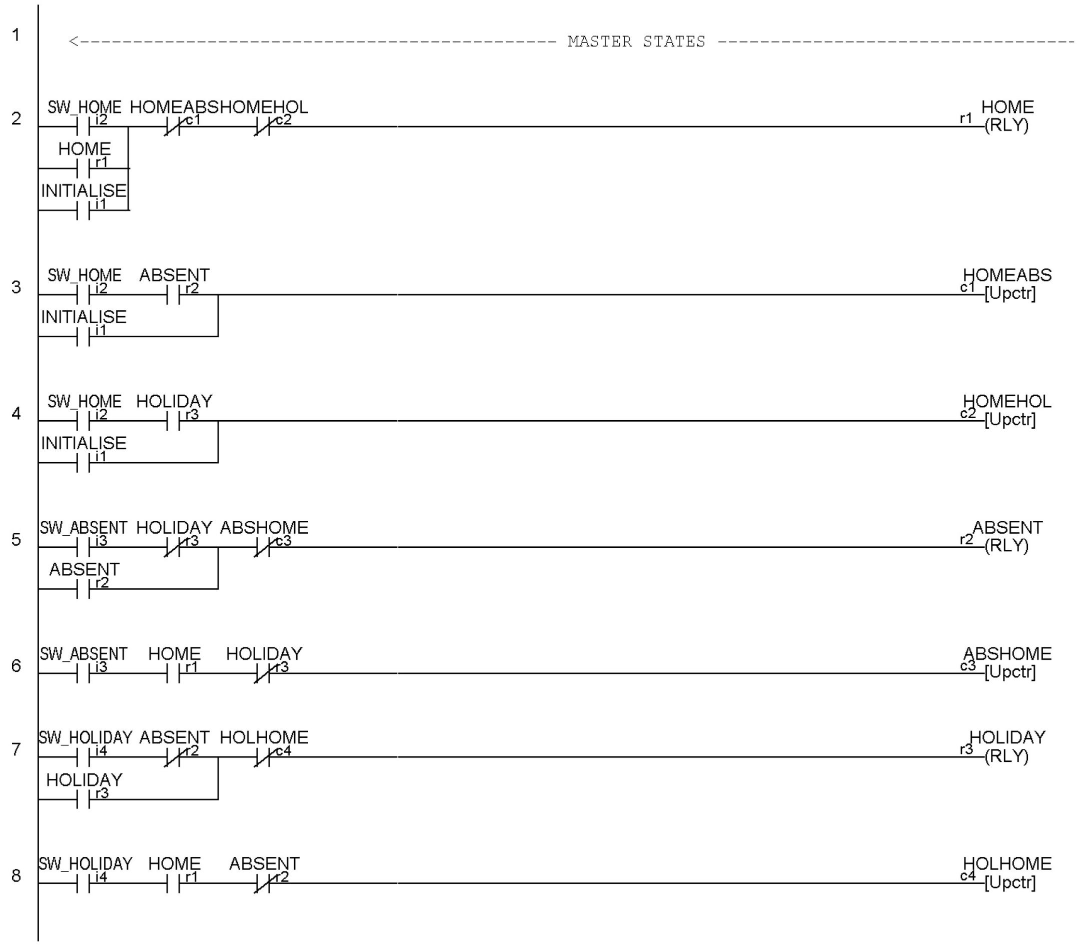


Figure 18 TRiLOGI Master States Control Program

Above the latching inputs were discussed, that same implementation is applied to **ABSENT** and **HOLIDAY** and is where the output is latched back in until one of the directional routine (**HOMEABS**, **HOLHOME** etc.) disconnects the relevant output.

5.3.2 Bedroom Occupied

Figure 19 TRiLOGI Bedroom Occupied Control Program

Figure 19 TRiLOGI Bedroom Occupied Control Program shows the parallel OR statement from **B_HEAT_SEN** OR **B_MOVE_SEN** to activate **B_OCCUPIED**. To deactivate requires the OR above to be replaced by an AND.

5.3.3 Bedroom Main Light

Figure 20 shows the series AND statement from **B_OCCUPIED** AND **HOME** AND **OUT_LIGHT_SEN** AND **B_ML_SW** to activate **B_ML**. To deactivate requires the AND above to be replaced by an OR.

Figure 20 TRiLOGI Bedroom Main Light Control Program

6.0 Methodical Process

The Methodical process is to show how a control system is developed from a model design. The model design is originally implemented on the basis of a written functional specification.

One of the most common logical occurrences in this project is an AND and OR function. For example if we take the **Bed Light** it has the following Functional Specification we get the following.

The bed light is turned **ON** by an AND expression that consists of the following being active to work.

- Bed light switch **Active**
- Home **Active**
- Outside Light Sensor **Dark**
- Room **Occupied**

The Bed light can be turned **OFF** by an OR expression that consists of the following:

- Bed light switch **Inactive**

OR

- Home **Inactive**

OR

- Outside Light Sensor **Light**

OR

- Room **Un-occupied**

From this we are required to generate a model that can correctly represent an AND for activating the bed light and therefore the OR for switching it off.

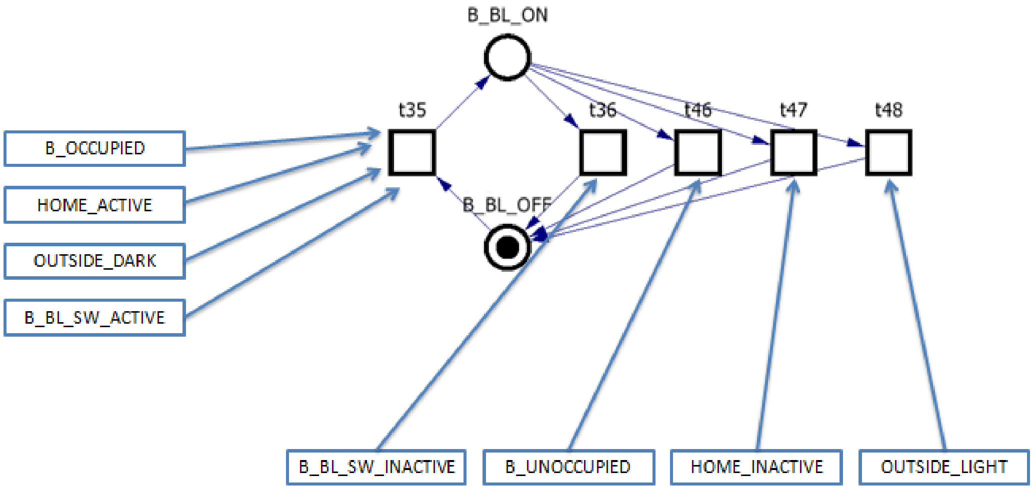


Figure 21 Methodical Process Model from FS

Figure 21 shows the AND logic is represented by all the wires from other systems and states being wired into one single transition to represent the AND logic and the OR is represented by individual transitions representing each individual required condition to turn the bed light off.

Now the model must be converted to a Ladder Logic representation in workable control system. Figure 22 shows the AND situation for turning **B_BL** on (**B_OCCUPIED** AND **HOME** AND **OUT_LIGHT_SEN** AND **B_BL_SW**) and the OR for turning B_BL off (**B_OCCUPIED** OR **HOME** OR **OUT_LIGHT_SEN** OR **B_BL_SW**)

Figure 22 Methodical Process Control from Model

We can therefore produce the following conversion table for the for ladder logic / model conversion:

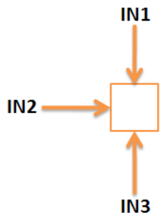
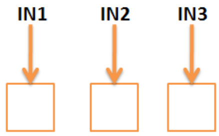
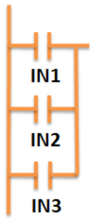
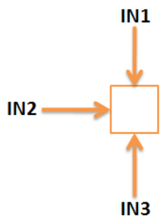
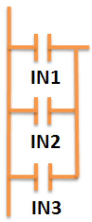
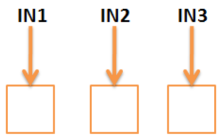
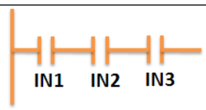
Functional Spec	Model Petri Net	Control Ladder Logic
Activating AND		
Activating OR		
Deactivating AND		
Deactivating OR		

Figure 23 Base Conversion FS to Model to Control

The conversion table in Figure 23 shows how the AND and the OR logic can be represented through an activating AND/OR and deactivating AND/OR, the activating and deactivating is referring to the Ladder Logic turning on and turning off possibilities when AND and OR occurs.

7.0 Conclusion

The demographics of the UK and US populations suggest that there is going to be an impact on the national services that take care of those who are elderly, disabled and unable to cope at home. Smart Homes give the opportunity for the individual to stay in the home for a longer period and thus do not impact the care home service directly. There are obvious factors to take into account, as a Smart Home is sometimes out of the reach to aid some individuals i.e. those with severe mental disorders, sever physical disabilities. However for the majority even some of the more minute tasks become very difficult to do, so therefore an automatic system can provide aid in such situations and make life that little bit easier.

To conclude, this project found new ways of developing a control system for a standard home.

Current markets look at smart homes as a multimedia interface and almost overlook some of the necessities in the home that are required to be automated, whether that be windows, doors, central heating, etc. The most prominent problem is that all these systems cannot be combined together as many of them are running different systems and interfaces. Having one standard technique to implement a Smart Home is unheard of, for which this project solves. This project shows that a bedroom window system can be copied across and implemented into other rooms and other smart homes, the program is meant to be re-used where necessary.

A standardised procedure is important and especially when it can allow hardware developers to further the access to automated systems. Having a valid model to test appropriate systems in a Smart Home is useful as it allows the designer to see exactly what is going on.

Petri Nets are an easy form of modelling and very easy to learn, so people who understand Petri Nets can discuss the systems easily using this common language. It is also able to represent almost all discrete systems very powerfully. During this project, it was found that Petri nets were a very good way of fault-finding, and the task was done effortlessly using them. Some Petri Net tools also allow the easy verification of properties and initiate system analysis and synthesis.

Once the model is finished and fully tested then a control system can be developed. During development of a control strategy in ladder logic, the designer can re-look at the model to confirm or make changes if a problem arises. Ladder is a standard industrial language throughout the automated world, so why not bring it into the home. The Project combines the use of some very popular tools to make sure that Smart Homes can be easily distributed to the masses. By using Ladder logic your saving the cost of developing highly specific code and therefore make it easier to keep costs down. Ladder Logic PLC's are vast, there are many different brands and models out there to suit almost every purpose. Allowing for this meant that were really taking the industrial environment to the home and making something that is reliable in the industrial work place, reliable in the home.

8.0 Evaluation

Looking at the modelling side of things the program itself was a difficult one to overcome. It was very useful to have the program developer within easy access if anything was to go wrong with WorkCraft; however WorkCraft was limited and came with its own problems. Developing the Petri Net to such a large scale caused some problems within the testing procedures of the program. The program has a dead-lock testing tool, however due to the vast program size that was developed for this project, it was unworkable.

Other problem hindered the timeframe, including the instability within Windows Vista.

Many other packages were tested, but all of which were from an educational background and therefore had some bugs, something of which was untrustworthy.

The project has the possibility of being taken further with the expansion of:

- More Rooms
- Extra Smart Home Variables (Washing Machine, Cooker, Fire Alarms, Gas Alarms, etc.)
- Added Continuous control (Implementing PID algorithms within the Ladder Structure, this is useful to control Temperature)
- Full implementation on an actual PLC,
- Investigate new scenarios for operating particular features (Door Opening Routines)

Overall this project ran extremely well given the limitations to Petri Net Development in Software Packages. The Ladder Logic Software was extremely useful to simulate all of the necessary Inputs / Outputs.

9.0 References

[1] Kris Sangani. “Home Automation: It’s no place like home”. Issues in IET Engineering & Technology Magazine, pgs 46 – 38, December 2006.

[2] Vasco Santos, Paulo Bartolomeu, José Fonseca and Alexandre Mota. “B-Live – A home automation system for disabled and elderly people”. Issues in Industrial Embedded Systems, IEEE Conference, pgs 333 – 336, July 2007.

[3] Juing-Huei Su, Chyi-Shyong Lee and Wei-Chen Wu. “The design and implementation of a low-cost and programmable home automation module”. Issues in Consumer Electronics, IEEE Journal, Volume 52, Issue 4, pgs 1239 – 1244, November 2006.

[4] Rune Torbensen. “OHAS: Open Home Automation System”. Issues in Consumer Electronics, IEEE Conference, pgs 1 – 4, April 2008.

[5] Panu Harmo, Tapio Taipalus, Jere Knuuttila, José Vallet and Aarne Halme. “Needs and Solutions – Home automation and service robots for the elderly and disabled”. Issues in Intelligent Robots and Systems, IEEE Conference, pgs 3201 – 3206, August 2005.

[6] Deborah Snoonian. “Smart Buildings”. Issues in IEEE Spectrum Magazine, Volume 40, Issue 8, pgs 18 – 23, August 2003.

[7] Dimitar Stefanov, Bien Zeungnam and Won-Chul Bang. “ The smart house for older persons and persons with disabilities: Structure, Technology, Arrangement and Perspectives”. Issues in Neural Systems and Rehabilitation, IEEE Journal, Volume 12, Issue , pgs 228 – 250, June 2004.

[8] Barry Flax. “Intelligent Buildings”. Issues in IEEE Communications Magazine, Volume 29, Issue 4, pgs 24 – 27, April 1991.

[9] UK National Statistics Website. http://www.statistics.gov.uk/statbase/explorer.asp?CTG=3&SL=4819&D=4912&DCT=32&DT=32#4912 , April 2006.

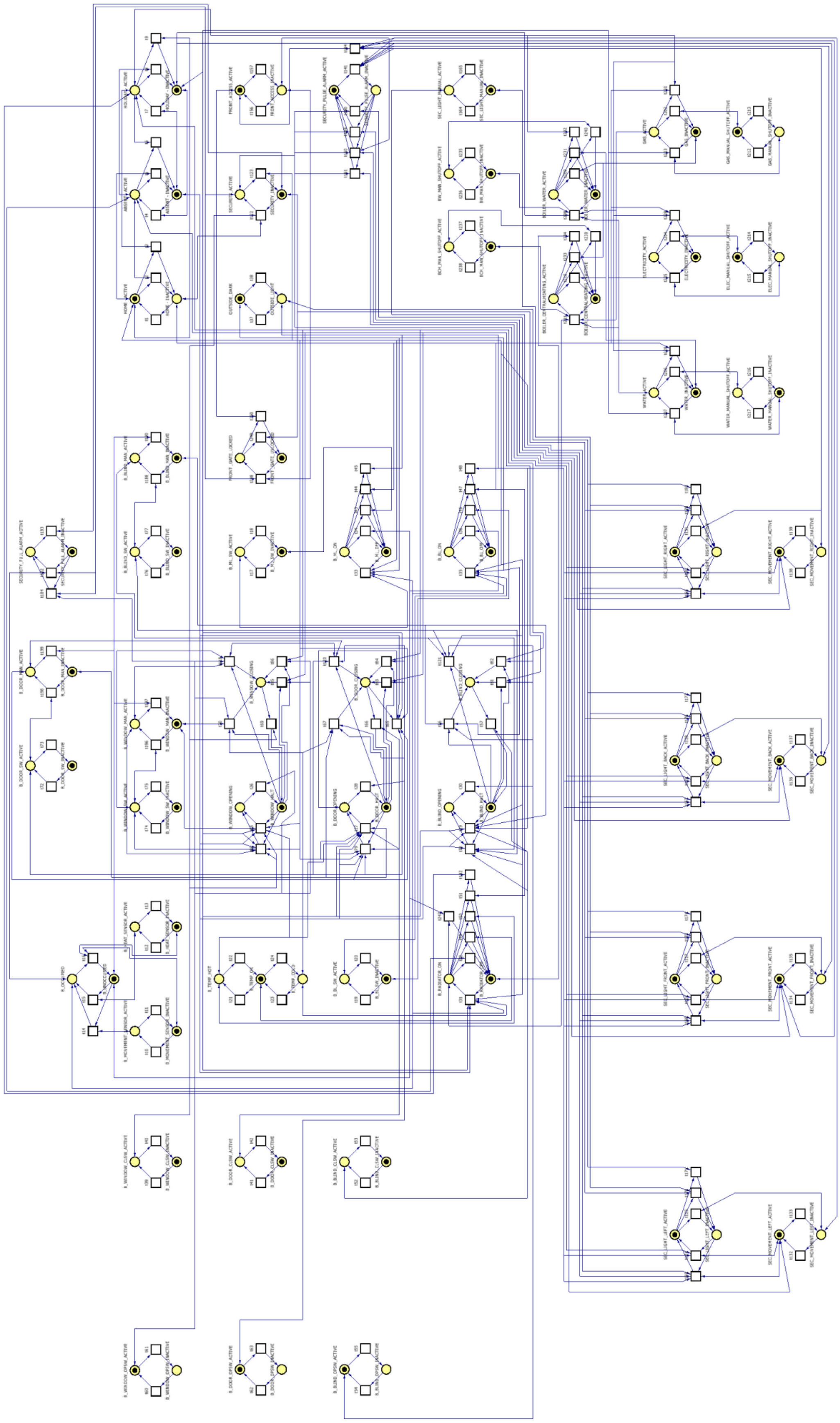
[10] US National Statistics Graphical Reference, http://www.agingstats.gov/agingstatsdotnet/Main_Site/Data/2008_Documents/images/Ind_1a.jpg , 2008.

[11] James Lyle Peterson. Petri Net Theory and the modelling of systems. Prentice Hall, 1981.

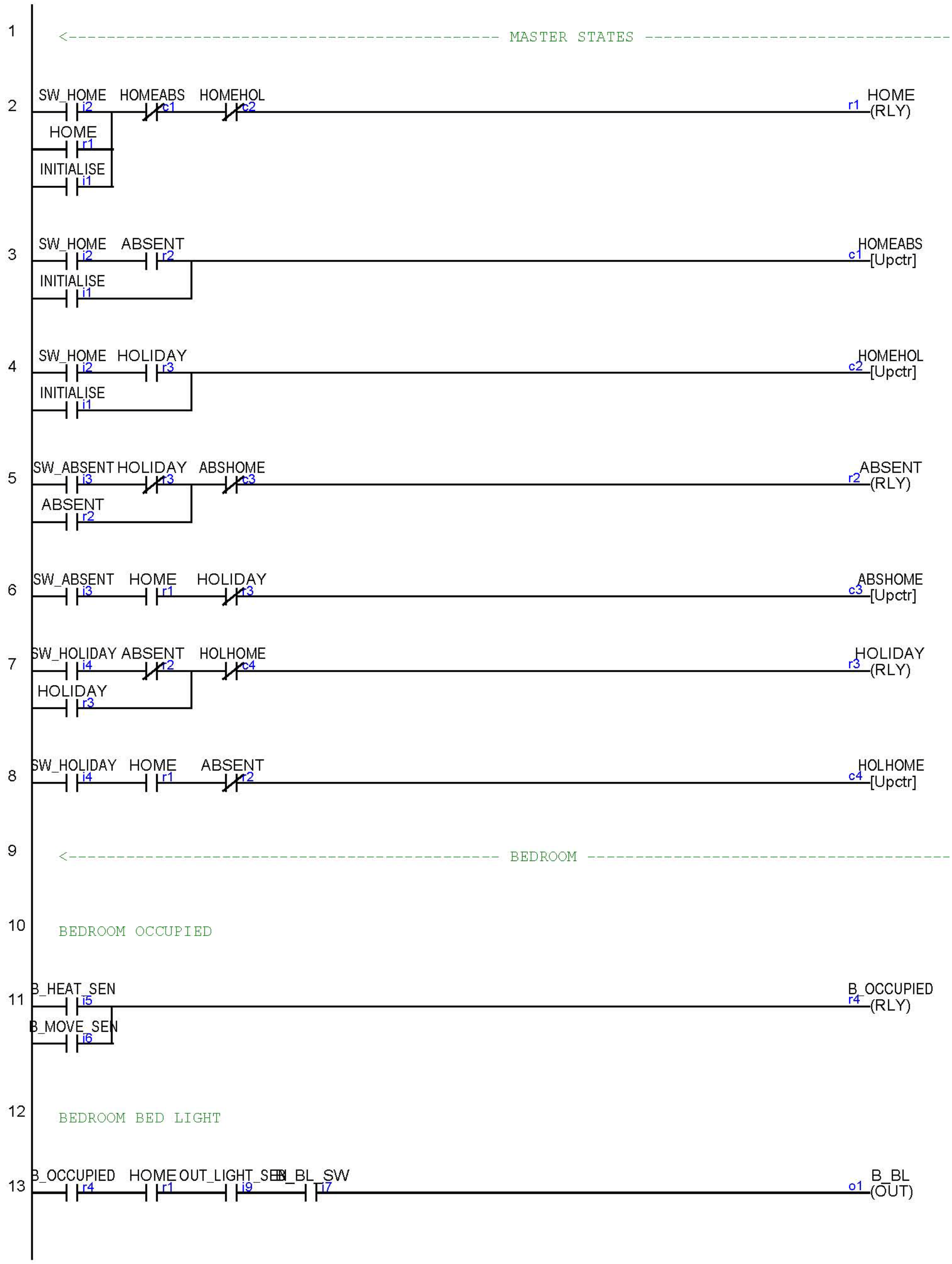
[12] Smart Home Ideas HVAC (Heating Ventilation & Air-Conditioning), http://www.smarthomeideas.com/images/hvac8.gif , 2007.

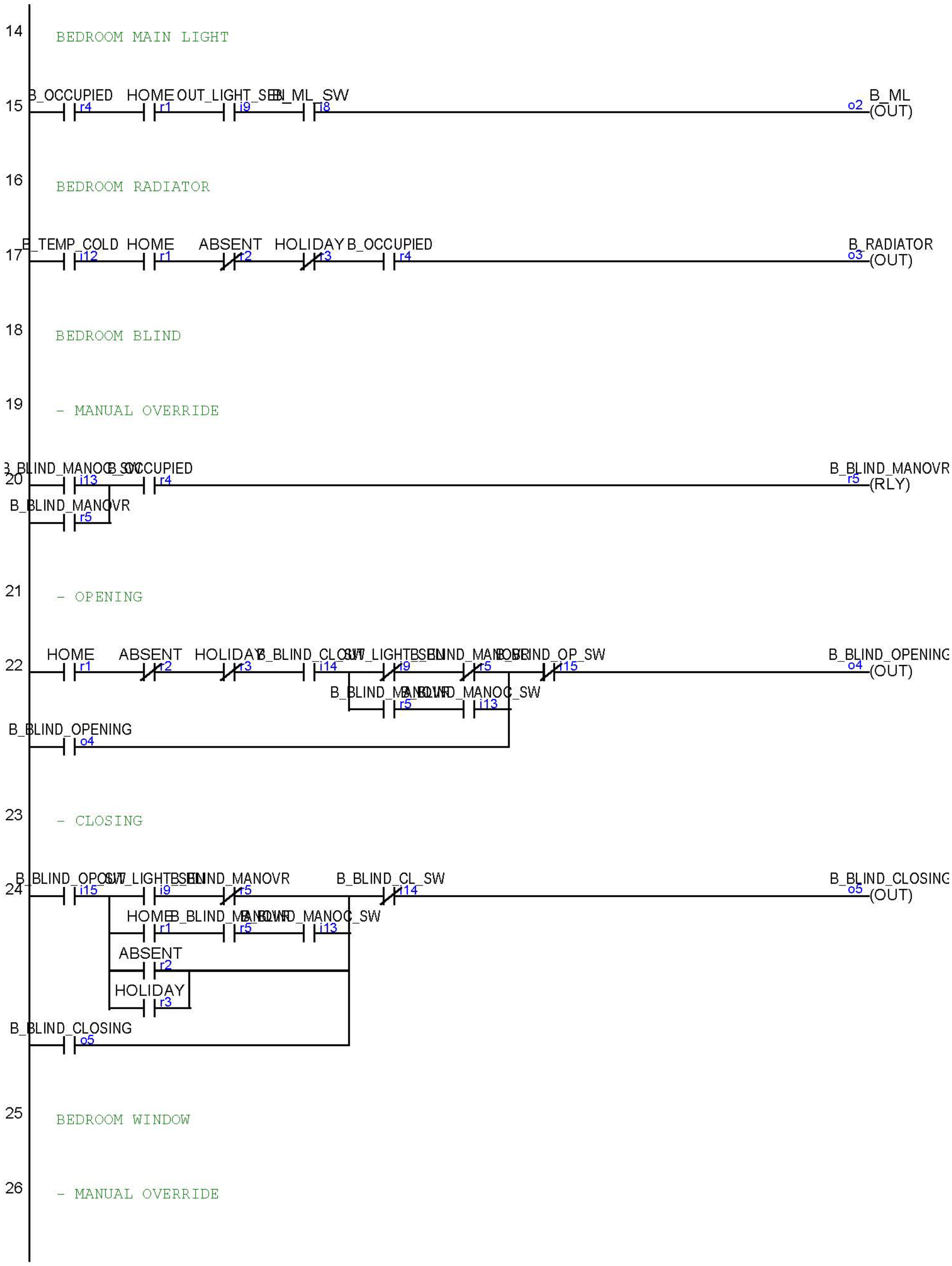
10.0 Appendices A

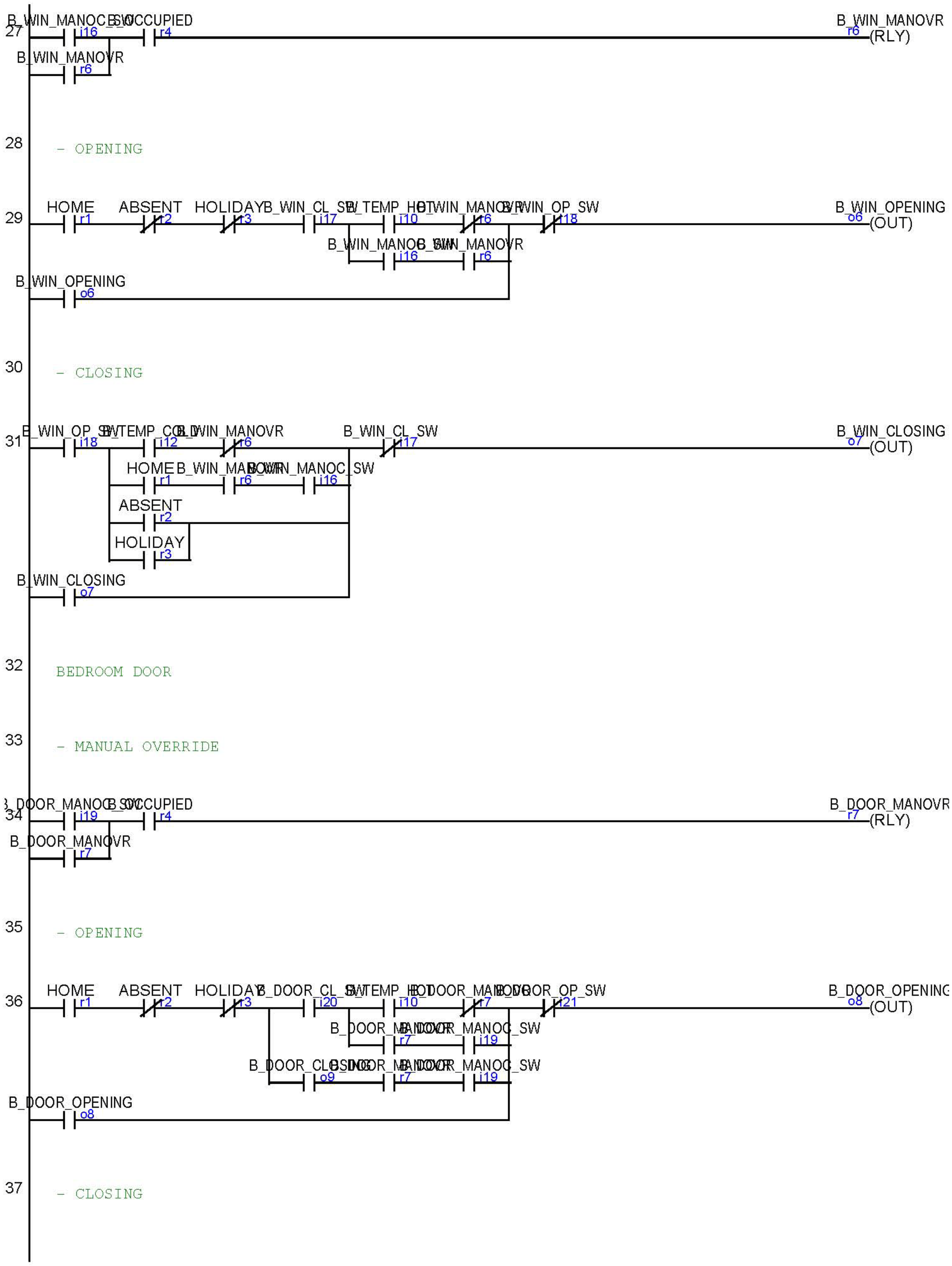
11.0 Appendices B

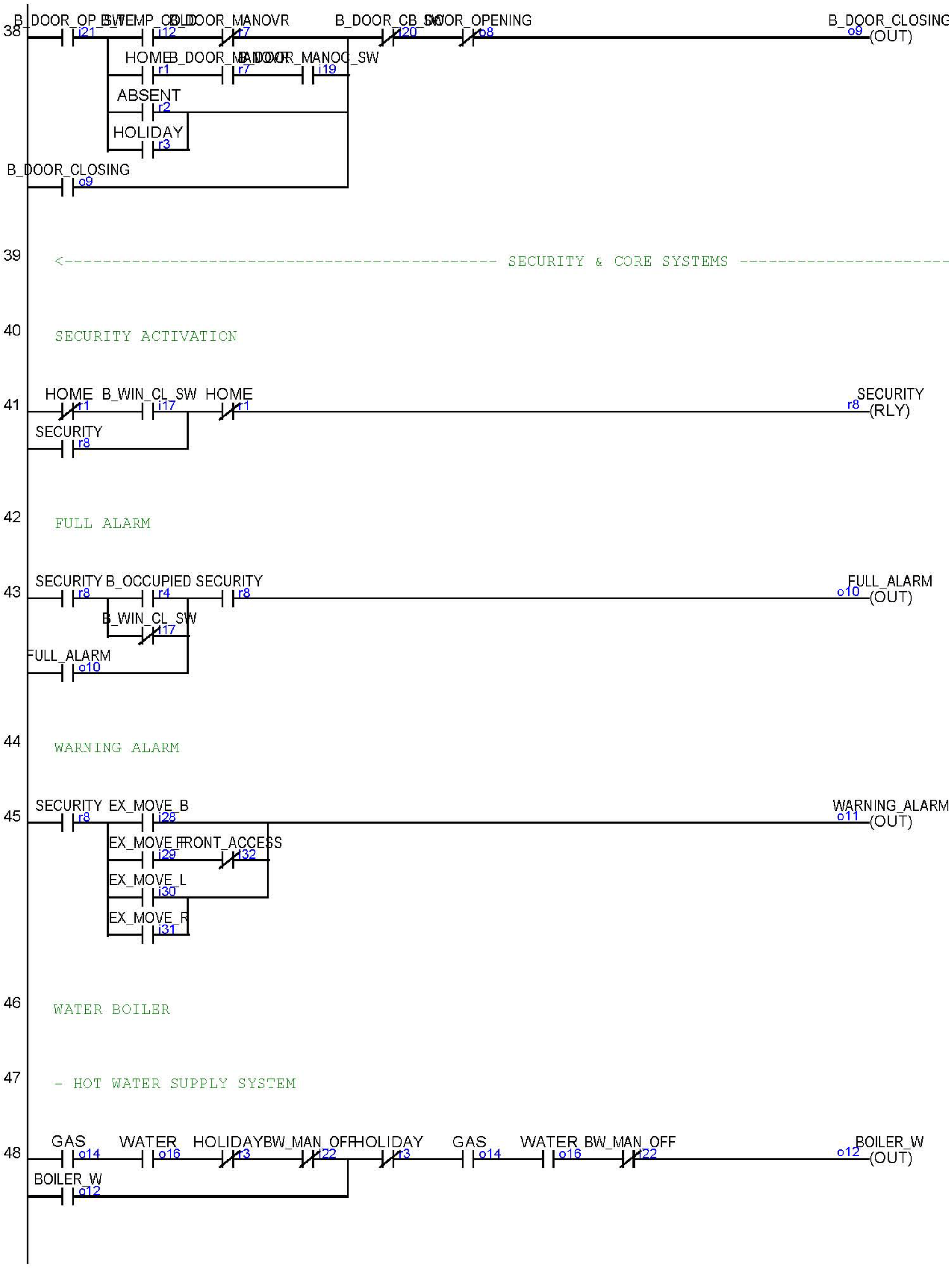


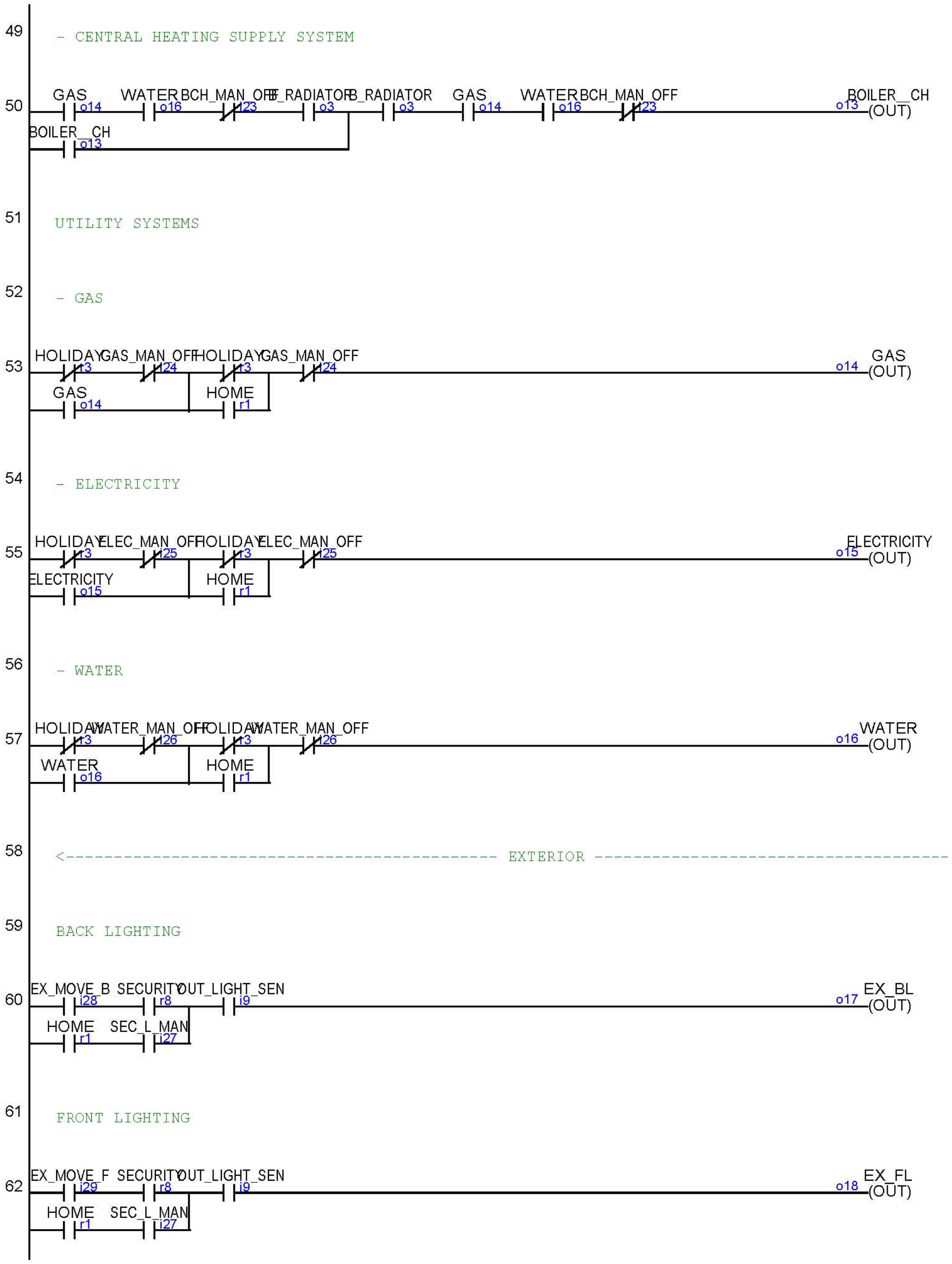
12.0 Appendices C

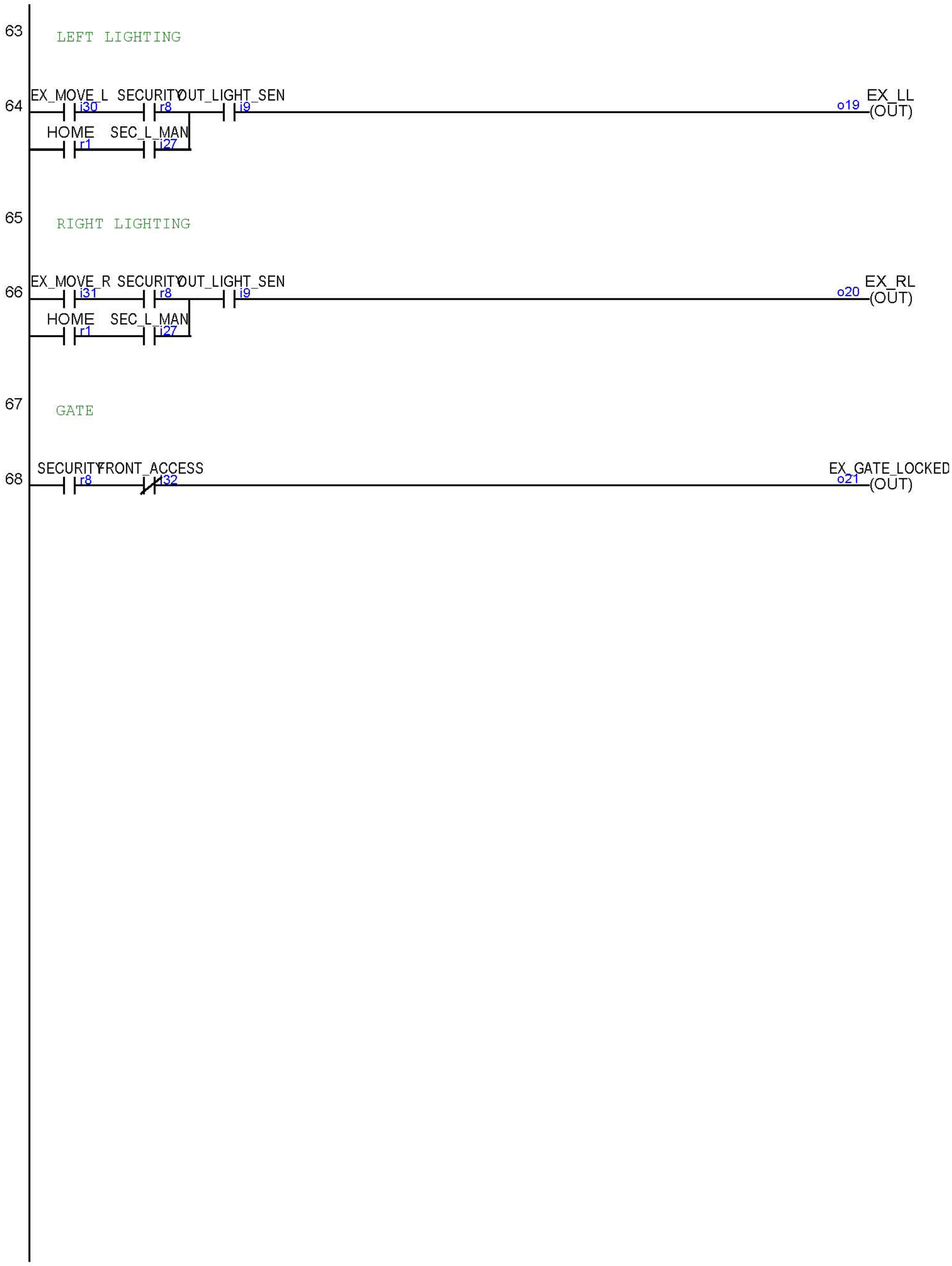



14 BEDROOM MAIN LIGHT
15 B_OCCUPIED r4 HOME r1 OUT_LIGHT_SEN i9 B_ML_SW i8 o2 B_ML (OUT)
16 BEDROOM RADIATOR
17 B_TEMP_COLD i12 HOME r1 ABSENT r2 HOLIDAY r3 B_OCCUPIED r4 o3 B_RADIATOR (OUT)
18 BEDROOM BLIND
19 - MANUAL OVERRIDE
20 B_BLIND_MANOC_SW i13 B_OCCUPIED r4 B_BLIND_MANOVR r5 r5 B_BLIND_MANOVR (RLY)
21 - OPENING
22 HOME r1 ABSENT r2 HOLIDAY r3 B_BLIND_CL_SW i14 OUT_LIGHT_SEN i9 B_BLIND_MANOVR r5 B_BLIND_OP_SW i15 B_BLIND_MANOVR r5 B_BLIND_MANOC_SW i13 B_BLIND_OPENING o4 o4 B_BLIND_OPENING (OUT)
23 - CLOSING
24 B_BLIND_OP_SW i15 OUT_LIGHT_SEN i9 B_BLIND_MANOVR r5 B_BLIND_CL_SW i14 HOME r1 B_BLIND_MANOVR r5 B_BLIND_MANOC_SW i13 ABSENT r2 HOLIDAY r3 B_BLIND_CLOSING o5 o5 B_BLIND_CLOSING (OUT)
25 BEDROOM WINDOW
26 - MANUAL OVERRIDE


27
B_WIN_MANOC_SW
B_OCCUPIED
i16
r4
B_WIN_MANOVR
r6
(RLY)
B_WIN_MANOVR
r6
28
- OPENING
29
HOME
ABSENT
HOLIDAY
B_WIN_CL_SW
B_TEMP_HOT
B_WIN_MANOVR
B_WIN_OP_SW
r1
r2
r3
i17
i10
r6
i18
B_WIN_OPENING
o6
(OUT)
B_WIN_MANOC_SW
B_WIN_MANOVR
i16
r6
B_WIN_OPENING
o6
30
- CLOSING
31
B_WIN_OP_SW
B_TEMP_COLD
B_WIN_MANOVR
B_WIN_CL_SW
i18
i12
r6
i17
B_WIN_CLOSING
o7
(OUT)
HOME
B_WIN_MANOVR
B_WIN_MANOC_SW
r1
r6
i16
ABSENT
r2
HOLIDAY
r3
B_WIN_CLOSING
o7
32
BEDROOM DOOR
33
- MANUAL OVERRIDE
34
B_DOOR_MANOC_SW
B_OCCUPIED
i19
r4
B_DOOR_MANOVR
r7
(RLY)
B_DOOR_MANOVR
r7
35
- OPENING
36
HOME
ABSENT
HOLIDAY
B_DOOR_CL_SW
B_TEMP_HOT
B_DOOR_MANOVR
B_DOOR_OP_SW
r1
r2
r3
i20
i10
r7
i21
B_DOOR_OPENING
o8
(OUT)
B_DOOR_MANOVR
B_DOOR_MANOC_SW
r7
i19
B_DOOR_CLOSING
B_DOOR_MANOVR
B_DOOR_MANOC_SW
o9
r7
i19
B_DOOR_OPENING
o8
37
- CLOSING


38
B_DOOR_OP_SW
i21
B_TEMP_COLD
i12
B_DOOR_MANOVR
r7
B_DOOR_CL_SW
i20
B_DOOR_OPENING
o8
B_DOOR_CLOSING
o9
(OUT)
HOME
r1
B_DOOR_MANOVR
r7
B_DOOR_MANOC_SW
i19
ABSENT
r2
HOLIDAY
r3
B_DOOR_CLOSING
o9
39
<--- SECURITY & CORE SYSTEMS ---------------------
40
SECURITY ACTIVATION
41
HOME
r1
B_WIN_CL_SW
i17
HOME
r1
SECURITY
r8
(RLY)
SECURITY
r8
42
FULL ALARM
43
SECURITY
r8
B_OCCUPIED
r4
SECURITY
r8
FULL_ALARM
o10
(OUT)
B_WIN_CL_SW
i17
FULL_ALARM
o10
44
WARNING ALARM
45
SECURITY
r8
EX_MOVE_B
i28
WARNING_ALARM
o11
(OUT)
EX_MOVE_F
i29
FRONT_ACCESS
i32
EX_MOVE_L
i30
EX_MOVE_R
i31
46
WATER BOILER
47
- HOT WATER SUPPLY SYSTEM
48
GAS
o14
WATER
o16
HOLIDAY
r3
BW_MAN_OFF
i22
HOLIDAY
r3
GAS
o14
WATER
o16
BW_MAN_OFF
i22
BOILER_W
o12
(OUT)
BOILER_W
o12


49 - CENTRAL HEATING SUPPLY SYSTEM
50 GAS o14 WATER o16 BCH_MAN_OFF i23 F_RADIATOR o3 B_RADIATOR o3 GAS o14 WATER o16 BCH_MAN_OFF i23 o13 BOILER__CH (OUT)
BOILER__CH o13
51 UTILITY SYSTEMS
52 - GAS
53 HOLIDAY r3 GAS_MAN_OFF i24 HOLIDAY r3 GAS_MAN_OFF i24 o14 GAS (OUT)
GAS o14 HOME r1
54 - ELECTRICITY
55 HOLIDAY r3 ELEC_MAN_OFF i25 HOLIDAY r3 ELEC_MAN_OFF i25 o15 ELECTRICITY (OUT)
ELECTRICITY o15 HOME r1
56 - WATER
57 HOLIDAY r3 WATER_MAN_OFF i26 HOLIDAY r3 WATER_MAN_OFF i26 o16 WATER (OUT)
WATER o16 HOME r1
58 <-- EXTERIOR ------------------------------------
59 BACK LIGHTING
60 EX_MOVE_B i28 SECURITY r8 OUT_LIGHT_SEN i9 o17 EX_BL (OUT)
HOME r1 SEC_L_MAN i27
61 FRONT LIGHTING
62 EX_MOVE_F i29 SECURITY r8 OUT_LIGHT_SEN i9 o18 EX_FL (OUT)
HOME r1 SEC_L_MAN i27


63
LEFT LIGHTING
64
EX_MOVE_L i30
SECURITY r8
OUT_LIGHT_SEN i9
HOME r1
SEC_L_MAN i27
o19 EX_LL (OUT)
65
RIGHT LIGHTING
66
EX_MOVE_R i31
SECURITY r8
OUT_LIGHT_SEN i9
HOME r1
SEC_L_MAN i27
o20 EX_RL (OUT)
67
GATE
68
SECURITY r8
FRONT_ACCESS i32
o21 EX_GATE_LOCKED (OUT)