

Universidad Complutense de Madrid

Universidad Nacional de Educación a Distancia



Máster en Ingeniería de Sistemas y de Control

**Simulación del Gasto Energético por medio del Modelado
de Sistemas Domóticos en Modelica**

Memoria presentada por

Sandra Viviana Deininger-Salcedo

Bajo la dirección de

Alfonso Urquía Moraleda

Curso Académico 2014/15

Septiembre 2015

Trabajo de fin de Máster

Máster en Ingeniería de Sistemas y de Control

**Simulación del Gasto Energético por medio del Modelado
de Sistemas Domóticos en Modelica**

Proyecto Tipo B

Proyecto específico propuesto por el alumno

Memoria presentada por

Sandra Viviana Deininger-Salcedo

Bajo la dirección de

Alfonso Urquía Moraleda



Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: Sandra Viviana Deininger-Salcedo

A handwritten signature in black ink, appearing to read "Sandra Viviana Deininger-S." The signature is fluid and cursive, with a large, stylized "S" at the beginning.

Resumen

En los últimos años la mayoría de los países Europeos y la industria tecnológica han puesto como objetivo principal el ahorro energético, ya que es significativo no solo en el aspecto financiero, sino que a su vez es ecológico. También el aumento del interés por las viviendas inteligentes por parte de la población y los avances tecnológicos, hacen que las viviendas que se construyen o se renuevan hoy en día, estén en parte implementadas con un sistema domótico. Debido a esto y a la extensa oferta de sistemas domóticos, nace la necesidad de definir criterios que ayuden de manera práctica a elegir un determinado sistema.

La librería DOMOSYS en Modelica contiene 3 package de sistemas domóticos que son Eltako, SBKNX y Wago. Estos representan los sistemas domóticos de Eltako inalámbrico, KNX (TP) y Wago SPS 750, respectivamente. Estos sistemas están actualmente presentes en el mercado y poseen gran demanda. Cada uno de los package de la librería está compuesto por los elementos básicos como son los sensores, actuadores y los elementos que desarrollan la comunicación y el control. Estos elementos varían según el sistema.

Para hacer posible una comparación entre el uso de cada uno de estos sistemas, se ha tomado una vivienda ejemplo, que sería una casa habitada por una familia, compuesta por sala, comedor, cocina, 2 baños, 3 alcobas y terraza. Se ha desarrollado una lista donde se establece los elementos que se van a automatizar.

El objetivo principal del desarrollo de la librería DOMOSYS es poder implementar cada uno de los sistemas en la vivienda ejemplo y observar el gasto energético del sistema en sí. De esta forma poder observar qué sistema tiene más ventajas con respecto al ahorro energético.

Palabras Clave: Ahorro Energético, Simulación de Sistemas domóticos, Modelica.

ABSTRACT

In the last years most of the European countries and the technical industry have set as main goal to save energy as this is not just significant in the financial aspect but also at the same time it is ecologic. Also the increasing interest in intelligent housing by the population and the technical advances in the last years make that the housing that are constructed or renovated nowadays are partly implemented with an home automation system. Because of this and because of the extensive supply of home automation systems arises the need to define criteria which help to chose in a practical way a determined system.

The DOMOSYS library in Modelica contains 3 home automation packages which are Eltako, SBKNX and Wago. They represent the home automation systems from Eltako wireless, KNX (TP) and Wago SPS 750 respectively. These systems are currently on the market and have high demand. Each of theses packages of the library is composed of the basic elements like sensors, actuators and the elements that develop communication and control. These elements vary according to the system.

To be able to make a comparison between the usage of each of this systems, an example housing has been chosen. It can accommodate one family and is composed of a living room, dining room, kitchen, two bathrooms, three alcoves and a terrace. A list has been developed where the elements that will be automated are established.

The principal objective of the development of the DOMOSYS library is to be able to implement each of the systems in the example housing and observe the energetic expenditure of the system per se. Thus it is possible to observe which system has more advantages with reference to energy saving.

Keywords: energy saving, simulation of home automation, Modelica.

Índice

1. Introducción Objetivos y Estructura.....	1
1.1 Introducción.....	1
1.2 Objetivos.....	2
1.3 Estructura.....	3
2. Domótica y Sistemas en la Actualidad.....	4
2.1 Introducción.....	4
2.2 Funciones de la Domótica.....	4
2.3 Medios de Transmisión de Sistemas Domóticos.....	7
2.3.1 Medio de Comunicación por Radiofrecuencia.....	7
2.3.2 433 MHz.....	8
2.3.3 868 MHz.....	8
2.3.4 2.4 GHz.....	10
2.3.5 Medio por Cable	10
2.4 Sistemas Domóticos.....	10
2.5 Topología.....	14
2.6 Sistemas Domóticos en el Mercado.....	17
2.7 Importancia de la Simulación de Sistemas Domóticos	22
2.8 Conclusiones.....	24
3. Librería DOMOSYS y Presentación de la Vivienda Ejemplo.....	26
3.1 Introducción.....	26
3.2 Arquitectura de DOMOSYS.....	26
3.3 Package Eltako.....	27
3.4 Package SBKNX (TP).....	28
3.5 Package Wago	30
3.6 Presentación de la Vivienda Ejemplo.....	31
3.7 Componentes de la Vivienda.....	34
3.8 Conclusiones.....	36
4. Sistema Eltako Inalámbrico	37
4.1 Introducción.....	37
4.2 Descripción del Protocolo.....	37
4.3 Descripción del Sistema Eltako inalámbrico.....	41
4.4 Comunicación entre Sensores y actuadores de el	

Sistema Eltako Inalámbrico.....	42
4.5 Descripción de la comunicación entre los Sensores y los Actuadores.....	42
4.6 Componentes del Package Eltako.....	43
4.6.1 Actuado Actorrele.....	43
4.6.2 Interruptor interruptorpasiv.....	46
4.6.3 Detector de humo Detectorhumo.....	47
4.6.4 Sensor de Temperatura stemperatura.....	48
4.6.5 Sensor de Contactos de Ventanas y Puertas kvp.....	49
4.6.6 Implementación del Sistema Eltako en la Vivienda Ejemplo.....	50
4.7 Conclusiones	63
5.Sistema KNX / EIB (TP).....	64
5.1 Introducción	64
5.2 Comunicación Básica de un Sensor y un Actuador	64
5.2.1 Descripción del Telegrama de Confirmación	68
5.2.2 Descripción del Telegrama de Datos	68
5.3 Descripción de la Librería SBKNX en Modelica.....	73
5.3.1 La fuente o el Alimentador FKNX.....	74
5.3.2 Acoplador de Línea ALKNX.....	76
5.3.3 El Acoplador de Bus ABKNX.....	78
5.4 Desarrollo de la Implementación del Sistema KNX/EIB en la Vivienda Ejemplo.....	80
5.4.1 Sótano.....	80
5.4.2 Primer Piso.....	84
5.4.3 Segundo Piso.....	86
5.4.4 Tercer Piso.....	88
5.5 Conclusiones.....	92
6.Sistema Wago SPS 750.....	93
6.1 Introducción.....	93
6.2 Controlador o Acoplador de Bus	94
6.2.1 Gasto Energético.....	96
6.3 Módulos del Controlador del Bus de Campo.....	97
6.4 División de los Módulos del Controlador de Bus.....	98

6.4.1 Módulos de Entrada Digital	98
6.4.2 Módulos de Entrada Analógica	99
6.4.3 Módulos de Salida Digital	99
6.4.4 Módulos de Salida Analógica.....	99
6.5 Descripción de la Librería Wago en Modelica.....	100
6.5.1 Modulo de Bus mb.....	100
6.5.2 Fuente de Alimentación para el controlador netzpower.....	103
6.5.3 Controlador de Bus cb.....	105
6.5.4 Modulo Final mf	107
6.6 Implementación del Sistema Wago en la Vivienda Ejemplo.....	108
6.6.1 Sótano	108
6.6.2 Primer Piso.....	110
6.6.3 Segundo Piso.....	112
6.6.4 Tercer Piso.....	114
6.7 Sistema Wago en Modelica.....	116
6.8 Conclusiones.....	118
7. Conclusiones y Trabajos Futuros.....	119
7.1 Conclusiones.....	119
7.2 Trabajos Futuros.....	120
Bibliografía.....	122
A) Librería DOMOSYS.....	127

Índice de Figuras

Figura 2.1: Subdivisión del Espacio Espectral de la Banda de 868 MHz (http://funkfernsteuerungindustrie.de/tyropedia/?page_id=13 , [12.01.2011]).....	9
Figura 2.2: Modelo de Referencia OSI (Offnfopt, 2015).....	11
Figura 2.3: Pirámide de los Niveles en la Automatización de edificios y viviendas. (Energiemanagement durch Gebäudeautomation p. 60).	13
Figura 2.4: Topología de Anillo.....	14
Figura 2.5: Topología en Estrella.....	15
Figura 2.6: Topología de Bus.....	15
Figura 2.7: Topología de Árbol.....	16
Figura 2.8: Topología Mixta.....	16
Figura 3.1: Arquitectura de DOMOSYS.....	26
Figura 3.2: Package Eltako.....	27
Figura 3.3: Package SBKNX.....	28
Figura 3.4: Package Wago.....	30
Figura 3.5: Sótano.....	31
Figura 3.6: Primer Piso.....	32
Figura 3.7: Segundo Piso.....	33
Figura 3.8: Tercer Piso.....	34
Figura 4.1: Especificaciones del Protocolo ENOCEAN en el modelo ISO.....	38
Figura 4.2: Estructura del Frame en ENOCEAN.....	39
Figura 4.3: Estructura del Subtelegrama en ENOCEAN.....	39
Figura 4.4 Actuador actorrele del Package Eltako.....	43
Figura 4.5: Diseño de actorrele en Dymola.....	44
Figura 4.6: Interruptorpasiv Package Eltako.....	46
Figura 4.7: Detectorhumo Package eltako.....	47
Figura 4.8: stemeratura Package eltako.....	48
Figura 4.9: Sensor de contactos de puertas y ventanas del Package eltako.....	49

Figura 4.10: Sótano con el Sistema Eltako Inalámbrico	51
Figura 4.11: Representación de la Configuración de los Componentes del Sistema Eltako en el Sótano.....	52
Figura 4.12: Simulación del Sistema Eltako en el Sótano.....	53
Figura 4.13: Primer Piso con el Sistema Eltako Inalámbrico.....	54
Figura 4.14: Representación de la Configuración de los Componentes del Sistema Eltako en el Primer Piso.....	55
Figura 4.15: Resultados de la Simulación del Sistema Eltako en el Primer Piso.....	56
Figura 4.16: Segundo Piso con el Sistema Eltako Inalámbrico	57
Figura 4.17: Representación de la Configuración de los Componentes del Sistema Eltako en el Segundo Piso.....	58
Figura 4.18: Simulación del Gasto Energético Segundo Piso.....	59
Figura 4.19: Tercer Piso con el Sistema Eltako Inalámbrico	60
Figura 4.20: Representación de la Configuración de los Componentes del Sistema Eltako en el Tercer Piso.....	61
Figura 4.21: Simulación del Gasto Energético Tercer Piso.....	62
Figura 4.22: Simulación del Gasto Energético de la Vivienda Ejemplo.....	62
Figura 5.1: Telegrama de Confirmación.....	68
Figura 5.2: Byte del Telegrama de Datos.....	68
Figura 5.3: Definición del Formato.....	69
Figura 5.4: Byte 1 y 2 del Telegrama de Datos.....	70
Figura 5.5: Dirección del Receptor del Telegrama de Datos.....	70
Figura 5.6: Byte 5 del Telegrama de Datos.....	71
Figura 5.7: Byte 6 y 7 del Telegrama de Datos.....	71
Figura 5.8: Fuente de Alimentación KNX/EIB en Modelica.....	74
Figura 5.9: Representación Gráfica en Modelica del Diseño de la Fuente en el Sistema KNX/EIB.....	75
Figura 5.10: Representación del Acoplador de Linea en Modelica.....	76
Figura 5.11: Representación Gráfica en Modelica del Diseño del Acoplador de Linea en el Sistema KNX/EIB.....	77
Figura 5.12: Representación del Acoplador de Bus en Modelica.....	79
Figura 5.13: Sótano con el Sistema KNX.....	81

Figura 5.14: Representación Gráfica de los Componentes del Sistema KNX en el Sótano en Modelica.....	82
Figura 5.15: Primer Piso con Sistema KNX	84
Figura 5.16: Representación Gráfica de los Componentes del Sistema KNX del Primer Piso en Modelica.....	85
Figura 5.17: Segundo Piso con Sistema KNX.....	86
Figura 5.18: Representación Gráfica del Segundo Piso con los Componentes del Sistema KNX en Modelica.....	87
Figura 5.19: Tercer Piso con el Sistema KNX.....	89
Figura 5.20: Representación Gráfica en Modelica del Sistema KNX en el Tercer Piso.....	90
Figura 5.21: Simulación del Gasto Energético de la Vivienda Ejemplo.....	91
Figura 6.1: Representación del Controlador SPS 750 de la Empresa WAGO (www.wago.com).....	96
Figura 6.2: Definición del Módulo de Bus.....	98
Figura 6.3: Representación del Módulo de Bus en Modelica.....	100
Figura 6.4: Esquema Gráfico de la Composición del Módulo de Bus en Modelica.....	101
Figura 6.5: Fuente de Alimentación en Modelica.....	103
Figura 6.6: Esquema gráfico en Modelica de netzpower.....	103
Figura 6.7: Controlador de Bus en Modelica.....	105
Figura 6.8: Representación Gráfica en Modelica de cb.....	106
Figura 6.9: Módulo Final mf.....	107
Figura 6.10: Sistema Wago Sótano.....	109
Figura 6.11: Sistema Wago Primer Piso.....	111
Figura 6.12: Sistema Wago Segundo Piso.....	113
Figura 6.13: Sistema Wago Tercer Piso.....	115
Figura 6.14: Esquema Gráfico de la Implementación del sistema Wago en la Vivienda Ejemplo.....	116
Figura 6.15: Resultado del Gasto de Potencia al Simular en Dymola el Sistema Wago en la Vivienda Ejemplo.....	117

Índice de Tablas

Tabla 2.1: Aspectos del Confort.....	5
Tabla 2.2: Aspectos de la Seguridad.....	6
Tabla 2.3: Aspectos de la visualización y el Ahorro Energético.....	6
Tabla 2.4: Aspectos de la Administración Multimedia.....	7
Tabla 2.5: Especificaciones de las Subdivisiones.....	9
Tabla 2.6: Sistemas Domóticos del mercado actual Alemán (Energiemanagement durch Gebäudeautomation).....	21
Tabla 3.1: Componentes para Automatizar en la Vivienda Ejemplo.....	36
Tabla 4.1: Clasificación de los Telegramas en ENOCEAN.....	40
Tabla 5.1: Definición de la Prioridad del Telegrama de Datos.....	69
Tabla 6.1: Elementos del Sótano a Automatizar con los Respectivos Módulos de Bus.....	108
Tabla 6.2: Elementos del Primer Piso a Automatizar con los Respectivos Módulos de Bus.....	110
Tabla 6.3: Elementos del Segundo Piso a Automatizar con los Respectivos Módulos de Bus.....	112
Tabla 6.4: Elementos del Tercer Piso a Automatizar con los Respectivos Módulos de Bus.....	114

Índice de Código Modelica

Código 4.1: Definición de actorrele en Modelica.....	45
Código 4.2: Definición de interruptorpasiv en Modelica.....	47
Código 4.3: Definición de Detectorhumo en Modelica.....	48
Código 4.4: Definición de stemperatura en Modelica.....	49
Código 4.5: Definición de kvp en Modelica.....	49
Código 4.6: Representación en Modelica del Gasto Energético del Sótano.....	53
Código 4.7: Ecuación del Gasto Energético del Primer Piso.....	56
código 4.8: Ecuación del Gasto Energético del Segundo Piso.....	58
Código 4.9: Ecuación del Gasto Energético del Tercer Piso.....	61
Código 5.1: Definición de FKNX en Modelica.....	76
Código 5.2: Definición del ALKNX en Modelica.....	78
código 5.3: Definición de ABKNX en Modelica.....	80
Código 5.4: Ecuación del Gasto Energético de los Componentes del Sótano en Modelica.....	83
Código 5.5: Ecuación del Gasto Energético Primer Piso.....	85
Código 5.6: Ecuación del Gasto Energético del Segundo Piso en Modelica.....	88
Código 5.7: Ecuación del Gasto Energético del Tercer piso.....	91
Código 6.1: Definición de mb en Modelica.....	102
Código 6.2: Definición de netzpower en Modelica.....	104
Código 6.3: Definición de cb en Modelica.....	106
Código 6.4: Definición de mf en Modelica.....	107
Código 6.5: Cálculo del Gasto de Potencia en la Vivien Ejemplo.....	117

1. Introducción

Objetivos y Estructura

1.1 Introducción

Uno de los motores más importantes en los últimos años para incentivar el desarrollo y comercialización de los sistemas domóticos, ha sido el ahorro energético como fundamental objetivo, ya que trae ventajas en el aspecto económico y ecológico. Debido a esto se reconoce la importancia a la hora de escoger un sistema domótico para una vivienda. El usuario normal se está concienciando acerca de la importancia del ahorro energético, pero le es abstracto los cálculos. Ahí nace la importancia de poder modelar y simular sistemas domóticos, para poder apreciar y establecer las posibles ventajas y desventajas que ofrecen a un determinado usuario un determinado sistema.

La Librería DOMOSYS se ha desarrollado para medir el consumo energético de sistemas domóticos, con énfasis en tres sistemas, que se han escogido porque son sistemas abiertos, tienen diferentes medios de comunicación y la naturaleza de estos tres sistemas es diferente. Así por ejemplo, Eltako es inalámbrico y descentralizado, KNX TP es cableado y descentralizado, Wago SPS 750 es cableado y centralizado. De esta forma se cubre un gran porcentaje de las características de los sistemas presentes en el mercado.

Para el desarrollo de la librería DOMOSYS se estudiaron los componentes de cada uno de los sistemas detalladamente en especial el consumo energético, para poder modelarlos en Modelica y así poder representar el determinado sistema en una vivienda unifamiliar y obte-

ner los resultados del consumo de potencia de cada uno de los sistemas.

1.2 Objetivos

- El Objetivo principal del trabajo fin de máster de informática y sistemas de control, es profundizar en temas de interés profesional y personal, por medio de los conocimientos adquiridos en este máster. Desarrollando el trabajo fin de máster, el cual debe entregar resultados puntuales acerca de un tema específico. Esto se logra en este caso, mediante el conocimiento de sistemas domóticos actuales y aplicando el diseño y simulación de sistemas en el lenguaje Modelica. Por tal motivo el objetivo es desarrollar la librería DOMOSYS, para saber qué sistema domótico de los propuestos, tiene menor consumo energético.
- Conocer la topología y el medio de transmisión de los sistemas domóticos actuales en el mercado.
- Por medio del conocimiento de la topología y el medio de transmisión de los diferentes sistemas domóticos que se encuentren en el mercado, se llega a escoger tres sistemas domóticos que son Eltako inalámbrico (descentralizado y radiofrecuencia), KNX (TP) (descentralizado y cable), Wago SPS 750 (centralizado y cable).
- Conocer de forma detallada los sistemas Eltako inalámbrico, KNX (TP) y Wago SPS 750, para modelar los componentes necesarios, para un funcionamiento lo más parecido a lo real del sistema, desde el punto de vista del gasto energético.
- Escoger los criterios para modelar cada uno de los componentes de los sistemas en el lenguaje Modelica.

- Escoger una vivienda ejemplo donde se puedan implementar los 3 sistemas,
- Observar el gasto energético de cada uno de los sistemas, por medio de la simulación de estos sistemas en la vivienda ejemplo, para así conocer el sistema que menos energía gasta.

1.3 Estructura

El siguiente trabajo está dividido en 7 capítulos.

Capítulo 2. En este capítulo se definen la domótica, cómo está formada, qué sistemas hay en el mercado y si se pueden simular.

Capítulo 3. Se presenta la Librería DOMOSYS y los respectivos package y los componentes.

Capítulo 4. Se presenta el sistema Eltako inalámbrico, se define la librería se explican los componentes y se simula el sistema en la vivienda ejemplo.

Capítulo 5. Se presenta el Sistema KNX (TP), se explica el funcionamiento del mismo. Se explican los componentes de la librería y se simula el sistema en la vivienda ejemplo.

Capítulo 6. Se presenta el sistema Wago SPS 750. Se explica la Librería den Modelica, se presentan los componentes y se observa el gasto energético en la vivienda ejemplo.

Capítulo 7. Conclusiones y trabajos futuros.

ANEXO A. Contiene el código en Modelica de la librería DOMOSYS.

2. Domótica y Sistemas en la Actualidad

2.1 Introducción

La domótica, según el diccionario de la lengua española, es un conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda. En esta rama de la informática y la electrónica se intenta aplicar los avances técnicos al día a día del hogar. Por lo tanto es un campo extenso que cubre innumerables ramas del conocimiento. En este capítulo se definen las funciones de la domótica sus medios de transporte y diferentes topologías, al igual que los principales sistemas domóticos en el mercado actual.

2.2 Funciones de la Domótica

Los sistemas que componen la domótica de una vivienda unifamiliar cubren las funciones de confort, seguridad, administración de la multimedia y visualización del ahorro energético.

En las siguientes Tablas se puede observar los diferentes aspectos de la domótica con las funciones principales y los elementos que desarrollan las funciones.

Objetivo	Elementos Específicos
<ul style="list-style-type: none"> • Programación de encendido automático con determinada intensidad de la luz eléctrica y programación de ambientes. 	<ul style="list-style-type: none"> • Interruptores y sensores para poder manejar la intensidad de la luz y tiempo de encendido. • Control sobre las bombillas o leds.
<ul style="list-style-type: none"> • Control de las sombras de la vivienda de acuerdo al gusto personal, dependiendo de la iluminación solar. 	<ul style="list-style-type: none"> • Sensores de luminosidad y motores para las persianas toldos y cortinas.
<ul style="list-style-type: none"> • Control del ambiente, manteniendo una temperatura adecuada con un determinado grado de humedad. Al igual que el control sobre la temperatura del agua, dependiendo de la época del año y del gusto personal. 	<ul style="list-style-type: none"> • Sensores de contactos, sensores de temperatura, reguladores de la calefacción.

Tabla 2.1: Aspectos del Confort

Objetivo	Elementos Específicos
<ul style="list-style-type: none"> • Detector de movimiento y observar los alrededores de la vivienda y determinados cuartos de la vivienda. 	<ul style="list-style-type: none"> • Sensores de movimientos • Cámaras conectadas a la red.
<ul style="list-style-type: none"> • Detector de ventanas rotas. 	<ul style="list-style-type: none"> • Sensores de vibraciones.
<ul style="list-style-type: none"> • Detector de ventanas y puertas abiertas. 	<ul style="list-style-type: none"> • Sensores de contactos que indican si las ventanas o puertas están abiertos o cerrados.
<p>Detector de Peligros como:</p> <ul style="list-style-type: none"> • Fuego. • Humo. • Hornillas encendidas. • Detector de corto circuito. • Detector de inundación. 	<ul style="list-style-type: none"> • Sensores de fuego, humo y de nivel. • Control sobre los actuadores.

Tabla 2.2: Aspectos de la Seguridad

Objetivo	Elementos Específicos
<ul style="list-style-type: none"> • Control sobre el gasto energético de la vivienda. Por medio de la visualización y control se puede observar el gasto eléctrico, gas, agua. 	<ul style="list-style-type: none"> • Por medio de un contador inteligente se puede observar y manipular el gasto energético.

Tabla 2.3: Aspectos de la visualización y el Ahorro Energético

Objetivo	Elementos Específicos
<ul style="list-style-type: none"> En cualquier momento y lugar de la casa por medio de los dispositivos electrónicos adecuados se puede acceder a la televisión, internet, radio, biblioteca de música, biblioteca de videos, información personal de los integrantes de la vivienda, calendario familiar, libro de noticias y planificación de la semana. 	<ul style="list-style-type: none"> El servidor para acceder a las diferentes bibliotecas y organización del sistema de contenidos. Elementos para comunicarse a la red para acceder a radio o televisión.

Tabla 2.4: Aspectos de la Administración Multimedia

2.3 Medios de Transmisión de Sistemas Domóticos

El medio de transmisión es fundamental en la comunicación de los sistemas, ya que dependiendo de ellos se desarrollan los protocolos de comunicación y se determina la velocidad y el volumen de la transmisión. En las siguientes secciones se explican el medio por radiofrecuencia y el medio por cable.

2.3.1 Medio de Comunicación por Radiofrecuencia

Utilizan el medio ambiente para difundir ondas electromagnéticas. Tiene la ventaja que se puede implementar sobre viviendas ya construidas sin necesidad de romper paredes. La emisión de las ondas funcionan sobre unas frecuencias específicas que son conocidas como las bandas ISM (Industrial, Scientific and Medical). Estas bandas son abiertas y de libre acceso pero tienen normas que deben cumplirse. Como es que, la transmisión de datos debe realizarse en distancias cortas y la potencia de transmisión no debe sobrepasar 1 watt. Las frecuencias de la banda ISM más usadas en domótica en Alemania son 433 MHz, 868 MHz y 2.4 GHz.

2.3.2 433 MHz

El ancho de banda está comprendido entre las frecuencias (433.05 – 434.79) y es de 1.74 MHz, la potencia máxima para transmitir en esta banda es de 10 mW. Es de uso libre, no se necesitan antenas, no tiene límites de tiempo para la transmisión, tampoco hay subdivisión de espacios espectrales, por tal motivo tiene una alta probabilidad de tener problemas de interferencia.

2.3.3 868 MHz

El ancho de banda es de 2 MHz y esta comprendido entre las frecuencias 868 – 870 MHz. Esta frecuencia es libre. Tiene restricciones de espacios espectrales y tiempos de transmisión, lo que la hace más robusta a las interferencias. La máxima potencia de transmisión es de 500 mW y no necesitan antenas.

La Figura 2.1 muestra la subdivisión del espacio spectral entre las frecuencias de 868 MHz y 870 MHz, esta dividido en 8 subbandas, las cuales tienen una potencia máxima de emisión, así por ejemplo la banda 6 , está comprendida en 869,4 y 869,65 MHz y la potencia máxima de emisión de los transmisores en esta banda es de 500 mW.

La tabla 2.1 aclara la función de cada una de estas divisiones, el porcentaje del tiempo de transmisión en una hora y cada subdivisión puede tener un ancho de canal determinado para la transmisión de datos. Por ejemplo la subdivisión 2 esta predefinida para alarmas, el ancho de banda esta entre las frecuencias 868,6 y 868,7 o sea un ancho de banda de 100 KHz, cuando se usa esta subdivisión el ancho de canal de transmisión es de 25 KHz. De esta forma se disminuye el número de posibilidades para que hayan interferencias.

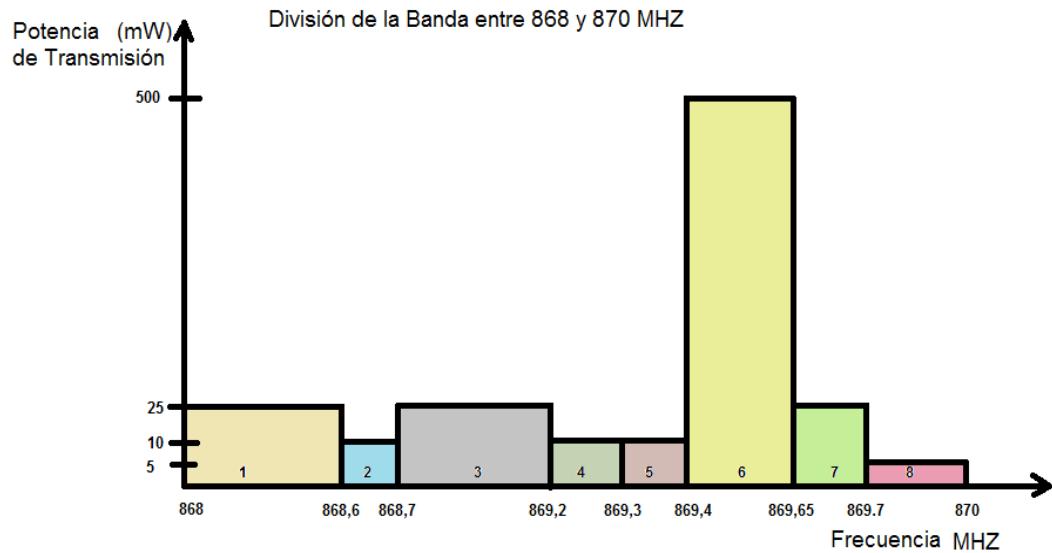


Figura 2.1: Subdivisión del Espacio Espectral de la Banda de 868 MHz (http://funkfernsteuerungindustrie.de/tyropedia/?page_id=13, [12.01.2011])

Subdivisión	Función	Ancho del Canal	Tiempo de Transmisión en 1 hora
1	Para todo uso	Todo el ancho	<1%
2	Alarmas	25 KHz	<0.1%
3	Para todo uso	Todo el ancho	<0.1%
4	Alarmas	25 KHz	<0.1%
5	Abierto	Todo el ancho	Todo el tiempo
6	Para todo uso	25 KHz / Todo	<10%
7	Alarmas	25 KHz	<10% - 100%
8	Para todo uso	Todo el ancho	<10% - 100%

Tabla 2.5: Especificaciones de las Subdivisiones

2.3.4 2.4 GHz

Es una banda que esta entre las frecuencias 2400 MHz – 2483.5 MHz, con un ancho de banda de 83.5 MHz. La máxima potencia de transmisión que esta permitida es de 100 mW. Aquí se pueden usar 80 canales y cada uno tiene un ancho de banda de 1 MHz. Sobre esta banda se han desarrollado estándares como IEEE 802.11 que tiene que ver con la WLAN. También el Bluetooth trabaja sobre esta frecuencia y otras tecnologías como ZigBee.

2.3.5 Medio por Cable

Para la comunicación en red se utilizan diferentes medios como los hilos de cobre llamados par trenzado, el cable coaxial y la fibra óptica. El medio par trenzado es el más utilizado, debido a sus ventajas como el costo, es relativamente fácil de colocar y de conectar con otros elementos. Entre distancias en metros puede transmitir hasta 10 Mbps y en distancias de kilómetros hasta 10 Kbps. Aquí también cabe mencionar el sistema Powerline que utiliza los cables por donde fluye la energía eléctrica para transmitir información, la portadora es 230V 50 Hz.

2.4 Sistemas Domóticos

Desde los años 60 se ha observado el desarrollo paulatino de los sistemas domóticos. En la actualidad nos encontramos con una gran oferta de sistemas debido a los desarrollos en la tecnología, que hacen posible un gran desarrollo en este campo. Cuando se piensa en un estándar en domótica, se cree falsamente que hay un determinado protocolo para la comunicación de los componentes en Domótica. Pero es totalmente lo contrario, hay muchos protocolos algunos abiertos y otros pertenecen a determinadas empresas y se les conoce como cerrados.

Domótica y Sistemas en la Actualidad

Para entender los diferentes sistemas del mercado es importante entender que los sistemas domóticos son sistemas de comunicación que pueden utilizar el modelo de referencia de la ISO conocido como modelo de interconexión de sistemas abiertos (OSI) [Modelo OSI. es.wikipedia.org/wiki/Modelo_OSI. 02.07.2015].

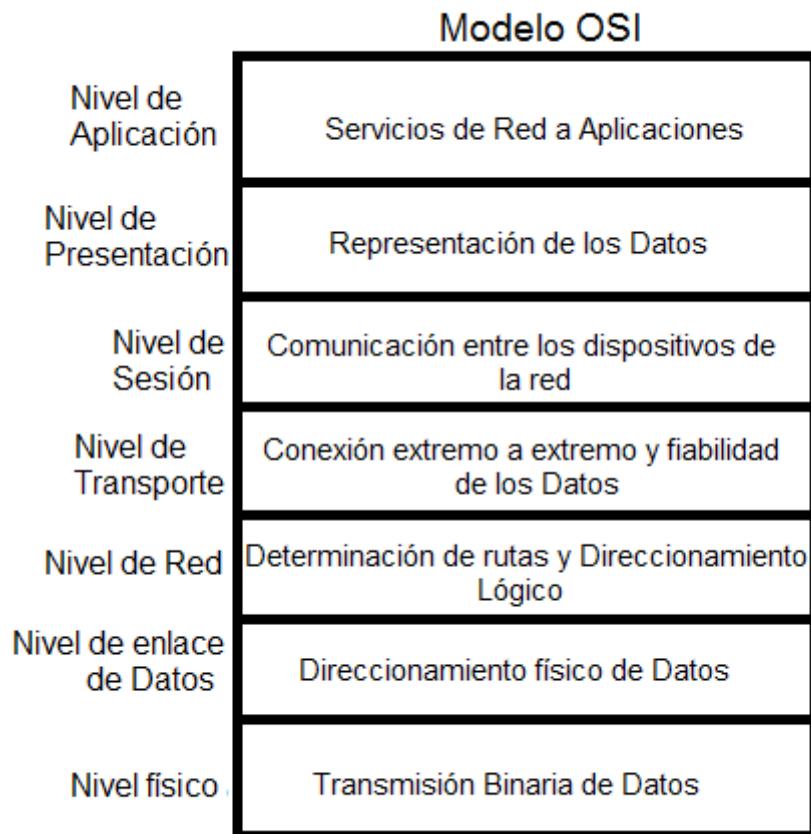


Figura 2.2: Modelo de Referencia OSI (Offnfopt, 2015)

En el nivel físico es donde se decide el medio por el cual se van a transmitir los bits de datos, la velocidad, la dirección como se inicia la comunicación, como se termina.

En el nivel de enlace de datos se segmentan los datos empaquetandolos, se encuentran los algoritmos para encontrar errores en la transmisión.

En el nivel de red aquí se establece cual va a ser la ruta de los datos y se direccionan.

En el nivel de transporte están los protocolos para enviar los datos en la red, así que se segmentan nuevamente.

Nivel de sesión aquí se mantiene la comunicación establecida entre uno o más componentes de la red hasta que la transmisión finalice.

Nivel de presentación se encarga en hacer reconocible los datos que llegaron a un determinado componente.

Nivel de aplicación aquí funcionan los software que manipulan la información y la hacen entendible para el usuario final.

Algunos sistemas domóticos se ciñen al modelo OSI, otros desarrollan sus propios estándares.

Teniendo como base las capas en un intercambio de información entre componentes de un sistema o entre sistemas diferentes, se puede ahora observar como esta compuesta la automatización de un sistema domótico. Se puede expresar como la pirámide de domótica representada en la figura 2.3.



Figura 2.3: Pirámide de los Niveles en la Automatización de edificios y viviendas.
(Energiemanagement durch Gebäudeautomation p. 60)

El nivel de bus de campo es el sistema que va a comunicar los componentes finales que son los sensores y los actuadores. El bus está unido a los componentes del nivel de automatización por medio de las interfaces que funcionan como traductores, puede darse el caso que los componentes del nivel de automatización utilicen otro sistema de bus para comunicarse entre ellos, vuelve y actúa el modelo OSI. Entre el nivel de automatización y el nivel de visualización y control puede también usarse otro sistema de bus que casi siempre es el Ethernet. También se da el caso que casi todos los niveles de la pirámide utilizan el mismo medio de transmisión como es el sistema KNX/EIB (TP), que utiliza el cable par trenzado, en todos menos el nivel de visualización y dirección.

La parte de la pirámide donde se encuentra el sistema de bus es muy importante, ya que aquí se define como va ocurrir el intercambio de

información entre los componentes finales (sensores y actuadores), si es serie o paralelo, si es bidireccional o no, el protocolo que se usa si es abierto o cerrado y la topología de la red.

Los sistemas domóticos que utilizan el Modelo OSI para intercambiar la información se les llama sistemas abiertos, a diferencia de los cerrados que son desarrollados y patentados por una empresa, la cual tiene el monopolio sobre los productos del sistema.

2.5 Topología

La Topología del sistema también ayuda a entender qué clase de sistema se observa, si es centralizado, descentralizado o medio centralizado.

- Anillo: Cada componente se comunica con otros dos componentes. La información viaja por cada componente hasta que llega al destino. La desventaja es que si falla la comunicación en alguno, entonces falla en todos.

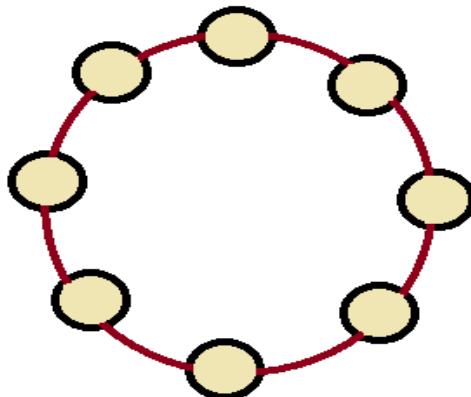


Figura 2.4: Topología de Anillo

- Estrella: Aquí existe un componente central que ejerce el control sobre los otros componentes. El componente central se encuentra

conectado a cada uno de los otros componentes. Esta topología es característica de sistemas centralizados.

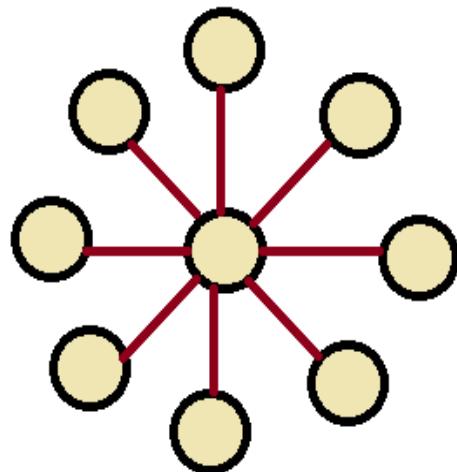


Figura 2.5: Topología en Estrella

- Bus: La información transita por una troncal y todos los componentes se conectan al cable central. Esta topología es característica de sistemas descentralizados. La información transita por el cable central hasta llegar al destino.

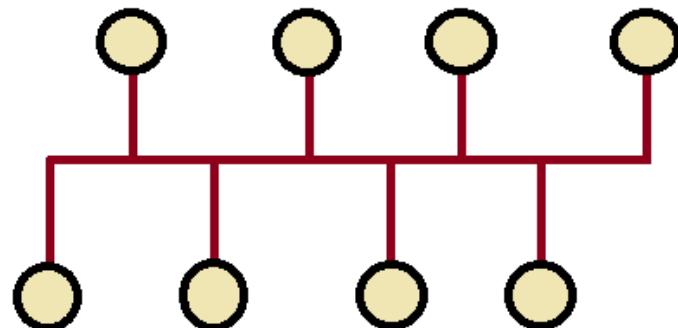


Figura 2.6: Topología de Bus

- Árbol: La información transita por determinados nodos hasta llegar al destino, puede usarse en sistemas centralizados o descentralizados.

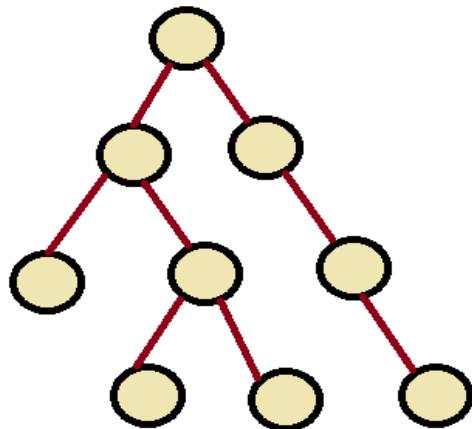


Figura 2.7: Topología de Árbol

- Mixta: Como se puede apreciar en la figura 2.8 es una mezcla de las topologías anteriores.

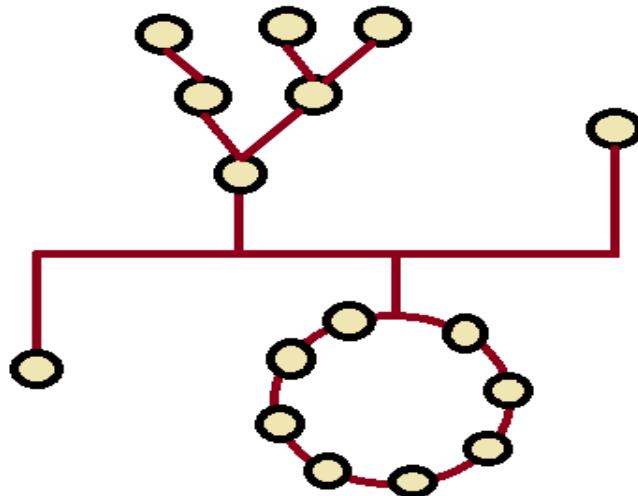


Figura 2.8: Topología Mixta

Según la topología del sistema se puede observar si el sistema es centralizado o descentralizado, así por ejemplo si es en estrella se tiene un sistema centralizado y si es de árbol entonces se observa un sistema descentralizado. Según diferentes autores como Aschendorf, Schneider están de acuerdo al afirmar que, no necesariamente un sistema es mejor que otro por el hecho de ser descentralizado, ya que

Domótica y Sistemas en la Actualidad

puede ser complicado para configurar o para buscar errores. Así que es más importante observar las necesidades de cada usuario.

2.6 Sistemas Domóticos en el Mercado

Sistemas domóticos más sobresalientes en el mercado Alemán según Aschendorf (Energiemanagement durch Gebäudeautomation, 2014).

Nombre del Sistema	Medio de Transmisión			Sistema	Software del sistema, para visualizar y configurar
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección		
Eltako	RF (868) RS485	RS485	RS485 Ethernet	Bidireccional/ unidireccional Abierto centralizado/descentralizado	GFVS IP-Symcon
KNX/EIB Powernet	Red eléctrica	Red eléctrica		Bidireccional descentralizado Abierto	ETS
DigitalStron	Red eléctrica	RS485	WLAN Ethernet	Cerrado Bidireccional	Internet browser IP-Symcon

Sistemas Domóticos en el Mercado

Nombre del Sistema	Medio de Transmisión			Sistema	Software del Sistema para visualizar y configurar.
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección.		
KNX/EIB (TP)	Par trenzado	Par trenzado	Par trenzado	Abierto Bidireccional descentralizado	ETS
LCN	Red eléctrica + 1 hilo	CAT 5 con conector LCN-IV	RS232, USB, TCP/IP, CAT 5.	Cerrado Bidireccional	LCN-PRO LCN-GVS
LON	Par trenzado	Par trenzado	Par trenzado	Abierto Bidireccional descentralizado	Alex Networker
PEHA PHC	RS485	RS485	RS232 RS485 USB	Abierto Bidireccional centralizado	PHC system-software
HomeMatic	RS485 RF	RS485	RS485 RF Ethernet	Cerrado Bidireccional Centralizado	Homeputer IP-SYMCON Tobit David
ELSO IHC	RS485 RF	RS485	RS485 RF	Centralizado	No tiene
Doepke	Espe-	RS485	RS485	Cerrado (Bid-	Software Do-

Domótica y Sistemas en la Actualidad

Nombre del Sistema	Medio de Transmisión			Sistema	Software del Sistema para visualizar y configurar.
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección.		
Dupline	cial para trenzado blindado	RS232	RS232 Ethernet Modbus	Cos) Bidireccional centralizado	epke Proline
Insta 433 MHz	RF (433)	RF (433)	RF (433)	Unidireccional centralizado	No tiene
ELV/FS20	RF (868)	RF (868)	USB Ethernet WLAN	Unidireccional centralizado	Homeputer IP-Symcon
EQ-3 (FS20) mejorado	RF (868)	RF (868) RS485 Ethernet	USB Ethernet IPv6	Centralizado Abierto bidireccional	IP-Symcon (web browser)
EATON xComfort	RF(868)	RF (868)	RS232 USB Ethernet	Central/descentral cerrado bidireccional	EATON-RF EATON APP
Merten Connect	RF (868)	RF(868)	USB	Bidireccional cerrado (z-wave) descentralizado	CONNECT
Z wave	RF	RF (868)	Interfaces	Cerrado	Browser

Sistemas Domóticos en el Mercado

Nombre del Sistema	Medio de Transmisión			Sistema	Software del Sistema para visualizar y configurar.
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección.		
	(868)			Conexion a internet	bidireccional descentralizado web z-web>me IP-Symcon
Eldat de Easywave	RF (868)	RF (868)	USB ethernet	Cerrado bidireccional descentralizado	Ccsoft IP Symcon
ENOCEAN	RF	RF	RF	Bidireccional/U nidireccional Abierto	Según la empresa que implemente el estándar, desarrolla el software.
Siemens S7 300	Profi-bus	Profibus	Profibus Instabus RS232	Bidireccional Abierto centralizado	Step 7
RWE Smart Home					
WAGO 750	Par trenza-doo	Ethernet	Ethernet	Bidireccional Centralizado	Wago-I/O check Wago-I/O

Domótica y Sistemas en la Actualidad

Nombre del Sistema	Medio de Transmisión			Sistema	Software del Sistema para visualizar y configurar.
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección.		
	RS485 RF y otros			Abierto(IEC 61131-3) PLC	PRO
Beckhoff	Diferentes buses.	Ethernet	Ethernet	Bidireccional Centralizado	Twin Cat Twin Cat Management Twin Cat PLC
Eaton Easy	Diferentes buses	Ethernet	Ethernet	Bidireccional descentralizado	
Schneider Electric Zelio	Diferentes buses	Ethernet	Ethernet USB	Bidireccional Centralizado	Zelio Soft 2
DALI sistema para manipular la luz.	Dali bus	Interfaces con sistemas como KNX, WAGO, etc;.	Depende del sistema que se tome	Bidireccional Centralizado	ETS WAGO-I/O PRO
IP-Symcon	La propuesta de empresa es el manipular sistemas domóticos como				

Nombre del Sistema	Medio de Transmisión			Sistema	Software del Sistema para visualizar y configurar.
	Nivel Bus de campo	Nivel de automatización	Nivel de visualización y dirección.		
los vistos por medio de la web, ellos desarrollan las interfaces, tienen los respectivos servidores y ofrecen seguridad.					

Tabla 2.6: Sistemas Domóticos del mercado actual Alemán (Energiemanagement durch Gebäudeautomation)

2.7 Importancia de la Simulación de Sistemas Domóticos

El modelado y simulación de sistemas es parte fundamental para desarrollar y mejorar no solo los componentes de un sistema, sino el sistema en sí.

El uso de un software de un sistema domótico determinado, es prácticamente la simulación y configuración de este. Para así observar, determinar y medir parámetros relevantes. Los cuales son dados por el desarrollador para utilidad del usuario.

Los programas de un sistema domótico pueden ser:

- Para desarrollar el sistema en sí.

Domótica y Sistemas en la Actualidad

- Para configurar el sistema, pueden ser usados por un instalador o el usuario.
- Para simular el sistema por medio de un software determinada y así utilizar el sistema.

En sí la simulación y configuración parte cuando el sistema ya está creado y definido.

La importancia de los programas para simular y configurar sistemas domóticos son:

- Facilitan al instalador y al usuario el uso del sistema, por lo tanto proveen al sistema con la capacidad de permanecer en el mercado.
- Entrega el control al usuario desde los puntos de vista de seguridad y gasto energético.
- Permiten flexibilidad al usuario al usar estos programas tanto dentro como fuera de la vivienda a controlar.

Hay una cantidad considerable de sistemas domóticos en el mercado, que ofrecen un software para configurar y usar el sistema, como se observa en la tabla 2.6. Estos software son de gran importancia porque establecen la relación entre el usuario y el sistema domótico.

También es muy interesante observar lo que propone la empresa IP-Symcon, que propone hacerse cargo del desarrollo de programas para el manejo de una vivienda a distancia. Para esto desarrolla programas para el control a través del Internet de sistemas domóticos ya establecidos en el mercado.

Lo que hacen es por medio de interfaces que ellos mismos desarrollan, se comunican con la parte de nivel de automatización (figura 2.3) de un sistema domótico determinado. Haciendo posible la comunicación con una determinada vivienda a través de internet y de forma segura, ya que ellos velarían por la seguridad de los servidores. Esta empresa ofrece en si, el manejo de la vivienda a distancia

Importancia de la Simulación de Sistemas Domóticos

con el apoyo de las empresas con los sistemas domóticos ya establecidos.

El gran interés de la simulación y configuración de sistemas domóticos se centra en la capacidad de manejar y controlar el sistema, por tal motivo los desarrollan los propietarios. Pero en la investigación bibliográfica no se observa programas de terceros que de forma abierta desarrollen la simulación de un determinado sistema.

2.8 Conclusiones

En este capítulo se puede observar cómo está formado un sistema domótico por medio de la pirámide de automatización. Aquí se observa que cada nivel puede tener diferentes medios de transmisión, siendo el medio y el estándar de la comunicación en el nivel de bus de campo el que caracteriza al sistema domótico. Por ejemplo hay sistemas domóticos por radiofrecuencia en el nivel de bus de campo, pero en el nivel de automatización se comunica por medio de cables e interfaces. A estos sistemas se les conoce sistemas domóticos por radio frecuencia.

El modelado y la respectiva simulación de sistemas domóticos cobra cada día más importancia, por tal motivo el auge de programas que manipulan estos sistemas. Hoy en día en el mercado las empresas que ofrecen sistemas domóticos completos, ofrecen programas desde la configuración del sistema hasta el manejo del sistema por medio de una aplicación en el teléfono inteligente. Por tal motivo el diseño y planificación de un sistema en una vivienda es muy importante. Pero no hay un programa, que ayude al usuario a definirse por un sistema determinado, ya que consistiría en tener todos los sistemas en un programa, lo cual es de poco interés para estas empresas.

3. Librería DOMOSYS y Presentación de la Vivienda Ejemplo

3.1 Introducción

En este capítulo se presenta la Arquitectura de la librería DOMOSYS, con sus respectivos package. Por tal motivo se han escogido 3 sistemas domóticos del mercado, que son Eltako, KNX (TP), Wago SPS 750. Estos package contienen los componentes básicos de cada sistema, al igual que el sistema configurado en la vivienda ejemplo. Los motivos por los cuales se han escogido estos sistemas es que son bidireccionales, con medio de transmisión muy comunes, además la comunicación con los componentes de nivel del bus de campo es abierta.

3.1

3.2 Arquitectura de DOMOSYS

DOMOSYS está compuesta de tres package que son Eltako, SBKNX y Wago. Cada package contiene los componentes básicos de cada sistema, poniendo énfasis en los sensores, actuadores y los componentes del nivel de automatización. El objetivo de la librería DOMOSYS es agrupar todos los posibles sistemas domóticos que se encuentran en el mercado con sus respectivos componentes que hacen posible la implementación del sistema en una vivienda. Por medio de la imple-

mentación observar el gasto energético de estos sistemas y poder observar qué sistema trae más ventajas desde este punto de vista.

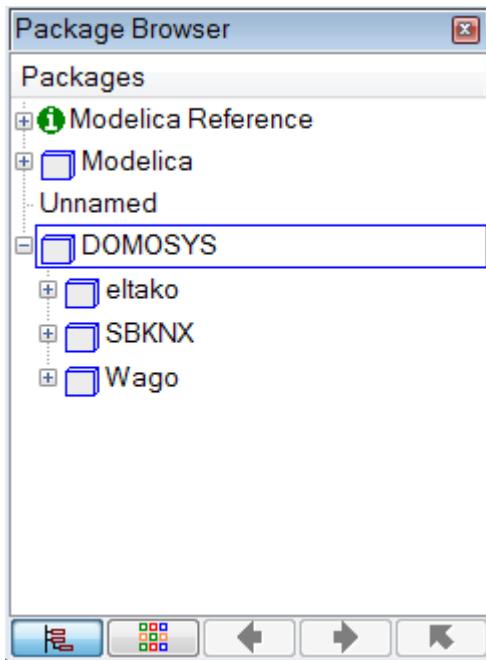


Figura 3.1: Arquitectura de DOMOSYS

3.3 Package Eltako

DOMOSYS es la librería principal, eltako es un package que está compuesto por los componentes básicos del sistema Eltako inalámbrico, véase la figura 3.1. Eltako inalámbrico en el nivel de bus de campo, el medio de transmisión es por radiofrecuencia. Los sensores emplean la tecnología de EnOcean, ya que tienen la capacidad de utilizar la energía del medio ambiente para enviar el mensaje al respectivo actor/actores. El actor es el mismo para todos los componentes de la casa, lo que cambia son las conexiones. Los sensores son interruptor pasiv, un interruptor para encender, apagar o regular (dimmen) un actor. El stemperatura, es un sensor de temperatura, El Detec-trohumo, que es un detector de humo como su nombre lo indica y por último tenemos el sensor de contacto de las ventanas y puertas llamado kvp. El actuador es el actorrele y es el mismo para todos, ya que la parte básica del actuador es igual para todos los actuadores de elta-

ko, esta compuesto por un relé, el actuador tiene un modulo para recibir y enviar señales al sensor correspondiente y a la central que puede ser el servidor. De esta forma ocurre la comunicación y se puede desarrollar la acción necesaria. Para manipular el sistema a distancia por medio de un teléfono inteligente se necesitan tener el servidor. El package vivienda esta compuesto por cuatro modelos que corresponden a cada piso de la vivienda. En cada piso de la vivienda se implemento el sistema, dando como resultado el modelo ge que contiene el gasto energético de todos los pisos de la vivienda más el gasto energético del servidor

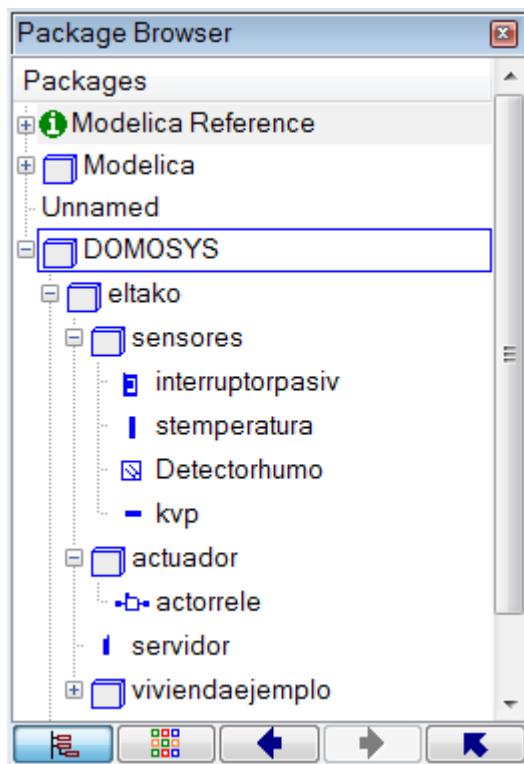


Figura 3.2: Package Eltako

3.4 Package SBKNX (TP)

Este package DOMOSYS, contiene los componentes básicos para implementar el sistema KNX (TP), véase la figura 3.2. El sistema básico KNX está formado por la fuente que es la que alimenta las líneas del sistema, dependiendo del número de líneas entonces se determina

Librería DOMOSYS y Presentación de la Vivienda Ejemplo

el número de fuentes. Cada linea está separada del resto del sistema por medio de un acoplador de línea que tiene a su vez la función de repetidor de los mensajes de entrada y salida. El acoplador de bus que forma parte de todos los sensores y actuadores, en sí es el lugar donde se originan las ordenes o se decodifican las ordenes recibidas. Todo acoplador de Bus tiene un módulo de aplicación que puede ser así por ejemplo, un sensor de temperatura o de movimiento o un actuador como un regulador de luz. Todos los componentes del nivel de bus de campo tienen como base un acoplador de bus, lo que varía es el módulo de aplicación. Por tal motivo no se dividieron los componente del bus de campo como sensores y actuadores. La carpeta vivienda que está dentro del sistema SBKNX contiene la implementación de este sistema en la vivienda ejemplo. Ésta está dividida en sótano KNXsotano, primer piso PPKNX, segundo piso SPKNX, tercer piso TPKNX. También contiene dentro de la carpeta vivienda un modelo que contiene el gasto total energético de todo el sistema implementado en la vivienda y se llama gt.

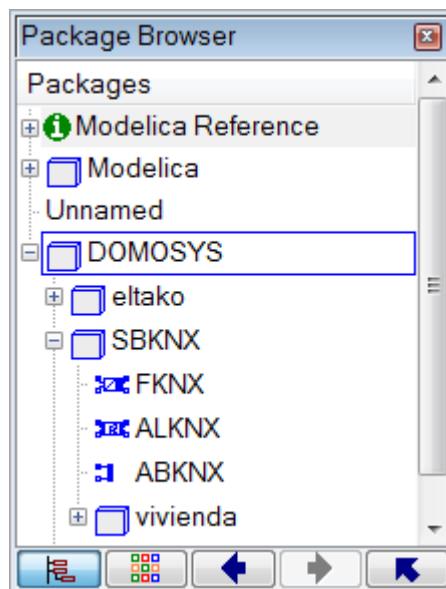


Figura 3.3: Package SBKNX

3.5 Package Wago

Este sistema a simular es un Controlador lógico programable dividido en módulos. Las normas para programar este sistema están definidas en la norma IEC61131-3. Los componentes básicos para formar la domótica en una vivienda son la fuente que convierte el voltaje de 230 V a en 24 V, este sirve para alimentar el controlador de bus y algunos módulos de bus de entrada y salida. El controlador de bus que es el corazón del sistema. Aquí se guardan las variables de cada uno de los módulos y los respectivos programas que pueden funcionar todo el tiempo o cuando alguna variable tome cierto valor, en sí depende todo del programa. Los módulos del controlador son como el traductor, ya que toman la información del sensor y se la entregan al controlador de forma tal que reconozca las variables y los programas unidos a ellas. También toma las ordenes del controlador y la lleva a los actuadores de manera que se realice la función esperada. Los módulos de bus se dividen en módulos de entrada y módulos de salida. Los módulos de entrada es donde están conectados los sensores y los módulos de salida es donde están conectados los actuadores, véase figura 3.4.

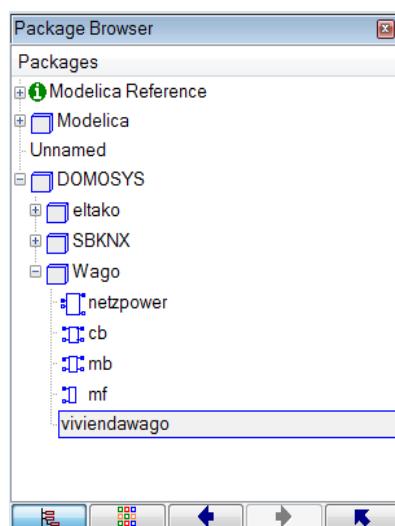


Figura 3.4: Package Wago

3.6 Presentación de la Vivienda Ejemplo

Las casas unifamiliares están compuestas en la mayoría de los casos por sótano, primer piso, segundo piso y pueden tener tercer piso. Para el desarrollo de este trabajo se toma como muestra una vivienda compuesta de 3 pisos con sótano. Como se observan en los siguientes gráficos.

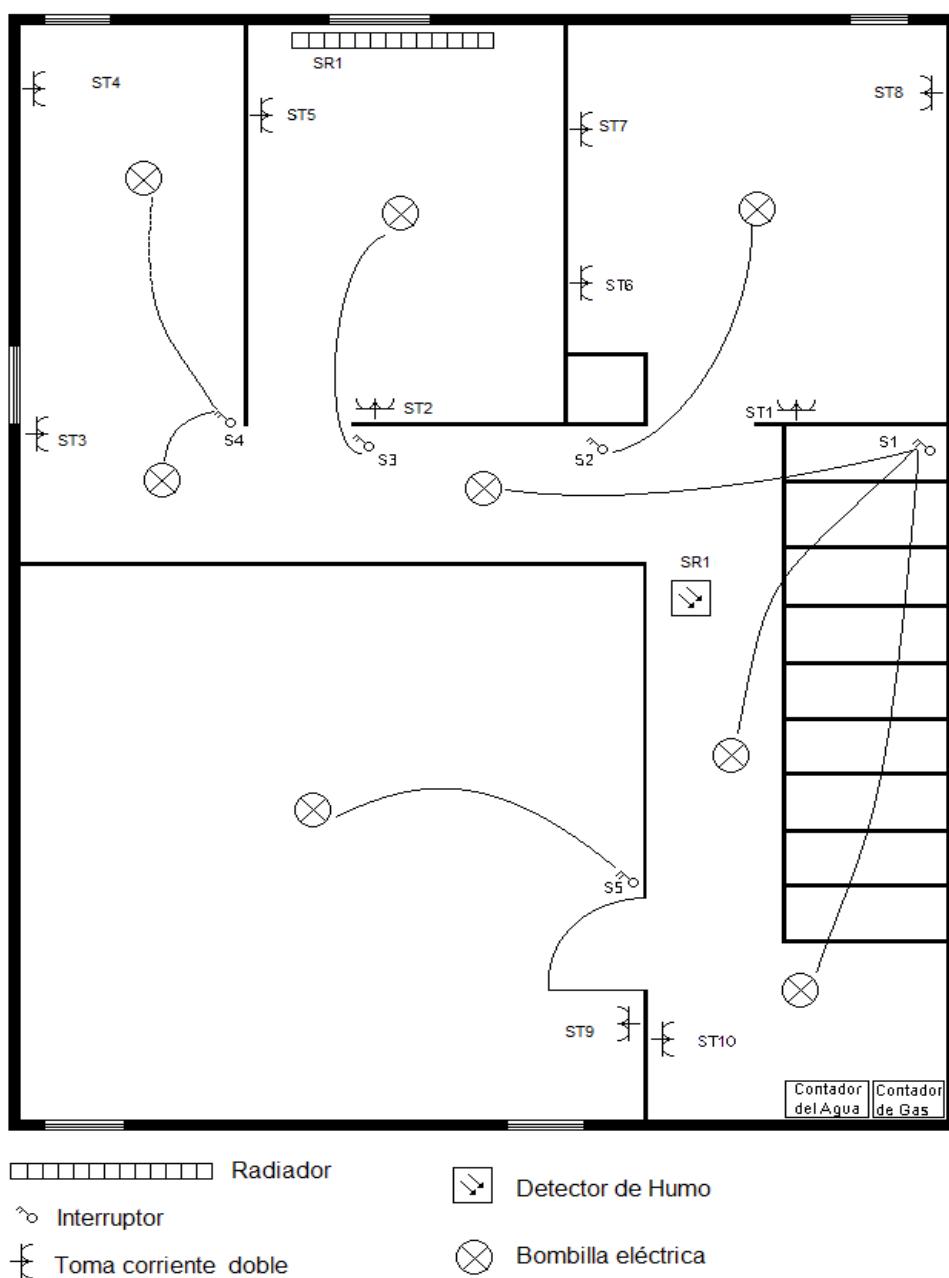


Figura 3.5: Sótano

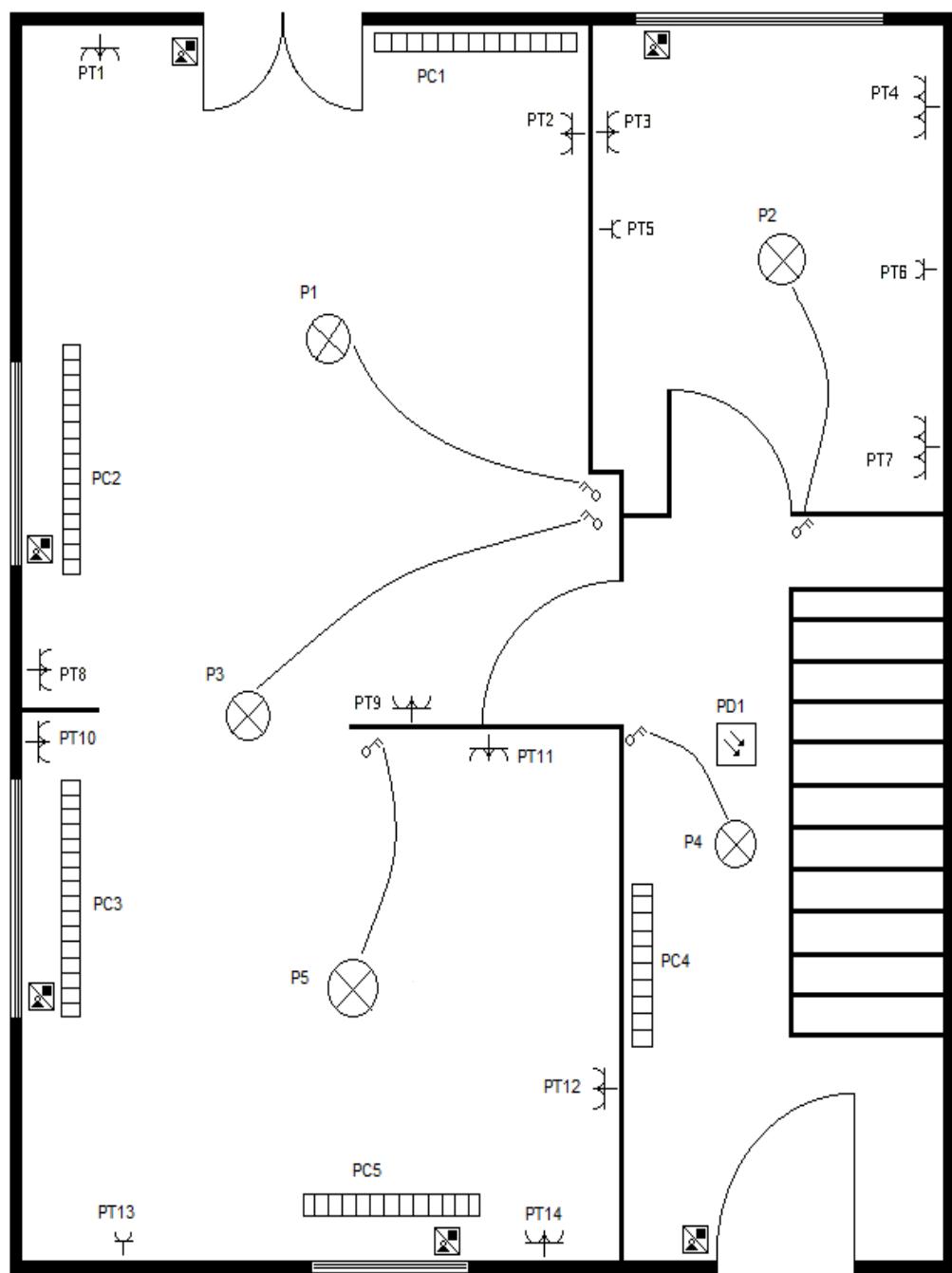


Figura 3.6: Primer Piso

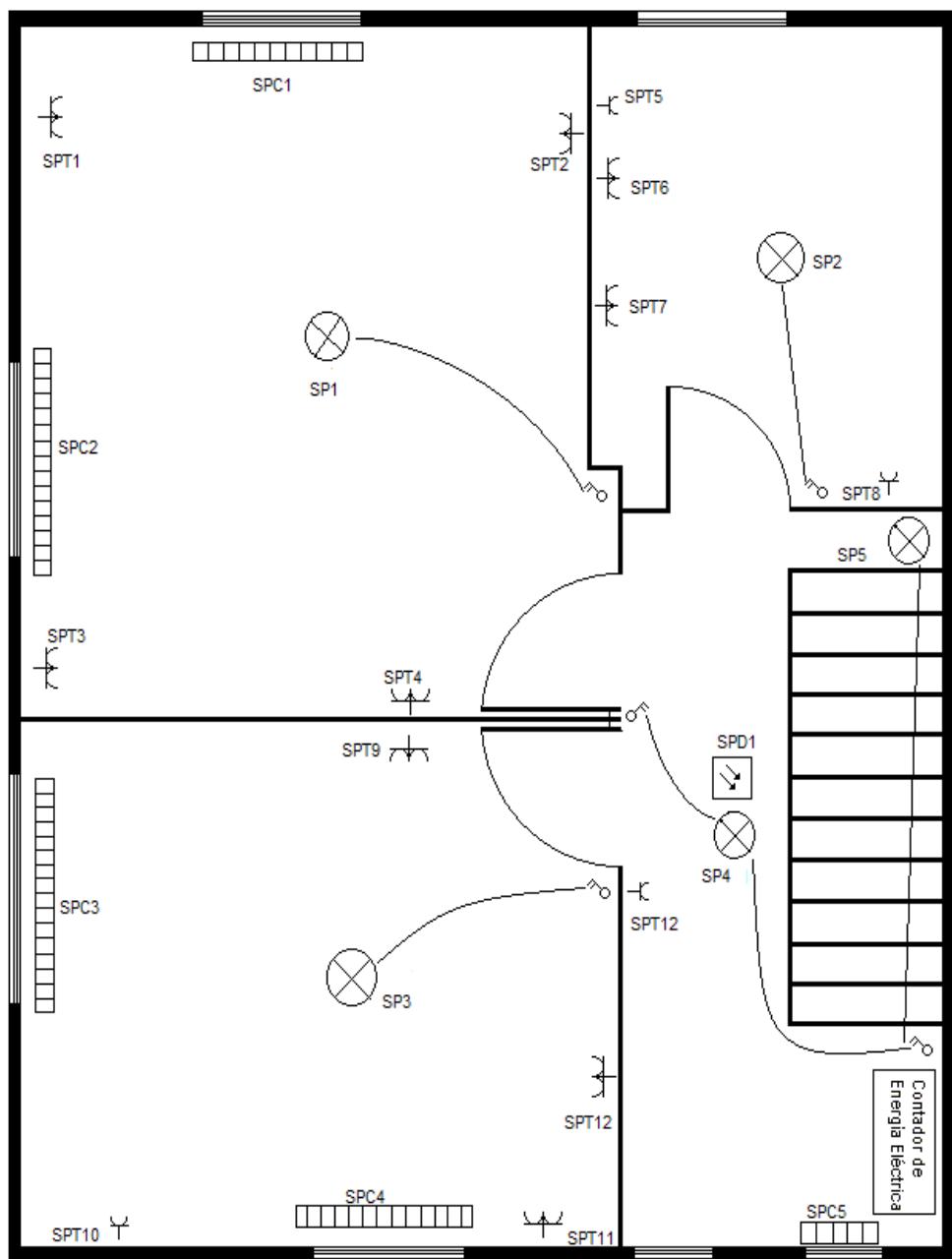


Figura 3.7: Segundo Piso

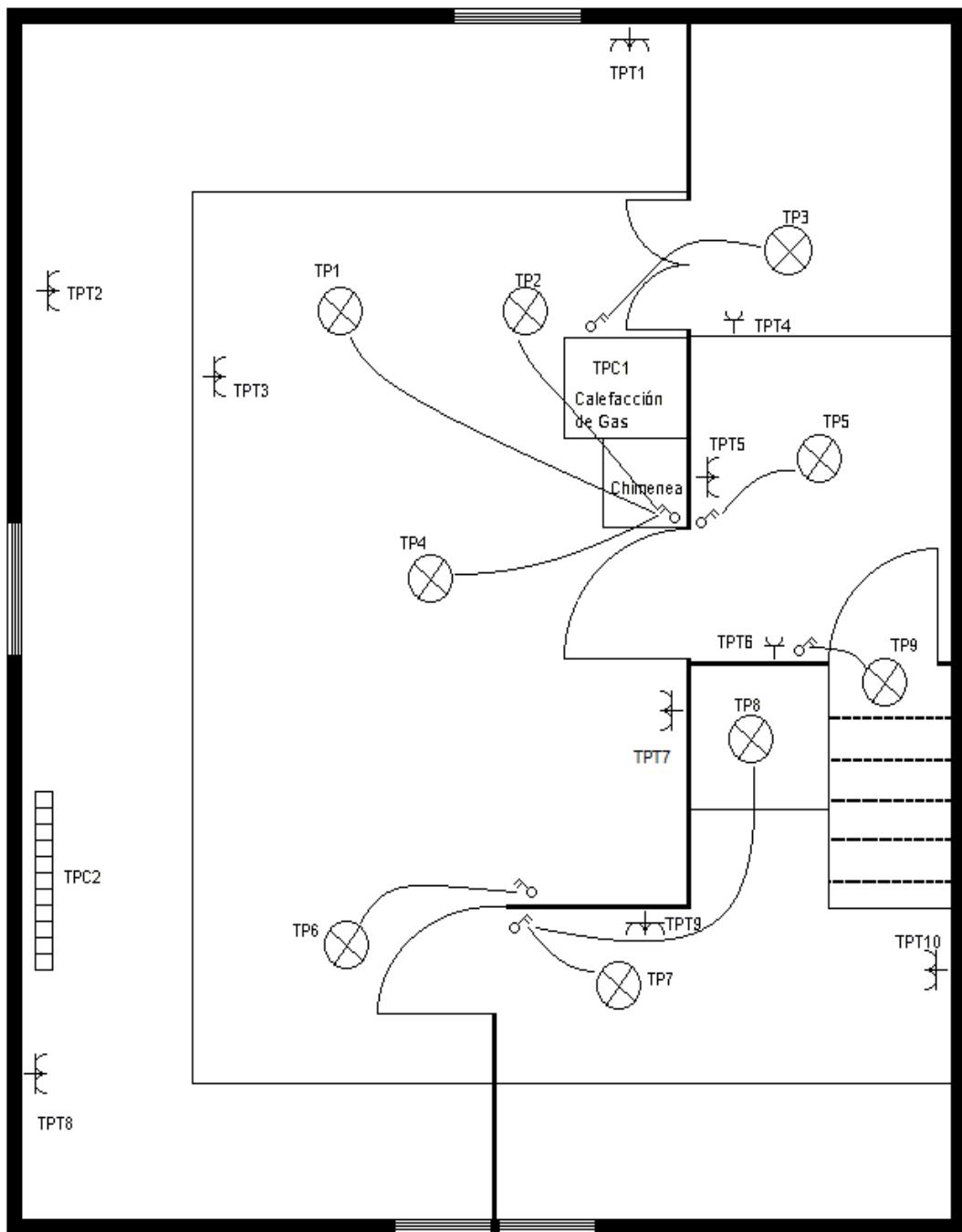


Figura 3.8: Tercer Piso

3.7 Componentes de la Vivienda

Librería DOMOSYS y Presentación de la Vivienda Ejemplo

Elementos para la Domótica	
Sótano	Cantidad
Detector de Humo	1
Detector de vibración en ventanas	6
Detector de ventanas abiertas	6
Detector de inundación	3
Lámparas en el techo o pared	8
Taster o Interruptor	5
Tomas de corriente	10
Sensor de Temperatura	1
Regulador de la temperatura	1
Primer Piso	
Detector de Humo	1
Detector de vibración en ventanas	6
Detector de ventanas abiertas	4
Detector de Puertas abiertas	2
Lámparas en el techo o pared	5
Taster o Interruptor	5
Tomas de Corriente	14
Sensor de temperatura	3
Regulador de la Temperatura	5
Cerradura electrónica	2
Segundo Piso	
Detector de Humo	1
Detector de vibración en ventanas	7
Detector de ventanas abiertas	7
Lámparas en el techo o pared	4
Taster o Interruptor	4
Tomas de corriente	13
Sensor de Temperatura	4
Regulador de Temperatura	7

Tercer Piso	
Detector de ventanas abiertas	4
Lamparas en el Techo o Pared	9
Taster o Interruptor	6
Tomas de Corriente	10
Sensor de Temperatura	2
Regulador de Temperatura	2

Tabla 3.1: Componentes para Automatizar en la Vivienda Ejemplo

3.8 Conclusiones

Gracias a las características del lenguaje orientado a objetos Modelica, se desarrollaron los 3 package que forman DOMOSYS utilizando la ayuda de la librería base de Modelica sin presentar problemas de dependencia. Los package son Eltako que contiene todos los componentes básicos para poder hacer funcionar el sistema Eltako inalámbrico; SBKNX (TP) contiene los componentes del Sistema KNX (TP) y Wago que contiene todos los componentes del sistema WAGO SPS 750.

Cada uno de los package contiene la vivienda ejemplo que esta compuesta de 3 pisos más el sótano. En esta vivienda se implementa el sistema por medio de la conexión de los componentes.

Por medio de el ordenamiento que se le ha dado a la librería se trata de entregar claridad, para que los posibles usuarios comprendan de una forma natural cómo funciona y cómo está compuesta la librería DOMOSYS.

4. Sistema Eltako

Inalámbrico

4.1 Introducción

En este capítulo se explica el sistema Eltako inalámbrico de la empresa Eltako. Este sistema es por radio frecuencia en la banda 868 MHz. El protocolo que usan para la comunicación es el que desarrolló la empresa ENOCEAN y se llama “ISO/IEC 14543-3-10 Information Technology”. Este protocolo inalámbrico está específicamente diseñado para mantener un consumo de energía de sensores e interruptores extremadamente bajo. Las ventajas más mencionadas es el uso de sensores y componentes de baja potencia. Tan baja es, que pueden tomar la energía del medio ambiente por medio de células fotovoltaicas, elementos piezoelectricos y otros convertidores de energía. También se explica el desarrollo del package Eltako en el lenguaje Modelica, con cada uno de sus componentes y la implementación en la vivienda ejemplo.

4.2 Descripción del Protocolo

Este protocolo utiliza casi todas las capas del Modelo OSI, excepto la capa sesión. Las más relevantes son la física, la de enlace de datos, la de red y la de aplicación (véase la figura 4.1).

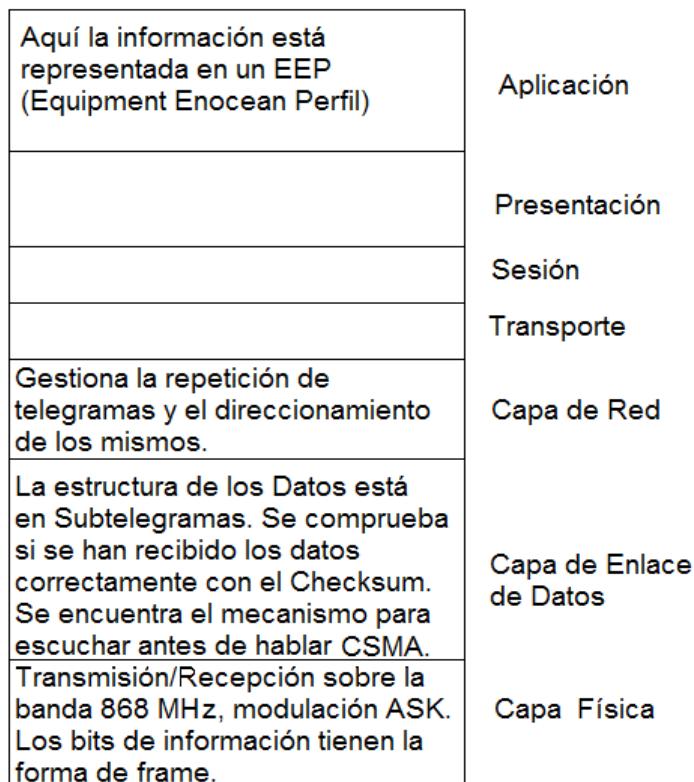


Figura 4.1: Especificaciones del Protocolo ENOCEAN en el modelo ISO.

Este protocolo está basado en paquetes que tienen las siguientes formas:

Frame: Es la unidad en la que se transmiten los bits de datos. Está determinada por una secuencia de bits, la componen símbolos especiales, que hacen posible que el receptor, detecte el inicio y el final del paquete.

Como se muestra en la figura 4.2, cada Frame esta compuesto de un preámbulo, que genera un bit de sincronización, después sigue la palabra de sincronización. Con ella el receptor puede prepararse para

Sistema Eltako Inalámbrico

recibir el tamaño de la información que está contenida en la longitud y después sigue la información.

Preambulo	Palabra de Sincronización	Longitud	Datos
-----------	---------------------------	----------	-------

Figura 4.2: Estructura del Frame en ENOCEAN.

- Subtelegrama: Al decodificar el frame, aparece el subtelegrama, que es procesado en la capa de enlace de datos.

Está compuesto (véase la figura 4.3):

- RORG: Identifica el tipo de Subtelegrama.
- DATA: La carga útil de subtelegrama.
- Sender ID: Identificación del transmisor.
- Status: Identifica si el subtelegrama es enviado por un repetidor y que tipo de control ha sido utilizado.
- Hash: es el valor de Checksum.

RORG	Data	Sender ID	Status	Hash
1 Byte	1.....X Bytes	4 Bytes	1 Byte	1 Byte

Figura 4.3: Estructura del Subtelegrama en ENOCEAN

Un telegrama puede estar conformado hasta por 3 subtelegramas.

La siguiente tabla muestra la clasificación de los telegramas de ENOCEAN.

Identificación del Telegrama	Identificación del Subtelegrama (RORG)	Función
RPS	F6	Comunicación repetida de un interruptor.
1BS	D5	Comunicación de 1 Byte.
4BS	A5	Comunicación de 4 Bytes.
VLD	D2	Variable de la longitud de Datos.
MSC	D1	Comunicación específica de fabricante.
ADT	A6	Telegrama de la dirección destino.
SM_LRН_REQ	C6	Solicitud de reconocimiento de aprendizaje.
SM_LRН_ANS	C7	Respuesta de solicitud de aprendizaje.
SM_REC	A7	Aprobación del reclamo.
SYS_EX	C5	Manejo remoto.
SEC	30	Telegrama seguro.
SEC_ENCAP	31	Telegrama encriptado.

Tabla 4.1: Clasificación de los Telegramas en ENOCEAN.

Para definir el perfil del equipo (EEP) el protocolo genera una estructura compuesta por 3 elementos:

- ERP: la clase de radiotelegrama, compuesto por 8 bits.
- FUNC: La función básica del contenido de datos, 6 bits.

- Type: Tipo de mecanismo, compuesto por 7 bits.

En el EEP están contenidos los datos exactos de lo que está pasando o quiere que pase con el sensor o el actuador.

4.3 Descripción del Sistema Eltako inalámbrico

Eltako es un sistema domótico por radiofrecuencia. Puede implementarse centralizado o descentralizado. Los sensores se comunican todos por radiofrecuencia por tal motivo no necesitan cables. Algunos tienen la tecnología de producir la energía necesaria para comunicarse, por lo cual no tienen baterías. Los que tienen baterías tienen un consumo bajo de energía, por esto el cambio de baterías no es frecuente.

Centralizado: esta opción la conforman los componentes de la Línea 14, todos los actuadores se encuentran ubicados en la caja de distribución eléctrica y se encuentran comunicados entre sí por medio del cable RS485. Modulo de Antena FAM14 recibe/envía mensajes por radiofrecuencia de/a los sensores y los envía/recibe a/de los actuadores por medio del cable RS485. Esta solución es eficiente y menos costosa. Para su implementación es necesario un distribuidor eléctrico bien planeado, ordenado y bien distribuido. Así se evita el romper paredes.

Descentralizado: está compuesta por la Línea 61 y 71, aquí los actuadores se encuentran cerca del elemento que se está automatizando. Para comunicarse con el software GFVS 3.0 necesitan un modulo de antena para comunicarse con los sensores y los actuadores. Aunque es más costoso, es más flexible y no es necesario romper paredes para su implementación.

En este trabajo se va a hacer énfasis en una configuración descentralizada.

Descripción del Sistema Eltako inalámbrico

Para conocer cada uno de los componentes del sistema domótico por radiofrecuencia Eltako, se puede bajar el catálogo por Internet en la dirección www.eltako.com.

Los componentes básicos del sistema son los sensores y los actuadores. Los sensores se diferencian dependiendo de su función, así que son más individuales. Los actuadores en cambio, manipulan un elemento de la corriente y por tal motivo casi todos los componen relés, que pueden graduar la corriente de salida dando paso a funciones como dimmer.

4.4 Comunicación entre Sensores y actuadores de el Sistema Eltako Inalámbrico

Los elementos que hacen posible la comunicación entre los sensores y los actuadores por radiofrecuencia son:

- Software GFVS 3.0
- Servidor con Antena incorporada.
- Interfaz IP de radiofrecuencia.

4.5 Descripción de la comunicación entre los Sensores y los Actuadores

Los sensores y actuadores se comunican con el protocolo desarrollado por la firma ENOCEAN, anteriormente descrito. Estos tienen una identificación única, para facilitar a los actuadores el proceso de aprendizaje para saber a qué sensor o sensores deben escuchar. El actuador puede tener uno o varios botones o ruedas de configuración para establecer el modo de trabajo. Para que un actuador obedezca a un determinado Sensor, se configura el actuador en modo de aprender “LRN”, aquí se pone a funcionar el sensor que debe ser escuchado y el actuador guarda la respectiva dirección. Los actuadores pue-

den escuchar hasta 35 sensores. Es decir, pueden guardar en su memoria la dirección de 35 sensores. Por medio del Software GFVS también se puede visualizar, configurar y accionar los actuadores y sensores. Para utilizar este software con los sensores y los actuadores, se debe tener un modulo de antena que facilita la comunicación bidireccional. Para una comunicación segura con el Internet y el sistema, se necesita el servidor GFVS.

4.6 Componentes del Package Eltako

Este package contiene 3 grandes grupos que son los sensores, actuador y la vivienda ejemplo. Además está el componente servidor que se encarga del manejo a distancia de la vivienda. El actuador es el acto-rele y es la parte básica de todos los actuadores del sistema Eltako. Los sensores están conformados por interruptorpassiv, stemperatura, Detectorhumo, kvp. Estos encierran los sensores básicos y más importantes del sistema Eltako inalámbrico. Todos ellos están configurados en la vivienda ejemplo, en donde se encuentra el modelo ge, que corresponde al gasto energético de toda la vivienda.

4.6.1 Actuado Actorrele

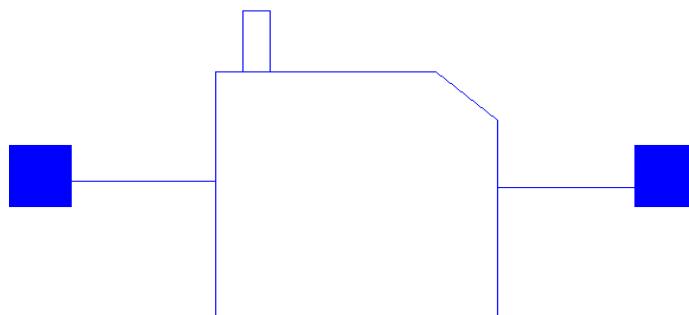


Figura 4.4 Actuador actorrele del Package Eltako

El actorrele se encuentra formado por una resistencia variable que tiene como fin representar el funcionamiento de la parte básica de

un actor del sistema Eltako. Cuando está en funcionamiento el actorrele tiene un gasto de potencia de 0.8W, por tal motivo el valor de la resistencia tiene 406 ohmios. Cuando el elemento que queremos controlar está en funcionamiento, entonces el gasto de potencia depende del consumo del elemento como el led o bombilla. En la figura 4.5 se puede observar los 2 pines por donde entra la información por medio de la corriente y el voltaje. La resistencia es variable, la flecha que se encuentra en la mitad de la misma, es la entrada del valor en ohmios de la resistencia. Esta variable se llama `variableResistor`. R y se observa la ecuación en el código 4.1, dependiendo si el actorrele tiene los parámetros booleanos activo como verdadero y an como falso, entonces significa que el actuador esta listo a ser usado pero no esta accionado y ese es el valor de potencia que estamos midiendo.

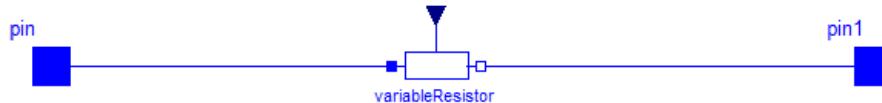


Figura 4.5: Diseño de actorrele en Dymola

- Los parámetros de el actorrele son:
- Parámetro `ident` que es el número de identificación del componente es integer.
- Parámetro `sensor` que es integer y significa la identificación del sensor que puede escuchar. Este actorrele puede escuchar hasta 9 sensores diferentes
- Parámetro `activ` que es un booleano y significa si está activo o no.
- Parámetro `temperaturadeseada` es con unidades celsius y es si está actuado como regulador de temperatura, entonces se puede configurar la temperatura deseada.

El parámetro de gasto energético es de 0.8 wattios.

Sistema Eltako Inalámbrico

```
model actorrele
  "Parte básica de todos los actuadores, aquí están almacenados todos los
  sensores que pueden ser escuchados."
  parameter Integer iden=111111 "Identificación del actor";//La identificación
  del actuador
  parameter Integer sensor1=111111
    "dirección del sensor 1, que lee para desarrollar acciones determinada";
  parameter Integer sensor2=000000
    "dirección del sensor 2, que lee para desarrollar acciones determinada.";
  parameter Integer sensor3=000000
    "dirección del sensor 3, que lee para desarrollar acciones determinada.";
  parameter Integer sensor4=000000
    "dirección del sensor 4, que lee para desarrollar acciones determinada.";
  parameter Integer sensor5=000000
    "dirección del sensor 5, que lee para desarrollar acciones determinada.";
  parameter Integer sensor6=000000
    "dirección del sensor 6, que lee para desarrollar acciones determinada.";
  parameter Integer sensor7=000000
    "dirección del sensor 7, que lee para desarrollar acciones determinada.";
  parameter Integer sensor8=000000
    "dirección del sensor 8, que lee para desarrollar acciones determinada.";
  parameter Integer sensor9=000000
    "dirección del sensor 9, que lee para desarrollar acciones determinada.";
  parameter Boolean activ=false
    "Significa si el actuador está activo en el sistema.";
  //Significa si el actuador está activo, si escucha a algún sensor.
  parameter Boolean an=false
    "Significa si el sensor está consumiendo o no energía";
  //Significa si está conectado a la energía eléctrica
  parameter Modelica.SIunits.Power gastoenergetico= 0.8
    "Cantidad de Potencia que consume el sistema";
  //La cantidad de gasto energético, por estar en funcionamiento
  parameter Modelica.SIunits.CelsiusTemperature temperaturadeseada=20 ;
  Modelica.Electrical.Analog.Interfaces.Pin pin;
  Modelica.Electrical.Analog.Interfaces.Pin pin1;

  Modelica.Electrical.Analog.Basic.VariableResistor variableResistor;

equation
  connect(pin, variableResistor.p);
  connect(variableResistor.n, pin1);
  variableResistor.R=if activ==true and an==true then 1000 else
    if activ==true and an== false then 406 else 1000000;
end actorrele;
```

Código 4.1: Definición de actorrele en Modelica.

4.6.2 Interruptor interruptorpasiv

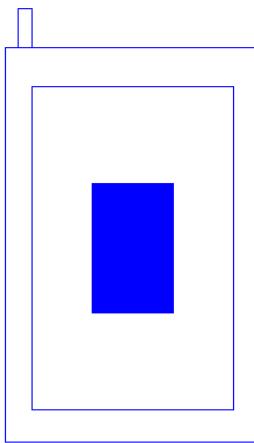


Figura 4.6: Interruptorpasiv
Package Eltako

Este elemento hace las veces de un interruptor. Es pasivo ya que produce la energía necesaria por sí mismo. Su función es enviar telegramas para avisarle a los actuadores una acción determinada. Tiene como parámetros:

- identificación (int) empieza siempre por uno, como su nombre lo indica es la identificación ante los actuadores.
- activ (BOOLEAN) que significa que el sensor está en funcionamiento.
- acción es un parámetro integer, que es un entero y relaciona una acción determinada con un número, de esta manera se puede regular la intensidad de la corriente o programar determinadas escenas.

Sistema Eltako Inalámbrico

```
model interruptorpasiv "Es un Interruptor inalambrico, que tiene dos estados y  
diferentes acciones que pueden hacer el papel de un regulador."  
parameter Integer identificacion=123456 "La identificación del Interruptor";  
parameter Boolean activo=false  
    "Sirve para saber si el Interruptor está en funcionamiento";  
parameter Integer accion=1  
    "Definir la acción que puede ser prender, apagar o dimmer";  
  
end interruptorpasiv;
```

Código 4.2: Definición de interruptorpasiv en Modelica.

4.6.3 Detector de humo Detectorhumo

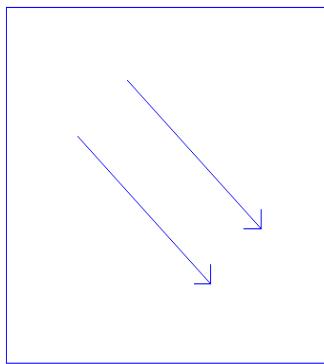


Figura 4.7: Detectorhumo Package eltako

Tiene como función detectar si hay fuego y humo. No está conectado a la energía eléctrica de la vivienda, se abastece de una pila de 9 v. Los parámetros son:

- ident significa un número entero de identificación.
- Activ es un booleano significa si está activo o no.
- mensaje es un número entero y significa el estado del sensor después del monitoreo.

```

model Detectorhumo "Sensor que detecta el humo"
  parameter Integer ident=111111 "Identificación del detector de humo";
  parameter Boolean activ=false
    "Booleano para saber si esta activo en el sistema";
  parameter Integer mensaje=1
    "Información que envia el detector de humo al actor";
  end Detectorhumo;

```

Código 4.3: Definición de Detectorhumo en Modelica.

4.6.4 Sensor de Temperatura stemperatura

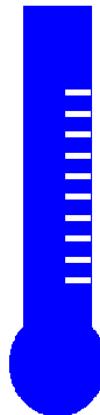


Figura 4.8: stemperatura Package eltako

Tiene como función enviar mensajes de la temperatura. Los parámetros son:

- identificacion es un parámetro entero que identifica al sensor.
- Temperatura es un parámetro en Celsius que toma la temperatura medida por el sensor y la envía al actuador.
- activo este parámetro es un booleano si es verdadero está activo.

Sistema Eltako Inalámbrico

```
model stemperatura
  "Sensor de Temperatura inalámbrico, mide la temperatura y la envía al regulador."
  parameter Integer identificacion=111111 "Identificación en el sistema";
  parameter Modelica.SIunits.CelsiusTemperature Temperatura=20
    "Temperatura que tiene el instrumento";
  parameter Boolean activo=false "Si el elemento esta activo en el sistema";
equation
  [ ]
end stemperatura;
```

Código 4.4: Definición de stemperatura en Modelica.

4.6.5 Sensor de Contactos de Ventanas y Puertas kvp



Figura 4.9: Sensor de contactos de puertas y ventanas del Package eltako

Sensor que tiene los siguientes parámetros:

- iden es un número entero, que significa la identificación del sensor.
- activo es un booleano, si es 1 significa que la puerta o ventana está cerrada.

```
model kvp
  "Contacto de puertas y ventanas, tiene dos estados que se identifican con el booleano activo."
  parameter Integer ident=1 "Identificación del contacto de puertas y ventanas";
  parameter Boolean activo=true
    "Booleano para saber si el sensor está activo en el sistema";
[ ]
end kvp;
```

Código 4.5: Definición de kvp en Modelica.

4.6.6 Implementación del Sistema Eltako en la Vivienda Ejemplo

Por medio de la implementación de los componentes anteriormente presentados, se pretende calcular el gasto energético del sistema eltako inalámbrico en la vivienda ejemplo. Se toma la tabla 3.1, donde se observan los elementos a automatizar, de esta forma se relaciona cada sensor con el respectivo actor. Teniendo en cuenta que los actores deben estar conectados a una fuente energética, por tal motivo se utiliza una de 230 voltios a 50 Hz. Así sucesivamente se desarrolla la representación en código Modelica de cada uno de los pisos de la vivienda ejemplo. De esta forma nacen los modelos estoano, eprimerpiso, esegundopiso, etercerpiso. Cada uno de estos modelos posee la variable gasto. Que representa la sumatoria del gasto energético de cada uno de los componentes del sistema eltako inalámbrico, que hacen posible la automatización de la misma. El modelo ge entrega el gasto energético de toda la vivienda ejemplo con la variable total.

Sistema Eltako Inalámbrico

- Sótano

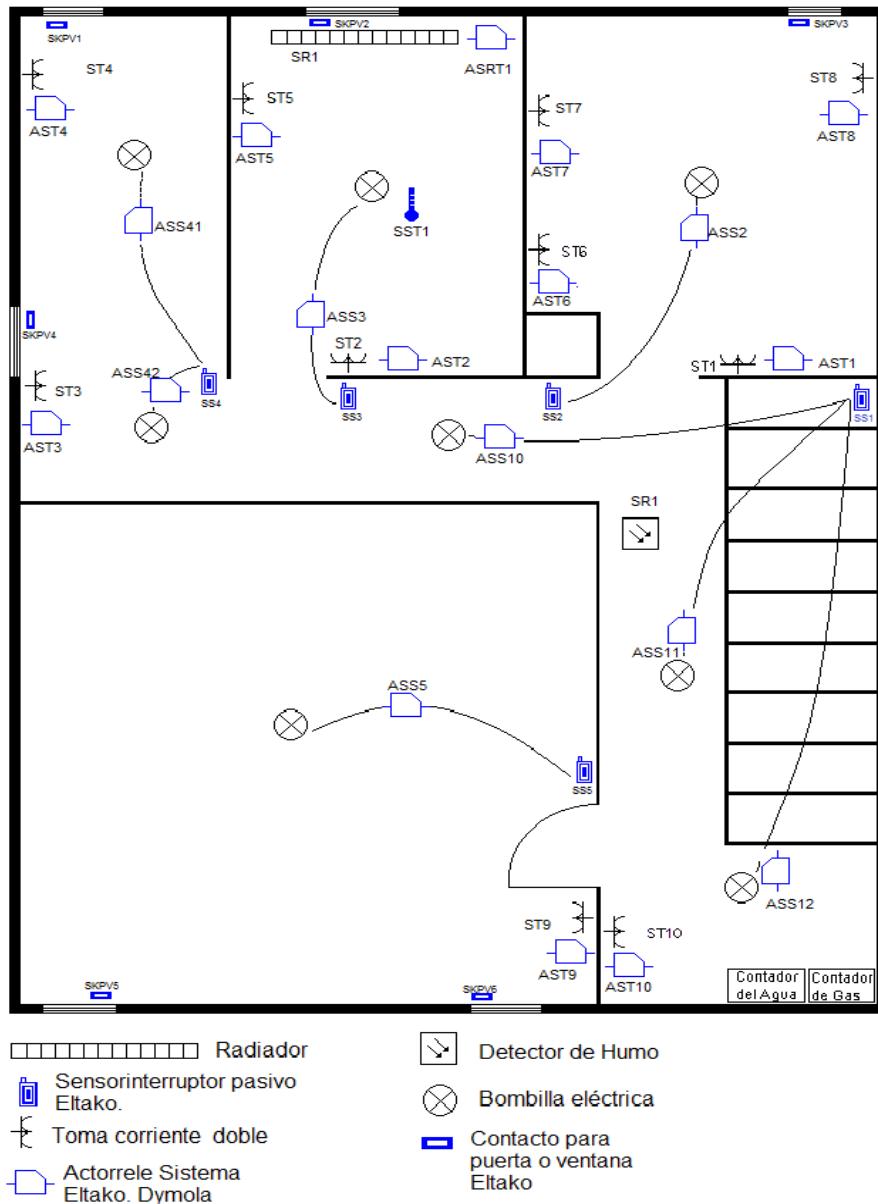


Figura 4.10: Sótano con el Sistema Eltako Inalámbrico

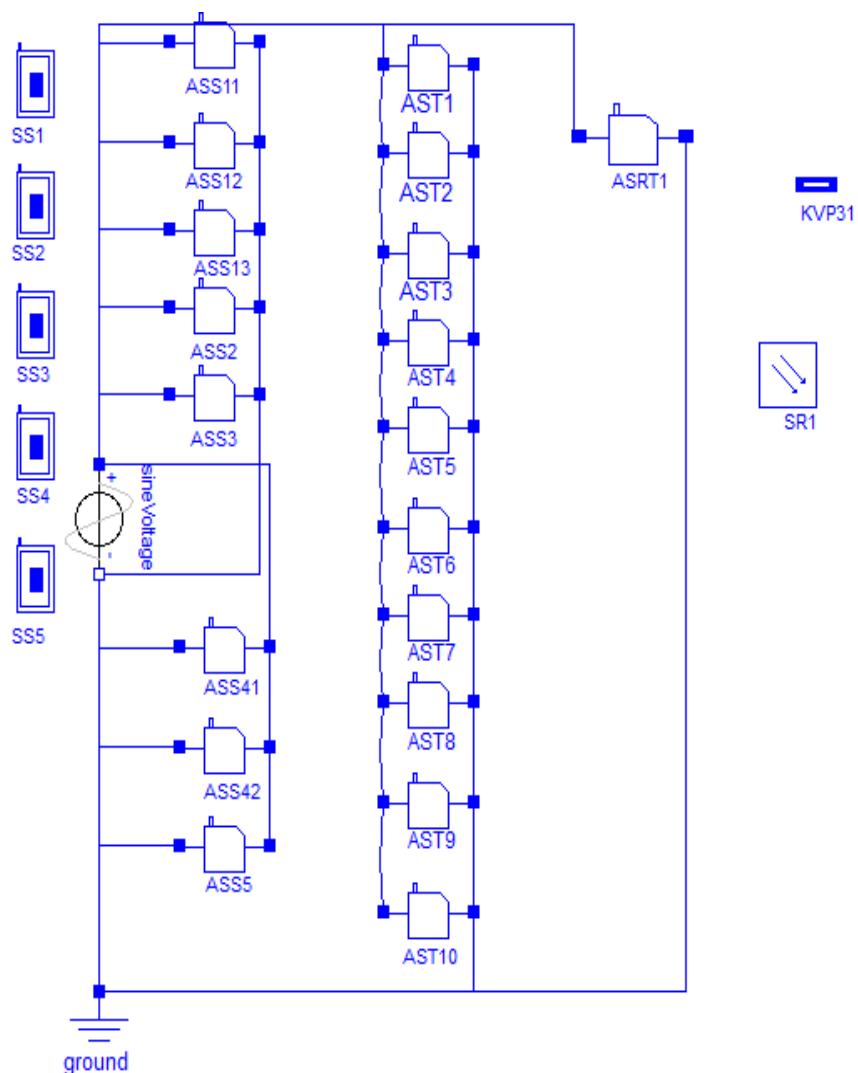


Figura 4.11: Representación de la Configuración de los Componentes del Sistema Eltako en el Sótano.

Sistema Eltako Inalámbrico

```

model esotano
    "Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el sótano."
    Modelica.SIunits.Power gastosotano;
    g
equation
    gastosotano=(if ASS11.activ==true then ASS11.gastoenergetico else 0) +
        (if ASS12.activ==true then ASS12.gastoenergetico else 0)+(if ASS13.activ==true then ASS13.gastoenergetico else 0) +
        (if ASS2.activ==true then ASS2.gastoenergetico else 0)+(if ASS3.activ==true then ASS3.gastoenergetico else 0) +
        (if ASS41.activ==true then ASS41.gastoenergetico else 0)+(if ASS42.activ==true then ASS42.gastoenergetico else 0) +
        (if ASS5.activ==true then ASS5.gastoenergetico else 0)+(if actorrele8.activ==true then
            actorrele8.gastoenergetico else 0)+(if actorrele9.activ==true then actorrele9.gastoenergetico else 0) +
        (if actorrele10.activ==true then actorrele10.gastoenergetico else 0)+(if actorrele11.activ==true then
            actorrele11.gastoenergetico else 0)+(if actorrele12.activ==true then actorrele12.gastoenergetico else 0) +
        (if actorrele13.activ==true then actorrele13.gastoenergetico else 0)+(if actorrele14.activ==true then
            actorrele14.gastoenergetico else 0)+(if actorrele15.activ==true then actorrele15.gastoenergetico else 0) +
        (if actorrele16.activ==true then actorrele16.gastoenergetico else 0)+(if actorrele17.activ==true then
            actorrele17.gastoenergetico else 0);
l
end esotano;

```

Código 4.6: Representación en Modelica del Gasto Energético del Sótano.

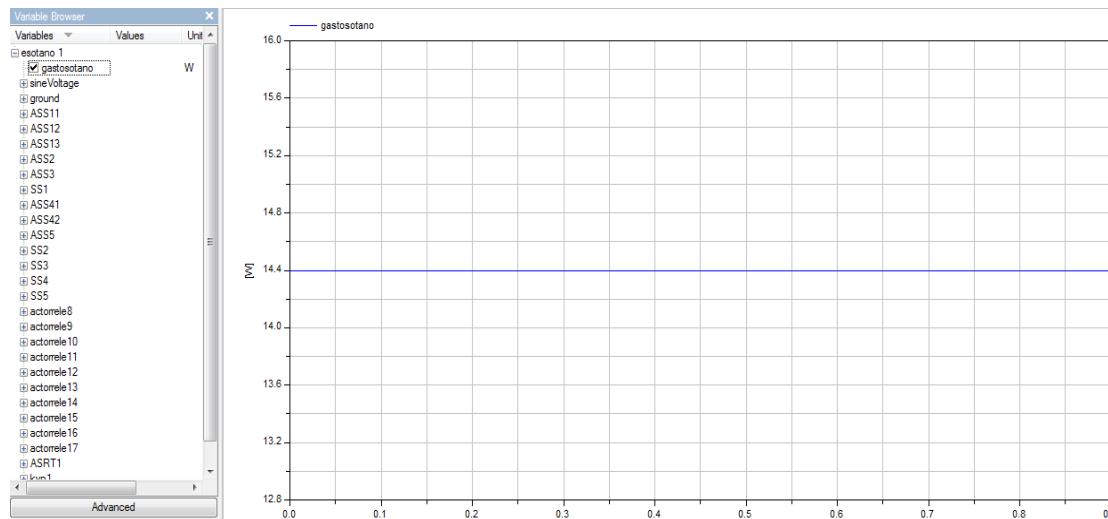


Figura 4.12: Simulación del Sistema Eltako en el Sótano.

- Primer Piso

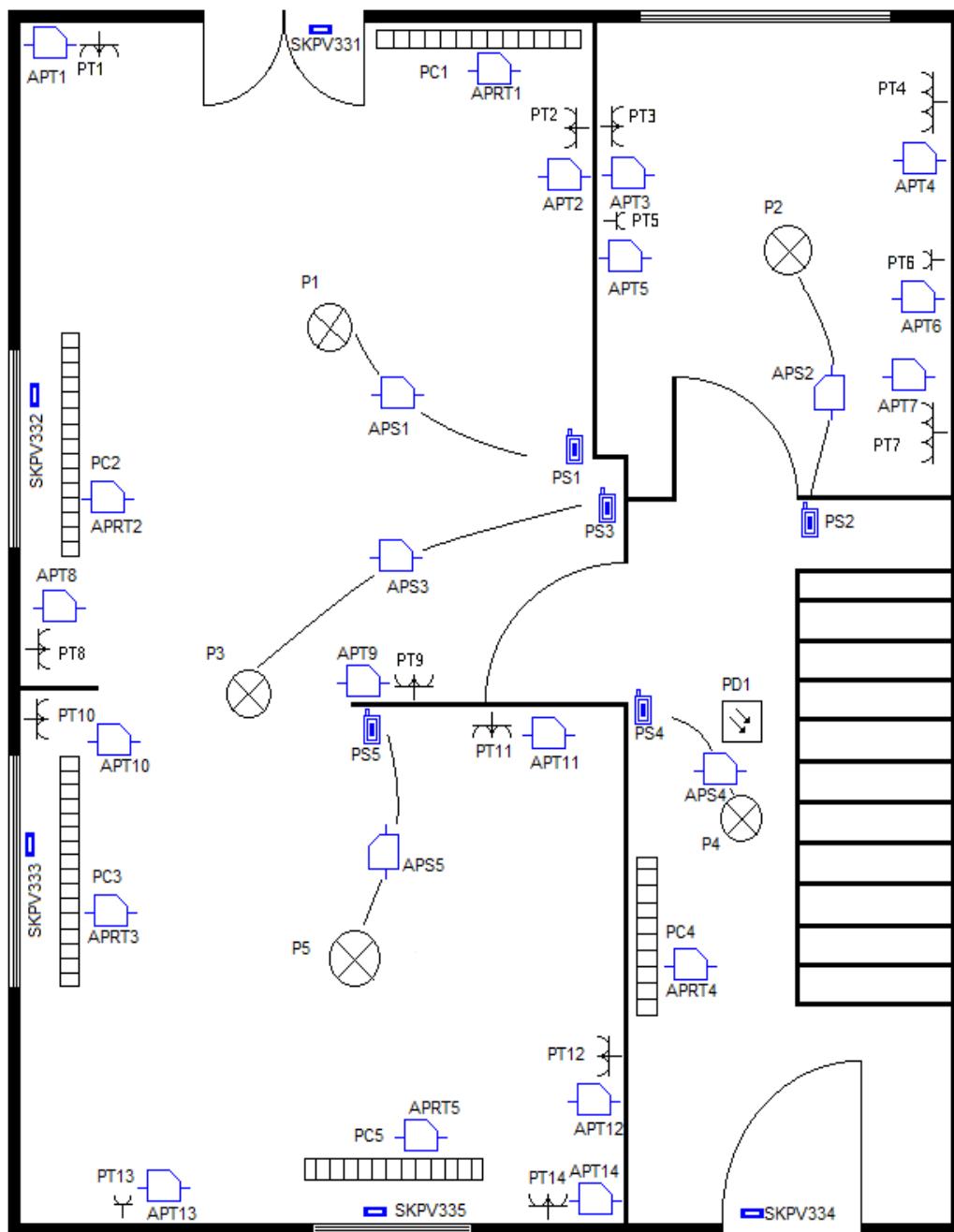


Figura 4.13: Primer Piso con el Sistema Eltako Inalámbrico.

Sistema Eltako Inalámbrico

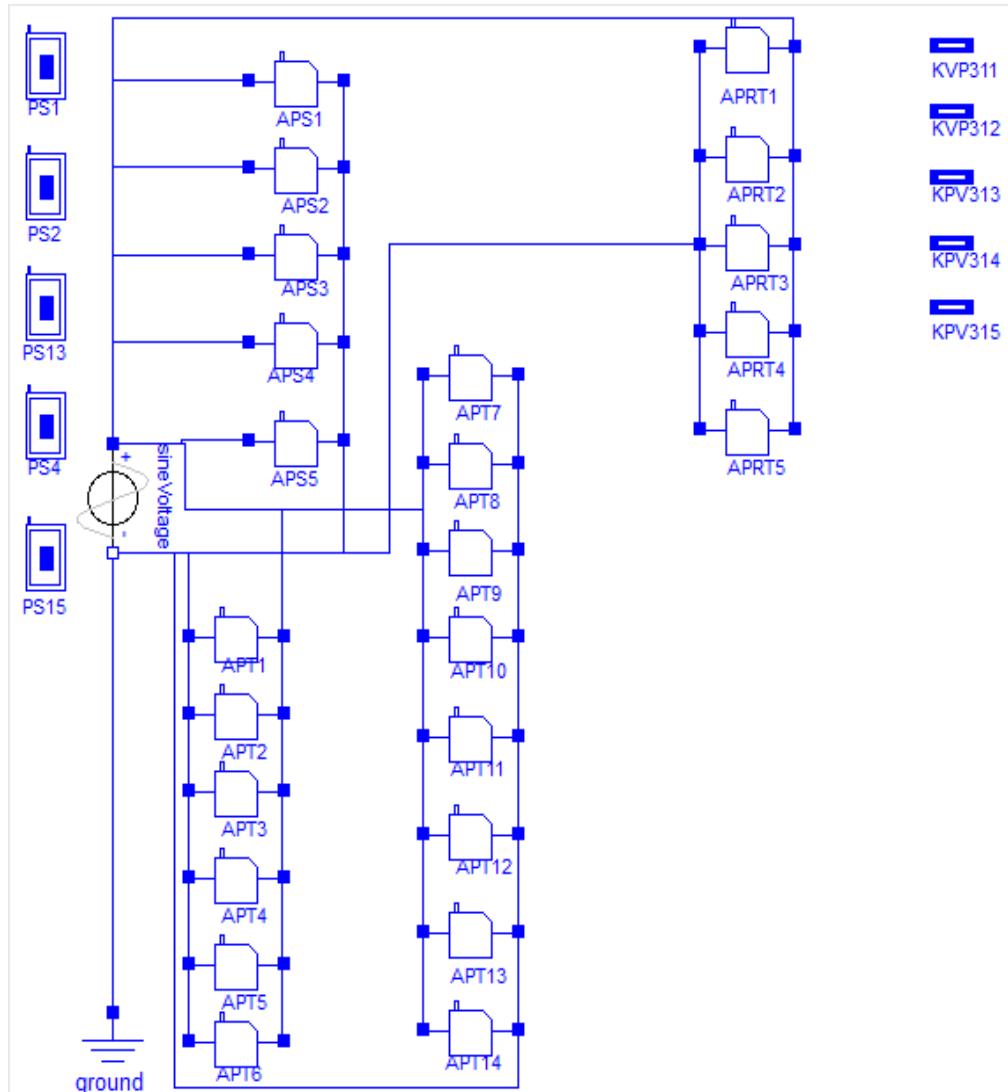


Figura 4.14: Representación de la Configuración de los Componentes del Sistema Eltako en el Primer Piso.

Componentes del Package Eltako

```

model eprimerpiso
Modelica.SIunits.Power gastoprimerpiso;
  equation
gastoprimerpiso=(if actorrele1.activ==true then actorrele1.gastoenergetico else 0)+(if actorrele2.activ==true then actorrele2.gastoenergetico else 0)+  

+(if actorrele3.activ==true then actorrele3.gastoenergetico else 0)+(if actorrele4.activ==true then actorrele4.gastoenergetico else 0)+  

(if actorrele5.activ==true then actorrele5.gastoenergetico else 0)+(if actorrele6.activ==true then actorrele6.gastoenergetico else 0)+  

(if actorrele7.activ==true then actorrele7.gastoenergetico else 0)+(if actorrele8.activ==true then actorrele8.gastoenergetico else 0)+  

(if actorrele9.activ==true then actorrele9.gastoenergetico else 0)+(if actorrele10.activ==true then actorrele10.gastoenergetico else 0)+  

(if actorrele11.activ==true then actorrele11.gastoenergetico else 0)+(if actorrele12.activ==true then actorrele12.gastoenergetico else 0)+  

(if actorrele13.activ==true then actorrele13.gastoenergetico else 0)+(if actorrele14.activ==true then actorrele14.gastoenergetico else 0)+  

(if actorrele15.activ==true then actorrele15.gastoenergetico else 0)+(if actorrele16.activ==true then actorrele16.gastoenergetico else 0)+  

(if actorrele17.activ==true then actorrele17.gastoenergetico else 0)+(if actorrele18.activ==true then actorrele18.gastoenergetico else 0)+  

(if actorrele19.activ==true then actorrele19.gastoenergetico else 0)+(if actorrele20.activ==true then actorrele20.gastoenergetico else 0)+  

(if actorrele21.activ==true then actorrele21.gastoenergetico else 0)+(if actorrele22.activ==true then actorrele22.gastoenergetico else 0)+  

(if actorrele23.activ==true then actorrele23.gastoenergetico else 0)+(if actorrele24.activ==true then actorrele24.gastoenergetico else 0);
  end eprimerpiso;

```

Código 4.7: Ecuación del Gasto Energético del Primer Piso



Figura 4.15: Resultados de la Simulación del Sistema Eltako en el Primer Piso.

Sistema Eltako Inalámbrico

- Segundo Piso

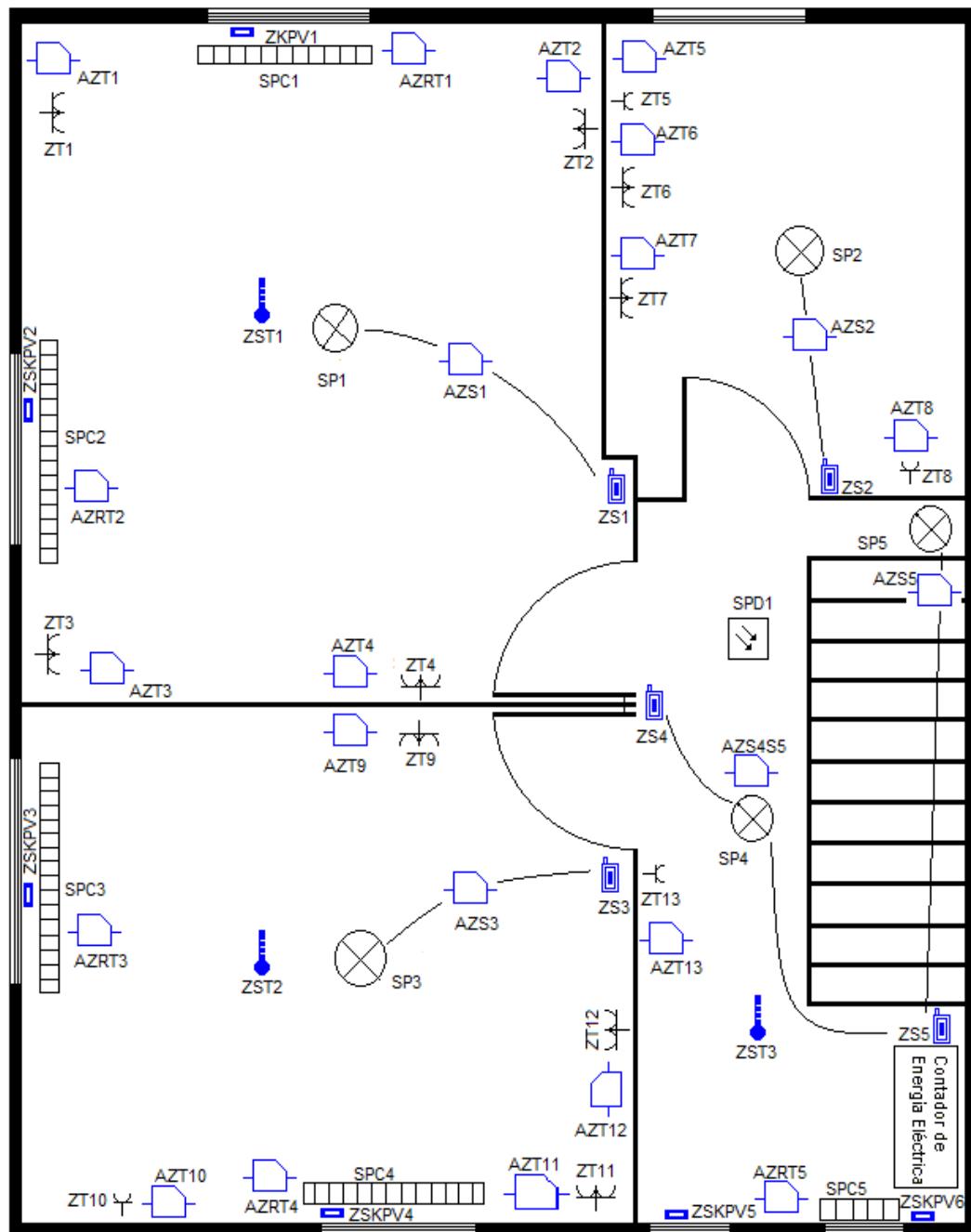


Figura 4.16: Segundo Piso con el Sistema Eltako Inalámbrico

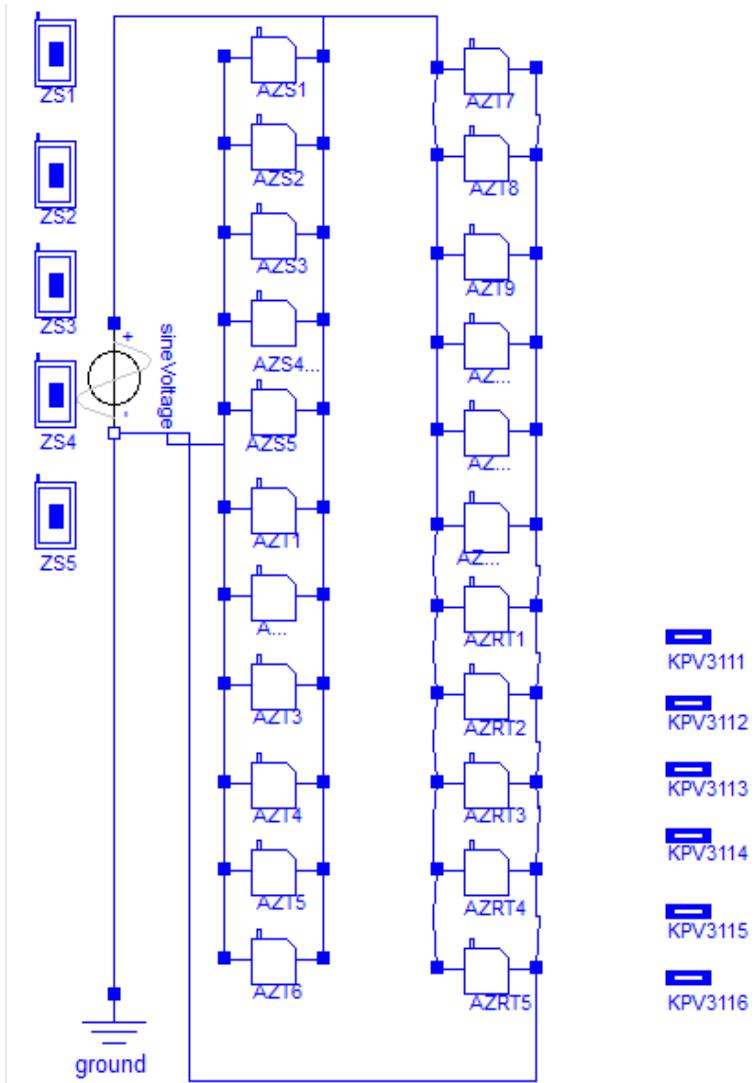


Figura 4.17: Representación de la Configuración de los Componentes del Sistema Eltako en el Segundo Piso.

```

model eseundopiso
    "Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el segundo piso."
    Modelica.SIunits.Power gastosegundopiso;
    if
equation
gastosegundopiso=(if actorrele1.activ==true then actorrele1.gastoenergetico else 0)+(if actorrele2.activ==true then
    actorrele2.gastoenergetico else 0)+(if actorrele3.activ==true then actorrele3.gastoenergetico else 0)+
    (if actorrele4.activ==true then actorrele4.gastoenergetico else 0)+(if actorrele5.activ==true then
    actorrele5.gastoenergetico else 0)+(if actorrele6.activ==true then actorrele6.gastoenergetico else 0)+
    (if actorrele7.activ==true then actorrele7.gastoenergetico else 0)+(if actorrele8.activ==true then
    actorrele8.gastoenergetico else 0)+(if actorrele9.activ==true then actorrele9.gastoenergetico else 0)+
    (if actorrele10.activ==true then actorrele10.gastoenergetico else 0)+(if actorrele11.activ==true then
    actorrele11.gastoenergetico else 0)+(if actorrele12.activ==true then actorrele12.gastoenergetico else 0)+
    (if actorrele13.activ==true then actorrele13.gastoenergetico else 0)+(if actorrele14.activ==true then
    actorrele14.gastoenergetico else 0)+(if actorrele15.activ==true then actorrele15.gastoenergetico else 0)+
    (if actorrele16.activ==true then actorrele16.gastoenergetico else 0)+(if actorrele17.activ==true then
    actorrele17.gastoenergetico else 0)+(if actorrele18.activ==true then actorrele18.gastoenergetico else 0)+
    (if actorrele19.activ==true then actorrele19.gastoenergetico else 0)+(if actorrele20.activ==true then
    actorrele20.gastoenergetico else 0)+(if actorrele21.activ==true then actorrele21.gastoenergetico else 0)+
    (if actorrele22.activ==true then actorrele22.gastoenergetico else 0);
end eseundopiso;

```

código 4.8: Ecuación del Gasto Energético del Segundo Piso

Sistema Eltako Inalámbrico

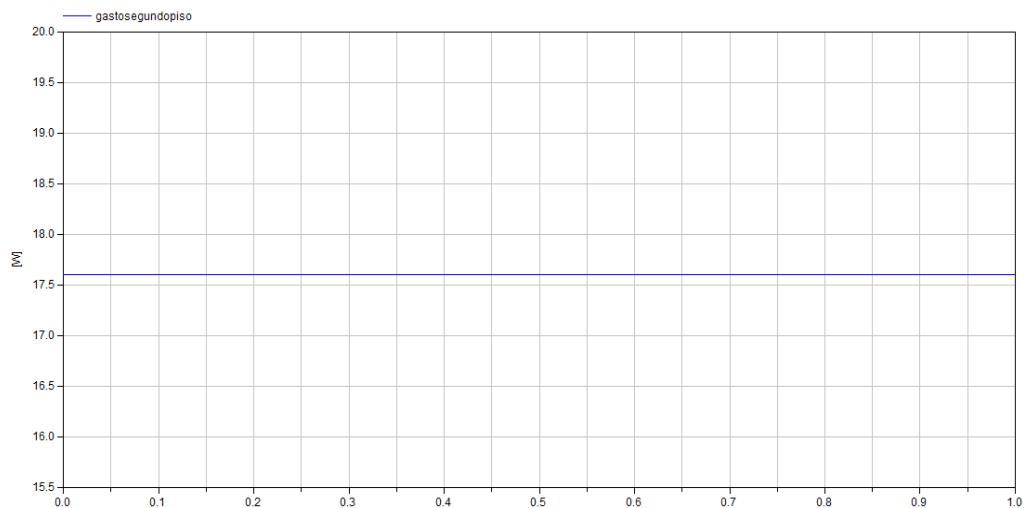


Figura 4.18: Simulación del Gasto Energético Segundo Piso.

- Tercer Piso

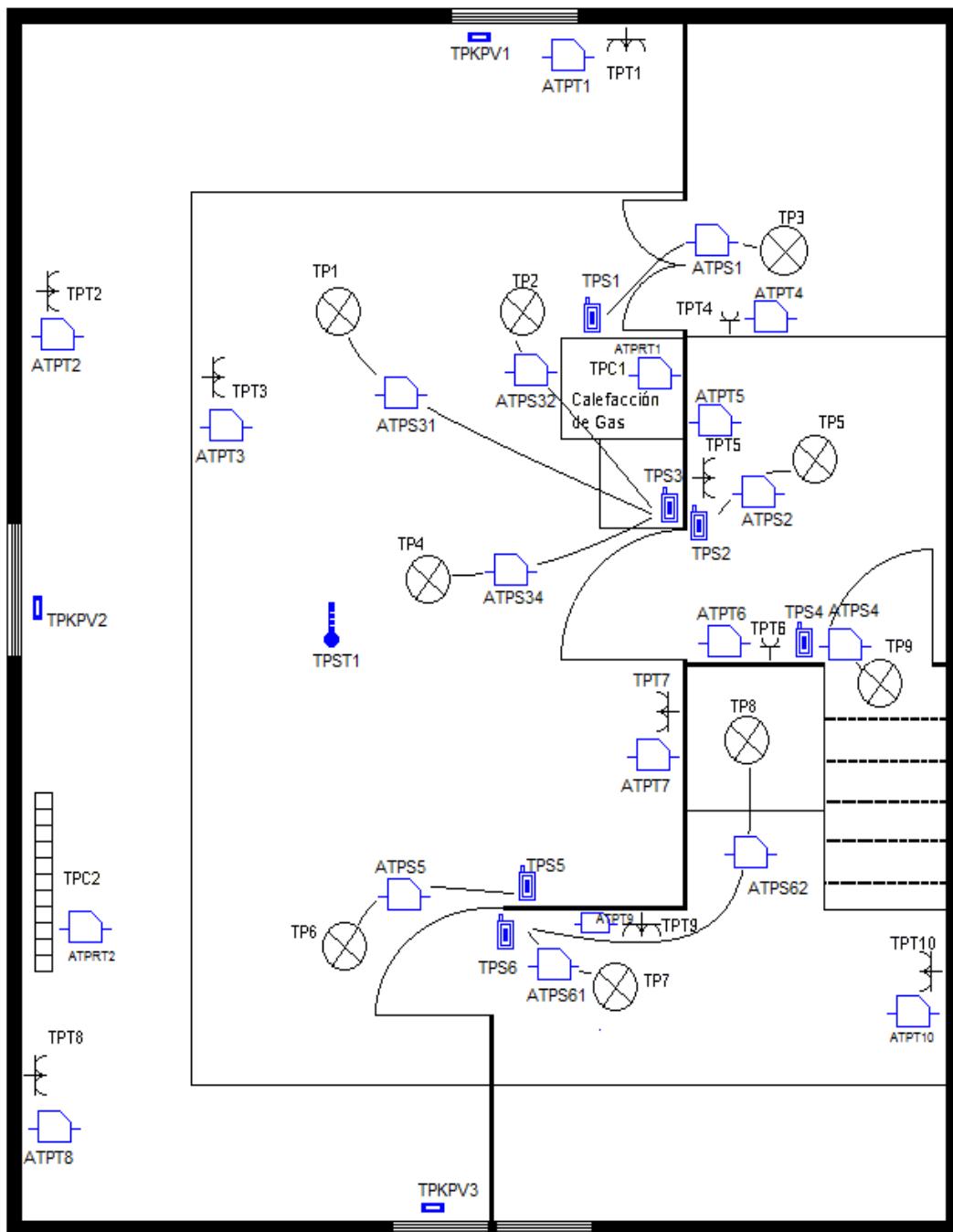


Figura 4.19: Tercer Piso con el Sistema Eltako Inalámbrico

Sistema Eltako Inalámbrico

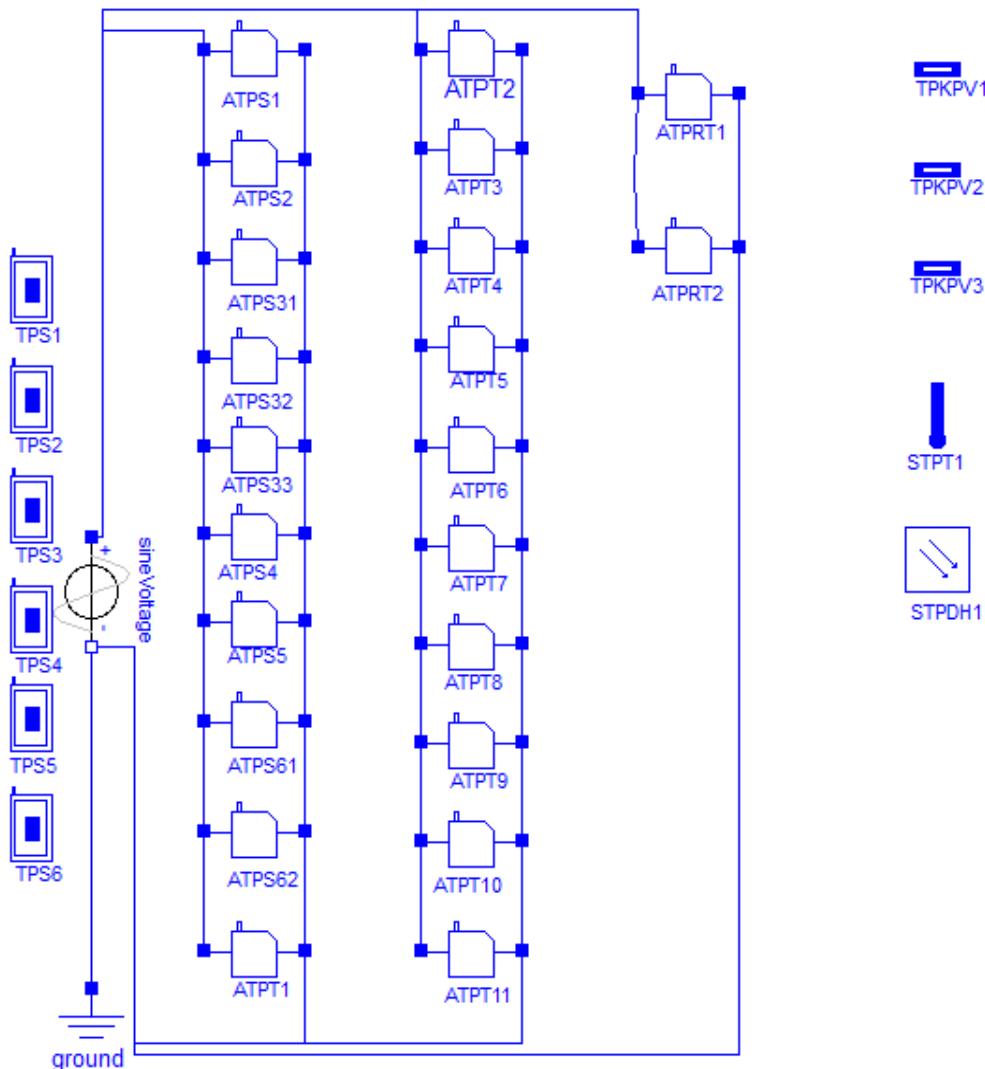


Figura 4.20: Representación de la Configuración de los Componentes del Sistema Eltako en el Tercer Piso.

```

model etercerpiso
  "Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se
  encuentran en el tercer piso."
  Modelica.SIunits.Power gastotercerpiso;
  equation
    gastotercerpiso=(if actorrele1.activ==true then actorrele1.gastoenergetico else 0)+(if actorrele2.activ==true
      then actorrele2.gastoenergetico else 0)+(if actorrele3.activ==true then actorrele3.gastoenergetico else 0)-
      +(if actorrele4.activ==true then actorrele4.gastoenergetico else 0)+(if actorrele5.activ==true then
      actorrele5.gastoenergetico else 0)+(if actorrele6.activ==true then actorrele6.gastoenergetico else 0)-
      +(if actorrele7.activ==true then actorrele7.gastoenergetico else 0)+(if actorrele8.activ==true then
      actorrele8.gastoenergetico else 0)+(if actorrele9.activ==true then actorrele9.gastoenergetico else 0)-
      +(if actorrele10.activ==true then actorrele10.gastoenergetico else 0)+(if actorrele11.activ==true then
      actorrele11.gastoenergetico else 0)+(if actorrele12.activ==true then actorrele12.gastoenergetico else 0)-
      +(if actorrele13.activ==true then actorrele13.gastoenergetico else 0)+(if actorrele14.activ==true then
      actorrele14.gastoenergetico else 0)+(if actorrele15.activ==true then actorrele15.gastoenergetico else 0)-
      +(if actorrele16.activ==true then actorrele16.gastoenergetico else 0)+(if actorrele17.activ==true then
      actorrele17.gastoenergetico else 0)+(if actorrele18.activ==true then actorrele18.gastoenergetico else 0)-
      +(if actorrele19.activ==true then actorrele19.gastoenergetico else 0)+(if actorrele20.activ==true then
      actorrele20.gastoenergetico else 0)+(if actorrele21.activ==true then actorrele21.gastoenergetico else 0)-
      +(if actorrele22.activ==true and sttemperatural.Temperatura<actorrele22.temperaturadeseada then
      actorrele22.gastoenergetico else 0);
end etercerpiso;

```

Código 4.9: Ecuación del Gasto Energético del Tercer Piso

Componentes del Package Eltako

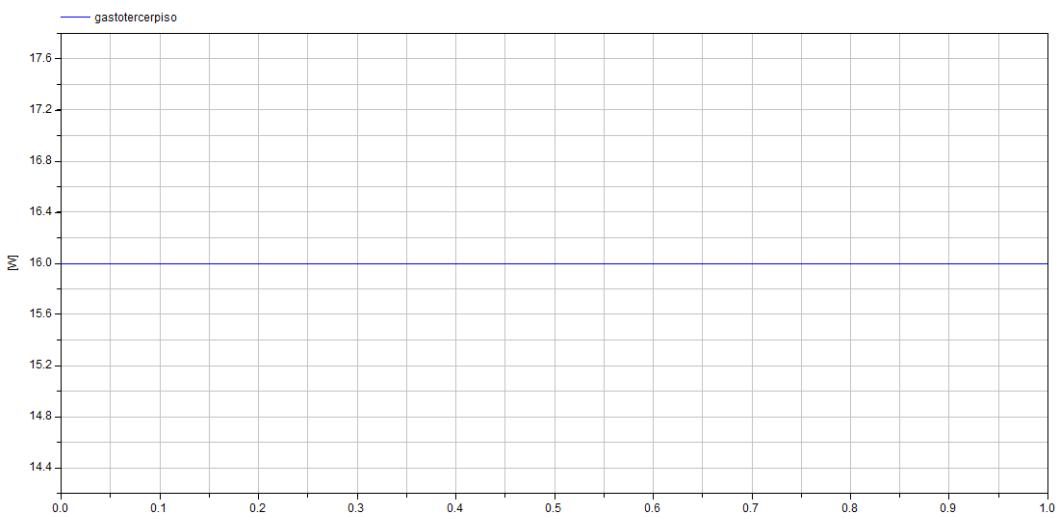


Figura 4.21: Simulación del Gasto Energético Tercer Piso

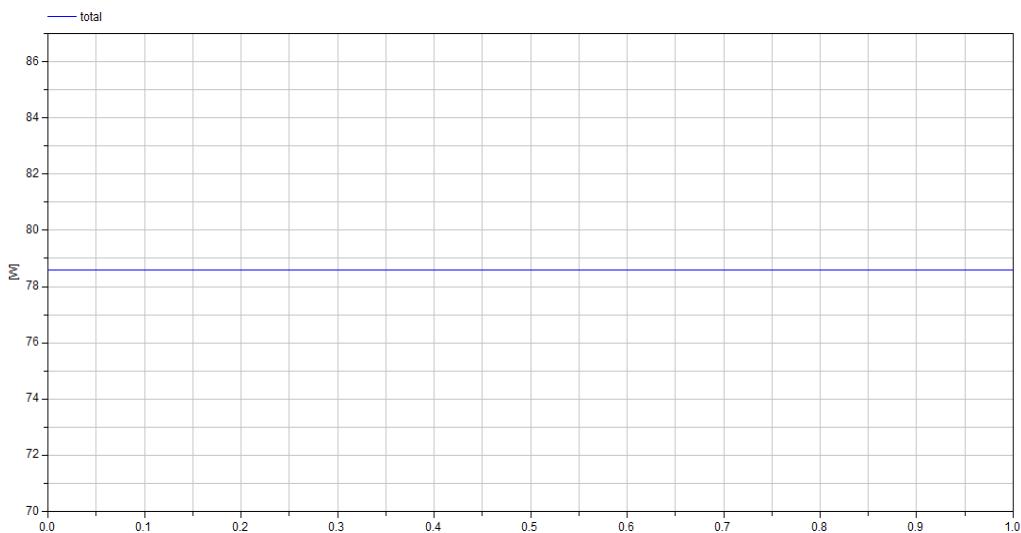


Figura 4.22: Simulación del Gasto Energético de la Vivienda Ejemplo.

4.7 Conclusiones

Por medio de la implementación del sistema eltako inalámbrico en la vivienda ejemplo, se simuló éste en Modelica, obteniendo el gasto energético del sistema. El sistema Eltako tiene sensores que utilizan la energía del medio para producir su energía (Harvesting Energy), pero también tiene sensores que toma energía de baterías, el ahorro energético de todas formas es alto, ya que el cambio ocurre como mínimo cada 3 años. Por tal motivo en Dymola se observan que los sensores no tienen conexión ninguna, solamente los actores están conectados a la energía eléctrica.

Para observar el gasto energético total de la vivienda se observa la variable total del modelo ge, en donde se encuentran los modelos de cada uno de los pisos y el servidor que hace posible la conexión del sistema con el mundo virtual. El resultado del gasto energético del sistema por estar listo para funcionar es de 78.6 W. Este resultado se puede ver en la figura 4.22, en la simulación de la variable total.

5. Sistema KNX / EIB (TP)

5.1 Introducción

Este sistema es descentralizado, lo que significa que la inteligencia está en cada uno de sus componentes. La Línea es la unidad más pequeña del sistema y está compuesta por un alimentador un sensor y un actuador. La estructura del sistema puede ser en línea, estrella y/o árbol. Cada componente posee una dirección que le sirve de identificación en todo el sistema, los componentes se comunican entre sí por medio de telegramas. Para evitar colisiones el sistema utiliza el protocolo CSMA/CA (acceso múltiple con escucha de portadora y evasión de colisiones), de esta forma todos los componentes escuchan la línea de transmisión, para así saber si el sistema está ocupado. En este capítulo se da a conocer con precisión el funcionamiento del sistema y se definen los componentes básicos del sistema desde el punto de vista de gasto energético, para así representar estos en Modelica y poder realizar la configuración del sistema en la vivienda ejemplo, para visualizar el gasto energético del sistema.

5.2 Comunicación Básica de un Sensor y un Actuador

El bus tiene un voltaje nominal de 24 voltios. El sistema funciona sin errores a partir de un voltaje de 21 voltios. Debido a que la información también transita por los mismos cables de la alimentación, el voltaje varía entre 29 y 21 voltios, así se representa la información

Sistema KNX / EIB (TP)

(unos y ceros). El participante es el que diferencia y separa la información de la alimentación.

Cada componente del sistema está dotado de un micro controlador y una memoria EEPROM, lo que hace que el sistema sea inteligente y descentralizado. Los componentes sensor son elementos que tienen como función el reconocer acontecimientos para transmitirlos en forma de telegramas.

El largo del bus y la distancia que deben tener los componentes entre sí es determinante para el buen funcionamiento del sistema. El máximo largo que puede tener una línea es de 1000 m. Entre dos participantes (sensor/actor) puede haber una distancia máxima de 700 m. Entre la alimentación y un participante (sensor/actor) puede haber una distancia máxima de 350m. Entre dos alimentadores debe haber una distancia mínima de 200m.

Por lo general un Sistema EIB/KNX está formado por varias líneas, para unir las diferentes líneas de un sistema se usa un acoplador de Línea. El acoplador puede ser de Linea, de campo o amplificador de Linea tiene como función separar las líneas galvánicamente y filtrar “direcciones”. Las Lineas quedan separadas físicamente, así si se presenta un cortocircuito entonces queda la linea inhabilitada y no todo el sistema.

Los participantes (sensor/actor) están compuestos de un acoplador de Bus (BA) y el terminal del bus. Es la parte inteligente que hace un sistema descentralizado, Aquí se envían y reciben datos y se asegura la alimentación para la parte electrónica del sistema.

Por medio de telegramas se difunde la información entre los participantes (sensor/actuador) del Sistema EIB/KNX. Los telegramas pueden ser de datos y de confirmación. El telegrama de datos está formado por un campo de control, la dirección de origen, la dirección final, un contador de ruta, el largo de la información, la información y el campo de comprobación.

Comunicación Básica de un Sensor y un Actuador

Cada participante (sensor/actuador) está equipado de un modulo de transmisión, en el medio par trenzado se conoce como TP UART, también conocido como Transceiver, porque transmite información al medio (TP) o recibe información del medio TP. Así que cuando envía bits al medio, tiene como función adecuar la señal de cada bit a una determinada frecuencia y una respectiva amplitud de voltaje para que estos viajen sobre la señal de 24 voltios.

Cuando recibe bits, su función es diferenciar el voltaje constante 24v de la información (bits).

Así que la función principal del transceiver es la de codificar o decodificar la señal que transita en el medio.

Cuando ocurre un acontecimiento como el movimiento de un interruptor, lo que hace el sensor es enviar un telegrama de datos. El receptor que puede ser una o varias bombillas, envían un telegrama de confirmación, para dar a entender que el telegrama de datos ha llegado a la dirección objetivo.

Cada componente tiene el modulo de transmisión que está equipado de un circuito integrado llamado TP UART. Este circuito integrado envía los telegramas de datos y de confirmación por medio de señales UART, las siglas de UART significan transmisión y recepción universal asíncrona. Las señales UART están formadas por:

- Bit de inicio SB 0
- 8 bits de Datos D0.....D7.
- Un bit de paridad PB.
- Un bit de parada EB 1.

Cuando se envía una señal UART, se esperan dos tiempos de envío de un bit, así que en total el envío de una señal UART se tarda 13 tiempos de envío de un bit. El tiempo que tarda entre la última señal

Sistema KNX / EIB (TP)

UART de un telegrama de Datos y el inicio del envío del telegrama de confirmación es de 15 tiempos del envío de un bit.

Para evitar conflictos a la hora de enviar información por el medio, se utiliza el protocolo CSMA/CA (acceso múltiple con escucha de portadora y evasión de colisiones), en el que todos los participantes pueden darse cuenta si el medio está ocupado. Si un participante va a enviar un telegrama de Datos, tiene que darse cuenta si el medio (bus) está libre.

Para ello el participante que vaya a enviar información pone atención durante 50 tiempos del envío de un bit. Si durante ese tiempo no ha habido transmisión de datos en el bus, es decir que se ha mantenido un voltaje de 24 voltios, significa que está libre “bus idle” así que se puede enviar la información. Puede darse el caso, en el que dos participantes accedan al bus al mismo tiempo. En este caso entran en el juego reglas que ayudan a definir, quien de los 2 participantes tiene mayor prioridad y por lo tanto gana el arbitraje del bus. Ganar el arbitraje significa ganarse el derecho de enviar el telegrama.

En el sistema EIB/KNX la prioridad del telegrama que se va a enviar, se define en el Byte del campo de control.

Para reducir los conflictos, se definen clases de acceso en el sistema EIB/KNX, para definir el arbitraje del bus. Las clases de acceso pueden ser 1 ó 2. Los telegramas con clase de acceso 1 son los de sistema y alarma, lo que significa que pueden enviar un telegrama cuando en el bus no hay actividad por un tiempo de envío de 50 bits.

Los telegramas de acceso 2 son los de alta prioridad y baja prioridad. Estos pueden enviar el telegrama cuando en el bus no hay actividad por un tiempo de envío de 53 bits.

5.2.1 Descripción del Telegrama de Confirmación

El Telegrama de confirmación lo compone un Byte véase figura 5.1, que sería el Byte del campo de control y puede tener tres posibilidades que son:

ACK el telegrama fue recibido correctamente.

BUSY el participante está ocupado.

NACK incorrecta recepción del telegrama.

	MSB D7 N1	D6 N0	D5	D4	D3 B1	D2 B0	D1	LSB D0
ACK	1	1	0	0	1	1	0	0
BUSY	1	1	0	0	0	0	0	0
NACK	0	0	0	0	1	1	0	0

Figura 5.1: Telegrama de Confirmación

- El Bit D5 (W) de repetición en el campo de control.
- Bits D3 (P1) y D2 (P0) del campo de control definen la prioridad.

5.2.2 Descripción del Telegrama de Datos

El telegrama de datos posee como Byte 0 el Byte del campo de control, en éste se define el formato del telegrama, la prioridad y si es un telegrama repetido, véase figura 5.2.

Byte del Campo de Control								
MSB								LSB
D7	D6	D5(W)	D4	D3 (P1)	D2(P0)	D1	D0	
1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	

Figura 5.2: Byte del Telegrama de Datos

Sistema KNX / EIB (TP)

Los bits D2 (P0) y D3(P1) definen la prioridad del telegrama como se muestra en la tabla 5.1.

D3(P1)	D2(P2)	Definición de la prioridad
0	0	Sistema
1	0	Alarma
0	1	Alta Prioridad
1	1	Baja Prioridad

Tabla 5.1: Definición de la Prioridad del Telegrama de Datos

El Bit W o bit 5 indica con un 1 que el telegrama se envía por primera vez. Con un 0 significa que es un telegrama repetido por lo tanto tiene una prioridad más alta.

Los demás bits del Byte del campo de control, definen el formato del Telegrama de Datos, véase figura 5.3:

- Formato Estándar: En el que el largo de la información puede ser hasta de 15 Bytes de datos.
- Formato Extendido: En el que el largo de la información puede ser hasta 255 Bytes de datos.
- Polling Frame: diferentes participantes pueden acceder a la información que se está enviando en el telegrama.

Byte 0 Campo de Control								
	7	6	5	4	3	2	1	0
Formato Estándar	1	0	x	1	x	x	0	0
Formato Extendido	0	0	x	1	x	x	0	0
Polling Frame	1	1	1	1	0	0	0	0

Figura 5.3: Definición del Formato

Comunicación Básica de un Sensor y un Actuador

Los Bytes 1 y 2 del Telegrama de Datos que contienen la dirección de origen, que esta escrita como dirección física, véase Figura 5.4.

	Byte 1								Byte 2							
	Dirección de origen								Dirección de Origen							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Participante (0...255)									1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
Línea (0....15)					1/0	1/0	1/0	1/0								
Campo (0....15)	1/0	1/0	1/0	1/0												

Figura 5.4: Byte 1 y 2 del Telegrama de Datos

El Byte 3 y 4 contienen la dirección del receptor. Esta puede ir escrita como dirección física o dirección lógica. Todo depende, si en el Byte 5 en el bit D7 o el bit más pesado se encuentra un uno o un cero. Si es la dirección lógica del receptor es porque se encuentra un 1, véase figura 5.5.

	Byte 3								Byte 4							
	Dirección del Receptor								Dirección del Receptor							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Grupo Secundario (0..255)									1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
Grupo de la mitad (0..7)						1/0	1/0	1/0								
Grupo principal (0...15)	0	1/0	1/0	1/0	1/0	1/0										

Figura 5.5: Dirección del Receptor del Telegrama de Datos

El Byte 5 también se le llama NPCI, que es el Protocolo de Red del Control de Información, véase figura 5.6.

Sistema KNX / EIB (TP)

Byte 5							
7	6	5	4	3	2	1	0
Números de Byte de la información. 0...15					n	n	n
Contador de Ruta 0 ...7	n	n	n	0..7			
Tipo de Dirección	1	Dirección Lógica					
	0	Dirección Física					

Figura 5.6: Byte 5 del Telegrama de Datos

Del bit 0 al 3 se define cuantos Bytes tiene la información (en formato estándar) que se va a utilizar, es decir la que va a definir qué es lo que se quiere hacer entre el sensor y el actuador.

Los bits 4 al 6 son el contador de ruta, normalmente al enviar el telegrama tienen el número 6, cada vez que pase por un acoplador de Linea o campo se disminuye en 1, cuando llega a cero no se sigue transmitiendo el telegrama. Si estos tres bits tiene el número 7, entonces significa que al pasar por los acopladores de linea y campo no se disminuye el valor.

Los Byte 6 y 7 véase figura 5.7. Aquí se encuentra la información que llegan a los actuadores, sensores y la información para controlar el sistema.

Byte 6								Byte 7									
								B3	B2	B1	B0	D5	D4	D3	D2	D1	D0
APCI	Lectura y los bits de Datos (D5...D0) no se toman en cuenta.	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Los bits (B3...B0) significan escritura, la información es D0.							0	0	0	0	x	x	x	x	x	x
	Los bits (B3..B0) significan Respuesta, la información (D5....D0)							1	0	0	0	0	0	0	0	0	D0
Tipo de UCD	Conectado							0	1								
	Desconectado							1	0								
Tipo de NCD	ACK (Reconocer)							1	0								
	NACK (Desconocer)							1	1								
No. de Bytes de Solo Datos (0...15)				n	n	n	n										
TCPI	Datos no numerados UDP	0	0														
	Datos enumerados NDP	0	1														
	Datos de Control no numerados UCD	1	0														
	Datos de Control enumerados NCD	1	1														

Figura 5.7: Byte 6 y 7 del Telegrama de Datos

Comunicación Básica de un Sensor y un Actuador

Los dos bits más pesados 7 y 6 del byte seis son del TCPI (Protocolo de Control del Transporte de la información) indican qué tipo de datos contiene la información. Estos pueden ser datos que contienen información (UDP y NDP) o datos que controlan la información (UCD y NCD). El bit 5, 4, 3 y 2 del byte 6 indican el número de bytes que contiene la información (B1, B0, D5, D4, D3, D2, D1, D0). Está activo solamente cuando están configurados en los bits 7 y 6 del byte seis el NDP o NCD. Si no, están con los números 0. Los bits 1 y 0 del byte 6 o B3 y B2 respectivamente, indican si con quien se comunican está conectado o no y si han reconocido la señal de confirmación o no. Los Bits B1 y B0 indican la orden que se está trabajando entre los participantes implicados, así puede ser:

Lectura, en la que se le ordena a los participantes implicados informar el estatus en el cual se encuentran, la información que se encuentra en los bits (D5, D4, D3, D2, D1, D0) no es relevante.

Escritura, significa que la información que deben tomar los participantes (actuadores) debe ser la que se encuentra en el bit D0, puede ser 1 que puede significar ON o un 0 que puede significar OFF.

Respuesta, los/el participante(s) que envía(n) la(s) orden(es) esta(n) esperando una respuesta que se encuentra en los bits D5, D4, D3, D2, D1, D0.

Si la información requiere más bits (D5....D0) entonces aquí se adicionan los Bytes necesarios para transportar la información en su totalidad.

El último Byte de un Telegrama es el campo de Comprobar o Comprobación, sirve para comprobar si la información que se envío está fuera de errores. Este byte se forma así, el bit D7 está formado por la paridad impar de todos los bits D7 de los Bytes del Telegrama, así consecutivamente se forma el Byte de comprobación.

La parte como se desarrolla la comunicación entre 2 participantes sensor y actuador, está simulada en Modelica en la librería EIB2. Esta no es una sublibreria de DOMOSYS.

5.3 Descripción de la Librería SBKNX en Modelica

La Librería SBKNX es una sub-librería de DOMOSYS y describe los componentes desde el punto de vista de gasto energético del sistema KNX/EIB TP. El sistema básico KNX/EIB está representado por la fuente que es la que alimenta las lineas del sistema. Dependiendo del número de lineas entonces se determina el número de fuentes. Cada línea está separada físicamente del resto del sistema por medio del acoplador de línea y el último componente de la librería es el acoplador de bus, que forma parte de los sensores y los actuadores. Es el lugar donde se originan ordenes o se decodifican las ordenes, para el desarrollo de una acción determinada. Es decir, un acoplador de bus puede tener un modulo de aplicación que puede ser un sensor de temperatura, así que allí se genera un telegrama en donde se envía los datos de la temperatura en un determinado momento. De esta forma los módulos que varían pueden ser sensores o actuadores, pero todos tienen un acoplador de bus.

- La fuente o el alimentador FKNX
- El acoplador de linea ALKNX
- El acoplador de bus. ABKNX. Este último lo contienen todos los participantes del sistema ya sean sensores o actuadores.

5.3.1 La fuente o el Alimentador FKNX

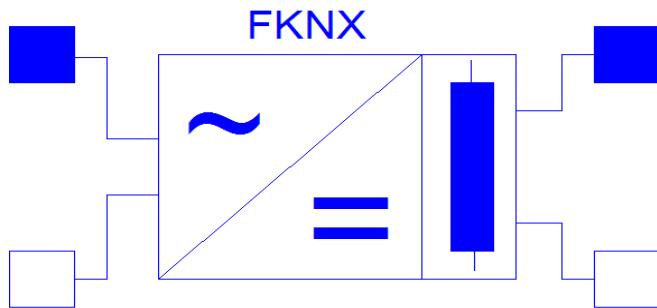


Figura 5.8: Fuente de Alimentación KNX/EIB en Modelica

Toma la corriente de la red eléctrica y por medio de un transformador y un corrector de onda entrega un voltaje DC, como se puede observar en el código 5.1. El voltaje nominal que entrega este elemento es de 24 voltios, pero este varia entre 21 y 30 voltios, así se transmite la información (unos y ceros). Su función principal es alimentar los acopladores de linea y los acopladores de bus.

Este elemento tiene la capacidad de alimentar hasta 10 acopladores de bus de una linea. Es un elemento estándar en el mercado de los componentes EIB y la capacidad de elementos que puede alimentar en una linea se mide por medio de la corriente que entrega, así los hay de 320 mA y de 640 mA. Cada componente de linea tiene un gasto de corriente de más o menos 10 mA. En la simulación con Dymola vamos a suponer que la fuente entrega 640 mA. La potencia perdida varía según en fabricante entre 4,5 vatios y 14 vatios, se escoge la media que es de 9,5 vatios.

Sistema KNX / EIB (TP)

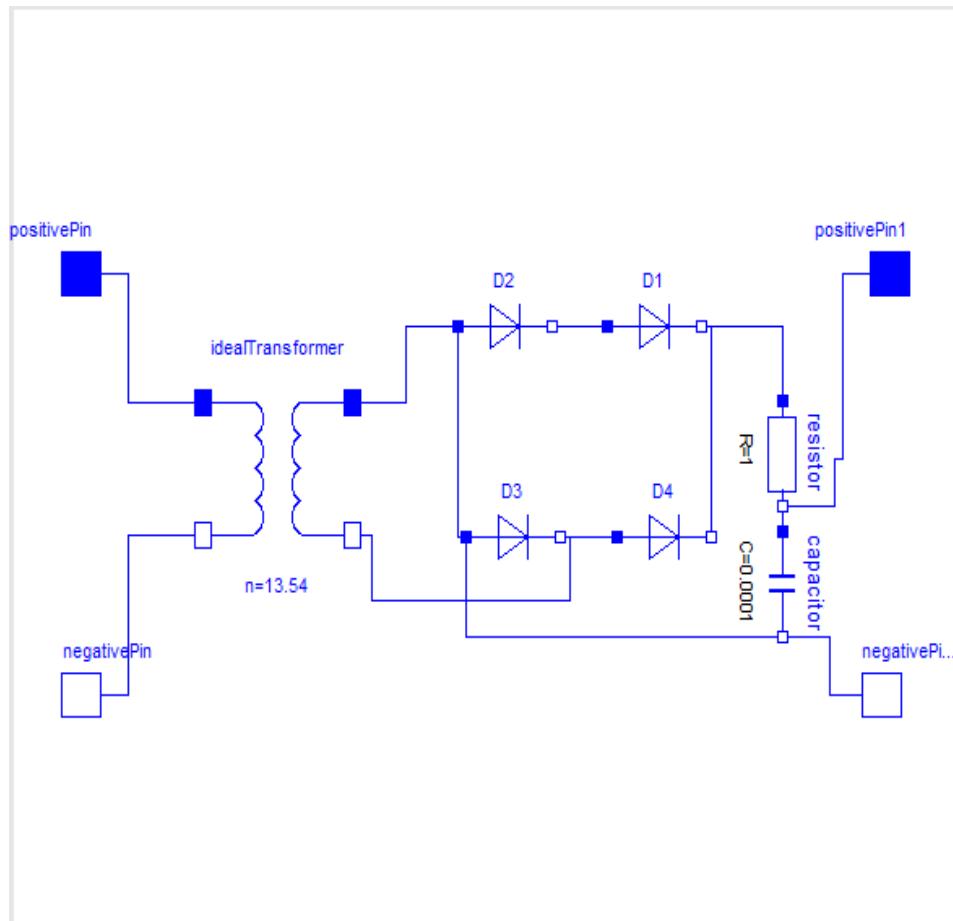


Figura 5.9: Representación Gráfica en Modelica del Diseño de la Fuente en el Sistema KNX/EIB

Descripción de la Librería SBKNX en Modelica

```
model FKNX "Fuente del Sistema KNX/EIB"

  Modelica.Electrical.Analog.Interfaces.PositivePin positivePin  a;
  Modelica.Electrical.Analog.Interfaces.NegativePin negativePin  a;
  a;

  Modelica.Electrical.Analog.Ideal.IdealDiode D2  a;
  Modelica.Electrical.Analog.Ideal.IdealDiode D1  a;
  Modelica.Electrical.Analog.Ideal.IdealDiode D3  a;
  Modelica.Electrical.Analog.Ideal.IdealDiode D4  a;
  Modelica.Electrical.Analog.Interfaces.PositivePin positivePin1  a;
  Modelica.Electrical.Analog.Interfaces.NegativePin negativePin1  a;
  Modelica.Electrical.Basic.Resistor resistor  a;
  Modelica.Electrical.Analog.Basic.Capacitor capacitor(C=0.0001)  a;
  Modelica.Electrical.Analog.Ideal.IdealTransformer idealTransformer(n=13.54)  a;
equation
  connect(D2.n, D1.p)  a;
  connect(D2.p, D3.p)  a;
  connect(D3.n, D4.p)  a;
  connect(D1.n, D4.n)  a;
  connect(resistor.n, capacitor.p)  a;
  connect(D1.n, resistor.p)  a;
  connect(D3.p, capacitor.n)  a;
  connect(capacitor.n, negativePin1)  a;
  connect(resistor.n, positivePin1)  a;
  connect(negativePin, idealTransformer.n1)  a;
  connect(positivePin, idealTransformer.p1)  a;
  connect(idealTransformer.p2, D2.p)  a;
  connect(idealTransformer.n2, D3.n)  a;
end FKNX;
```

Código 5.1: Definición de FKNX en Modelica

5.3.2 Acoplador de Línea ALKNX

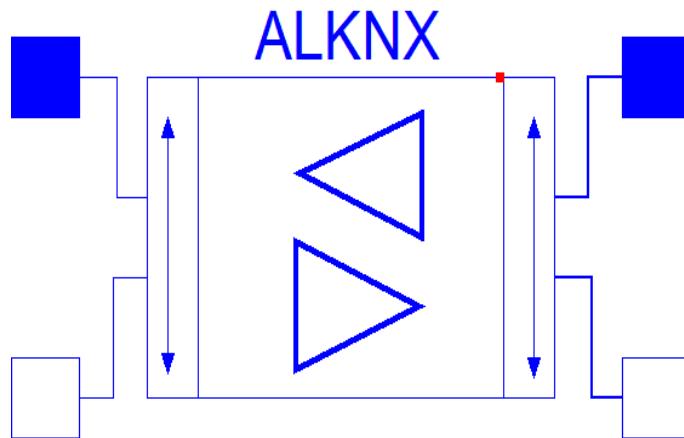


Figura 5.10: Representación del Acoplador de Linea en Modelica.

Se encuentra en cada línea como su nombre lo indica. Tiene como función la de separar las impedancias de la linea con el resto del

Sistema KNX / EIB (TP)

sistema y puede ser que tenga la función filtrar los datos. De esta forma repite y transmite la información, si va dirigida a participantes que se encuentren en la respectiva Linea. La perdida de potencia máxima de este elemento varía según el constructor. Por ejemplo el acoplador de Linea para el sistema TP de ABB tiene un gasto de potencia de 250 mW.

El diseño en Modelica se compone de un separador de impedancias con filtros tanto a la entrada como a la salida, para que la señal de entrada sea igual a la señal de salida, como se observa en el código

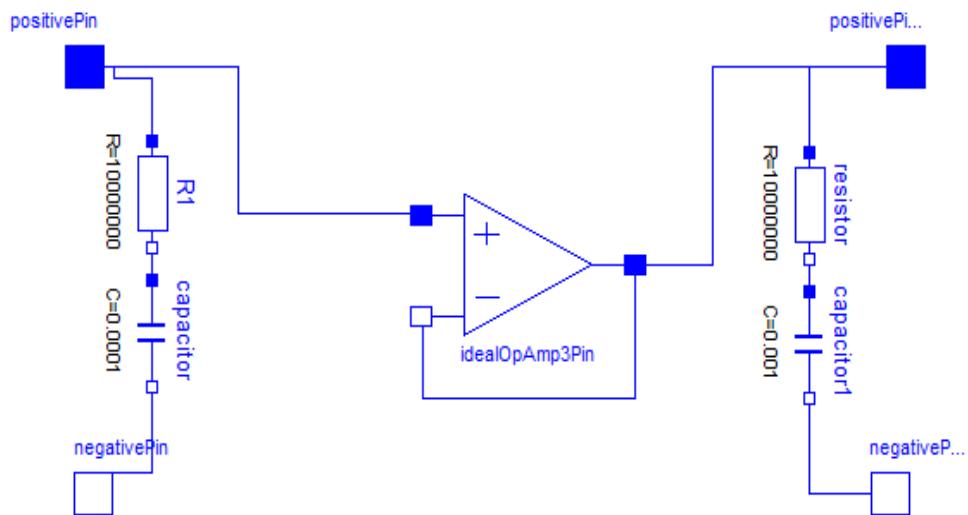


Figura 5.11: Representación Gráfica en Modelica del Diseño del Acoplador de Linea en el Sistema KNX/EIB

```
model ALKNX
    "Acoplador de Linea, separa las impedancias entre dos lineas."
    Modelica.Electrical.Analog.Interfaces.PositivePin positivePin  a;
    Modelica.Electrical.Analog.Interfaces.NegativePin negativePin  a;
    Modelica.Electrical.Analog.Interfaces.PositivePin positivePin1  a;
    Modelica.Electrical.Analog.Interfaces.NegativePin negativePin1  a;
    Modelica.Electrical.Analog.Ideal.IdealOpAmp3Pin idealOpAmp3Pin  a;
    a;

    Modelica.Electrical.Analog.Basic.Resistor R1(R=10000000)  a;
    Modelica.Electrical.Analog.Basic.Capacitor capacitor(C=0.0001)  a;
    Modelica.Electrical.Analog.Basic.Resistor resistor(R=10000000)  a;
    Modelica.Electrical.Analog.Basic.Capacitor capacitor1(C=0.001)  a;
equation
    connect(positivePin, idealOpAmp3Pin.in_p)  a;
    connect(idealOpAmp3Pin.out, positivePin1)  a;
    connect(idealOpAmp3Pin.in_n, idealOpAmp3Pin.out)  a;
    connect(positivePin, R1.p)  a;
    connect(R1.n, capacitor.p)  a;
    connect(negativePin, capacitor.n)  a;
    connect(positivePin1, resistor.p)  a;
    connect(resistor.n, capacitor1.p)  a;
    connect(capacitor1.n, negativePin1)  a;
end ALKNX;
```

Código 5.2: Definición del ALKNX en Modelica.

5.3.3 El Acoplador de Bus ABKNX

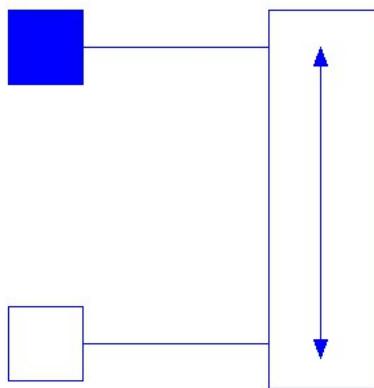


Figura 5.12: Representación del Acoplador de Bus en Modelica.

Este componente está representado en la librería SBKNX como una resistencia. Ya que para determinar la cantidad de potencia que se le entrega al sistema y la cantidad de potencia que se pierde se necesita tan solo una resistencia que represente la impedancia del sistema. El objetivo principal de la representación del acoplador de bus como de los otros componentes, es poder observar el gasto energético, de esta forma se observa el componente como un equivalente de Norton o Thévenin. La parte inteligente del sistema está compuesta por un modulo de transmisión y un controlador. El módulo de transmisión tiene como función: proteger el componente de polarizaciones inversas y altas temperaturas y entregar al controlador la información en 1 y 0.

El controlador tiene como función: se encarga de identificar si el mensaje va dirigido a la respectiva dirección. Leer/Enviar telegramas y realizar las instrucciones. La descripción del funcionamiento de la comunicación de este sistema está en la librería EIB en Dymola. El acoplador de bus está acompañado de un módulo de aplicación, éste es el que define si el componente del sistema es un sensor o un actuador.

La potencia que toma este componente para su funcionamiento es de 150 mW, por tal motivo se ha tomado la resistencia de 834 ohmios.

```

model ABKNX
    "Parte inteligente del sistema, decodifica y codifica la información. "
    Modelica.Electrical.Analog.Interfaces.PositivePin P a;
    Modelica.Electrical.Analog.Interfaces.NegativePin N a;
    a;
    Modelica.Electrical.Analog.Basic.Resistor RB(R=384) a;
    parameter Modelica.SIunits.Current corriente=0.0625;
    parameter Modelica.SIunits.Power Dp=0.150;
    Modelica.SIunits.Voltage voltaje;

equation
    voltaje=P.v-N.v;
    connect(P, RB.p) a;
    connect(RB.n, N) a;
end ABKNX;

```

código 5.3: Definición de ABKNX en Modelica.

5.4 Desarrollo de la Implementación del Sistema KNX/EIB en la Vivienda Ejemplo

En las gráficas de la representación de la vivienda se observa el cableado de color verde, que es el cable por donde se comunican todos los componentes del Sistema KNX. Los acopladores de linea y la fuente se observan en el código Modelica, que acompaña a las gráficas de cada uno de los pisos de la vivienda. Los elementos finales que son los sensores y los actuadores están enumerados tanto en la gráfica como en el código. Cada piso contiene la representación gráfica en el código Modelica, en donde se puede observar los acopladores de linea y las fuentes. Por medio de la

5.4.1 Sótano

Por medio de la representación de la vivienda, se puede observar la enumeración de los sensores y los actuadores. En la representación gráfica del código Modelica se observa que por cada 10 componentes, sean sensores o actores se debe suministrar una fuente que puede alimentar hasta 10 sensores y actuadores. El acoplador de linea

Sistema KNX / EIB (TP)

tiene como función separar galvánicamente el sistema, por lo tanto esta presente cada vez que haya una fuente de alimentación.

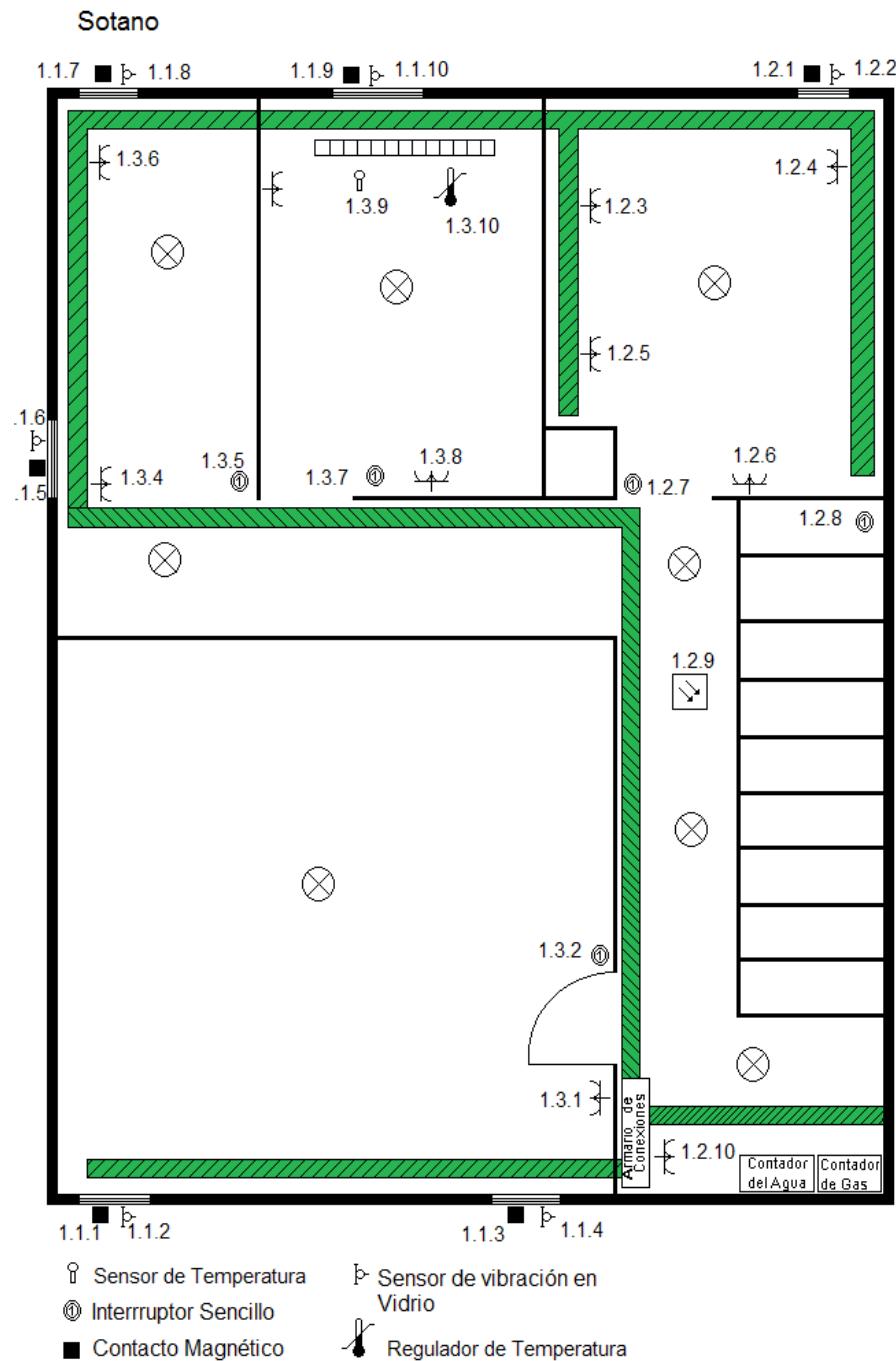


Figura 5.13: Sótano con el Sistema KNX

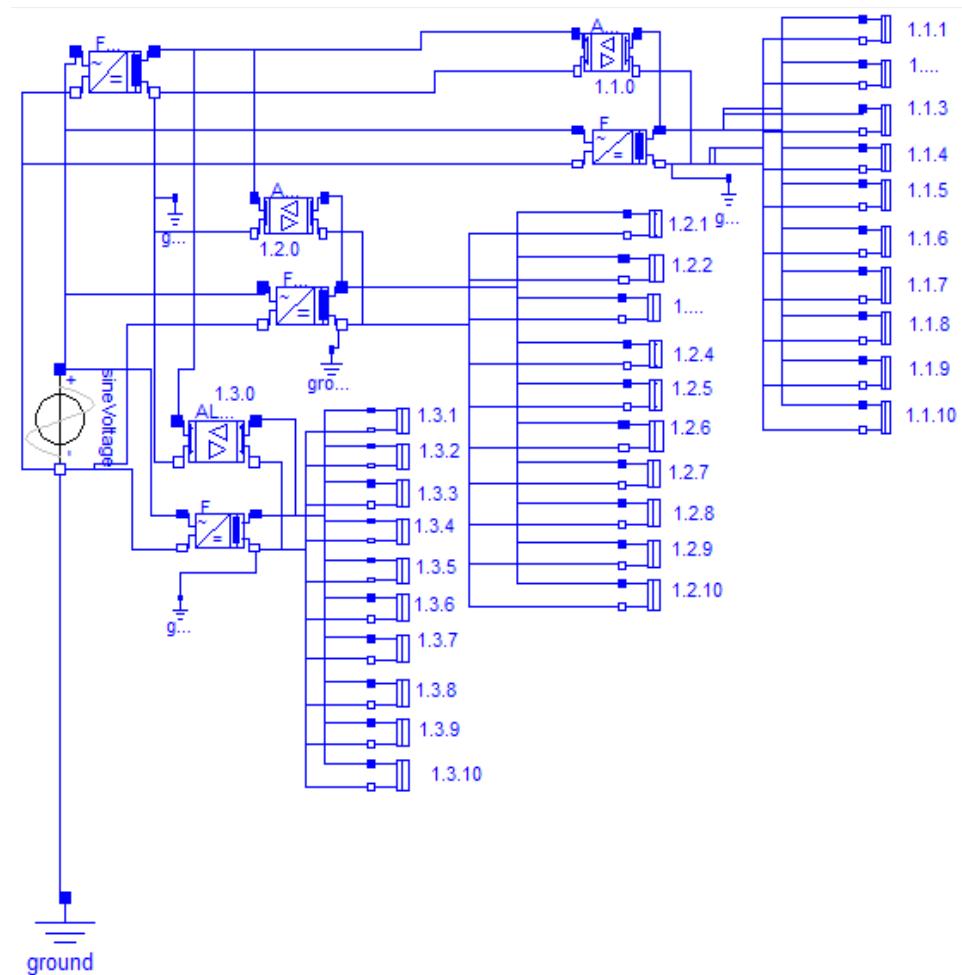


Figura 5.14: Representación Gráfica de los Componentes del Sistema KNX en el Sótano en Modelica.

```
model KNXsotano
  parameter Integer componentes=30;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;
equation
  NAL=componentes/10;
  AK1=integer(NAL);
  gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);
end KNXsotano;
```

Código 5.4: Ecuación del Gasto Energético de los Componentes del Sótano en Modelica.

5.4.2 Primer Piso

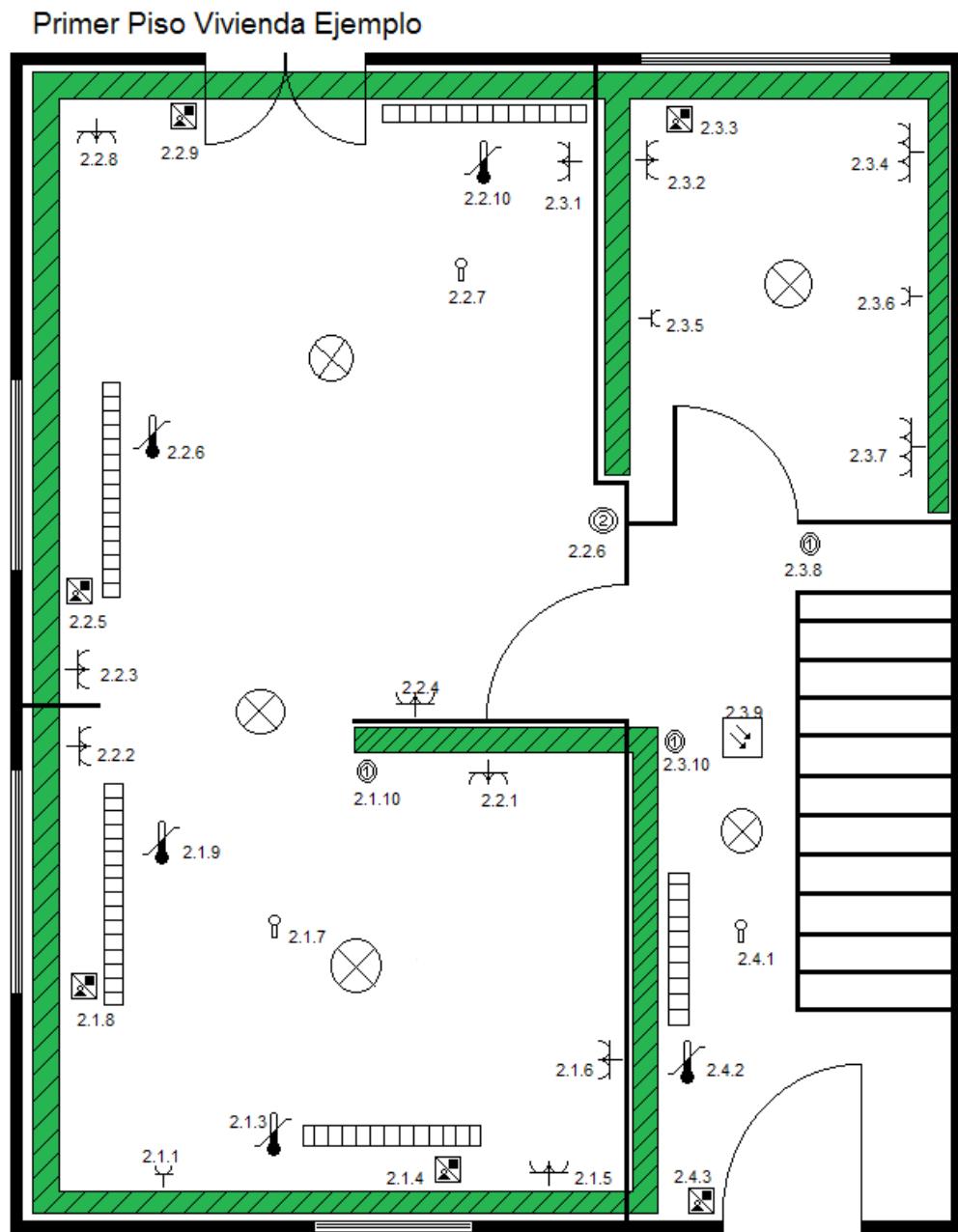


Figura 5.15: Primer Piso con Sistema KNX

Sistema KNX / EIB (TP)

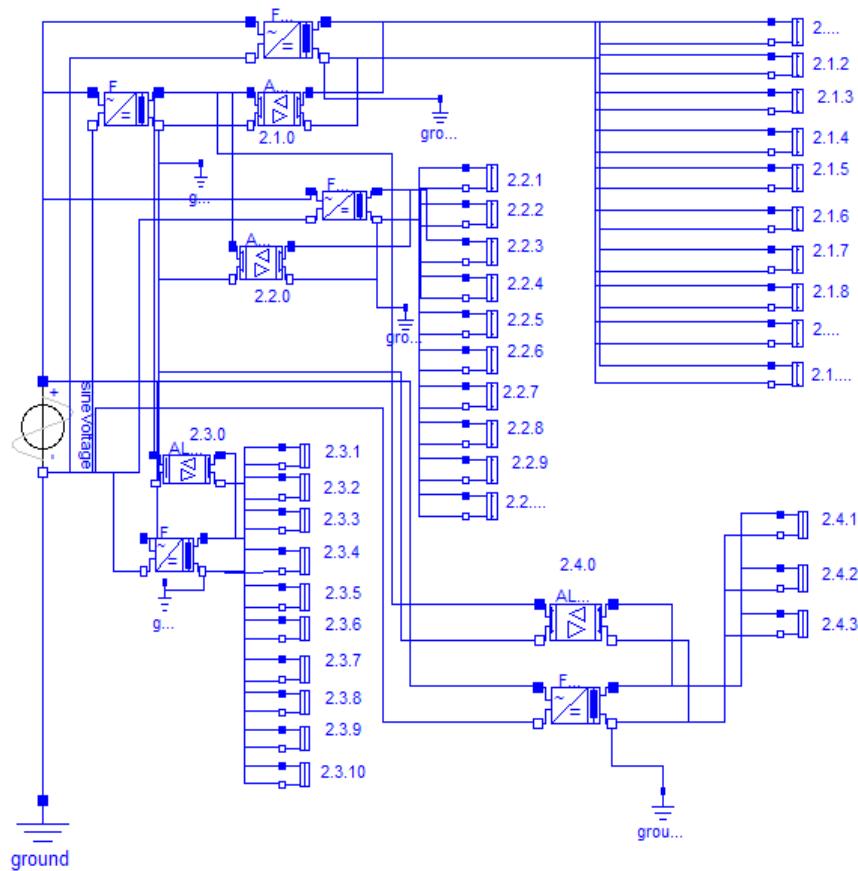


Figura 5.16: Representación Gráfica de los Componentes del Sistema KNX del Primer Piso en Modelica.

```

model PPKNX
  parameter Integer componentes=33;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;
  
equation
  NAL=componentes/10;
  AK1=integer(NAL);
  gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);

  
end PPKNX;

```

Código 5.5: Ecuación del Gasto Energético Primer Piso

5.4.3 Segundo Piso

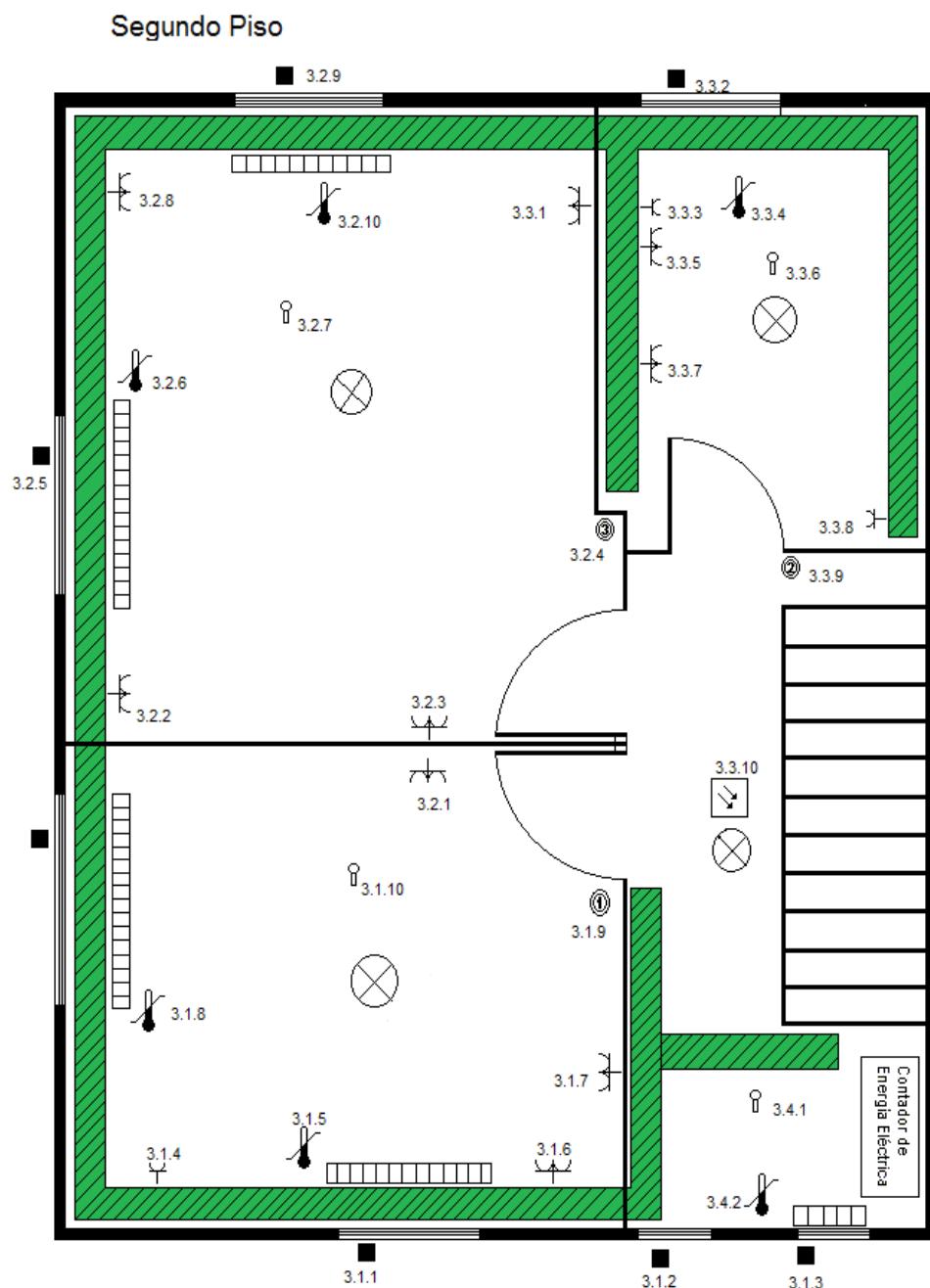


Figura 5.17: Segundo Piso con Sistema KNX

Sistema KNX / EIB (TP)

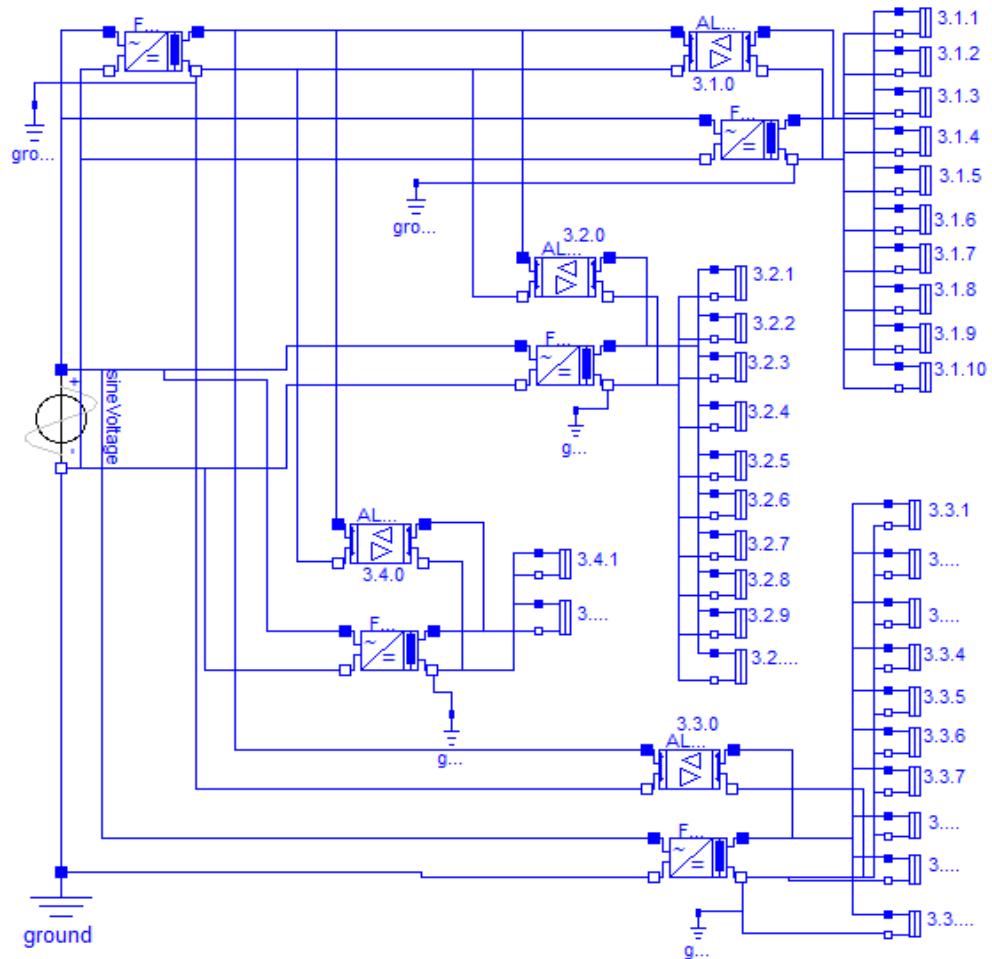


Figura 5.18: Representación Gráfica del Segundo Piso con los Componentes del Sistema KNX en Modelica.

```
model SPKNX
  parameter Integer componentes=32;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;
  
equation
  NAL=componentes/10;
  AK1=integer(NAL);
  gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);
  
end SPKNX;
```

Código 5.6: Ecuación del Gasto Energético del Segundo Piso
en Modelica.

5.4.4 Tercer Piso

Sistema KNX / EIB (TP)

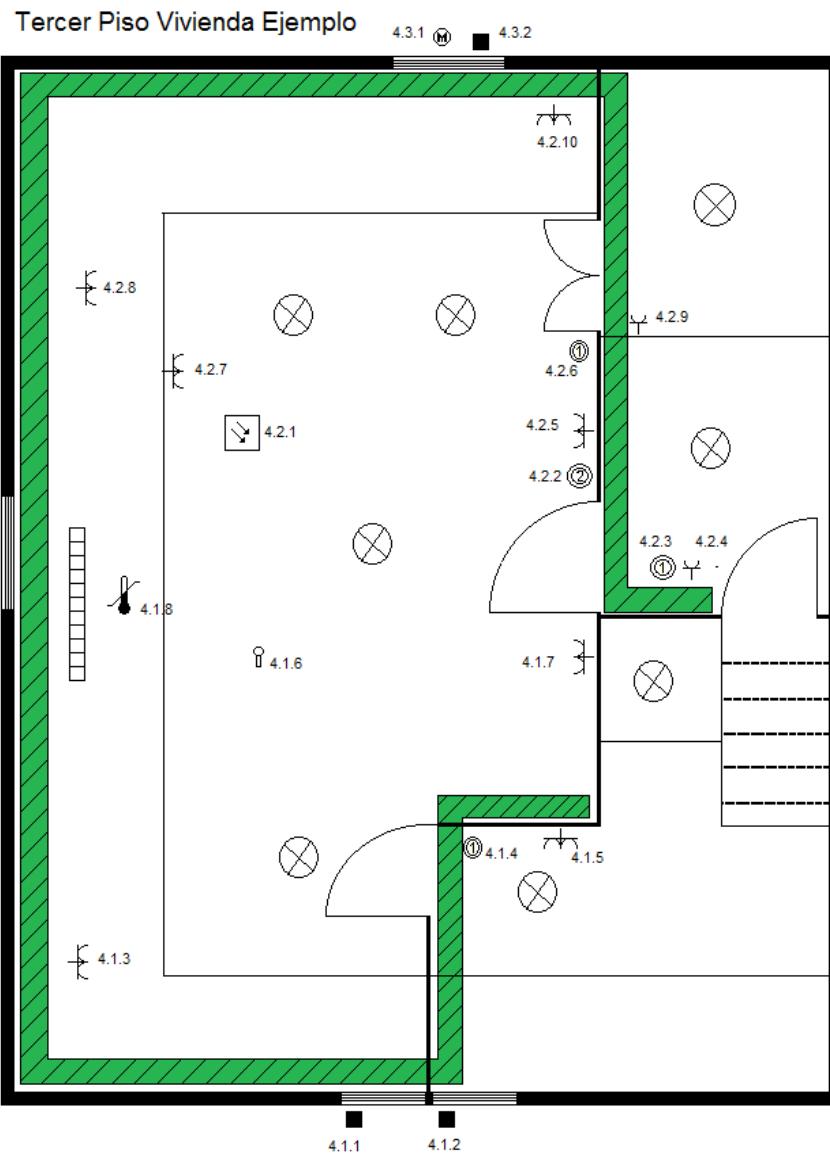


Figura 5.19: Tercer Piso con el Sistema KNX.

Desarrollo de la Implementación del Sistema KNX/EIB en la Vivienda Ejemplo

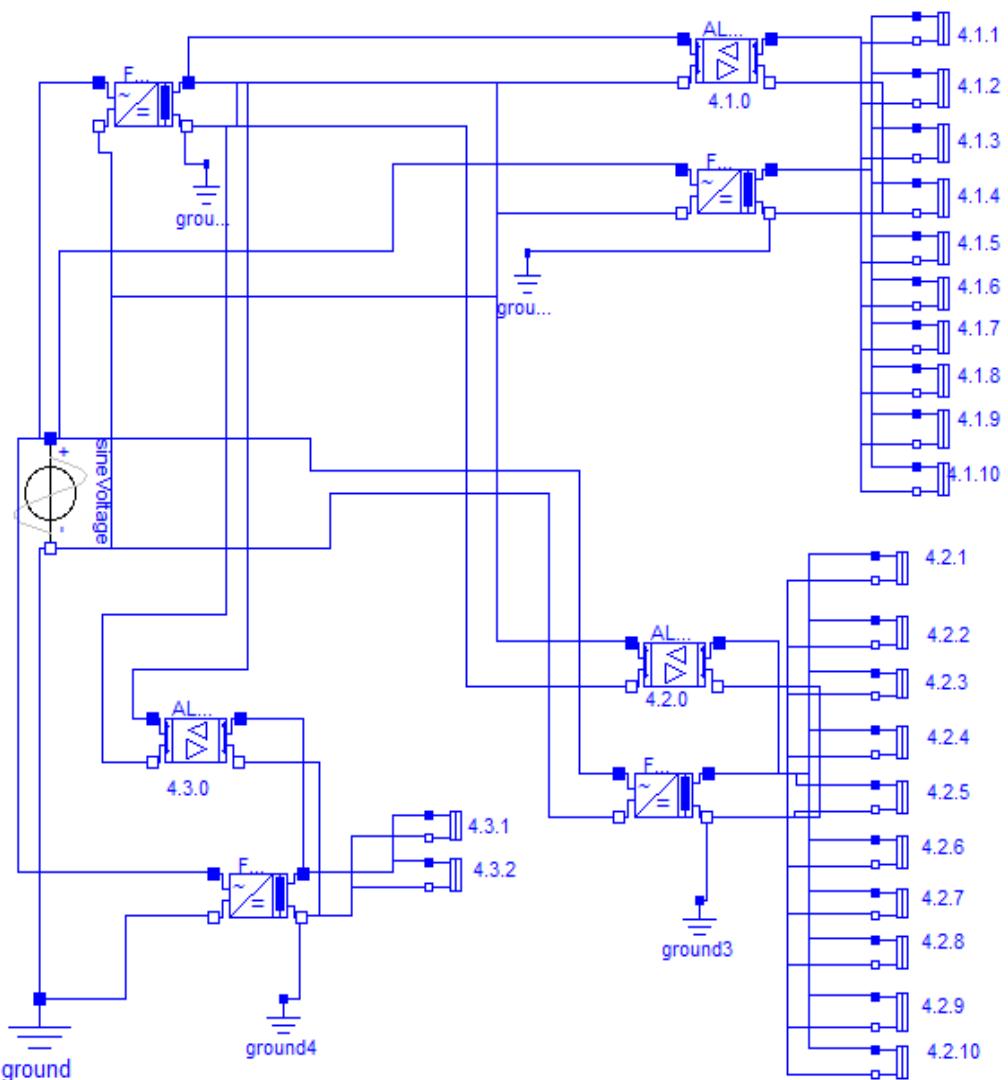


Figura 5.20: Representación Gráfica en Modelica del Sistema KNX en el Tercer Piso.

Sistema KNX / EIB (TP)

```
model TPKNX
  parameter Integer componentes=22;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;

  equation
    NAL=componentes/10;
    AK1=integer(NAL);
    gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);
end TPKNX;
```

Código 5.7: Ecuación del Gasto Energético del Tercer piso.

En las figuras anteriores se observa que el sistema KNX se ha implementado en cada piso, de la forma más cercana a la realidad, De esta forma se puede obtener el gasto energético del sistema implementado en toda la vivienda ejemplo. El modelo `gt` contiene la sumatoria del gasto energético de los cuatro pisos de la vivienda ejemplo. En la figura 5.21 está el resultado del valor del gasto energético de la vivienda ejemplo.



Figura 5.21: Simulación del Gasto Energético de la Vivienda Ejemplo.

5.5 Conclusiones

Los resultados dan a entender que el gasto energético del sistema KNX TP en la vivienda unifamiliar dada como ejemplo es de 162.8 W. Lo que significa que el sistema tiene un consumo energético no despreciable. Que equivale más o menos al encendido permanente de una bombilla de 150 W. Pero se recibe a cambio confort y seguridad.

6. Sistema Wago SPS

750

6.1 Introducción

La empresa Wago ubicada al norte de Alemania, conocida por sus innovadores conectores y los sistemas de controladores lógicos programables PLC. Estos se han utilizado en un principio para la automatización de procesos industriales y hoy en día también se utilizan en la domótica. Estos PLC están formados por un controlador y las características para programarlo están definidas en la norma IEC61131-3. Hay 5 lenguajes para programar estos sistemas, como el escalera, lista de instrucciones, el estructurado, diagrama de bloques y diagrama secuencial. El sistema que se va a analizar es el controlador lógico programable 750. Este sistema está compuesto por un controlador o acoplador de bus y los módulos que van a manejar los diferentes datos del sistema. El sistema Wago SPS 750 se caracteriza por poder manipular otros sistemas domóticos dentro de él, de esta forma sería un sistema que contendría subsistemas como:

- KNX
- DALI
- EnOcean-Funktechnik
- MODBUS
- M-Bus
- MP-Bus

- SMI
- LonWorks
- E-Mail
- SMS
- y otras aplicaciones

Este sistema es muy flexible, ya que por medio de programas se pueden realizar diferentes funciones que se acomodan a la necesidad de cada vivienda, la dificultad esta en el manejo de la programación del PLC. El controlador o acoplador de bus contiene librerías para desarrollar programas que se concentran a las necesidades de cada usuario.

6.2 Controlador o Acoplador de Bus

Está formado por una CPU que puede ser programada con las instrucciones para los controladores lógicos programables. Por una memoria, que está dividida en los datos de entrada y los datos de salida.

El intercambio de información ocurre entre las tres interfaces:

- Interfaz entre los módulos y el controlador del bus de campo.
- Entre el bus de campo maestro y el controlador de bus de campo.
- Los datos y la CPU.

El controlador de bus de campo actúa como un esclavo y es un software que por medio del bus de campo maestro dirige el comportamiento de este. En la figura 6.1 se observa la representación del sistema, el controlador lógico programable es el controlador de bus SPS 750. La comunicación con el maestro, que en la mayoría de los casos es un computador, ocurre por medio de la comunicación con los protocolos Ethernet, MODBUS/TCP o MODBUS/UDP.

Sistema Wago SPS 750

El controlador de bus de campo en sí contiene programas que están ejecutándose todo el tiempo. Éste tiene un contacto permanente con los módulos de bus, ya que dependiendo del valor de las entradas, estos programas reaccionan con datos de salida. Por tal motivo es un esclavo ya que necesita de una conexión de bus con un computador y el respectivo programa como WAGO-I/O PRO para cambiarlos o realizar nuevos programas.

El controlador de bus de campo se comunica directamente con los módulos de bus por medio de 6 contactos que tienen cada uno de ellos, en donde se transmiten los datos y se alimenta el sistema interno de los módulos de Bus. El máximo número de módulos de bus que pueden estar unidos al controlador es de 64 sin extensión, pero si hay extensión entonces pueden ser hasta 250.

En sí el intercambio de información ocurre entre el bus de campo maestro y los módulos de bus a través del controlador del bus de campo.

El controlador de bus 750 tiene las siguientes posibles conexiones y éstas pueden ocurrir al mismo tiempo:

- 3 Conexiones HTTP
- 15 Conexiones sobre el MODBUS
- 128 Conexiones Ethernet
- 2 Conexiones con el software WAGO-I/O-PRO
- 2 Conexiones FTP
- 2 Conexiones SNMP

Controlador o Acoplador de Bus

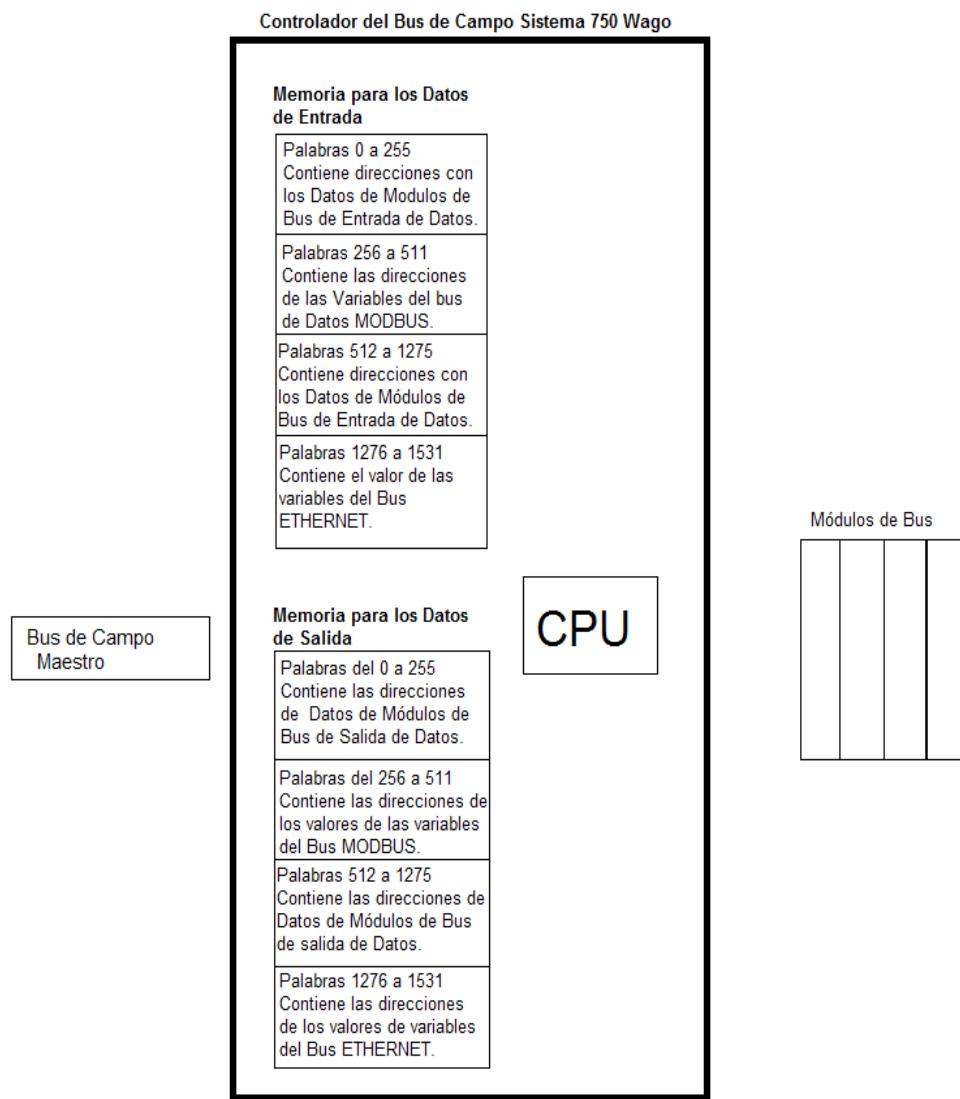


Figura 6.1: Representación del Controlador SPS 750 de la Empresa WAGO (www.wago.com).

6.2.1 Gasto Energético

El acoplador de bus o controlador de bus de campo se alimenta de 24 voltios dc. Según la serie puede ser por ejemplo el 750-881 que toma 380 mA y 5 voltios. También tiene la capacidad de alimentar un número determinado de módulos ya que puede entregar hasta 1620 mA. Si la cantidad de módulos sobrepasa esta cantidad, entonces se debe recurrir a un módulo de alimentación. Éste va a entregar hasta una corriente determinada, para la alimentación de cierto nú-

mero de módulos. Para determinar cuántos módulos de alimentación se necesitan para un ejemplo concreto, se tiene que tener en cuenta que cada entrada digital consume 2 mA y cada relé consume 90 mA. De esta forma al planificar la cantidad de módulos necesarios para una vivienda determinada, hay que tener en cuenta la alimentación de los mismos.

6.3 Módulos del Controlador del Bus de Campo

Los módulos del controlador de bus, se comunican con el controlador lógico programable por medio de los protocolos MODBUS/TCP o Ethernet/IP. Estos módulos pueden ser de entrada o de salida de datos, los datos de salida llegan a los actuadores para desarrollar una orden y los de entrada son en la mayoría los sensores e interruptores. Por ejemplo para mover una persiana, primero tiene que nacer la orden, que puede darse desde un interruptor o un sensor, estos producen datos de entrada y sería un módulo específico. El que realiza la acción, sería otro módulo, el cual contiene los datos de salida.

Todo controlador de bus 750 debe tener un módulo final que sirve para cerrar el nodo y de esta forma se asegura y garantiza la transmisión de los datos, este se localiza siempre al final de todos los módulos de bus que contenga el controlador de bus 750.

Los módulos de entrada/salida pueden ser digitales o análogos, depende de las señales que reciban/envíen de/a los sensores/actuadores.

Para implementar un sistema simple de encender o apagar un bombillo se necesita básicamente dos módulos de bus, el que recibe la señal del interruptor y el que da la orden de encender.

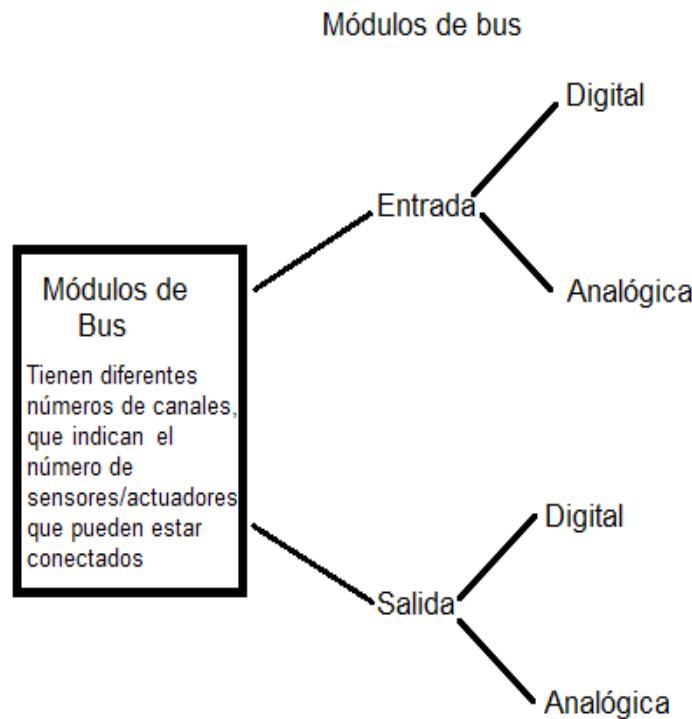


Figura 6.2: Definición del Módulo de Bus.

6.4 División de los Módulos del Controlador de Bus

Todos los módulos de bus poseen contactos donde se transmiten los datos y se alimenta el sistema. El sistema se alimenta de 24 voltios. Los módulos tienen 6 contactos por los cuales se transporta la información y se alimenta el sistema con 5 voltios, que son tomados de los 24 voltios.

6.4.1 Módulos de Entrada Digital

Los hay de 2, 4, 8 ó 16 canales. Por ejemplo un módulo con 2 canales, podría tener conectado un sensor de contacto de ventana, y si los contactos no están unidos es 5 Voltios que sería un 1 y si están unidos es 0 que serían 0 voltios. Como este sistema de controladores lógicos programables nacen para automatizar procesos industria-

Sistema Wago SPS 750

les, los niveles altos pueden estar dados con otros voltajes como de 12, 24, 42, 48, 60, 110, 220 voltios DC y también AC de 120/230 voltios.

6.4.2 Módulos de Entrada Analógica

La mayoría de los módulos están compuestos de 2 canales, donde pueden entrar señales de 0 a 20 mA, 4 a 20 mA, 0 a 10 voltios, 0 a 30 voltios y se pueden conectar termocuplas y termo resistencias. La resolución con la que se transmiten estos datos al sistema es de 12 bits.

6.4.3 Módulos de Salida Digital

Pueden tener 1, 2, 4, 6, 8 ó 16 canales donde se puede accionar un interruptor o manipular determinado componente.

6.4.4 Módulos de Salida Analógica

Pueden tener 2 o 4 canales de salida y puede salir una señal de 0 a 20 mA, 4 a 20 mA, 0 a 10 voltios.

El gasto energético de los módulos por estar listo para funcionar depende de la cantidad de canales que tengan.

Así los módulos de entradas digitales si tienen 2 canales entonces tiene una gasto de 3,7 mA por 5 voltios. Si tiene 16 canales entonces tendrá un gasto de 25 mA por 5 voltios. Los módulos de salida digital tienen gasto de potencia de 3 mA a 100 mA por 5 voltios. Los módulos análogos de entrada tienen un gasto de potencia de 70 mA por 5 voltios, casi todos son de 2 canales.

6.5 Descripción de la Librería Wago en Modelica

El sistema Wago de la librería en Modelica, contiene los componentes necesarios para desarrollar la domótica de la vivienda ejemplo. Sus componentes se asemejan a los reales en el aspecto de la cantidad de potencia que toma el sistema para estar listo para funcionar. El principal objetivo es conocer por medio de la implementación del sistema en una vivienda ejemplo, cuanta potencia consume el sistema por estar listo para funcionar.

6.5.1 Modulo de Bus mb

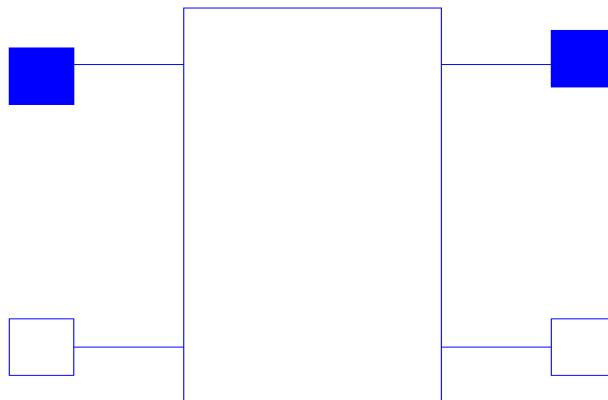


Figura 6.3: Representación del Módulo de Bus en Modelica

Este componente define los módulos de buses a nivel general del sistema Wago SPS 750. Con los parámetros booleanos digital y salida, se define qué clase de modulo es. Si digital es verdadero “true”, significa que el modulo de bus es digital, si digital es falso “false”, significa que el módulo de bus es análogo. Si salida es verdadero, significa que está conectado a un actuador (bombilla, motor de persianas, regulador de temperatura, regulador de luz, etc;). Si salida es falso, significa que está conectado a un sensor (interruptor, sen-

Sistema Wago SPS 750

sor de temperatura, sensor de luminosidad, contactos de ventanas, etc.). El parámetro Isis, es la corriente del sistema y está se define dependiendo de lo que el modulo sea (digital/análogo y salida/entrada). Por esto hay 4 posibilidades de Isis que son: Ied (corriente módulo digital y de entrada), Iea (corriente módulo de entrada análogo), Isd (corriente módulo de salida digital) y Isa (corriente módulo de salida análoga). La corriente del sistema Isis se le resta a la corriente de entrada y si ese resultado es positivo entonces sería el valor de la corriente para el siguiente módulo, si es negativo entonces se le entrega al siguiente módulo 2 amperios . El gasto de potencia es el parámetro gasto. Éste es igual a la multiplicación de Isis por el vsis. Se le suma una potencia de 4 W si el resultado de la resta de la corriente de entrada y el sistema han dado negativo. De esta forma se agregaría una fuente virtual. Cada módulo de la librería contiene 2 canales, que son la cantidad de sensores o actuadores que estarían conectados.

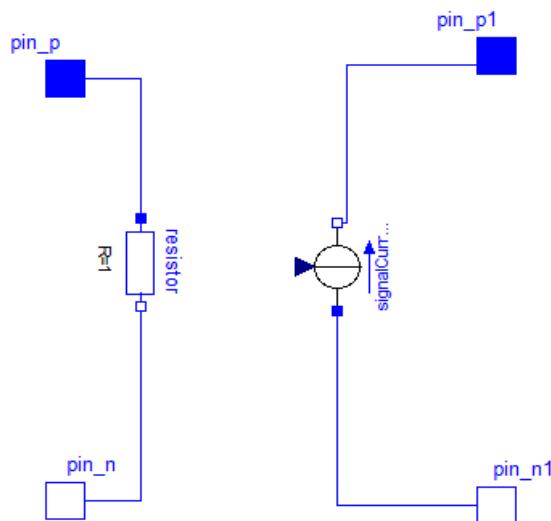


Figura 6.4: Esquema Gráfico de la Composición del Módulo de Bus en Modelica

En la figura 6.4 se observa la separación de impedancias del sistema. Entre el pin_p y el pin_n se conecta al controlador del bus y el pin_p1 y el pin_n1 se conecta a los elementos primarios.

Descripción de la Librería Wago en Modelica

```
model mb "Es la parte básica de todos los sensores y actuadores"
  import D = Modelica.Electrical.Digital;
  import L = Modelica.Electrical.Digital.Interfaces.LogicValue;
  Modelica.SIunits.Power gasto;
  //parameter Integer canales=2 annotation 0;
  parameter Modelica.SIunits.Voltage vsis=5; //El voltaje de alimentación del sistema interno
  parameter Modelica.SIunits.Current Ied=0.004;
  parameter Modelica.SIunits.Current Iea=0.06;
  parameter Modelica.SIunits.Current Isd=0.015;
  parameter Modelica.SIunits.Current Isa=0.07;
  parameter Boolean digital=true;
  parameter Boolean salida=true;
  Modelica.SIunits.Current Isis;

  equation

    /*con la siguiente ecuación se pretende saber que tanta corriente consume
      el sistema en si, pero esa respuesta depende de si el sistema es análogo
      o digital y si es de entrada o salida.*/
    Isis=if digital==true and salida==true then Isd else if
      digital==true and salida==false then Ied else if
      digital==false and salida==true then Isa else if
      digital==false and salida==false then Ied else 1;
    signalCurrent.i=if (resistor.p.i-Isis)>0 then (resistor.p.i-Isis) else 2; /*Si la corriente
      que entra al modulo de bus menos la corriente que necesita el sistema es mayor que cero
      significa que el sistema puede consumir corriente de la que hay en la linea, si no entonces
      se debe dar corriente*/
    gasto=if (pin_p.i-Isis)>0 then Isis*vsis else (Isis*vsis)+4;

  end mb;
```

Código 6.1: Definición de mb en Modelica

6.5.2 Fuente de Alimentación para el controlador netzpower

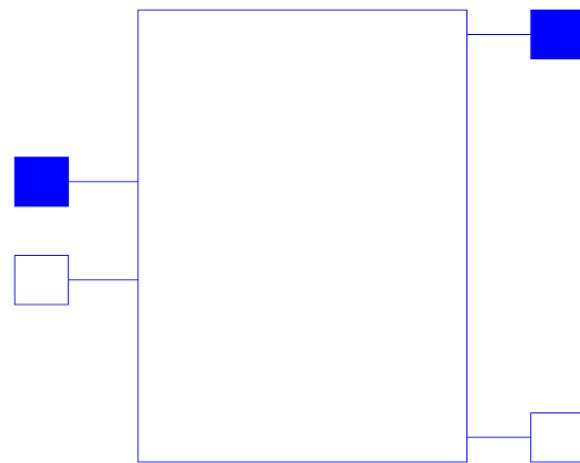


Figura 6.5: Fuente de Alimentación en Modelica

Es la fuente que alimenta los controladores y módulos de Bus. Su función principal es la de entregar al sistema corriente directa por medio de la corriente alterna que entra. El gasto energético de este elemento es de 8 W.

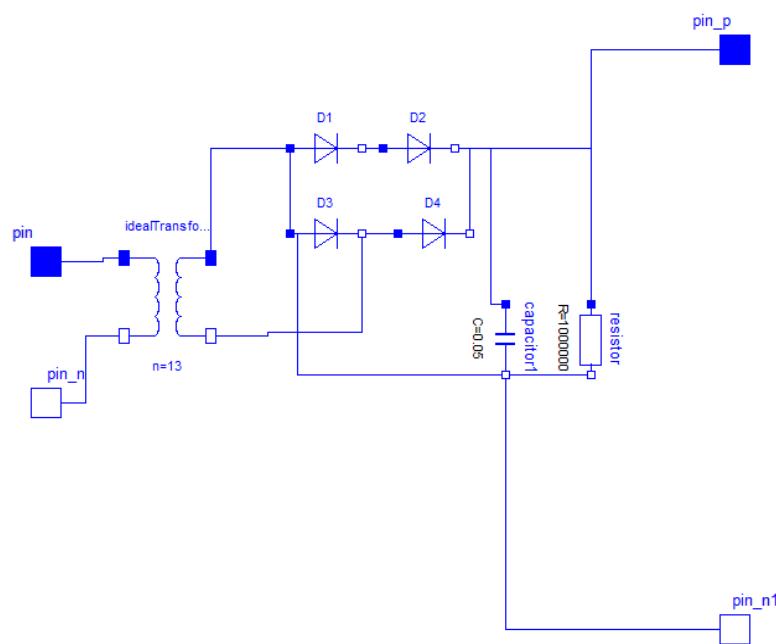


Figura 6.6: Esquema gráfico en Modelica de netzpower.

Descripción de la Librería Wago en Modelica

En la figura 6.2 se puede observar la composición del componente netzpower. Por pin y pin_n se conecta a la energía de la vivienda que es de 230 voltios, por medio del transformador queda separado galvánicamente la entrada de la salida. Por medio de un corrector de onda y un filtro a la salida, se entrega en el pin_p y el pin_n1 de 24 a 30 voltios de voltaje directo, que es fundamental para alimentar el sistema.

```
model netzpower "Esta formada por un corrector de onda completa para que entre
230 voltios con 50HZ salgan 26 voltios DC. Para alimentar el controlador de bus."
parameter Modelica.SIunits.Power gasto=4; //El gasto de potencia
Modelica.Electrical.Analog.Interfaces.Pin pin a;
a;

Modelica.Electrical.Analog.Ideal.IdealDiode D1 a;
Modelica.Electrical.Analog.Ideal.IdealDiode D2 a;
Modelica.Electrical.Analog.Ideal.IdealDiode D3 a;
Modelica.Electrical.Analog.Ideal.IdealDiode D4 a;
Modelica.Electrical.Analog.Basic.Resistor resistor(R=1000000) a;
Modelica.Electrical.Analog.Ideal.IdealTransformer idealTransformer(n=13) a;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n a;
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p a;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n1 a;
Modelica.Electrical.Analog.Basic.Capacitor capacitor1(C=0.05) a;
equation
connect(pin, pin) a;
connect(D1.n, D2.p) a;
connect(D3.n, D4.p) a;
connect(D4.n, D2.n) a;
connect(D2.n, resistor.p) a;
connect(idealTransformer.p1, pin) a;
connect(idealTransformer.p2, D1.p) a;
connect(idealTransformer.n2, D3.n) a;
connect(D1.p, D3.p) a;
connect(pin_p, pin_p) a;
connect(pin_n, idealTransformer.n1) a;
connect(capacitor1.p, D2.n) a;
connect(D2.n, pin_p) a;
connect(capacitor1.n, resistor.n) a;
connect(D3.p, capacitor1.n) a;
connect(capacitor1.n, pin_n1) a;
end netzpower;
```

Código 6.2: Definición de netzpower en Modelica.

6.5.3 Controlador de Bus cb

Controlador de Bus 750-881

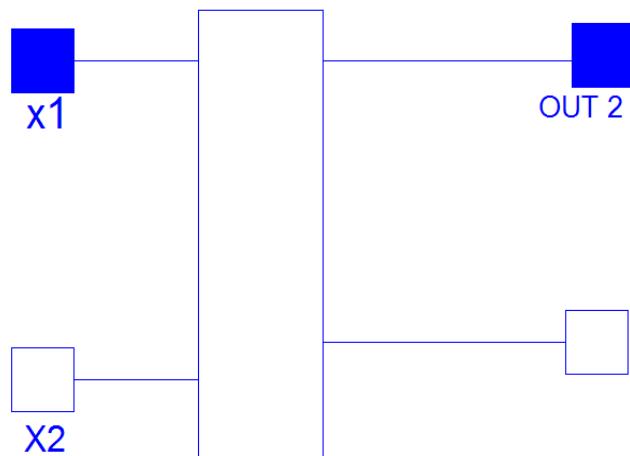


Figura 6.7: Controlador de Bus en Modelica

El controlador de bus en Modelica tiene 4 conectores, de los cuales x1 y x2 son la entrada del voltaje de la fuente, el controlador de bus alimenta los módulos de bus entregando una corriente determinada, que varía según el modelo del controlador de bus, por tal motivo éste es un parámetro de corriente que se llama Ientrega. En el controlador de Bus 750-881 Ientrega equivale a 1.2 A. El voltaje del sistema interno está definido en el parámetro voltaje vsis y equivale a 5 voltios. El gasto energético del sistema se determina por medio de la multiplicación de los parámetros vsis por Isis, donde Isis es la corriente que consume el sistema interno.

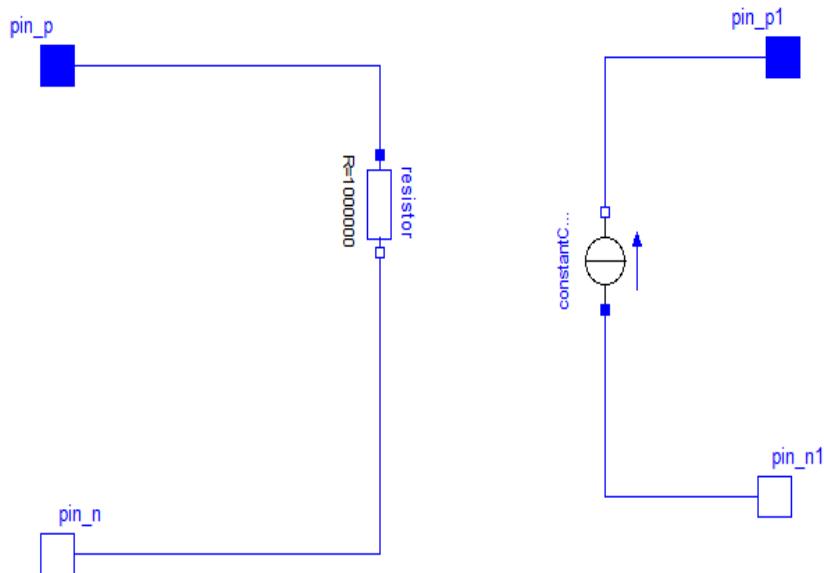


Figura 6.8: Representación Gráfica en Modelica de cb.

Este componente está compuesto por un resistor y una fuente de corriente, están separados galvánicamente. Por el pin_p y pin_n, entra el voltaje de alimentación del la fuente de voltaje. Por pin_{p1} y pin_{n1}, sale el voltaje información para un determinado número de módulos, véase figura 6.8.

```

model cb "La función principal de este componente en Dymola es abastecer el mismo controlador con una determinada corriente para que el sistema interno funcione, también entrega una corriente determinada al sistema interno de los módulos de buses."
import D = Modelica.Electrical.Digital;
import L = Modelica.Electrical.Digital.Interfaces.LogicalValue;
parameter Modelica.SIunits.Current Iisis=0.380; //corriente que consume el sistema del controlador para funcionar
parameter Modelica.SIunits.Current Ientrega=1.2; //Corriente para entregar a los modulos
parameter Modelica.SIunits.Voltage vsis=5; //Voltaje que toma el sistema para funcionar
Modelica.SIunits.Power gasto; //Gasto de potencia de un controlador de bus 750-881
Boolean ON; //Es una variable que esta activa si el voltaje de entrada es mayo de 24 voltios

Modelica.Electrical.Analog.Interfaces.NegativePin pin_n a;
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p a;
Modelica.Electrical.Analog.Basic.Resistor resistor(R=1000000) a;

Modelica.Electrical.Analog.Interfaces.PositivePin pin_p1 a;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n1 a;
Modelica.Electrical.Analog.Sources.ConstantCurrent constantCurrent(I=1.2) a;
equation
gasto=Isis*vsi; // Es el gasto que tiene el sistema por funcionar
ON=if resistor.p>24 then true else false;
connect(pin_p, resistor.p) a;
connect(pin_n, resistor.n) a;
connect(constantCurrent.n, pin_p1) a;
connect(constantCurrent.p, pin_n1) a;
end cb;

```

Código 6.3: Definición de cb en Modelica.

6.5.4 Modulo Final mf

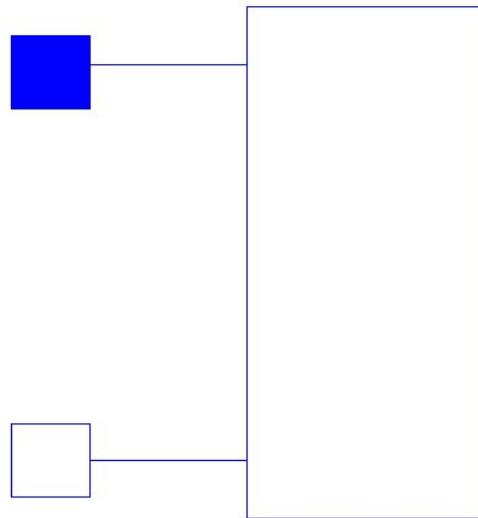


Figura 6.9: Módulo Final mf

El módulo mf sirve para cerrar el sistema. Esta compuesto por una resistencia. El gasto Energético es de 1000 mW, véase figura 6.9.

```
model mf
  "Es necesario en el sistema para saber que el sistema esta cerrado."
  parameter Modelica.SIunits.Power gasto=0.1;
  Modelica.Electrical.Analog.Interfaces.PositivePin pin_p a;
  Modelica.Electrical.Analog.Interfaces.NegativePin pin_n a;
  Modelica.Electrical.Analog.Basic.Resistor resistor a;
  a;

equation
  connect(pin_p, resistor.p) a;
  connect(pin_n, resistor.n) a;
end mf;
```

Código 6.4: Definición de mf en Modelica.

6.6 Implementación del Sistema Wago en la Vivienda Ejemplo.

La vivienda ejemplo necesita un controlador de bus 750 para poder implementar el sistema Wago. Los módulos de bus necesarios, dependen de la cantidad de actores y sensores que la vivienda necesite. Cada módulo de bus está compuesto de dos canales, lo que significa que se pueden conectar ya sea dos sensores o actuadores.

6.6.1 Sótano

Elementos	Características del Módulo de Bus
5 Interruptores	3 mb de entrada digital Ied.
10 Toma corrientes	5 mb de salida digital Isd
5 bombillas	3 mb de salida análoga Isa
1 Sensor de temperatura	1 mb de entrada análoga Iea
1 Regulador de temperatura	Se toma un canal de los utilizados en las bombillas Isa.
1 Sensor de fuego y humo	Se toma un canal del utilizado en el sensor de temperatura.
6 Sensores de Contactos de puertas y ventanas.	3 mb de entrada digital Ied
1 Sirena de alerta.	1 mb de salida digital Isd

Tabla 6.1: Elementos del Sótano a Automatizar con los Respectivos Módulos de Bus

Para implementar el sistema Wago en el sótano, además del controlador de Bus, se necesitan 16 módulos de bus mb.

Sistema Wago SPS 750

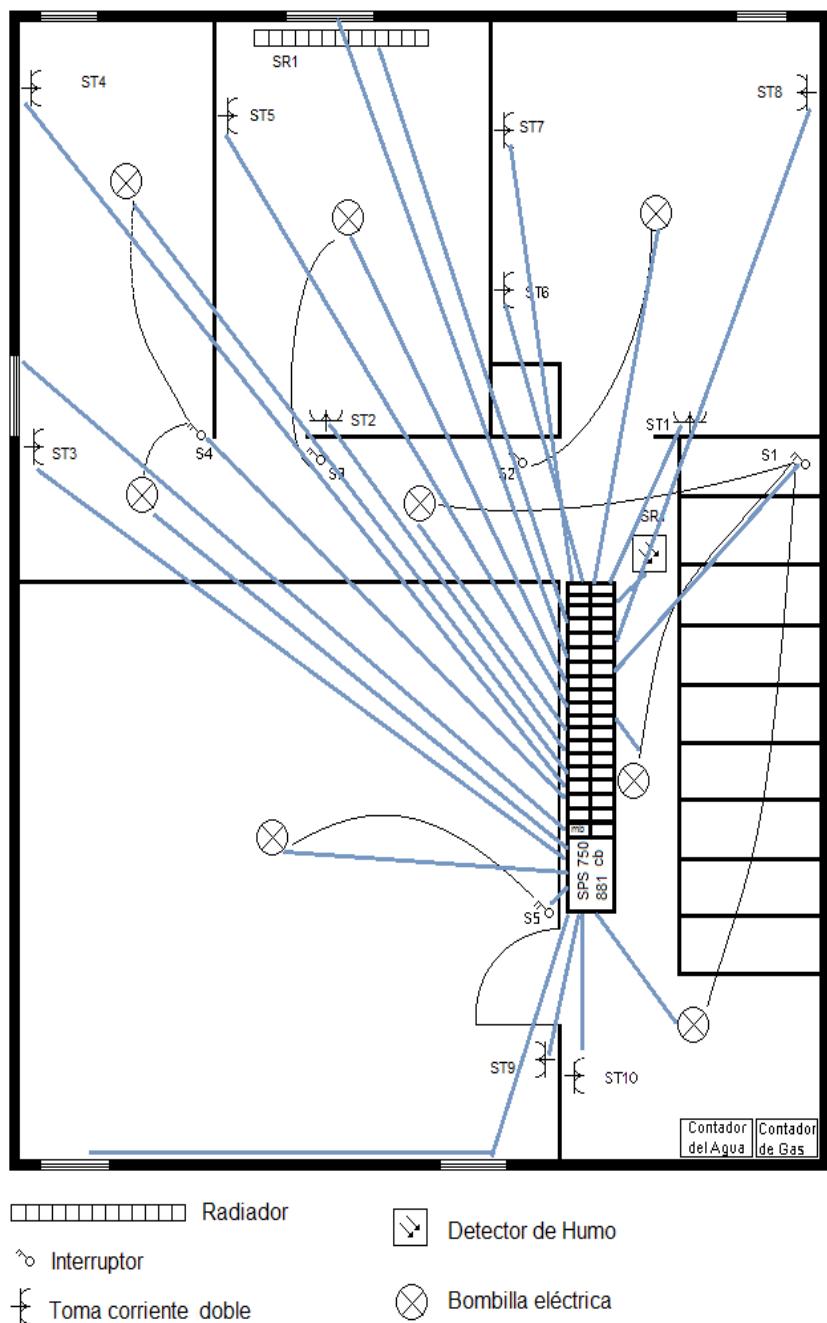


Figura 6.10: Sistema Wago Sótano

6.6.2 Primer Piso

Elementos	Características del módulo de bus
5 Interruptores o taster de Luz	3 mb de entrada digital, Ied.
14 Toma corrientes	7 mb de entrada digital, Isd.
6 sensores contactos de puertas y ventanas	3 mb de entrada digital, Ied.
3 sensores de temperatura	2 mb de entrada análoga, Iea.
5 reguladores de temperatura	3 mb de salida análoga, Isa.
5 bombillas	2 mb de salida análoga, Isa.
5 Motores de persianas	3 mb de salida análoga, Isa.
1 Detector de humo	Hay 1 canal que no se utiliza en los interruptores de luz.

Tabla 6.2: Elementos del Primer Piso a Automatizar con los Respectivos Módulos de Bus.

Por medio de la tabla, se puede hacer el cálculo para observar la cantidad de módulos de bus, necesarios para automatizar el primer piso. Son necesarios 23 módulos.

Sistema Wago SPS 750

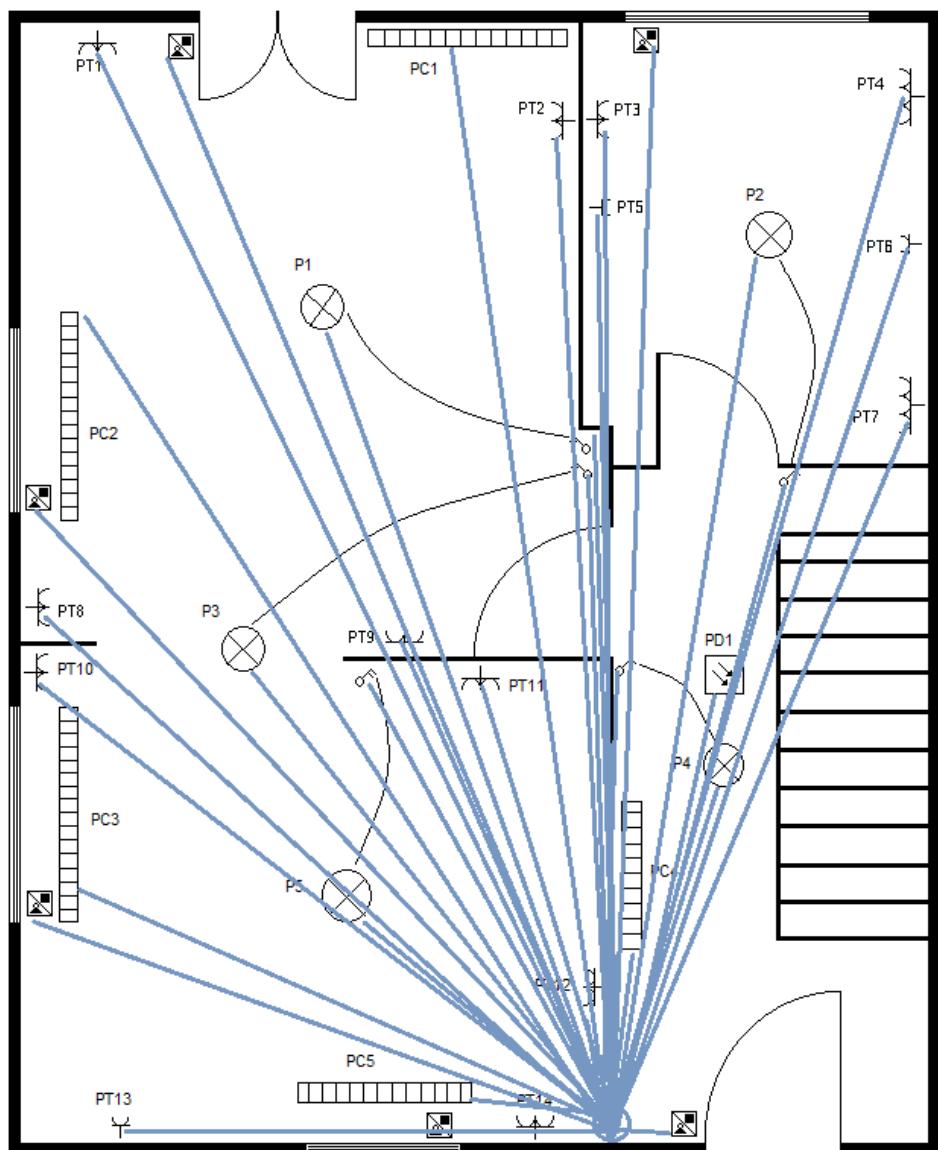


Figura 6.11: Sistema Wago Primer Piso

6.6.3 Segundo Piso

Elementos	Características del módulo de bus.
4 salidas de bombillas (Regulador de luz)	2 mb de salida digital, Isd.
7 sensores de contactos de puertas y ventanas	4 mb de entrada digital, Ied.
4 Sensores de temperatura	2 mb de entrada analógica, Iea.
13 Toma corrientes	7 mb de salida digital, Isd.
5 Interruptores de Luz (Dimmer)	2 mb de entrada digital, Ied.
5 Reguladores de temperatura	3 mb de salida analógica, Isa.
1 Detector de humo y fuego	Se toma un canal de un modulo de entrada digital.

Tabla 6.3: Elementos del Segundo Piso a Automatizar con los Respectivos Módulos de Bus.

Sistema Wago SPS 750

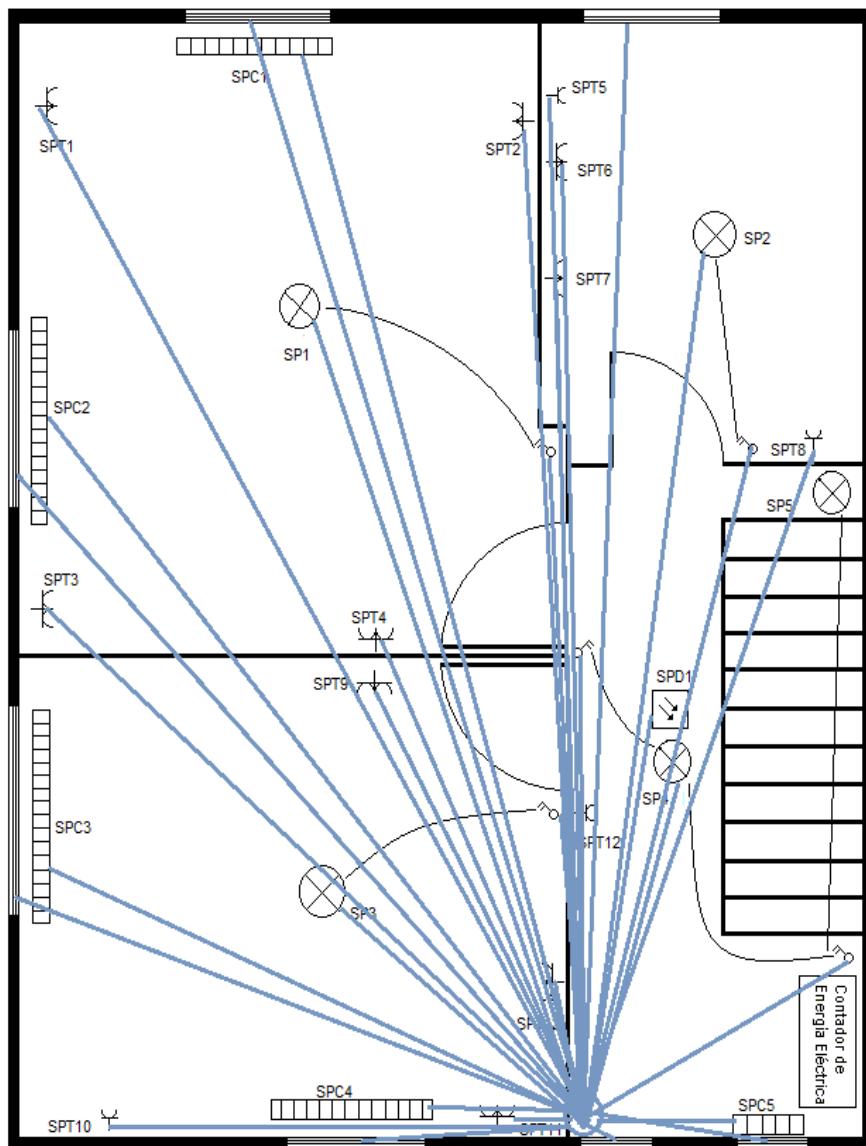


Figura 6.12: Sistema Wago Segundo Piso

6.6.4 Tercer Piso

Elementos	Características del Módulo de Bus
4 sensores de puertas y ventanas	2 mb de entrada digital, Ied.
9 bombillos con regulación	5 mb de salida digital, Isd.
6 interruptores (Dimmer)	3 mb de entrada digital, Ied.
10 Toma corrientes	5 mb de salida digital, Isd.
2 sensores de Temperatura	1 mb de entrada analógica, Iea.
2 Reguladores de temperatura	1 mb de salida analógica, Isa.

Tabla 6.4: Elementos del Tercer Piso a Automatizar con los Respectivos Módulos de Bus

Se necesitan 17 módulos de bus.

Sistema Wago SPS 750

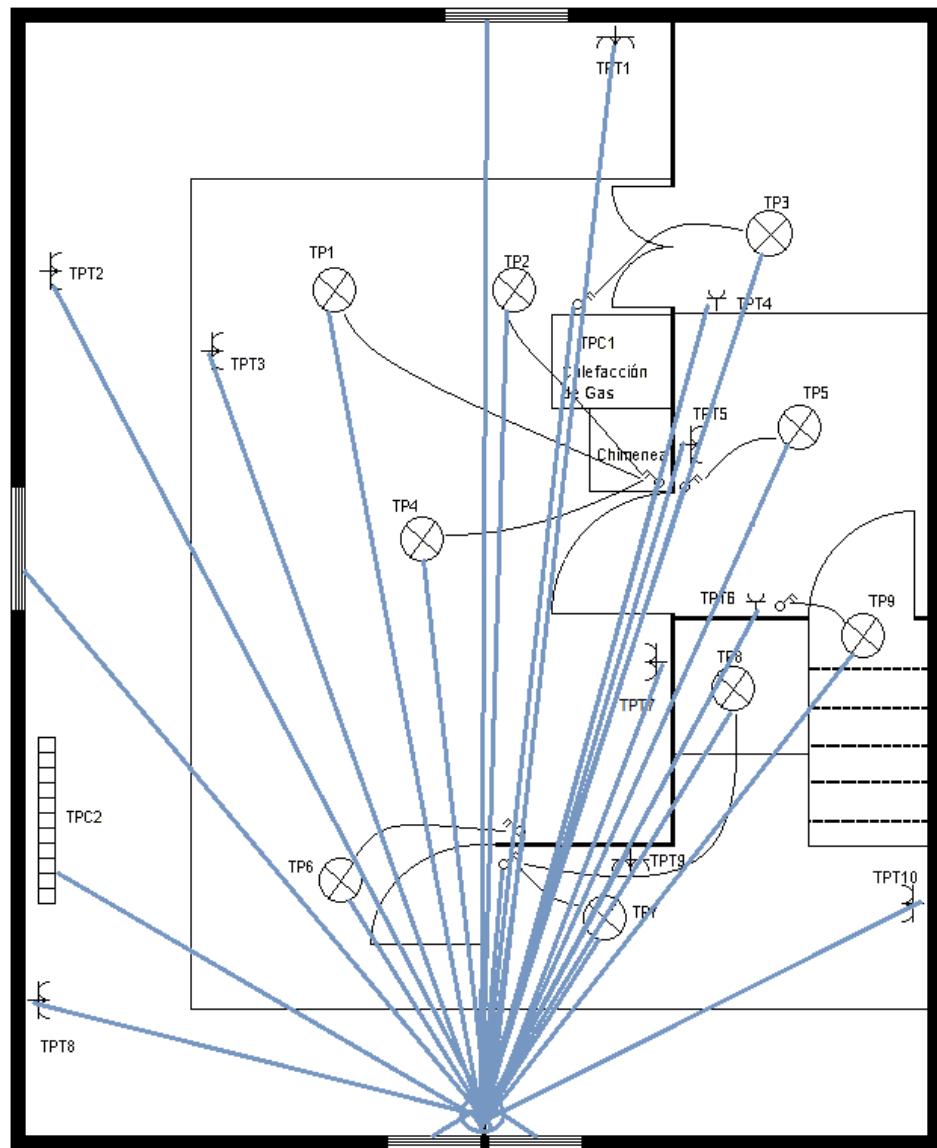


Figura 6.13: Sistema Wago Tercer Piso

6.7 Sistema Wago en Modelica

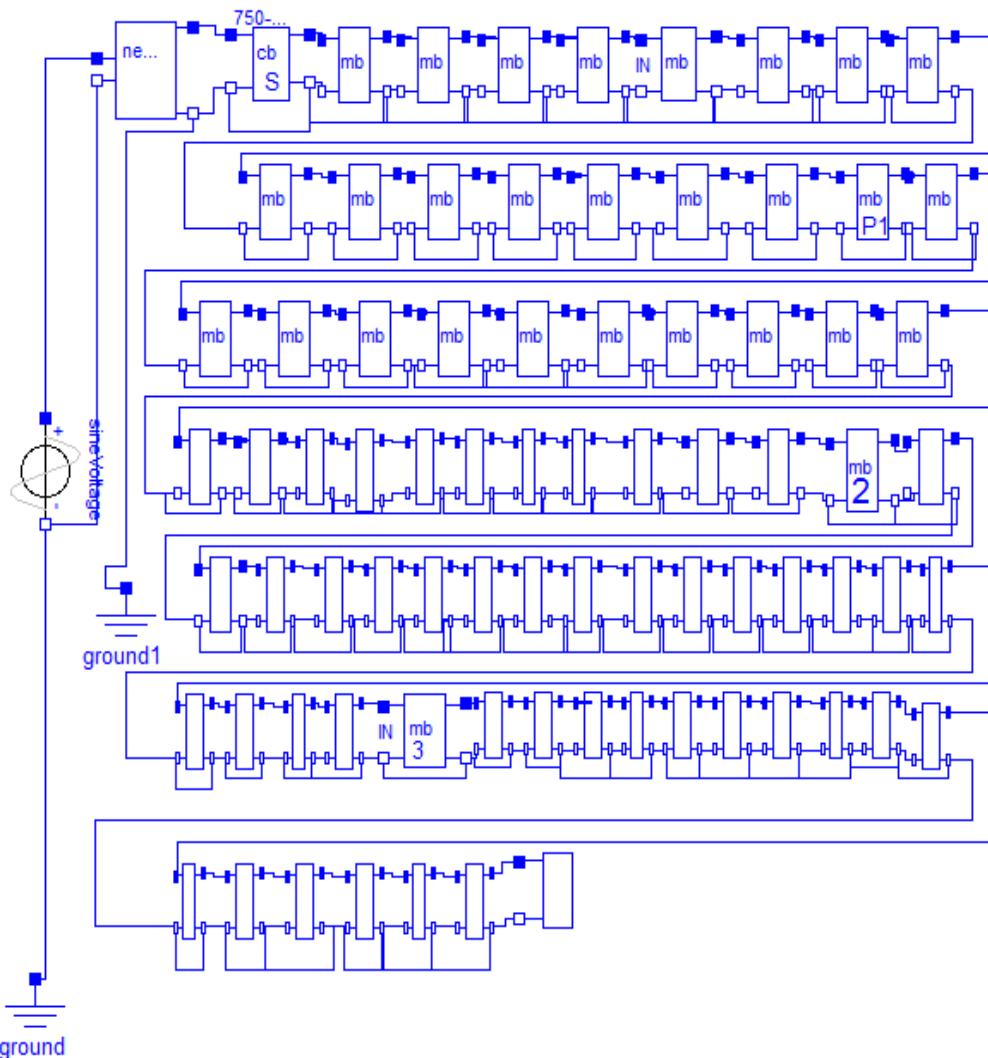


Figura 6.14: Esquema Gráfico de la Implementación del sistema Wago en la Vivienda Ejemplo.

Se puede observar que el sistema Wago es centralizado, que toda la información tiene que llegar al controlador de bus para poder ser procesada y vuelta a enviar. Por medio del cálculo de la sumatoria

Sistema Wago SPS 750

de los gastos de potencia, se llega al resultado del gasto energético del sistema el Wago en la vivienda ejemplo.

```
model viviendawago
  "Implementación del sistema el Wago en la vivienda ejemplo. "
  Modelica.SIunits.Power gasto;
  equation
    gasto=netzpower1.gasto+cbl.gasto+mb1.gasto+mb2.gasto+mb3.gasto+mb4.gasto+mb5.gasto+mb6.gasto+mb7.gasto+mb8.gasto+mb9.gasto
      +mb10.gasto+mb11.gasto+mb12.gasto+mb13.gasto+mb14.gasto+mb15.gasto+mb16.gasto+mb17.gasto+mb18.gasto+mb19.gasto
      +mb20.gasto+mb21.gasto+mb22.gasto+mb23.gasto+mb24.gasto+mb25.gasto+mb26.gasto+mb27.gasto+mb28.gasto+mb29.gasto
      +mb30.gasto+mb31.gasto+mb32.gasto+mb33.gasto+mb34.gasto+mb35.gasto+mb36.gasto+mb37.gasto+mb38.gasto+mb39.gasto
      +mb40.gasto+mb41.gasto+mb42.gasto+mb43.gasto+mb44.gasto+mb45.gasto+mb46.gasto+mb47.gasto+mb48.gasto+mb49.gasto
      +mb50.gasto+mb51.gasto+mb52.gasto+mb53.gasto+mb54.gasto+mb55.gasto+mb56.gasto+mb57.gasto+mb58.gasto+mb59.gasto
      +mb60.gasto+mb61.gasto+mb62.gasto+mb63.gasto+mb64.gasto+mb65.gasto+mb66.gasto+mb67.gasto+mb68.gasto+mb69.gasto
      +mb70.gasto+mb71.gasto+mb72.gasto+mb73.gasto+mb74.gasto+mb75.gasto+mb76.gasto+mb77.gasto+mb78.gasto+mf1.gasto;
  end viviendawago;
```

Código 6.5: Cálculo del Gasto de Potencia en la Vivien Ejemplo.

El siguiente código es la simulación del sistema wago en la vivienda ejemplo, se observa el valor de gasto, que es de 18,065 vatios.

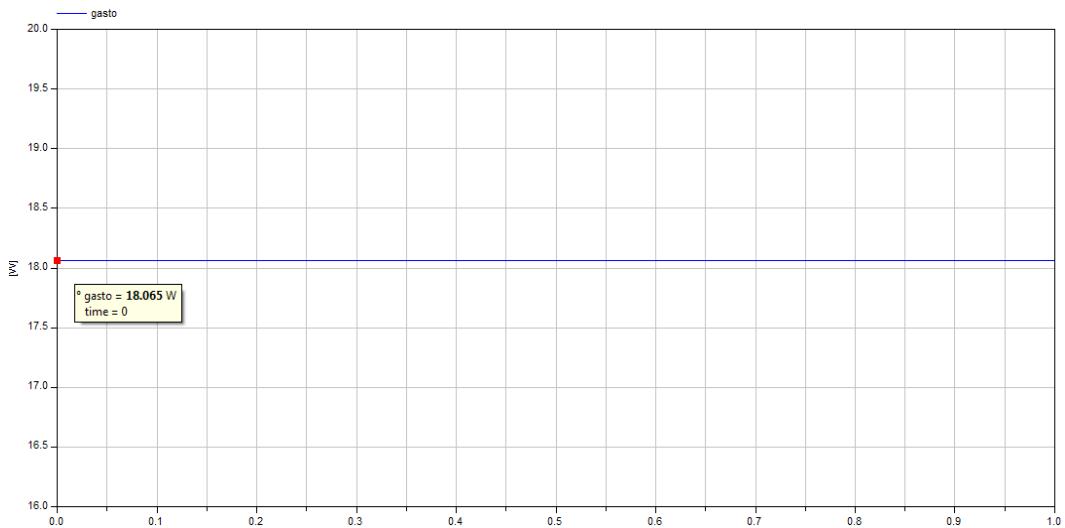


Figura 6.15: Resultado del Gasto de Potencia al Simular en Dymola el Sistema Wago en la Vivienda Ejemplo

6.8 Conclusiones

Como se puede apreciar el sistema de controladores lógicos programables Wago SPS 750 tiene un cableado exagerado, lo que representa una desventaja al implementar el sistema, pero de manera clara se puede observar que es un sistema centralizado, ya que los programas que administran las funciones de los elementos de la vivienda se encuentran en el controlador de bus. Para hallar el gasto energético que genera este sistema, se implementa el sistema Wago en la vivienda ejemplo, tomando los componentes desarrollados en la sublibreria Wago de la librería DOMOSYS en Modelica. Así se crea el modelo viviendawago, donde se representa todos los componentes necesarios para automatizar la vivienda ejemplo (fuentes, controlador de bus, módulos de bus). Se simulan y se obtiene el gasto energético del sistema, el cual es de 18.065 W.

7. Conclusiones y

Trabajos Futuros

7.1 Conclusiones

- En el diseño de la librería DOMOSYS se aprecia el beneficio de utilizar el lenguaje Modelica por la ventaja de ser orientado a objetos, lo que simplifica el trabajo de diseñar por medio de la reutilización de otras librerías y componentes. A su vez Modelica es interesante y exigente ya que el diseño de los componentes se realizan por medio del planteamiento de ecuaciones, que definen de forma matemática las relaciones con otros posibles componentes por medio del principio constitutivo de los mismos. Esta característica hace posible un enfoque práctico y un conocimiento detallado de los sistemas domóticos KNX (TP), Eltako inalámbrico y Wago SPS 750, en la parte de consumo energético.
- Además del beneficio del confort y seguridad que brindan los sistemas domóticos, en este trabajo se ahonda en el estudio detallado de los sistemas Eltako inalámbrico, KNX (TP), Wago SPS 750 desde el punto de vista del gasto energético de cada sistema. Por tal motivo se tomó una vivienda ejemplo, para poder medir el consumo de cada uno de estos sistemas cuando están listos para ser usados, obteniendo como resultado que el sistema que tiene más bajo consumo es el Wago SPS 750 con un gasto de 19.2 vatios en la vivienda ejemplo. Con este resultado se puede apreciar que los sistemas centralizados

tienen un gasto energético claramente menor que los sistemas descentralizados como se puede observar en el sistema KNX (TP) y Eltako inalámbrico. Aunque podemos apreciar en los gráficos de la implementación del sistema en la vivienda ejemplo que el cableado en el sistema Wago SPS 750 es exagerado.

- La Librería DOMOSYS ofrece el conocimiento claro y objetivo del consumo energético de tres sistemas domóticos, de forma segura e independiente, a usuarios interesados en implementar un sistema domótico ya sea Eltako inalámbrico, KNX (TP) y Wago SPS 750, pero también da una idea del consumo que pueda tener un sistema centralizado o descentralizado. Al contrario de los software que se ofrecen en el mercado que son de un sistema determinado, este sistema da cabida para poder evaluar 3 diferentes sistemas en su aspecto de consumo energético.

7.2 Trabajos Futuros

La Librería Domosys es muy interesante para personas que estén buscando un sistema domótico para una vivienda determinada y busquen el comportamiento de éste desde el punto de vista de ahorro energético. Así que se pueden fijar en estos tres sistemas. Por la naturaleza abierta de el lenguaje Modelica que es orientado a objetos se puede seguir completando esta librería con otros sistemas domóticos relevantes del mercado.

Conclusiones y Trabajos Futuros

Bibliografía

- [1] Aschendorf, Bernd. *Energiemanagement durch Gebäudeautomation: Grundlagen - Technologien - Anwendungen.* Dortmund: Springer, 2013.
- [2] Harke, Werner. *Domótica para viviendas y edificios.* 1^a ed. Barcelona: Marcombo, 2010.
- [3] Völkel, Frank. *Smart Home mit KNX.* Munchen: FRANZIS, 2012.
- [4] Schneider, Wolfgang. *Praxiswissen Digitale Gebäudeautomation: Planen, Konfigurieren, Betreiben.* Deutschland: Viewweg+Taubner, 1997.
- [5] Scherg, Reiner. *EIB Planen und Installieren.* Deutschland: Vogel, 2002.
- [6] Leidenroth, Hannes. *EIB Anwenderhandbuch: Planung, Projektierung, Inbetriebnahmen, Kundenberatung.* 3^a ed. Deutschland:Huss-Medien, 2003.
- [7] Jens, Klaus. Pech, Anton. *Elektro und Regeltechnik.* Wien: Springer, 2007.
- [8] Waldmann, Jan. Messsystema zur Untersuchung eine möglichen Beeinflus-
sung des Menschen durch elektromagnetische Felder.
- [9] Hansemann, Thomas. Hübner, Christof. Merz, Hermann. *Gebäudeauto-
mation Kommunikationssysteme mit EIB/KNX, LON und BACnet.* Deutsch-
land: Carl Hanser, 2009.

[10] Bohne, Dirk. *Technischer Ausbau von Gebäuden und nachhaltige Gebäudetechnik*. 3^a ed. Hannover: Springer Vieweg, 2009.

[11] GIRA. *Catálogo 2014/2015. Deutschland*:GIRA, 2014.

[12] Eltako Electronics. *Catálogo Eltako – Der Gebäudefunk, Productos y precios*. Fellbach: Eltako Electronics, 2015.

[13] “Generic Profiles” https://www.enocean-alliance.org/de/enocean_standard/ [25.04.2015].

[14] “EnOcean_Equipment Profiles EEP V2.6.2-public-2.pdf”
https://www.enocean-alliance.org/de/enocean_standard/
[25.04.2015].

[15] “EnOceanRadioProtocol.pdf, EnOceanRadioProtocol2.pdf, EnOceanRadioProtocol3.pdf”

<http://www.enocean.com> [07.05.2015]

[16] “Handbuch Automation”

http://global.wago.com/media/2_products/m07500881_00000000_0de.pdf

[17.05.2015]

[17] Tipler, Paul A. *Física II*. España: Reverté S. A. 1993.

[18] ICNIRP “International Commission on Non-Ionizing Radiation Protection”. *Guidelines for Limiting Exposure to Time-Varying Electric, Magnetic and Electromagnetic fields (up to 300 GHz)*. 1998

[19] Campo Franco, Juan Fernando. Marulanda Meza, Juan Sebastian. *Desarrollador de un prototipo de simulador de un sistema domótico para hogares, basado en redes de protocolo X10.* 2010

[20] Pacheco Rocamora, Eufemio. *Diseño y Simulación de un sistema domótico para una vivienda unifamiliar.* 2008

[21] Schnell, Gerhard. Wiedemann Bernhard. *Bussysteme in der Automatisierungs- und Prozesstechnik.* Wiesbaden, Deutschland: Vieweg. 2006.

[22] Issendorf KG. *lcn-systembeschreibung.pdf.* www.lcn.de/downloads.
Deutschland 05.07.2015

[23] Empresa ELV/EQ-3. *Die neue Smart Generation.* www.hausautomation-homematic-ip.html. Deutschland 06.07.2015

[24] Empresa ELDAT. *Katalog 2014.*
www.eldat.de/produkte/_katalog/katalog_easywave_0114_de.pdf. Deutschland
06.07.2015.

[25] Empresa Lenze. *Handbuch IEC61131-3 inside.*
http://download.lenze.com/TD/DDS__Drive%20PLC%20Developer%20Studio%20IEC61131-3%20programming__v2-0__DE.pdf.
[10.07.2015]

[26] Empresa WAGO. *Automatisierungstechnik. Hauptkatalog Band 3 Ausgabe 2015.* <http://www.wago.de/produkte/produktinformationen-infomaterial/infomaterial/uebersicht/index.jsp>.

[11.07.2015]

[27] 868 MHz

http://funkfernsteuerungindustrie.de/tyropedia/?page_id=13

[10.03.2015]

[28] Allgemeinzuteilung

<http://www.mikrocontroller.net/articles/Allgemeinzuteilung>

[15.03.2015]

[29] Funkanwendungen auf den ISM-Bändern

http://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/allgemeinzuteilungen-node.htm.

[5.05.2015]

A) Librería DOMOSYS

El siguiente anexo muestra la información generada por Dymola sobre la librería DOMOSYS.

DOMOSYS

Librería con tres Sistema Domóticos para medir el Gasto Energético.

Package Content

Name	Description
eltako	Paquete con el Sistema Eltako Inalámbrico
<input type="checkbox"/> SBKNX	Paquete con el Sistema KNX/EIB de la forma TP por cable par trenzado.
<input type="checkbox"/> Wago	Paquete con el Sistema de la Empresa Wago que es un Controlador Lógico Programable PLC

HTML-documentation generated by [Dymola](#) Mon Aug 31 10:08:11 2015.

DOMOSYS_eltako

Paquete con el Sistema Eltako Inalámbrico

Package Content

Name	Description
sensores	Los sensores del Sistema Eltako Inalámbrico.
actuador	Se encuentran los actuadores básicos inalámbricos
servidor	Por medio del Servidor el sistema tiene la capacidad de ser controlado desde fuera de la vivienda
viviendaejemplo	Implementación del Sistema en la vivienda ejemplo.

DOMOSYS_eltako.servidor

Por medio del Servidor el sistema tiene la capacidad de ser controlado desde fuera de la vivienda



Parameters

Type	Name	Default	Description
Power	potencia	13	[W]

Modelica definition

```

model servidor
    "Por medio del Servidor el sistema tiene la capacidad de ser
controlado desde fuera de la vivienda"
    parameter Modelica.SIunits.Power potencia=13;

end servidor;

```

HTML-documentation generated by [Dymola](#) Mon Aug 31 10:59:13 2015.

DOMOSYS_eltako.sensores

Los sensores del Sistema Eltako Inalámbrico.

Package Content

Name	Description
 interruptorpasiv	Es un Interruptor inalambrico, que tiene dos estados y diferentes acciones que pueden hacer el papel de un regulador.
 sttemperatura	Sensor de Temperatura inalámbrico, mide la temperatura y la envía al regulador.
Detectorhumo	Sensor que detecta el humo
 kvp	Contacto de puertas y ventanas, tiene dos estados que se identifican con el booleano activo.

DOMOSYS_eltako.sensores.interruptorpasiv



Es un Interruptor inalámbrico, que tiene dos estados y diferentes acciones que pueden hacer el papel de un regulador.

Parameters

Type	Name	Default	Description
Integer	identificacion	123456	La identificación del Interruptor
Boolean	activo	false	Sirve para saber si el Interruptor está en funcionamiento
Integer	accion	1	Definir la acción que puede ser prender, apagar o dimmer

Modelica definition

```
model interruptorpasiv
```

```

"Es un Interruptor inalambrico, que tiene dos estados y diferentes acciones que pueden hacer el papel de un regulador."
parameter Integer identificacion=123456 "La identificación del
Interruptor";
parameter Boolean activo=false
    "Sirve para saber si el Interruptor está en funcionamiento";
parameter Integer accion=1
    "Definir la acción que puede ser prender, apagar o dimmer";

end interruptorpasiv;

```

DOMOSYS_eltako.sensores.stemperatura

Sensor de Temperatura inalámbrico, mide la temperatura y la envía al regulador.



Parameters

Type	Name	Default	Description
Integer	identificacion	111111	Identificación en el sistema
CelsiusTemperature	Temperatura	20	Temperatura que tiene el instrumento [degC]
Boolean	activo	false	Si el elemento esta activo en el sistema

Modelica definition

```

model stemperatura
    "Sensor de Temperatura inalámbrico, mide la temperatura y la
envía al regulador."
    parameter Integer identificacion=111111 "Identificación en el
sistema";
    parameter Modelica.SIunits.CelsiusTemperature Temperatura=20
        "Temperatura que tiene el instrumento";

```

```

parameter Boolean activo=false "Si el elemento esta activo en
el sistema";
equation

end stemeratura;

```

DOMOSYS_eltako.sensores.Detectorhumo



Sensor que detecta el humo

Parameters

Type	Name	Default	Description
Integer	ident	111111	Identificación del detector de humo
Boolean	activ	false	Booleano para saber si esta activo en el sistema
Integer	mensaje	1	Información que envia el detector de humo al actor

Modelica definition

```

model Detectorhumo "Sensor que detecta el humo"
  parameter Integer ident=111111 "Identificación del detector de
humo";
  parameter Boolean activ=false
    "Booleano para saber si esta activo en el sistema";
  parameter Integer mensaje=1
    "Información que envia el detector de humo al actor";

end Detectorhumo;

```



DOMOSYS_eltako.sensores.kvp

Contacto de puertas y ventanas, tiene dos estados que se identifican con el booleano activo.

Parameters

Type	Name	De-fault	Description
Integer	ident	1	Identificación del contacto de puertas y ventanas
Boolean	activo	true	Booleano para saber si el sensor está activo en el sistema

Modelica definition

```
model kvp
  "Contacto de puertas y ventanas, tiene dos estados que se identifican con el booleano activo."
  parameter Integer ident=1 "Identificación del contacto de puertas y ventanas";
  parameter Boolean activo=true
  "Booleano para saber si el sensor está activo en el sistema";
end kvp;
```

HTML-documentation generated by [Dymola](#) Mon Aug 31 13:21:07 2015.

DOMOSYS.eltako.actuador

Se encuentran los actuadores básicos inalámbricos

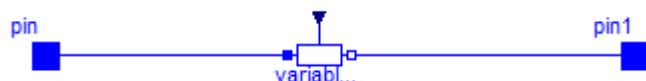
Package Content

Name	Description
actorrele	Parte básica de todos los actuadores, aqui estan almacenados todos los sensores que pueden ser escuchados.

DOMOSYS.eltako.actuador.actorrele



Parte básica de todos los actuadores, aqui estan almacenados todos los sensores que pueden ser escuchados.



Parameters

Type	Name	Default	Description
Integer	iden	111111	Identificación del actor
Integer	sensor1	111111	dirección del sensor 1, que lee para desarrollar acciones determinada
Integer	sensor2	000000	dirección del sensor 2, que lee para desarrollar acciones determinada.
Integer	sensor3	000000	dirección del sensor 3, que lee para desarrollar acciones determinada.
Integer	sensor4	000000	dirección del sensor 4, que lee para desarrollar acciones determinada.
Integer	sensor5	000000	dirección del sensor 5, que lee para desarrollar acciones determinada.

Integer	sensor6	000000	dirección del sensor 6, que lee para desarrollar acciones determinada.
Integer	sensor7	000000	dirección del sensor 7, que lee para desarrollar acciones determinada.
Integer	sensor8	000000	dirección del sensor 8, que lee para desarrollar acciones determinada.
Integer	sensor9	000000	dirección del sensor 9, que lee para desarrollar acciones determinada.
Boolean	activ	false	Significa si el actuador esta activo en el sistema.
Boolean	an	false	Significa si el sensor esta consumiendo o no energía
Power	gastoenergetico	0.8	Cantidad de Potencia que consume el sistema [W]
CelsiusTemperature	temperaturade-seada	20	Si forma parte de un actor regulador de temperatura, es la temperatura deseada. [degC]

Connectors

Type	Name	Description
Pin	pin	
Pin	pin1	

Modelica definition

```

model actorrele
    "Parte básica de todos los actuadores, aquí están almacenados
    todos los sensores que pueden ser escuchados."
    parameter Integer iden=111111 "Identificación del actor"; //La
    identificación del actuador
    parameter Integer sensor1=111111
        "dirección del sensor 1, que lee para desarrollar acciones
        determinada"; //El sensor del 1 al

```

Librería DOMOSYS

```
9, significa el número de sensores, de los cuales ordenes aceptan.

parameter Integer sensor2=000000
    "dirección del sensor 2, que lee para desarrollar acciones determinada./";

parameter Integer sensor3=000000
    "dirección del sensor 3, que lee para desarrollar acciones determinada./";

parameter Integer sensor4=000000
    "dirección del sensor 4, que lee para desarrollar acciones determinada./";

parameter Integer sensor5=000000
    "dirección del sensor 5, que lee para desarrollar acciones determinada./";

parameter Integer sensor6=000000
    "dirección del sensor 6, que lee para desarrollar acciones determinada./";

parameter Integer sensor7=000000
    "dirección del sensor 7, que lee para desarrollar acciones determinada./";

parameter Integer sensor8=000000
    "dirección del sensor 8, que lee para desarrollar acciones determinada./";

parameter Integer sensor9=000000
    "dirección del sensor 9, que lee para desarrollar acciones determinada./";

parameter Boolean activ=false
    "Significa si el actuador esta activo en el sistema.";
//Significa si el actuador está activo, si escucha a algún sensor.

parameter Boolean an=false
    "Significa si el sensor esta consumiendo o no energía";
//Significa si está conectado a la energía eléctrica

parameter Modelica.SIunits.Power gastoenergetico= 0.8
    "Cantidad de Potencia que consume el sistema";
//La cantidad de gasto energético, por estar en funcionamiento

parameter Modelica.SIunits.CelsiusTemperature temperaturadeseada=20
    "Si forma parte de un actor regulador de temperatura, es la temperatura deseada.";
//Temperatura deseada
```

```
Modelica.Electrical.Analog.Interfaces.Pin pin;
Modelica.Electrical.Analog.Interfaces.Pin pin1;

Modelica.Electrical.Analog.Basic.VariableResistor variableResistor;
equation
  connect(pin, variableResistor.p);
  connect(variableResistor.n, pin1);
  variableResistor.R = if activ==true and an==true then 1000 else
    if activ==true and an== false then 406 else
      1000000;                                //pin donde entra la corriente
                                              //eléctrica de alimentación.

//pin donde sale la corriente eléctrica de alimentación

//La ecuación que permite el paso de la corriente.
end actorrele;
```

HTML-documentation generated by [Dymola](#) Mon Aug 31 13:27:35 2015.

DOMOSYS_eltako.viviendaejemplo

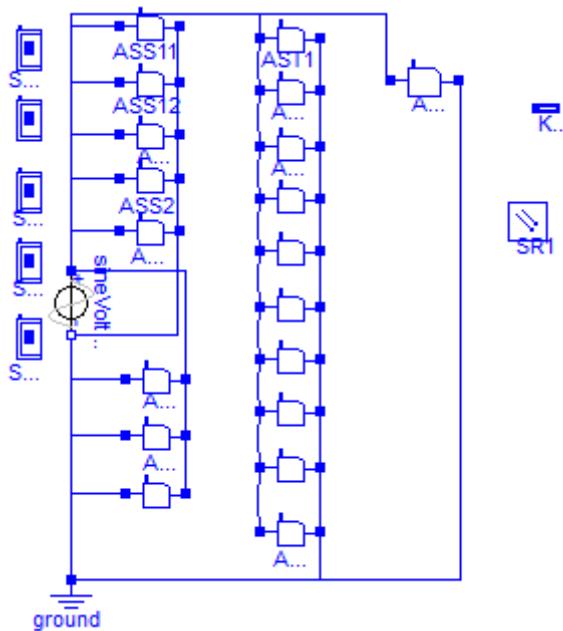
Implementación del Sistema en la vivienda ejemplo.

Package Content

Name	Description
<u>esotano</u>	Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el sótano.
<u>eprimerpiso</u>	Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el primer piso.
<u>esegundopiso</u>	Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el segundo piso.
<u>etercerpiso</u>	Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el tercer piso.
<u>ge</u>	Sumatoria del gasto energético en toda la vivienda ejemplo.

DOMOSYS_eltako.viviendaejemplo.esotano

Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el sótano.



Modelica definition

```

model esotano
  "Sumatoria del gasto energético de los componentes del sistema
  Eltako inalambrico, que se encuentran en el sótano."
  Modelica.SIunits.Power gastosotano;
  Modelica.Electrical.Analog.Sources.SineVoltage
  sineVoltage(V=325, freqHz=60);
  Modelica.Electrical.Analog.Basic.Ground ground;
  DOMOSYS.eltako.actuador.actorrele ASS11(
    activ=true,
    an=false,
    iden=21,
    sensor1=11);
  DOMOSYS.eltako.actuador.actorrele ASS12(
    iden=22,
    sensor1=11,
    activ=true);
  DOMOSYS.eltako.actuador.actorrele ASS13(
    iden=23,
    sensor1=11,
    activ=true);
  DOMOSYS.eltako.actuador.actorrele ASS2(
    iden=24,
    sensor1=12,
    activ=true);

```

Librería DOMOSYS

```
    activ=true);

DOMOSYS_eltako.actuador.actorrele ASS3(
    iden=25,
    sensor1=13,
    activ=true);

DOMOSYS_eltako.sensores.interruptorpasiv SS1(
    identificacion=11);

DOMOSYS_eltako.actuador.actorrele ASS41(
    iden=26,
    sensor1=14,
    activ=true);

DOMOSYS_eltako.actuador.actorrele ASS42(
    iden=27,
    sensor1=14,
    activ=true);

DOMOSYS_eltako.actuador.actorrele ASS5(
    iden=28,
    sensor1=15,
    activ=true);

DOMOSYS_eltako.sensores.interruptorpasiv SS2(
    identificacion=12);

DOMOSYS_eltako.sensores.interruptorpasiv SS3(
    identificacion=13);

DOMOSYS_eltako.sensores.interruptorpasiv SS4(
    identificacion=14);

DOMOSYS_eltako.sensores.interruptorpasiv SS5(
    identificacion=15);

DOMOSYS_eltako.actuador.actorrele actorrele8(
    iden=29,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele9(
    sensor1=555,
    iden=210,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele10(
    iden=211,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele11(
    iden=212,
```

```

    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele12(
    iden=213,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele13(
    iden=214,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele14(
    iden=215,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele15(
    iden=217,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele16(
    iden=216,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele17(
    iden=218,
    sensor1=555,
    activ=true);

DOMOSYS_eltako.actuador.actorrele ASRT1(
    iden=219,
    activ=true,
    sensor1=31);

DOMOSYS_eltako.sensores.kvp kvp1(
    ident=31);

sensores.Detectorthumo detectorthumo;

equation
    gastosotano=(if ASS11.activ==true then ASS11.gastoenergetico
else 0)+(if ASS12.activ==true then ASS12.gastoenergetico else 0)+
        (if ASS13.activ==true then ASS13.gastoenergetico else 0)+
        (if ASS2.activ==true then ASS2.gastoenergetico else 0)+
            (if ASS3.activ==true then ASS3.gastoenergetico else 0)+(if
ASS41.activ==true then ASS41.gastoenergetico else 0)+

```

Librería DOMOSYS

```
(if ASS42.activ==true then ASS42.gastoenergetico else 0)+  
(if ASS5.activ==true then ASS5.gastoenergetico else 0)+  
    (if actorrele8.activ==true then actorrele8.gastoenergetico  
else 0)+(if actorrele9.activ==true then actorrele9.gastoenergeti-  
co else 0)+  
    (if actorrele10.activ==true then actorrele10.gastoenerge-  
tico else 0)+(if actorrele11.activ==true then actorrele11.gastoe-  
nergetico else 0)+  
    (if actorrele12.activ==true then actorrele12.gastoenerge-  
tico else 0)+(if actorrele13.activ==true then actorrele13.gastoe-  
nergetico else 0)+  
    (if actorrele14.activ==true then actorrele14.gastoenerge-  
tico else 0)+(if actorrele15.activ==true then actorrele15.gastoe-  
nergetico else 0)+  
    (if actorrele16.activ==true then actorrele16.gastoenerge-  
tico else 0)+(if actorrele17.activ==true then actorrele17.gastoe-  
nergetico else 0);  
connect(sineVoltage.n, ground.p);  
connect(sineVoltage.p, ASS11.pin);  
connect(ASS12.pin, sineVoltage.p);  
connect(sineVoltage.p, ASS13.pin);  
connect(sineVoltage.n, ASS13.pin1);  
connect(ASS13.pin1, ASS12.pin1);  
connect(ASS12.pin1, ASS11.pin1);  
connect(sineVoltage.p, ASS2.pin);  
connect(sineVoltage.n, ASS2.pin1);  
connect(sineVoltage.p, ASS3.pin);  
connect(sineVoltage.n, ASS3.pin1);  
connect(sineVoltage.n, ASS41.pin);  
connect(sineVoltage.n, ASS42.pin);  
connect(sineVoltage.n, ASS5.pin);  
connect(sineVoltage.p, ASS42.pin1);  
connect(sineVoltage.p, ASS41.pin1);  
connect(sineVoltage.p, ASS5.pin1);  
connect(ground.p, actorrele17.pin1);  
connect(ground.p, actorrele15.pin1);  
connect(ground.p, actorrele16.pin1);  
connect(ground.p, actorrele14.pin1);  
connect(ground.p, actorrele13.pin1);  
connect(ground.p, actorrele12.pin1);  
connect(ground.p, actorrele11.pin1);
```

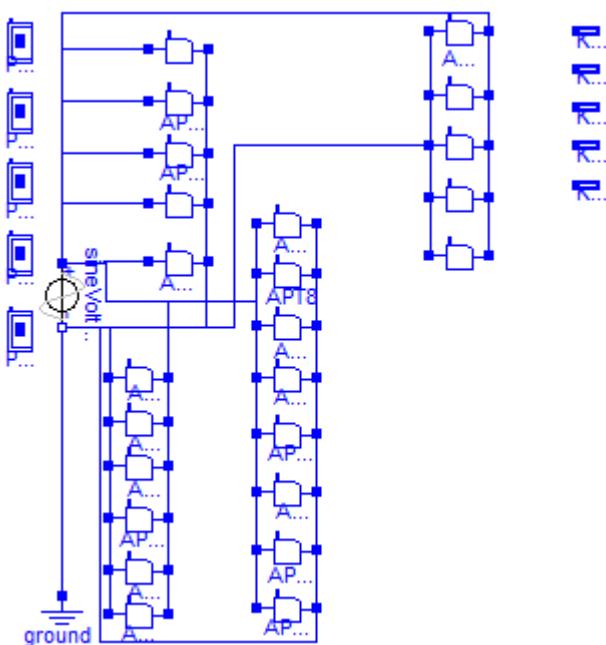
```

connect(ground.p, actorrele10.pin1);
connect(ground.p, actorrele9.pin1);
connect(ground.p, actorrele8.pin1);
connect(sineVoltage.p, actorrele8.pin);
connect(sineVoltage.p, ASRT1.pin);
connect(ground.p, ASRT1.pin1);
connect(actorrele8.pin, actorrele9.pin);
connect(actorrele9.pin, actorrele10.pin);
connect(actorrele10.pin, actorrele11.pin);
connect(actorrele11.pin, actorrele12.pin);
connect(actorrele12.pin, actorrele13.pin);
connect(actorrele13.pin, actorrele14.pin);
connect(actorrele14.pin, actorrele16.pin);
connect(actorrele16.pin, actorrele15.pin);
connect(actorrele15.pin, actorrele17.pin);
end esotano;

```

DOMOSYS_eltako.viviendaejemplo.eprimerpiso

Sumatoria del gasto energético de los componentes del sistema Eltako inalámbrico, que se encuentran en el primer piso.



Modelica definition

```

model eprimerpiso
    "Sumatoria del gasto energético de los componentes del sistema
    Eltako inalambrico, que se encuentran en el primer piso."
    Modelica.SIunits.Power gastoprimerpiso;
        Modelica.Electrical.Analog.Sources.SineVoltage sineVoltage;
        Modelica.Electrical.Analog.Basic.Ground ground;
        DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv1(
            identificacion=111);
        DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv2(
            identificacion=112);
        DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv3(
            identificacion=113);
        DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv4(
            identificacion=114);
        DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv5(
            identificacion=115);
        DOMOSYS.eltako.actuador.actorrele actorrele1(
            iden=211,
            sensor1=111,
            activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele2(
            iden=212, sensor1=112,
            activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele3(
            iden=213, sensor1=113,
            activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele4(
            iden=214, sensor1=114,
            activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele5(
            iden=215, sensor1=115,
            activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele6(
            iden=216, activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele7(
            iden=217, activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele8(
            iden=218, activ=true);
        DOMOSYS.eltako.actuador.actorrele actorrele9(

```

```

        iden=219, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele10(
        iden=220, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele11(
        iden=221, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele12(
        iden=223, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele13(
        iden=224, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele14(
        iden=225, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele15(
        iden=226, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele16(
        iden=227, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele17(
        iden=228, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele18(
        iden=229, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele19(
        iden=222, activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele20(
    sensor1=311,
    sensor2=312,
    sensor3=313,
    sensor4=315,
    iden=211);

DOMOSYS_eltako.actuador.actorrele actorrele21(
    iden=212,
    sensor1=312,
    sensor2=311,
    sensor3=315,
    sensor4=313);

DOMOSYS_eltako.actuador.actorrele actorrele22(
    iden=213,
    sensor1=311,
    sensor2=312,
    sensor3=313,
    sensor4=315,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele23(

```

Librería DOMOSYS

```
        iden=234, sensor1=334,
        activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele24(
    iden=235,
    sensor1=331,
    sensor2=332,
    sensor3=333,
    sensor4=335,
    activ=true);

DOMOSYS_eltako.sensores.kvp kvp1(
    ident=311, activo=true);

DOMOSYS_eltako.sensores.kvp kvp2(
    ident=312, activo=true);

DOMOSYS_eltako.sensores.kvp kvp3(
    ident=313, activo=true);

DOMOSYS_eltako.sensores.kvp kvp4(
    ident=314);

DOMOSYS_eltako.sensores.kvp kvp5(
    ident=315, activo=true);

equation

gastoprimerpiso=(if actorrele1.activ==true then actorrele1.gasto-
energetico else 0)+(if actorrele2.activ==true then
actorrele2.gastoenergetico else 0)
            +(if actorrele3.activ==true then actorrele3.gasto-
energetico else 0)+(if actorrele4.activ==true then actorrele4.gas-
toenergetico else 0) +
            (if actorrele5.activ==true then actorrele5.gastoener-
getico else 0)+(if actorrele6.activ==true then actorrele6.gasto-
energetico else 0) +
            (if actorrele7.activ==true then actorrele7.gastoener-
getico else 0)+(if actorrele8.activ==true then actorrele8.gasto-
energetico else 0) +
            (if actorrele9.activ==true then actorrele9.gastoener-
getico else 0)+(if actorrele10.activ==true then actorrele10.gas-
toenergetico else 0) +
            (if actorrele11.activ==true then actorrele11.gasto-
energetico else 0)+(if actorrele12.activ==true then
actorrele12.gastoenergetico else 0) +
            (if actorrele13.activ==true then actorrele13.gasto-
energetico else 0)+(if actorrele14.activ==true then
actorrele14.gastoenergetico else 0) +
```

```

(if actorrele15.activ==true then actorrele15.gastoenergetico
else 0)+(if actorrele16.activ==true then
actorrele16.gastoenergetico else 0)+

(if actorrele17.activ==true then actorrele17.gastoenergetico
else 0)+(if actorrele18.activ==true then
actorrele18.gastoenergetico else 0)+

(if actorrele19.activ==true then actorrele19.gastoenergetico
else 0)+(if actorrele20.activ==true then
actorrele20.gastoenergetico else 0)+

(if actorrele21.activ==true then actorrele21.gastoenergetico
else 0)+(if actorrele22.activ==true then
actorrele22.gastoenergetico else 0)+

(if actorrele23.activ==true then actorrele23.gastoenergetico
else 0)+(if actorrele24.activ==true then
actorrele24.gastoenergetico else 0);

connect(sineVoltage.n, ground.p);
connect(sineVoltage.p, actorrele1.pin);
connect(sineVoltage.p, actorrele2.pin);
connect(sineVoltage.p, actorrele3.pin);
connect(sineVoltage.p, actorrele4.pin);
connect(sineVoltage.p, actorrele5.pin);
connect(sineVoltage.n, actorrele5.pin1);
connect(sineVoltage.n, actorrele4.pin1);
connect(sineVoltage.n, actorrele3.pin1);
connect(sineVoltage.n, actorrele2.pin1);
connect(sineVoltage.n, actorrele1.pin1);
connect(sineVoltage.n, actorrele6.pin);
connect(sineVoltage.n, actorrele7.pin);
connect(sineVoltage.n, actorrele8.pin);
connect(sineVoltage.n, actorrele9.pin);
connect(sineVoltage.n, actorrele10.pin);
connect(sineVoltage.n, actorrele11.pin);
connect(sineVoltage.n, actorrele18.pin1);
connect(sineVoltage.n, actorrele17.pin1);
connect(sineVoltage.n, actorrele16.pin1);
connect(sineVoltage.n, actorrele15.pin1);
connect(sineVoltage.n, actorrele14.pin1);
connect(sineVoltage.n, actorrele13.pin1);
connect(sineVoltage.n, actorrele12.pin1);
connect(sineVoltage.n, actorrele19.pin1);
connect(sineVoltage.p, actorrele6.pin1);

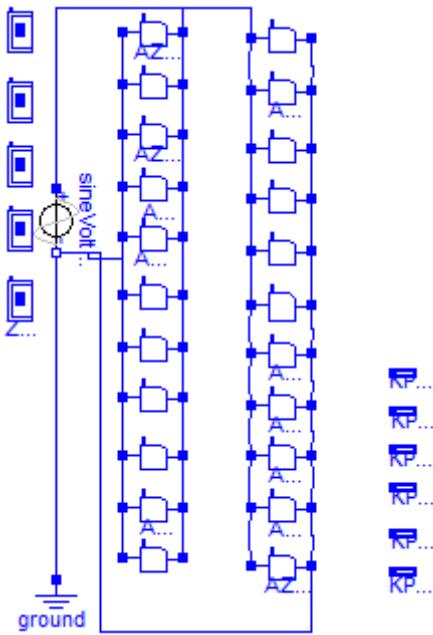
```

Librería DOMOSYS

```
connect(sineVoltage.p, actorrele7.pin1);
connect(sineVoltage.p, actorrele8.pin1);
connect(sineVoltage.p, actorrele9.pin1);
connect(sineVoltage.p, actorrele10.pin1);
connect(sineVoltage.p, actorrele11.pin1);
connect(sineVoltage.p, actorrele12.pin1);
connect(sineVoltage.p, actorrele19.pin1);
connect(sineVoltage.p, actorrele13.pin1);
connect(sineVoltage.p, actorrele14.pin1);
connect(sineVoltage.p, actorrele15.pin1);
connect(sineVoltage.p, actorrele16.pin1);
connect(sineVoltage.p, actorrele17.pin1);
connect(sineVoltage.p, actorrele18.pin1);
connect(sineVoltage.n, actorrele22.pin);
connect(actorrele24.pin, actorrele23.pin);
connect(actorrele23.pin, actorrele22.pin);
connect(actorrele22.pin, actorrele21.pin);
connect(actorrele21.pin, actorrele20.pin);
connect(actorrele24.pin1, actorrele23.pin1);
connect(actorrele23.pin1, actorrele22.pin1);
connect(actorrele22.pin1, actorrele21.pin1);
connect(actorrele21.pin1, actorrele20.pin1);
connect(sineVoltage.p, actorrele20.pin1);
end eprimerpiso;
```

DOMOSYS.eltako.viviendaejemplo.esegundopiso

Sumatoria del gasto energético de los componentes del sistema Eltako inalámbrico, que se encuentran en el segundo piso.



Modelica definition

```

model eseundopiso
  "Sumatoria del gasto energético de los componentes del sistema
  Eltako inalambrico, que se encuentran en el segundo piso."
  Modelica.SIunits.Power gastosegundopiso;
  DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv1(
    identificacion=1111,
    activo=true);
  Modelica.Electrical.Analog.Sources.SineVoltage sineVoltage;
  Modelica.Electrical.Analog.Basic.Ground ground;
  DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv2(
    identificacion=1112,
    activo=true);
  DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv3(
    identificacion=1113,
    activo=true);
  DOMOSYS.eltako.sensores.interruptorpasiv interruptorpasiv4(
    identificacion=1114,
    activo=true);
  DOMOSYS.eltako.actuador.actorrele actorrele1(
    sensor1=1111,
    iden=2111,
    activ=true);
  DOMOSYS.eltako.actuador.actorrele actorrele2(

```

Librería DOMOSYS

```
iden=2112,  
sensor1=1112,  
activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele3(  
iden=2113,  
sensor1=1113,  
activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele4(  
iden=2114,  
sensor1=1114,  
sensor2=1115,  
activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele5(  
iden=2115,  
sensor1=1115,  
activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele6(  
iden=2116, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele7(  
iden=2117, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele8(  
iden=2118, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele9(  
iden=2119, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele10(  
iden=2120, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele11(  
iden=2121, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele12(  
iden=2122, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele13(  
iden=2123, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele14(  
iden=2124, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele15(  
iden=2125, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele16(  
iden=2126, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele17(  
iden=2127, activ=true);  
DOMOSYS.eltako.actuador.actorrele actorrele18(
```

```

iden=2128,
sensor1=3111,
activ=true,
sensor2=3112);

DOMOSYS_eltako.actuador.actorrele actorrele19(
    activ=true,
    iden=2129,
    sensor1=3111,
    sensor2=3112);

DOMOSYS_eltako.actuador.actorrele actorrele20(
    activ=true,
    sensor1=3113,
    sensor2=3114,
    iden=2130);

DOMOSYS_eltako.actuador.actorrele actorrele21(
    activ=true,
    iden=2131,
    sensor1=3114,
    sensor2=3113);

DOMOSYS_eltako.actuador.actorrele actorrele22(
    activ=true,
    iden=2132,
    sensor1=3115,
    sensor2=3116);

DOMOSYS_eltako.sensores.kvp kvp1(
    ident=3111, activo=true);

DOMOSYS_eltako.sensores.kvp kvp2(
    ident=3112, activo=true);

DOMOSYS_eltako.sensores.kvp kvp3(
    ident=3113, activo=true);

DOMOSYS_eltako.sensores.kvp kvp4(
    ident=3114, activo=true);

DOMOSYS_eltako.sensores.kvp kvp5(
    ident=3115, activo=true);

DOMOSYS_eltako.sensores.interruptorpasiv interruptorpasiv5(
    identificacion=1115,
activo=true);

DOMOSYS_eltako.sensores.kvp kvp6(
    ident=3116, activo=true);

equation

```

Librería DOMOSYS

```
gastosegundopiso=(if actorrele1.activ==true then actorrele1.gastoenergetico else 0)+(if actorrele2.activ==true then actorrele2.gastoenergetico else 0)+  
          (if actorrele3.activ==true then actorrele3.gastoenergetico else 0)+(if actorrele4.activ==true then actorrele4.gastoenergetico else 0)+  
          (if actorrele5.activ==true then actorrele5.gastoenergetico else 0)+(if actorrele6.activ==true then actorrele6.gastoenergetico else 0)+  
          (if actorrele7.activ==true then actorrele7.gastoenergetico else 0)+(if actorrele8.activ==true then actorrele8.gastoenergetico else 0)+  
          (if actorrele9.activ==true then actorrele9.gastoenergetico else 0)+(if actorrele10.activ==true then actorrele10.gastoenergetico else 0)+  
          (if actorrele11.activ==true then actorrele11.gastoenergetico else 0)+(if actorrele12.activ==true then actorrele12.gastoenergetico else 0)+  
          (if actorrele13.activ==true then actorrele13.gastoenergetico else 0)+(if actorrele14.activ==true then actorrele14.gastoenergetico else 0)+  
          (if actorrele15.activ==true then actorrele15.gastoenergetico else 0)+(if actorrele16.activ==true then actorrele16.gastoenergetico else 0)+  
          (if actorrele17.activ==true then actorrele17.gastoenergetico else 0)+(if actorrele18.activ==true then actorrele18.gastoenergetico else 0)+  
          (if actorrele19.activ==true then actorrele19.gastoenergetico else 0)+(if actorrele20.activ==true then actorrele20.gastoenergetico else 0)+  
          (if actorrele21.activ==true then actorrele21.gastoenergetico else 0)+(if actorrele22.activ==true then actorrele22.gastoenergetico else 0);  
connect(sineVoltage.n, ground.p);  
connect(sineVoltage.n, actorrele5.pin);  
connect(sineVoltage.p, actorrele1.pin1);  
connect(actorrele1.pin1, actorrele2.pin1);  
connect(actorrele2.pin1, actorrele3.pin1);  
connect(actorrele3.pin1, actorrele4.pin1);  
connect(actorrele4.pin1, actorrele5.pin1);  
connect(actorrele5.pin, actorrele4.pin);
```

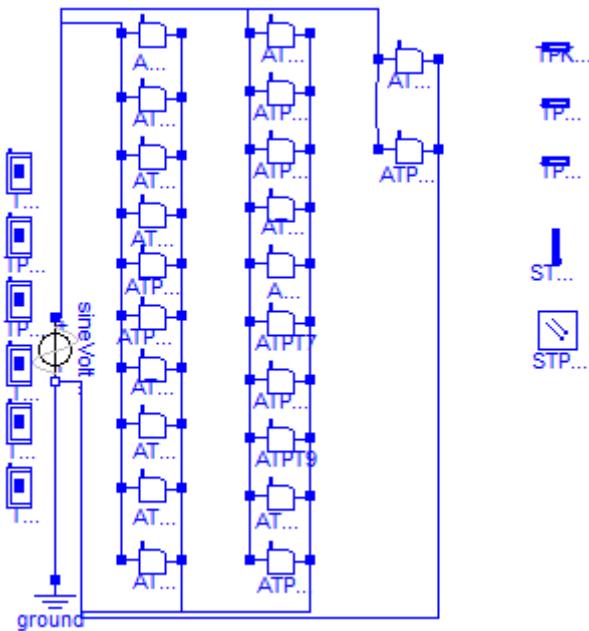
```

connect(actorrele4.pin, actorrele3.pin);
connect(actorrele3.pin, actorrele2.pin);
connect(actorrele2.pin, actorrele1.pin);
connect(sineVoltage.p, actorrele12.pin);
connect(actorrele5.pin, actorrele6.pin);
connect(actorrele6.pin, actorrele7.pin);
connect(actorrele7.pin, actorrele8.pin);
connect(actorrele8.pin, actorrele9.pin);
connect(actorrele9.pin, actorrele10.pin);
connect(actorrele5.pin1, actorrele6.pin1);
connect(actorrele6.pin1, actorrele7.pin1);
connect(actorrele7.pin1, actorrele8.pin1);
connect(actorrele8.pin1, actorrele9.pin1);
connect(actorrele9.pin1, actorrele10.pin1);
connect(actorrele10.pin, actorrele11.pin);
connect(actorrele10.pin1, actorrele11.pin1);
connect(actorrele12.pin, actorrele13.pin);
connect(actorrele13.pin, actorrele14.pin);
connect(actorrele14.pin, actorrele15.pin);
connect(actorrele15.pin, actorrele16.pin);
connect(actorrele16.pin, actorrele17.pin);
connect(actorrele17.pin, actorrele18.pin);
connect(actorrele18.pin, actorrele19.pin);
connect(actorrele19.pin, actorrele20.pin);
connect(actorrele20.pin, actorrele21.pin);
connect(actorrele21.pin, actorrele22.pin);
connect(sineVoltage.n, actorrele22.pin1);
connect(actorrele22.pin1, actorrele21.pin1);
connect(actorrele21.pin1, actorrele20.pin1);
connect(actorrele20.pin1, actorrele19.pin1);
connect(actorrele19.pin1, actorrele18.pin1);
connect(actorrele18.pin1, actorrele17.pin1);
connect(actorrele17.pin1, actorrele16.pin1);
connect(actorrele16.pin1, actorrele15.pin1);
connect(actorrele15.pin1, actorrele14.pin1);
connect(actorrele14.pin1, actorrele13.pin1);
connect(actorrele13.pin1, actorrele12.pin1);
end esegundopiso;

```

DOMOSYS_eltako.viviendaejemplo.etercerpiso

Sumatoria del gasto energético de los componentes del sistema Eltako inalambrico, que se encuentran en el tercer piso.



Modelica definition

```
model etercerpiso
  "Sumatoria del gasto energético de los componentes del sistema
  Eltako inalambrico, que se encuentran en el tercer piso."
  Modelica.SIunits.Power gastotercepiso;
  Modelica.Electrical.Analog.Sources.SineVoltage sineVoltage;
  Modelica.Electrical.Analog.Basic.Ground ground;
  DOMOSYS_eltako.sensores.interruptorpasiv1(
    identificacion=11111, acti-
    vo=true);
  DOMOSYS_eltako.sensores.interruptorpasiv2(
    identificacion=11112, acti-
    vo=true);
  DOMOSYS_eltako.sensores.interruptorpasiv3(
    identificacion=11113, acti-
    vo=true);
  DOMOSYS_eltako.sensores.interruptorpasiv4(
    identificacion=11114, acti-
    vo=true);
```

```
DOMOSYS_eltako.sensores.interruptorpasiv interruptorpasiv5(
    identificacion=11115, acti-
vo=true);

DOMOSYS_eltako.sensores.interruptorpasiv interruptorpasiv6(
    identificacion=11116, acti-
vo=true);

DOMOSYS_eltako.actuador.actorrele actorrele1(
    iden=21111,
    sensor1=11111,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele2(
    iden=21111,
    sensor1=11112,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele3(
    iden=21113,
    activ=true,
    sensor1=11113);

DOMOSYS_eltako.actuador.actorrele actorrele4(
    iden=21114,
    sensor1=11113,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele5(
    iden=21115,
    sensor1=11113,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele6(
    iden=21116,
    activ=true,
    sensor1=11114);

DOMOSYS_eltako.actuador.actorrele actorrele7(
    iden=21117,
    sensor1=11115,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele8(
    iden=21118,
    sensor1=11116,
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele9(
    iden=21119,
    sensor1=11116,
```

Librería DOMOSYS

```
    activ=true);

DOMOSYS_eltako.actuador.actorrele actorrele10(
    iden=21111, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele11(
    iden=21112, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele12(
    activ=true, iden=21113);
DOMOSYS_eltako.actuador.actorrele actorrele13(
    activ=true, iden=21114);
DOMOSYS_eltako.actuador.actorrele actorrele14(
    activ=true, iden=21115);
DOMOSYS_eltako.actuador.actorrele actorrele15(
    iden=21116, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele16(
    iden=21117, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele17(
    activ=true, iden=21118);
DOMOSYS_eltako.actuador.actorrele actorrele18(
    iden=21119, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele19(
    iden=21120, activ=true);
DOMOSYS_eltako.actuador.actorrele actorrele20(
    iden=21121);

DOMOSYS_eltako.actuador.actorrele actorrele21(
    activ=true, iden=21123);
DOMOSYS_eltako.actuador.actorrele actorrele22(
    iden=21122, activ=true);

DOMOSYS_eltako.sensores.kvp kvp1(
    ident=31111, activo=true);
DOMOSYS_eltako.sensores.kvp kvp2(
    ident=31112, activo=true);
DOMOSYS_eltako.sensores.kvp kvp3(
    ident=31113, activo=true);

DOMOSYS_eltako.sensores.stemperatura stemperatural1(
    identificacion=41111, activo=true);
DOMOSYS_eltako.sensores.Detectorhumo detectorhumo(
    ident=51111, activ=true);

equation
    gastotercerpiso=(if actorrele1.activ==true then actorrele1.gastoenergetico
        else 0)+(if actorrele2.activ==true then
            actorrele2.gastoenergetico else 0)+
```

```

(if actorrele3.activ==true then actorrele3.gastoener-
getico else 0)+(if actorrele4.activ==true then actorrele4.gastoene-
rgetico else 0)+

(if actorrele5.activ==true then actorrele5.gastoener-
getico else 0)+(if actorrele6.activ==true then actorrele6.gastoene-
rgetico else 0)+

(if actorrele7.activ==true then actorrele7.gastoener-
getico else 0)+(if actorrele8.activ==true then actorrele8.gastoene-
rgetico else 0)+

(if actorrele9.activ==true then actorrele9.gastoener-
getico else 0)+(if actorrele10.activ==true then actorrele10.gas-
toenergetico else 0)+

(if actorrele11.activ==true then actorrele11.gastoene-
rgetico else 0)+(if actorrele12.activ==true then
actorrele12.gastoenergetico else 0)+

(if actorrele13.activ==true then actorrele13.gastoene-
rgetico else 0)+(if actorrele14.activ==true then
actorrele14.gastoenergetico else 0)+

(if actorrele15.activ==true then actorrele15.gastoene-
rgetico else 0)+(if actorrele16.activ==true then
actorrele16.gastoenergetico else 0)+

(if actorrele17.activ==true then actorrele17.gastoene-
rgetico else 0)+(if actorrele18.activ==true then
actorrele18.gastoenergetico else 0)+

(if actorrele19.activ==true then actorrele19.gastoene-
rgetico else 0)+(if actorrele20.activ==true then
actorrele20.gastoenergetico else 0)+

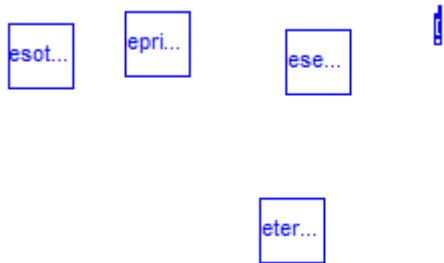
(if actorrele21.activ==true then actorrele21.gastoene-
rgetico else 0)+(if actorrele22.activ==true and
stemeratural.Temperatura<actorrele22.temperaturade-
seada then actorrele22.gastoenergetico else 0);

connect(sineVoltage.n, ground.p);
connect(sineVoltage.p, actorrele1.pin);
connect(actorrele1.pin, actorrele2.pin);
connect(actorrele2.pin, actorrele3.pin);
connect(actorrele3.pin, actorrele4.pin);
connect(actorrele4.pin, actorrele5.pin);
connect(actorrele5.pin, actorrele6.pin);
connect(actorrele6.pin, actorrele7.pin);
connect(actorrele7.pin, actorrele8.pin);
connect(actorrele8.pin, actorrele9.pin);

```

Librería DOMOSYS

```
connect(actorrele9.pin, actorrele10.pin);
connect(sineVoltage.n, actorrele10.pin1);
connect(actorrele10.pin1, actorrele9.pin1);
connect(actorrele9.pin1, actorrele8.pin1);
connect(actorrele8.pin1, actorrele7.pin1);
connect(actorrele7.pin1, actorrele6.pin1);
connect(actorrele6.pin1, actorrele5.pin1);
connect(actorrele5.pin1, actorrele4.pin1);
connect(actorrele4.pin1, actorrele3.pin1);
connect(actorrele3.pin1, actorrele2.pin1);
connect(actorrele2.pin1, actorrele1.pin1);
connect(sineVoltage.p, actorrele11.pin);
connect(actorrele11.pin, actorrele12.pin);
connect(actorrele12.pin, actorrele13.pin);
connect(actorrele13.pin, actorrele14.pin);
connect(actorrele14.pin, actorrele15.pin);
connect(actorrele15.pin, actorrele16.pin);
connect(actorrele16.pin, actorrele17.pin);
connect(actorrele17.pin, actorrele18.pin);
connect(actorrele18.pin, actorrele19.pin);
connect(actorrele19.pin, actorrele20.pin);
connect(sineVoltage.n, actorrele20.pin1);
connect(actorrele20.pin1, actorrele19.pin1);
connect(actorrele19.pin1, actorrele18.pin1);
connect(actorrele18.pin1, actorrele17.pin1);
connect(actorrele17.pin1, actorrele16.pin1);
connect(actorrele16.pin1, actorrele15.pin1);
connect(actorrele15.pin1, actorrele14.pin1);
connect(actorrele14.pin1, actorrele13.pin1);
connect(actorrele13.pin1, actorrele12.pin1);
connect(actorrele12.pin1, actorrele11.pin1);
connect(sineVoltage.p, actorrele22.pin);
connect(actorrele22.pin, actorrele21.pin);
connect(sineVoltage.n, actorrele21.pin1);
connect(actorrele21.pin1, actorrele22.pin1);
end etercerpiso;
```

Sumatoria del gasto energético en toda la vivienda ejemplo.**Modelica definition**

```

model ge
    "Sumatoria del gasto energético en toda la vivienda ejemplo."
    Modelica.SIunits.Power total;
    esotano esotanol1;
    eprimerpiso eprimerpisol1;
    eseundopiso esegundopisol1;
    etercerpiso etercerpisol1;
    servidor servidor1;
equation
    total= esotanol1.gastosotano+eprimerpisol1.gastoprimerpisot+esegundopisol1.gastosegundopiso+etercerpisol1.gastotercerpiso+servidor1.potencia;

end ge;

```

HTML-documentation generated by [Dymola](#) Mon Aug 31 13:36:03 2015.

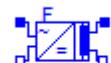
DOMOSYS.SBKNX

Paquete con el Sistema KNX/EIB de la forma TP por el medio cable par trenzado.

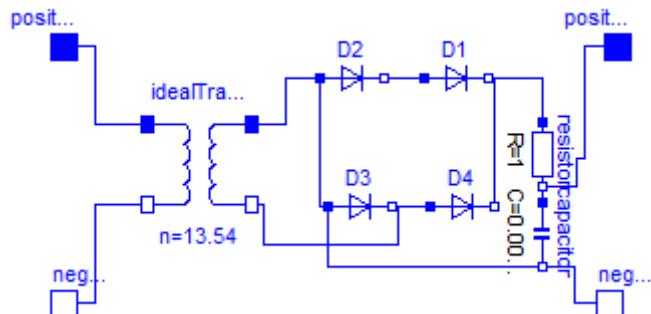
Package Content

Name	Description
FKNX	Fuente del Sistema KNX/EIB
ALKNX	Acoplador de Linea, separa las impedancias entre dos lineas.
ABKNX	Parte inteligente del sistema, decodifica y codifica la información.
vivienda	Implementación del sistema KNX en la vivienda ejemplo.

[DOMOSYS.SBKNX.FKNX](#)



Fuente del Sistema KNX/EIB



Connectors

Type	Name	Description
PositivePin	positivePin	
NegativePin	negativePin	
PositivePin	positivePin1	
NegativePin	negativePin1	

Modelica definition

```
model FKNX "Fuente del Sistema KNX/EIB"
```

```

Modelica.Electrical.Analog.Interfaces.PositivePin positivePin;
Modelica.Electrical.Analog.Interfaces.NegativePin negativePin;

Modelica.Electrical.Analog.Ideal.IdealDiode D2;
Modelica.Electrical.Analog.Ideal.IdealDiode D1;
Modelica.Electrical.Analog.Ideal.IdealDiode D3;
Modelica.Electrical.Analog.Ideal.IdealDiode D4;
Modelica.Electrical.Analog.Interfaces.PositivePin positivePin1;
Modelica.Electrical.Analog.Interfaces.NegativePin negativePin1;
Modelica.Electrical.Analog.Basic.Resistor resistor;
Modelica.Electrical.Analog.Basic.Capacitor capacitor(C=0.0001);
Modelica.Electrical.Analog.Ideal.IdealTransformer idealTransformer(n=13.54);

equation
  connect(D2.n, D1.p);
  connect(D2.p, D3.p);
  connect(D3.n, D4.p);
  connect(D1.n, D4.n);
  connect(resistor.n, capacitor.p);
  connect(D1.n, resistor.p);
  connect(D3.p, capacitor.n);
  connect(capacitor.n, negativePin1);
  connect(resistor.n, positivePin1);
  connect(negativePin, idealTransformer.n1);
  connect(positivePin, idealTransformer.p1);
  connect(idealTransformer.p2, D2.p);
  connect(idealTransformer.n2, D3.n);
end FKNX;

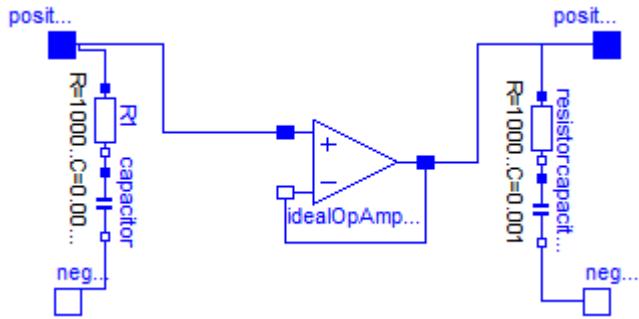
```

DOMOSYS.SBKNX.ALKNX



Acoplador de Linea, separa las impedancias entre dos lineas.

Librería DOMOSYS



Connectors

Type	Name	Description
PositivePin	positive-Pin	
NegativePin	negative-Pin	
PositivePin	positive-Pin1	
NegativePin	negative-Pin1	

Modelica definition

```

model ALKNX
    "Acoplador de Linea, separa las impedancias entre dos lineas."
    Modelica.Electrical.Analog.Interfaces.PositivePin positivePin;
    Modelica.Electrical.Analog.Interfaces.NegativePin negativePin;
    Modelica.Electrical.Analog.Interfaces.PositivePin positivePin1;
    Modelica.Electrical.Analog.Interfaces.NegativePin negativePin1;
    Modelica.Electrical.Analog.Ideal.IdealOpAmp3Pin idealOpAmp3Pin;

    Modelica.Electrical.Analog.Basic.Resistor R1(R=10000000);
    Modelica.Electrical.Analog.Basic.Capacitor capacitor(C=0.0001);
    Modelica.Electrical.Analog.Basic.Resistor resistor(R=10000000);

```

```

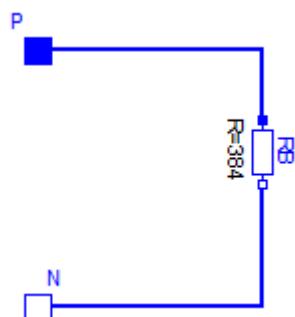
Modelica.Electrical.Analog.Basic.Capacitor capacitor1(C=0.001);
equation
  connect(positivePin, idealOpAmp3Pin.in_p);
  connect(idealOpAmp3Pin.out, positivePin1);
  connect(idealOpAmp3Pin.in_n, idealOpAmp3Pin.out);
  connect(positivePin, R1.p);
  connect(R1.n, capacitor.p);
  connect(negativePin, capacitor.n);
  connect(positivePin1, resistor.p);
  connect(resistor.n, capacitor1.p);
  connect(capacitor1.n, negativePin1);
end ALKNX;

```

DOMOSYS.SBKNX.ABKNX



Parte inteligente del sistema, decodifica y codifica la información.



Parameters

Type	Name	De-fault	Description
Cu-current	corriente	0.0625	[A]
Power	Dp	0.150	[W]

Connectors

Type	Name	Descrip-
------	------	----------

Librería DOMOSYS

		tion
<u>PositivePin</u>	P	
<u>NegativePin</u>	N	

Modelica definition

```
model ABKNX
  "Parte inteligente del sistema, decodifica y codifica la información."
  Modelica.Electrical.Analog.Interfaces.PositivePin P;
  Modelica.Electrical.Analog.Interfaces.NegativePin N;
  Modelica.Electrical.Analog.Basic.Resistor RB(R=384);
  parameter Modelica.SIunits.Current corriente=0.0625;
  parameter Modelica.SIunits.Power Dp=0.150;
  Modelica.SIunits.Voltage voltaje;

  equation
    voltaje=P.v-N.v;
    connect(P, RB.p);
    connect(RB.n, N);
  end ABKNX;
```

HTML-documentation generated by [Dymola](#) Mon Aug 31 23:08:53 2015.

DOMOSYS.SBKNX.vivienda

Implementación del sistema KNX en la vivienda ejemplo.

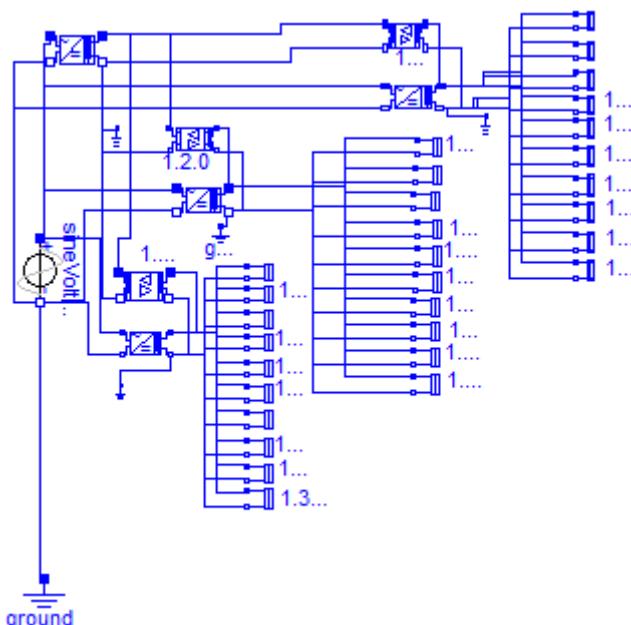
Package Content

Name	Description
KNXsotano	Implementación del Sistema KNX en el sótano.
PPKNX	Implementación del sistema KNX en el primer piso.

SPKNX	Implementación del sistema KNX/EIB en el segundo piso.
TPKNX	Implementación del sistema KNX/EIB del tercer piso.
gt	La sumatoria del gasto energético de cada uno de los niveles de la vivienda ejemplo.

DOMOSYS.SBKNX.vivienda.KNXsotano

Implementación del Sistema KNX en el sótano.



Parameters

Type	Name	Default	Description
Integer	componentes	30	

Modelica definition

```
model KNXsotano "Implementación del Sistema KNX en el sótano."
parameter Integer componentes=30;
Integer AK1;
Real NAL;
```

Librería DOMOSYS

```
Modelica.Electrical.Analog.Sources.SineVoltage
sineVoltage(V=325, freqHz=50);
Modelica.Electrical.Analog.Basic.Ground ground;
Modelica.SIunits.Power gasto;
FKNX fKNX;
ALKNX aLKNX;
FKNX fKNX1;
ABKNX aBKNX;
ABKNX aBKNX1;
ABKNX aBKNX2;
ABKNX aBKNX3;
Modelica.Electrical.Analog.Basic.Ground ground1;
Modelica.Electrical.Analog.Basic.Ground ground2;
ABKNX aBKNX4;
ABKNX aBKNX5;
ABKNX aBKNX6;
ABKNX aBKNX7;
ABKNX aBKNX8;
ABKNX aBKNX9;
ALKNX aLKNX1;
FKNX fKNX2;
ABKNX aBKNX10;
ABKNX aBKNX11;
ABKNX aBKNX12;
ABKNX aBKNX13;
ABKNX aBKNX14;
ABKNX aBKNX15;
ABKNX aBKNX16;
ABKNX aBKNX17;
ABKNX aBKNX18;
ABKNX aBKNX19;
ALKNX aLKNX2;
FKNX fKNX3;
ABKNX aBKNX20;
ABKNX aBKNX21;
ABKNX aBKNX22;
Modelica.Electrical.Analog.Basic.Ground ground4;
ABKNX aBKNX23;
ABKNX aBKNX24;
ABKNX aBKNX25;
ABKNX aBKNX26;
```

```

ABKNX aBKNX27;
ABKNX aBKNX28;
ABKNX aBKNX29;
Modelica.Electrical.Analog.Basic.Ground ground3;
equation

    NAL=componentes/10;
    AK1=integer(NAL);

    gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);

    connect(fKNX3.negativePin1, ground4.p);
    connect(fKNX3.positivePin1, aBKNX20.P);
    connect(fKNX3.positivePin1, aBKNX21.P);
    connect(fKNX3.positivePin1, aBKNX22.P);
    connect(fKNX3.positivePin1, aBKNX23.P);
    connect(fKNX3.positivePin1, aBKNX24.P);
    connect(fKNX3.positivePin1, aBKNX26.P);
    connect(fKNX3.positivePin1, aBKNX25.P);
    connect(fKNX3.positivePin1, aBKNX27.P);
    connect(fKNX3.positivePin1, aBKNX28.P);
    connect(fKNX3.positivePin1, aBKNX29.P);
    connect(fKNX3.negativePin1, aBKNX20.N);
    connect(fKNX3.negativePin1, aBKNX21.N);
    connect(fKNX3.negativePin1, aBKNX22.N);
    connect(fKNX3.negativePin1, aBKNX24.N);
    connect(fKNX3.negativePin1, aBKNX26.N);
    connect(fKNX3.negativePin1, aBKNX25.N);
    connect(fKNX3.negativePin1, aBKNX27.N);
    connect(fKNX3.negativePin1, aBKNX28.N);
    connect(fKNX3.negativePin1, aBKNX29.N);
    connect(fKNX.positivePin1, aLKNX.positivePin);
    connect(fKNX.negativePin1, aLKNX.negativePin);
    connect(aLKNX.negativePin1, fKNX1.negativePin1);
    connect(aLKNX.positivePin1, fKNX1.positivePin1);
    connect(fKNX1.negativePin1, ground2.p);
    connect(fKNX1.positivePin1, aBKNX.P);
    connect(fKNX1.positivePin1, aBKNX1.P);
    connect(fKNX1.positivePin1, aBKNX2.P);
    connect(fKNX1.positivePin1, aBKNX3.P);
    connect(fKNX1.positivePin1, aBKNX4.P);

```

Librería DOMOSYS

```
connect(fKNX1.positivePin1, aBKNX5.P);
connect(fKNX1.positivePin1, aBKNX6.P);
connect(fKNX1.positivePin1, aBKNX7.P);
connect(fKNX1.positivePin1, aBKNX8.P);
connect(fKNX1.positivePin1, aBKNX9.P);
connect(fKNX1.negativePin1, aBKNX.N);
connect(fKNX1.negativePin1, aBKNX1.N);
connect(fKNX1.negativePin1, aBKNX2.N);
connect(fKNX1.negativePin1, aBKNX3.N);
connect(fKNX1.negativePin1, aBKNX4.N);
connect(fKNX1.negativePin1, aBKNX5.N);
connect(fKNX1.negativePin1, aBKNX6.N);
connect(fKNX1.negativePin1, aBKNX7.N);
connect(fKNX1.negativePin1, aBKNX8.N);
connect(fKNX1.negativePin1, aBKNX9.N);
connect(sineVoltage.n, ground.p);
connect(sineVoltage.n, fKNX.negativePin);
connect(sineVoltage.n, fKNX1.negativePin);
connect(sineVoltage.p, fKNX.positivePin);
connect(sineVoltage.p, fKNX1.positivePin);
connect(aLKNX1.negativePin, fKNX.negativePin1);
connect(fKNX.positivePin1, aLKNX1.positivePin);
connect(ground1.p, fKNX.negativePin1);
connect(sineVoltage.p, fKNX2.positivePin);
connect(sineVoltage.n, fKNX2.negativePin);
connect(aLKNX1.positivePin1, fKNX2.positivePin1);
connect(aLKNX1.negativePin1, fKNX2.negativePin1);
connect(fKNX2.positivePin1, aBKNX10.P);
connect(fKNX2.positivePin1, aBKNX11.P);
connect(fKNX2.positivePin1, aBKNX12.P);
connect(fKNX2.positivePin1, aBKNX13.P);
connect(fKNX2.positivePin1, aBKNX14.P);
connect(fKNX2.positivePin1, aBKNX15.P);
connect(fKNX2.positivePin1, aBKNX16.P);
connect(fKNX2.positivePin1, aBKNX17.P);
connect(fKNX2.positivePin1, aBKNX18.P);
connect(fKNX2.positivePin1, aBKNX19.P);
connect(fKNX2.negativePin1, aBKNX10.N);
connect(fKNX2.negativePin1, aBKNX11.N);
connect(aBKNX12.N, fKNX2.negativePin1);
connect(fKNX2.negativePin1, aBKNX13.N);
```

```

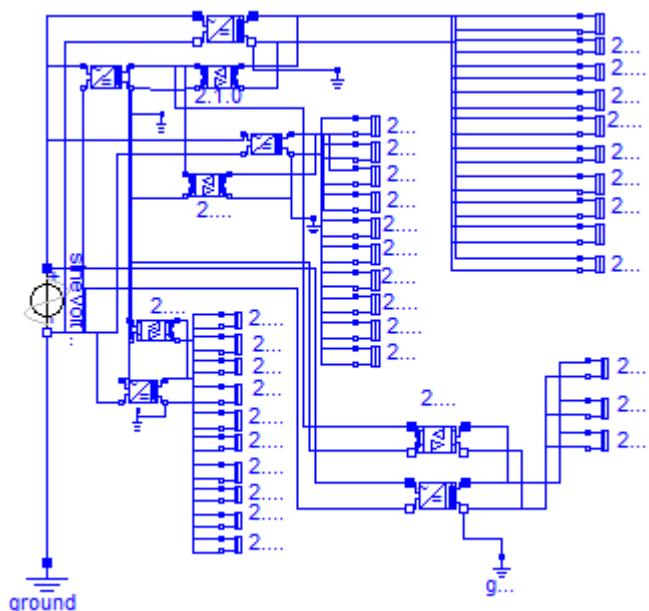
connect(fKNX2.negativePin1, aBKNX14.N);
connect(fKNX2.negativePin1, aBKNX15.N);
connect(fKNX2.negativePin1, aBKNX16.N);
connect(fKNX2.negativePin1, aBKNX17.N);
connect(fKNX2.negativePin1, aBKNX18.N);
connect(fKNX2.negativePin1, aBKNX19.N);
connect(aBKNX23.N, fKNX3.negativePin1);
connect(aLKNX2.positivePin1, fKNX3.positivePin1);
connect(aLKNX2.negativePin1, fKNX3.negativePin1);
connect(aLKNX2.negativePin, fKNX.negativePin1);
connect(fKNX.positivePin1, aLKNX2.positivePin);
connect(sineVoltage.p, fKNX3.positivePin);
connect(sineVoltage.n, fKNX3.negativePin);
connect(ground3.p, fKNX2.negativePin1);

end KNXsotano;

```

DOMOSYS.SBKNX.vivienda.PPKNX

Implementación del sistema KNX en el primer piso.



Parameters

Type	Name	Default	Description
Integer	componentes	33	

Modelica definition

```

model PPKNX "Implementación del sistema KNX en el primer piso."
  parameter Integer componentes=33;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;
  Modelica.Electrical.Analog.Sources.SineVoltage
  sineVoltage(V=325, freqHz=50);
  Modelica.Electrical.Analog.Basic.Ground ground;
  FKNX fKNX;
  ALKNX aKNX;
  Modelica.Electrical.Analog.Basic.Ground ground1;
  ABKNX aBKNX;
  ABKNX aBKNX1;
  ABKNX aBKNX2;
  ABKNX aBKNX3;
  ABKNX aBKNX4;
  ABKNX aBKNX5;
  ABKNX aBKNX6;
  ABKNX aBKNX7;
  ABKNX aBKNX8;
  ABKNX aBKNX9;
  FKNX fKNX1;
  Modelica.Electrical.Analog.Basic.Ground ground2;
  ALKNX aBKNX1;
  FKNX fKNX2;
  ABKNX aBKNX10;
  ABKNX aBKNX11;
  ABKNX aBKNX12;
  ABKNX aBKNX13;
  ABKNX aBKNX14;
  ABKNX aBKNX15;
  ABKNX aBKNX16;
  ABKNX aBKNX17;

```

```

ABKNX aBKNX18;
ABKNX aBKNX19;
ALKNX aLKNX2;
FKNX fKNX3;
ABKNX aBKNX20;
ABKNX aBKNX21;
ABKNX aBKNX22;
ABKNX aBKNX23;
ABKNX aBKNX24;
ABKNX aBKNX25;
ABKNX aBKNX26;
ABKNX aBKNX27;
ABKNX aBKNX28;
ABKNX aBKNX29;
Modelica.Electrical.Analog.Basic.Ground ground3;
Modelica.Electrical.Analog.Basic.Ground ground4;
FKNX fKNX4;
ALKNX aLKNX3;
Modelica.Electrical.Analog.Basic.Ground ground5;
ABKNX aBKNX30;
ABKNX aBKNX31;
ABKNX aBKNX32;

equation
    NAL=componentes/10;
    AK1=integer(NAL);
    gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);

    connect(sineVoltage.n, ground.p);
    connect(sineVoltage.p, fKNX.positivePin);
    connect(sineVoltage.n, fKNX.negativePin);
    connect(fKNX.negativePin1, ground1.p);
    connect(fKNX.negativePin1, aLKNX.negativePin);
    connect(sineVoltage.p, fKNX1.positivePin);
    connect(sineVoltage.n, fKNX1.negativePin);
    connect(aLKNX.negativePin1, fKNX1.negativePin1);
    connect(aLKNX.positivePin1, fKNX1.positivePin1);
    connect(fKNX1.negativePin1, aBKNX.N);
    connect(fKNX1.negativePin1, aBKNX1.N);
    connect(fKNX1.positivePin1, aBKNX1.P);
    connect(fKNX1.negativePin1, aBKNX2.N);
    connect(fKNX1.positivePin1, aBKNX2.P);

```

Librería DOMOSYS

```
connect(fKNX1.negativePin1, aBKNX3.N);
connect(fKNX1.positivePin1, aBKNX3.P);
connect(fKNX1.negativePin1, aBKNX4.N);
connect(fKNX1.positivePin1, aBKNX4.P);
connect(fKNX1.negativePin1, aBKNX5.N);
connect(fKNX1.positivePin1, aBKNX5.P);
connect(fKNX1.negativePin1, aBKNX6.N);
connect(fKNX1.positivePin1, aBKNX6.P);
connect(fKNX1.negativePin1, aBKNX7.N);
connect(fKNX1.positivePin1, aBKNX7.P);
connect(fKNX1.negativePin1, aBKNX8.N);
connect(fKNX1.positivePin1, aBKNX8.P);
connect(fKNX1.negativePin1, aBKNX9.N);
connect(fKNX1.positivePin1, aBKNX9.P);
connect(ground2.p, fKNX1.negativePin1);
connect(fKNX.positivePin1, aLKNX.positivePin);
connect(aLKNX1.positivePin1, fKNX2.positivePin1);
connect(fKNX2.negativePin1, aLKNX1.negativePin1);
connect(sineVoltage.p, fKNX2.positivePin);
connect(sineVoltage.n, fKNX2.negativePin);
connect(aLKNX1.negativePin, fKNX.negativePin1);
connect(fKNX.positivePin1, aLKNX1.positivePin);
connect(fKNX2.positivePin1, aBKNX10.P);
connect(fKNX2.negativePin1, aBKNX10.N);
connect(fKNX2.positivePin1, aBKNX11.P);
connect(fKNX2.negativePin1, aBKNX11.N);
connect(fKNX2.positivePin1, aBKNX12.P);
connect(fKNX2.negativePin1, aBKNX12.N);
connect(fKNX2.positivePin1, aBKNX13.P);
connect(fKNX2.negativePin1, aBKNX13.N);
connect(fKNX2.positivePin1, aBKNX14.P);
connect(fKNX2.negativePin1, aBKNX14.N);
connect(fKNX2.positivePin1, aBKNX15.P);
connect(fKNX2.negativePin1, aBKNX15.N);
connect(fKNX2.positivePin1, aBKNX16.P);
connect(fKNX2.negativePin1, aBKNX16.N);
connect(fKNX2.positivePin1, aBKNX17.P);
connect(fKNX2.negativePin1, aBKNX17.N);
connect(fKNX2.positivePin1, aBKNX18.P);
connect(fKNX2.negativePin1, aBKNX18.N);
connect(fKNX2.positivePin1, aBKNX19.P);
```

```

connect(fKNX2.negativePin1, aBKNX19.N);
connect(fKNX.positivePin1, aLKNX2.positivePin);
connect(aLKNX2.negativePin, fKNX.negativePin1);
connect(sineVoltage.n, fKNX3.negativePin);
connect(sineVoltage.p, fKNX3.positivePin);
connect(fKNX3.negativePin1, aLKNX2.negativePin1);
connect(fKNX3.positivePin1, aLKNX2.positivePin1);
connect(fKNX3.positivePin1, aBKNX20.P);
connect(fKNX3.negativePin1, aBKNX20.N);
connect(fKNX3.positivePin1, aBKNX21.P);
connect(fKNX3.negativePin1, aBKNX21.N);
connect(fKNX3.positivePin1, aBKNX22.P);
connect(fKNX3.negativePin1, aBKNX22.N);
connect(fKNX3.positivePin1, aBKNX23.P);
connect(fKNX3.negativePin1, aBKNX23.N);
connect(fKNX3.positivePin1, aBKNX24.P);
connect(fKNX3.negativePin1, aBKNX24.N);
connect(fKNX3.positivePin1, aBKNX25.P);
connect(fKNX3.negativePin1, aBKNX25.N);
connect(fKNX3.positivePin1, aBKNX26.P);
connect(fKNX3.negativePin1, aBKNX26.N);
connect(fKNX3.positivePin1, aBKNX27.P);
connect(fKNX3.negativePin1, aBKNX27.N);
connect(fKNX3.positivePin1, aBKNX28.P);
connect(fKNX3.negativePin1, aBKNX28.N);
connect(fKNX3.positivePin1, aBKNX29.P);
connect(fKNX3.negativePin1, aBKNX29.N);
connect(ground3.p, fKNX2.negativePin1);
connect(fKNX3.negativePin1, ground4.p);
connect(aLKNX3.negativePin1, fKNX4.negativePin1);
connect(aLKNX3.positivePin1, fKNX4.positivePin1);
connect(fKNX.positivePin1, aLKNX3.positivePin);
connect(aLKNX3.negativePin, fKNX.negativePin1);
connect(sineVoltage.p, fKNX4.positivePin);
connect(fKNX4.negativePin, sineVoltage.n);
connect(fKNX4.negativePin1, ground5.p);
connect(fKNX4.positivePin1, aBKNX30.P);
connect(fKNX4.negativePin1, aBKNX30.N);
connect(fKNX4.positivePin1, aBKNX31.P);
connect(fKNX4.positivePin1, aBKNX32.P);
connect(fKNX4.negativePin1, aBKNX31.N);

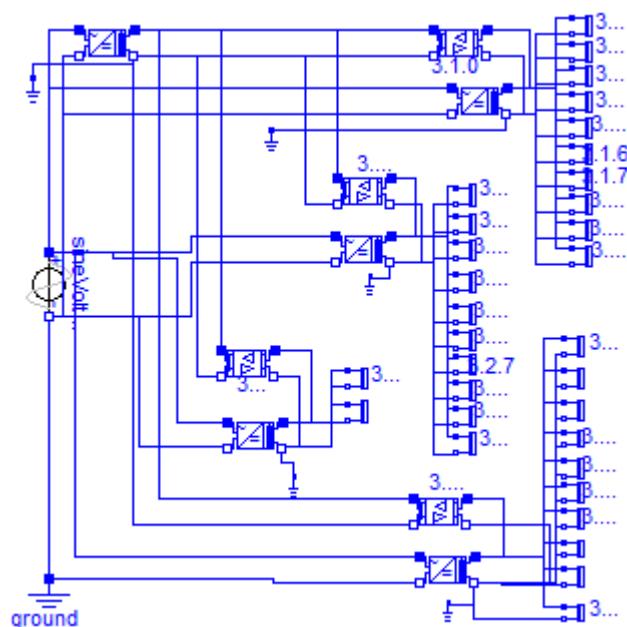
```

Librería DOMOSYS

```
connect (fKNX4.negativePin1, aBKNX32.N);
connect (aBKNX.P, fKNX1.positivePin1);
end PPKNX;
```

DOMOSYS.SBKNX.vivienda.SPKNX

Implementación del sistema KNX/EIB en el segundo piso.



Parameters

Type	Name	Default	Description
Integer	componentes	32	

Modelica definition

```
model SPKNX "Implementación del sistema KNX/EIB en el segundo
piso."
parameter Integer componentes=32;
Integer AK1;
Real NAL;
```

```
Modelica.SIunits.Power gasto;
Modelica.Electrical.Analog.Basic.Ground ground;
    Modelica.Electrical.Analog.Sources.SineVoltage
sineVoltage(V=325, freqHz=50);
    FKNX fKNX;
    ALKNX aLKNX;
    FKNX fKNX1;
    Modelica.Electrical.Analog.Basic.Ground ground1;
    ABKNX aBKNX;
    ABKNX aBKNX1;
    ABKNX aBKNX2;
    ABKNX aBKNX3;
    ABKNX aBKNX4;
    ABKNX aBKNX5;
    ABKNX aBKNX6;
    ABKNX aBKNX7;
    ABKNX aBKNX8;
    ABKNX aBKNX9;
    Modelica.Electrical.Analog.Basic.Ground ground2;
    FKNX fKNX2;
    ALKNX aLKNX1;
    ABKNX aBKNX10;
    ABKNX aBKNX11;
    ABKNX aBKNX12;
    ABKNX aBKNX13;
    ABKNX aBKNX14;
    ABKNX aBKNX15;
    ABKNX aBKNX16;
    ABKNX aBKNX17;
    ABKNX aBKNX18;
    ABKNX aBKNX19;
    Modelica.Electrical.Analog.Basic.Ground ground3;
    ALKNX aLKNX2;
    FKNX fKNX3;
    ABKNX aBKNX20;
    Modelica.Electrical.Analog.Basic.Ground ground4;
    ABKNX aBKNX21;
    ABKNX aBKNX22;
    ABKNX aBKNX23;
    ABKNX aBKNX24;
    ABKNX aBKNX25;
```

Librería DOMOSYS

```
ABKNX aBKNX26;
ABKNX aBKNX27;
ABKNX aBKNX28;
ABKNX aBKNX29;
ALKNX aLKNX3;
FKNX fKnx4;
ABKNX aBKNX30;
ABKNX aBKNX31;
Modelica.Electrical.Analog.Basic.Ground ground5;
equation
NAL=componentes/10;
AK1=integer(NAL);
gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);
connect(ground.p, sineVoltage.n);
connect(sineVoltage.p, fKnx4.positivePin);
connect(sineVoltage.n, fKnx4.negativePin);
connect(fKnx4.positivePin1, aLKNX.positivePin);
connect(fKnx4.negativePin1, aLKNX.negativePin);
connect(sineVoltage.p, fKnx1.positivePin);
connect(fKnx1.negativePin, sineVoltage.n);
connect(aLKNX.positivePin1, fKnx1.positivePin1);
connect(aLKNX.negativePin1, fKnx1.negativePin1);
connect(fKnx1.negativePin1, ground1.p);
connect(fKnx1.positivePin1, aBKNX.P);
connect(fKnx1.positivePin1, aBKNX1.P);
connect(fKnx1.positivePin1, aBKNX2.P);
connect(fKnx1.positivePin1, aBKNX3.P);
connect(fKnx1.positivePin1, aBKNX4.P);
connect(fKnx1.positivePin1, aBKNX5.P);
connect(fKnx1.positivePin1, aBKNX6.P);
connect(fKnx1.positivePin1, aBKNX7.P);
connect(fKnx1.positivePin1, aBKNX8.P);
connect(fKnx1.positivePin1, aBKNX9.P);
connect(fKnx1.negativePin1, aBKNX.N);
connect(fKnx1.negativePin1, aBKNX1.N);
connect(fKnx1.negativePin1, aBKNX2.N);
connect(fKnx1.negativePin1, aBKNX3.N);
connect(fKnx1.negativePin1, aBKNX4.N);
connect(fKnx1.negativePin1, aBKNX5.N);
connect(fKnx1.negativePin1, aBKNX6.N);
connect(fKnx1.negativePin1, aBKNX7.N);
```

```

connect(fKNX1.negativePin1, aBKNX8.N);
connect(fKNX1.negativePin1, aBKNX9.N);
connect(fKNX.negativePin1, ground2.p);
connect(sineVoltage.p, fKNX2.positivePin);
connect(ground.p, fKNX2.negativePin);
connect(aLKNX1.negativePin1, fKNX2.negativePin1);
connect(fKNX.negativePin1, aLKNX1.negativePin);
connect(fKNX.positivePin1, aLKNX1.positivePin);
connect(fKNX2.negativePin1, aBKNX19.N);
connect(fKNX2.negativePin1, aBKNX17.N);
connect(fKNX2.negativePin1, aBKNX16.N);
connect(fKNX2.negativePin1, aBKNX15.N);
connect(fKNX2.negativePin1, aBKNX14.N);
connect(fKNX2.negativePin1, aBKNX13.N);
connect(fKNX2.negativePin1, aBKNX12.N);
connect(fKNX2.negativePin1, aBKNX11.N);
connect(fKNX2.negativePin1, aBKNX10.N);
connect(fKNX2.positivePin1, aBKNX19.P);
connect(fKNX2.positivePin1, aBKNX18.P);
connect(fKNX2.positivePin1, aBKNX17.P);
connect(fKNX2.positivePin1, aBKNX16.P);
connect(fKNX2.positivePin1, aBKNX15.P);
connect(fKNX2.positivePin1, aBKNX14.P);
connect(fKNX2.positivePin1, aBKNX13.P);
connect(fKNX2.positivePin1, aBKNX12.P);
connect(fKNX2.positivePin1, aBKNX11.P);
connect(fKNX2.positivePin1, aBKNX10.P);
connect(fKNX2.negativePin1, ground3.p);
connect(aLKNX2.negativePin1, fKNX3.negativePin);
connect(aLKNX2.positivePin1, fKNX3.positivePin);
connect(fKNX.negativePin1, aLKNX2.negativePin);
connect(aLKNX2.positivePin, fKNX.positivePin);
connect(sineVoltage.n, fKNX3.negativePin);
connect(sineVoltage.p, fKNX3.positivePin);
connect(fKNX3.positivePin1, aBKNX20.P);
connect(fKNX3.positivePin1, aBKNX21.P);
connect(fKNX3.positivePin1, aBKNX22.P);
connect(fKNX3.positivePin1, aBKNX23.P);
connect(fKNX3.positivePin1, aBKNX24.P);
connect(fKNX3.positivePin1, aBKNX25.P);
connect(fKNX3.positivePin1, aBKNX26.P);

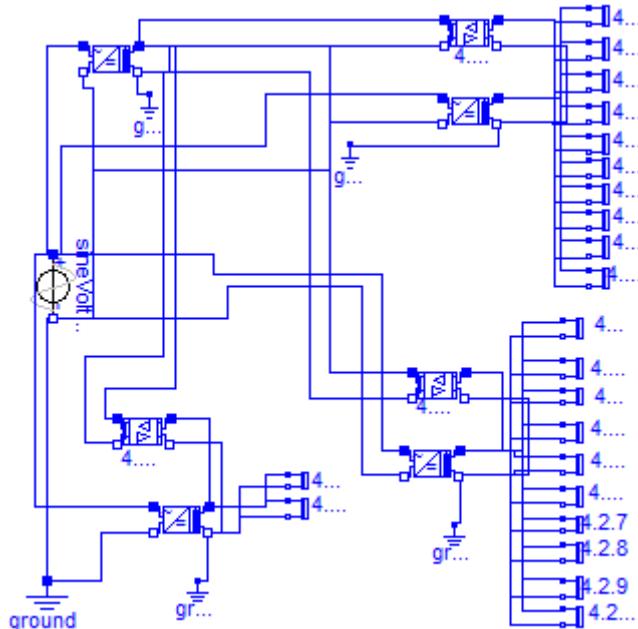
```

Librería DOMOSYS

```
connect(fKNX3.positivePin1, aBKNX27.P);
connect(fKNX3.positivePin1, aBKNX28.P);
connect(fKNX3.positivePin1, aBKNX29.P);
connect(fKNX3.negativePin1, aBKNX20.N);
connect(fKNX3.negativePin1, aBKNX21.N);
connect(fKNX3.negativePin1, aBKNX22.N);
connect(fKNX3.negativePin1, aBKNX23.N);
connect(fKNX3.negativePin1, aBKNX24.N);
connect(fKNX3.negativePin1, aBKNX25.N);
connect(fKNX3.negativePin1, aBKNX26.N);
connect(fKNX3.negativePin1, aBKNX27.N);
connect(fKNX3.negativePin1, aBKNX28.N);
connect(fKNX3.negativePin1, aBKNX29.N);
connect(fKNX.negativePin1, aLKNX3.negativePin);
connect(fKNX.positivePin1, aLKNX3.positivePin);
connect(sineVoltage.n, fKNX4.negativePin);
connect(sineVoltage.p, fKNX4.positivePin);
connect(fKNX4.negativePin1, ground5.p);
connect(fKNX4.positivePin1, aLKNX3.positivePin1);
connect(aLKNX3.negativePin1, fKNX4.negativePin1);
connect(fKNX4.positivePin1, aBKNX30.P);
connect(fKNX4.positivePin1, aBKNX31.P);
connect(fKNX4.negativePin1, aBKNX30.N);
connect(fKNX4.negativePin1, aBKNX31.N);
connect(fKNX3.negativePin1, ground4.p);
connect(aBKNX18.N, fKNX2.negativePin1);
connect(aLKNX1.positivePin1, fKNX2.positivePin1);
end SPKNX;
```

DOMOSYS.SBKNX.vivienda.TPKNX

Implementación del sistema KNX/EIB del tercer piso.



Parameters

Type	Name	Default	Description
Integer	componentes	22	

Modelica definition

```

model TPKNX "Implementación del sistema KNX/EIB del tercer piso."
  parameter Integer componentes=22;
  Integer AK1;
  Real NAL;
  Modelica.SIunits.Power gasto;

  Modelica.Electrical.Analog.Sources.SineVoltage
  sineVoltage(V=325, freqHz=50);
  Modelica.Electrical.Analog.Basic.Ground ground;
  FKNX fKNX;

```

Librería DOMOSYS

```
FKNX fKNX1;
ALKNX aLKNX;
Modelica.Electrical.Analog.Basic.Ground ground1;
ABKNX aBKNX;
ABKNX aBKNX1;
ABKNX aBKNX2;
ABKNX aBKNX3;
ABKNX aBKNX4;
ABKNX aBKNX5;
ABKNX aBKNX6;
ABKNX aBKNX7;
ABKNX aBKNX8;
ABKNX aBKNX9;
Modelica.Electrical.Analog.Basic.Ground ground2;
FKNX fKNX2;
ALKNX aLKNX1;
Modelica.Electrical.Analog.Basic.Ground ground3;
ABKNX aBKNX10;
ABKNX aBKNX11;
ABKNX aBKNX12;
ABKNX aBKNX13;
ABKNX aBKNX14;
ABKNX aBKNX15;
ABKNX aBKNX16;
ABKNX aBKNX17;
ABKNX aBKNX18;
ABKNX aBKNX19;
FKNX fKNX3;
ALKNX aLKNX2;
Modelica.Electrical.Analog.Basic.Ground ground4;
ABKNX aBKNX22;
ABKNX aBKNX23;

equation
NAL=componentes/10;
AK1=integer(NAL);
gasto=(9.5*(AK1+1))+(0.150*componentes)+(0.250*AK1);
connect(sineVoltage.n, ground.p);
connect(sineVoltage.p, fKNX.positivePin);
connect(sineVoltage.n, fKNX.negativePin);
connect(fKNX.positivePin1, aLKNX.positivePin);
connect(fKNX.negativePin1, aLKNX.negativePin);
```

```

connect(ground1.p, fKNX.negativePin1);
connect(sineVoltage.p, fKNX1.positivePin);
connect(sineVoltage.n, fKNX1.negativePin);
connect(aLKNX.positivePin1, fKNX1.positivePin1);
connect(aLKNX.negativePin1, fKNX1.negativePin1);
connect(ground2.p, fKNX1.negativePin1);
connect(fKNX1.positivePin1, aBKNX.P);
connect(fKNX1.positivePin1, aBKNX1.P);
connect(fKNX1.positivePin1, aBKNX2.P);
connect(aBKNX3.P, fKNX1.positivePin1);
connect(fKNX1.positivePin1, aBKNX4.P);
connect(fKNX1.positivePin1, aBKNX5.P);
connect(fKNX1.positivePin1, aBKNX6.P);
connect(fKNX1.positivePin1, aBKNX7.P);
connect(fKNX1.positivePin1, aBKNX8.P);
connect(fKNX1.positivePin1, aBKNX9.P);
connect(fKNX1.negativePin1, aBKNX.N);
connect(fKNX1.negativePin1, aBKNX1.N);
connect(fKNX1.negativePin1, aBKNX2.N);
connect(fKNX1.negativePin1, aBKNX3.N);
connect(fKNX1.negativePin1, aBKNX4.N);
connect(fKNX1.negativePin1, aBKNX5.N);
connect(fKNX1.negativePin1, aBKNX6.N);
connect(fKNX1.negativePin1, aBKNX7.N);
connect(fKNX1.negativePin1, aBKNX8.N);
connect(fKNX1.negativePin1, aBKNX9.N);
connect(aLKNX1.positivePin1, fKNX2.positivePin1);
connect(aLKNX1.negativePin1, fKNX2.negativePin1);
connect(fKNX.negativePin1, aLKNX1.negativePin);
connect(fKNX.positivePin1, aLKNX1.positivePin);
connect(sineVoltage.p, fKNX2.positivePin);
connect(sineVoltage.n, fKNX2.negativePin);
connect(fKNX2.positivePin1, aBKNX10.P);
connect(fKNX2.positivePin1, aBKNX11.P);
connect(fKNX2.positivePin1, aBKNX12.P);
connect(fKNX2.positivePin1, aBKNX13.P);
connect(fKNX2.positivePin1, aBKNX15.P);
connect(fKNX2.positivePin1, aBKNX14.P);
connect(fKNX2.positivePin1, aBKNX16.P);
connect(fKNX2.positivePin1, aBKNX17.P);
connect(fKNX2.positivePin1, aBKNX18.P);

```

Librería DOMOSYS

```
connect(fKNX2.positivePin1, aBKNX19.P);
connect(fKNX2.negativePin1, aBKNX10.N);
connect(fKNX2.negativePin1, aBKNX11.N);
connect(fKNX2.negativePin1, aBKNX12.N);
connect(fKNX2.negativePin1, aBKNX13.N);
connect(aBKNX14.N, fKNX2.negativePin1);
connect(fKNX2.negativePin1, aBKNX15.N);
connect(fKNX2.negativePin1, aBKNX16.N);
connect(fKNX2.negativePin1, aBKNX17.N);
connect(fKNX2.negativePin1, aBKNX18.N);
connect(fKNX2.negativePin1, aBKNX19.N);
connect(aLKNX2.negativePin1, fKNX3.negativePin1);
connect(aLKNX2.positivePin1, fKNX3.positivePin1);
connect(fKNX3.negativePin, ground.p);
connect(fKNX3.positivePin, sineVoltage.p);
connect(ground4.p, fKNX3.negativePin1);
connect(fKNX.positivePin1, aLKNX2.positivePin);
connect(aLKNX2.negativePin, fKNX.negativePin1);
connect(fKNX3.positivePin1, aBKNX22.P);
connect(fKNX3.positivePin1, aBKNX23.P);
connect(fKNX3.negativePin1, aBKNX23.N);
connect(fKNX3.negativePin1, aBKNX22.N);
connect(ground3.p, fKNX2.negativePin1);
end TPKNX;
```

DOMOSYS.SBKNX.vivienda.gt

La sumatoria del gasto energético de cada uno de los niveles de la vivienda ejemplo.

Modelica definition

```
model gt
  "La sumatoria del gasto energético de cada uno de los niveles
  de la vivienda ejemplo."
  Modelica.SIunits.Power gastototal;
  KNXsotano kNXsotano;
  PPKNX pPKNX;
  SPKNX sPKNX;
```

```

TPKNX tPKNX;
equation
  gastototal=kNXsotano.gasto+pPKNX.gasto+sPKNX.gasto+tPKNX.gasto;
end gt;

```

HTML-documentation generated by [Dymola](#) Mon Aug 31 23:16:43 2015.

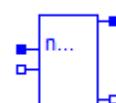
DOMOSYS.Wago

Paquete con el Sistema de la Empresa Wago que es un Controlador Lógico Programable PLC

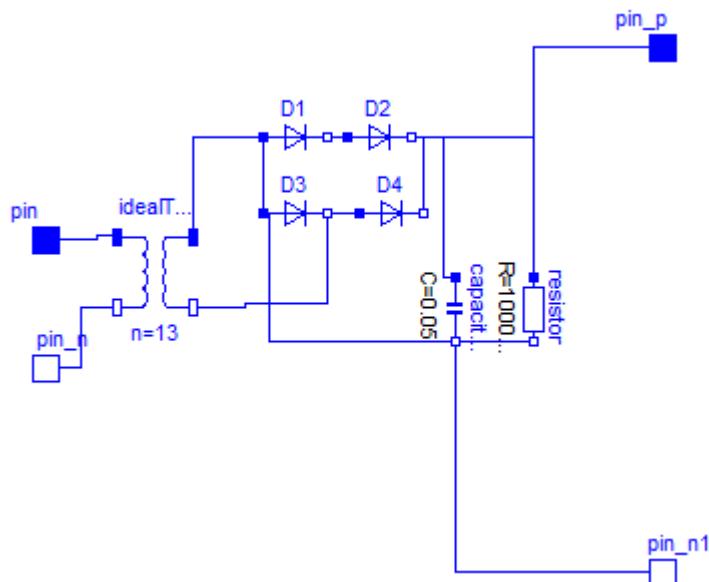
Package Content

Name	Description
 netzpower	Esta formada por un corrector de onda completa para que cuando entre 230 voltios con 50HZ salgan 26 voltios DC. Para alimentar el controlador de bus.
 cb	La función principal de este componente en Dymola es abastecer el mismo controlador con una determinada corriente para que el sistema interno funcione, también entrega una corriente determinada al sistema interno de los módulos de buses.
mb	Es la parte básica de todos los sensores y actuadores
 mf	Es necesario en el sistema para saber que el sistema esta cerrado.
viviendawago	Implementación del sistema el Wago en la vivienda ejemplo.

[DOMOSYS.Wago.netzpower](#)



Esta formada por un corrector de onda completa para que cuando entre 230 voltios con 50HZ salgan 26 voltios DC. Para alimentar el controlador de bus.



Parameters

Type	Name	Default	Description
Power	gasto	4	[W]

Connectors

Type	Name	Description
Pin	pin	
NegativePin	pin_n	
PositivePin	pin_p	
NegativePin	pin_n1	

Modelica definition

```
model netzpower
    "Esta formada por un corrector de onda completa para que cuando entre 230 voltios con 50HZ salgan 26 voltios DC. Para alimentar el controlador de bus."
```

```

parameter Modelica.SIunits.Power gasto=4; //El gasto de potencia
Modelica.Electrical.Analog.Interfaces.Pin pin;

Modelica.Electrical.Analog.Ideal.IdealDiode D1;
Modelica.Electrical.Analog.Ideal.IdealDiode D2;
Modelica.Electrical.Analog.Ideal.IdealDiode D3;
Modelica.Electrical.Analog.Ideal.IdealDiode D4;
Modelica.Electrical.Analog.Basic.Resistor resistor(R=1000000);
Modelica.Electrical.Analog.Ideal.IdealTransformer idealTransformer(n=13);

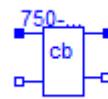
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n;
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n1;
Modelica.Electrical.Analog.Basic.Capacitor capacitor1(C=0.05);

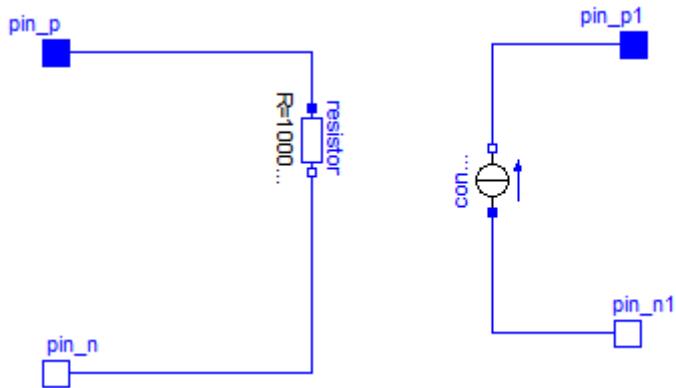
equation
  connect(pin, pin);
  connect(D1.n, D2.p);
  connect(D3.n, D4.p);
  connect(D4.n, D2.n);
  connect(D2.n, resistor.p);
  connect(idealTransformer.p1, pin);
  connect(idealTransformer.p2, D1.p);
  connect(idealTransformer.n2, D3.n);
  connect(D1.p, D3.p);
  connect(pin_p, pin_p);
  connect(pin_n, idealTransformer.n1);
  connect(capacitor1.p, D2.n);
  connect(D2.n, pin_p);
  connect(capacitor1.n, resistor.n);
  connect(D3.p, capacitor1.n);
  connect(capacitor1.n, pin_n1);
end netzpower;

```

DOMOSYS.Wago.cb

La función principal de este componente en Dymola es abastecer el mismo controlador con una determinada corriente para que el sistema interno funcione, también entrega una corriente determinada al sistema interno de los módulos de buses.





Parameters

Type	Name	Default	Description
<u>Current</u>	Isis	0.380	[A]
<u>Current</u>	Ientrega	1.2	[A]
<u>Voltage</u>	vsis	5	[V]

Connectors

Type	Name	Description
<u>NegativePin</u>	pin_n	
<u>PositivePin</u>	pin_p	
<u>PositivePin</u>	pin_p1	
<u>NegativePin</u>	pin_n1	

Modelica definition

```

model cb "La función principal de este componente en Dymola es
abastecer el mismo controlador con una determinada corriente para
que el sistema interno funcione, también entrega una corriente
determinada al sistema interno de los módulos de buses."
import D = Modelica.Electrical.Digital;
import L = Modelica.Electrical.Digital.Interfaces.LogicValue;
parameter Modelica.SIunits.Current Isis=0.380; //corriente que
consume el sistema del controlador para funcionar

```

```

parameter Modelica.SIunits.Current Ientrega=1.2; //Corriente
para entregar a los modulos
parameter Modelica.SIunits.Voltage vsis=5; //Voltaje que toma
el sistema para funcionar
Modelica.SIunits.Power gasto; //Gasto de potencia de un control-
lador de bus 750-881
Boolean ON; //Es una variable que esta activa si el voltaje de
entrada es mayo de 24 voltios

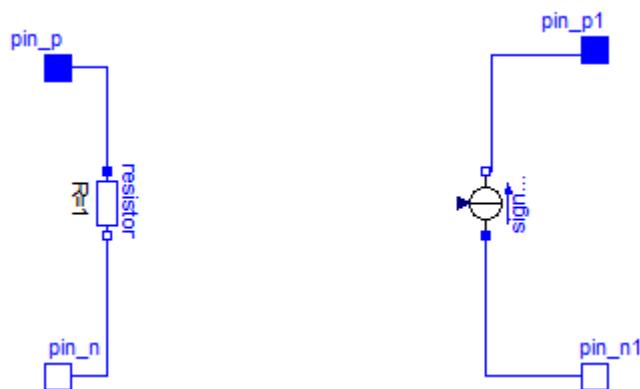
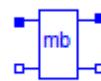
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n;
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p;
Modelica.Electrical.Analog.Basic.Resistor resistor(R=1000000);

Modelica.Electrical.Analog.Interfaces.PositivePin pin_p1;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n1;
Modelica.Electrical.Analog.Sources.ConstantCurrent constantCu-
rrent(I=1.2);
equation
gasto=Isis*vsis; // Es el gasto que tiene el sistema por funcio-
nar
ON=if resistor.p.v>24 then true else false;
connect(pin_p, resistor.p);
connect(pin_n, resistor.n);
connect(constantCurrent.n, pin_p1);
connect(constantCurrent.p, pin_n1);
end cb;

```

DOMOSYS.Wago.mb

Es la parte básica de todos los sensores y actuadores



Parameters

Type	Name	Default	Description
Voltage	vsis	5	[V]
Current	Ied	0.004	[A]
Current	Iea	0.06	[A]
Current	Isd	0.015	[A]
Current	Isa	0.07	[A]
Boolean	digital	true	
Boolean	salida	true	

Connectors

Type	Name	Description
PositivePin	pin_p	
NegativePin	pin_n	
PositivePin	pin_p1	
NegativePin	pin_n1	

Modelica definition

```

model mb "Es la parte básica de todos los sensores y actuadores"
  import D = Modelica.Electrical.Digital;
  import L = Modelica.Electrical.Digital.Interfaces.LogicValue;
  Modelica.SIunits.Power gasto;
  //parameter Integer canales=2 annotation 0;
  parameter Modelica.SIunits.Voltage vsis=5; //El voltaje de alimentación del sistema interno
  parameter Modelica.SIunits.Current Ied=0.004;
  parameter Modelica.SIunits.Current Iea=0.06;
  parameter Modelica.SIunits.Current Isd=0.015;
  parameter Modelica.SIunits.Current Isa=0.07;
  parameter Boolean digital=true;
  parameter Boolean salida=true;
  Modelica.SIunits.Current Isis;

```

```

Modelica.Electrical.Analog.Basic.Resistor resistor(R=1);
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n;
Modelica.Electrical.Analog.Interfaces.PositivePin pin_p1;
Modelica.Electrical.Analog.Interfaces.NegativePin pin_n1;
Modelica.Electrical.Analog.Sources.SignalCurrent signalCurrent;
equation

    /*con la siguiente ecuación se pretende saber que tanta co-
rriente consume
    el sistema en si, pero esa respuesta depende de si el
    sistema es análogo
    o digital y si es de entrada o salida.*/
Isis=if digital==true and salida==true then Isd else if
    digital==true and salida==false then Ied else if
    digital==false and salida==true then Isa else if
    digital==false and salida==false then Ied else 1;
    signalCurrent.i=if (resistor.p.i-Isis)>0 then (resistor.p.i-I-
    sis) else 2; //Si la corriente que entra al modulo de bus menos
    la corriente que necesita el sistema
//es mayor que cero significa que el sistema puede consumir co-
rriente de la que hay en la linea, si no entonces se debe dar co-
rriente

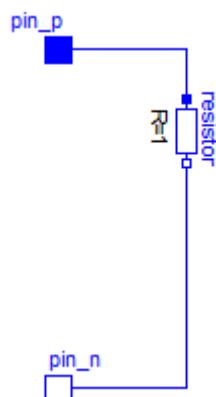
gasto=if (pin_p.i-Isis)>0 then Isis*vsis else (Isis*vsis)+4;

connect(pin_p, resistor.p);
connect(pin_n, resistor.n);
connect(signalCurrent.n, pin_p1);
connect(signalCurrent.p, pin_n1);
end mb;

```

DOMOSYS.Wago.mf

Es necesario en el sistema para saber que el sistema esta cerrado.

**Parameters**

Type	Name	Default	Description
<u>Power</u>	gasto	0.1	[W]

Connectors

Type	Name	Description
<u>PositivePin</u>	pin_p	
<u>NegativePin</u>	pin_n	

Modelica definition

```

model mf
    "Es necesario en el sistema para saber que el sistema esta cerrado."
    parameter Modelica.SIunits.Power gasto=0.1;
    Modelica.Electrical.Analog.Interfaces.PositivePin pin_p;
    Modelica.Electrical.Analog.Interfaces.NegativePin pin_n;
    Modelica.Electrical.Analog.Basic.Resistor resistor;

equation
    connect (pin_p, resistor.p);
    
```

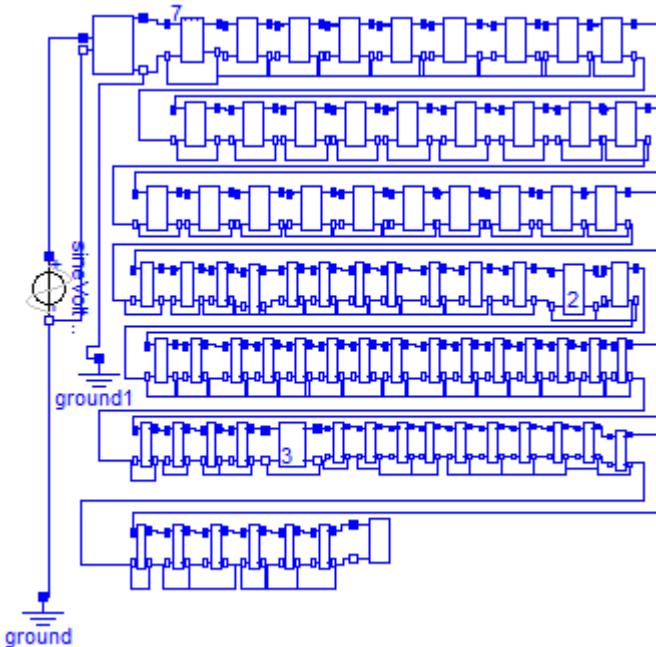
```

connect (pin_n, resistor.n);
end mf;

```

DOMOSYS.Wago.viviendawago

Implementación del sistema el Wago en la vivienda ejemplo.



Modelica definition

```

model viviendawago
  "Implementación del sistema el Wago en la vivienda ejemplo. "
  Modelica.SIunits.Power gasto;
  Modelica.Electrical.Analog.Sources.SineVoltage  sineVoltage(
    V=324, freqHz=
      50);
  Modelica.Electrical.Analog.Basic.Ground ground;
  netzpower netzpower1;
  cb cb1;
  mb mb1(salida=false);
  mb mb2(salida=false);
  mb mb3(salida=false);
  mb mb4(salida=false);
  mb mb5(salida=false);
  mb mb6(salida=false);

```

Librería DOMOSYS

```
mb mb7(salida=false);
mb mb8(salida=true);
mb mb9(salida=true);
mb mb10(salida=true);
mb mb11(salida=true);
mb mb12(digital=false);
Modelica.Electrical.Analog.Basic.Ground ground1;
mb mb13(digital=false);
mb mb14(digital=false);
mb mb15;
mb mb16(salida=false);
mb mb17(salida=false);
mb mb18(salida=false);
mb mb19;
mb mb20;
mb mb21;
mb mb22;
mb mb23;
mb mb24;
mb mb25;
mb mb26;
mb mb27;
mb mb28;
mb mb29(digital=false, salida=false);
mb mb30(digital=false, salida=false);
mb mb31(digital=false);
mb mb32(digital=false);
mb mb33(digital=false);
mb mb34(digital=false);
mb mb35(digital=false);
mb mb36(digital=false);
mb mb37(digital=false);
mb mb38(digital=false);
mb mb39;
mb mb40;
mb mb41;
mb mb42;
mb mb43;
mb mb44;
mb mb45;
mb mb46;
```

```

mb mb47;
mb mb48;
mb mb49;
mb mb50;
mb mb51;
mb mb52;
mb mb53;
mb mb54;
mb mb55;
mb mb56;
mb mb57;
mb mb58;
mb mb59;
mb mb60;
mb mb61;
mb mb62;
mb mb63;
mb mb64;
mb mb65;
mb mb66;
mb mb67;
mb mb68;
mb mb69;
mb mb70;
mb mb71;
mb mb72;
mb mb73;
mb mb74;
mb mb75;
mb mb76;
mf mf1;

equation
    gasto=netzpower1.gasto+cb1.gasto+mb1.gasto+mb2.gasto+mb3.gasto+
mb4.gasto+mb5.gasto+mb6.gasto+mb7.gasto+mb8.gasto+mb9.gasto
        +mb10.gasto+mb11.gasto+mb12.gasto+mb13.gasto+mb14.gasto+m
b15.gasto+mb16.gasto+mb17.gasto+mb18.gasto+mb19.gasto
        +mb20.gasto+mb21.gasto+mb22.gasto+mb23.gasto+mb24.gasto+m
b25.gasto+mb26.gasto+mb27.gasto+mb28.gasto+mb29.gasto
        +mb30.gasto+mb31.gasto+mb32.gasto+mb33.gasto+mb34.gasto+m
b35.gasto+mb36.gasto+mb37.gasto+mb38.gasto+mb39.gasto

```

Librería DOMOSYS

```
+mb40.gasto+mb41.gasto+mb42.gasto+mb43.gasto+mb44.gasto+m  
b45.gasto+mb46.gasto+mb47.gasto+mb48.gasto+mb49.gasto  
+mb50.gasto+mb51.gasto+mb52.gasto+mb53.gasto+mb54.gasto+m  
b55.gasto+mb56.gasto+mb57.gasto+mb58.gasto+mb59.gasto  
+mb60.gasto+mb61.gasto+mb62.gasto+mb63.gasto+mb64.gasto+m  
b65.gasto+mb66.gasto+mb67.gasto+mb68.gasto+mb69.gasto  
+mb70.gasto+mb71.gasto+mb72.gasto+mb73.gasto+mb74.gasto+m  
b75.gasto+mb76.gasto+mf1.gasto;  
connect(sineVoltage.n, ground.p);  
connect(sineVoltage.p, netzpower1.pin);  
connect(sineVoltage.n, netzpower1.pin_n);  
connect(cb1.pin_n1, mb1.pin_n);  
connect(cb1.pin_p1, mb1.pin_p);  
connect(mb1.pin_p1, mb2.pin_p);  
connect(mb1.pin_n1, mb2.pin_n);  
connect(mb2.pin_n1, mb3.pin_n);  
connect(mb2.pin_p1, mb3.pin_p);  
connect(mb3.pin_p1, mb4.pin_p);  
connect(mb3.pin_n1, mb4.pin_n);  
connect(mb4.pin_p1, mb5.pin_p);  
connect(mb4.pin_n1, mb5.pin_n);  
connect(mb5.pin_p1, mb6.pin_p);  
connect(mb5.pin_n1, mb6.pin_n);  
connect(mb6.pin_n1, mb7.pin_n);  
connect(mb6.pin_p1, mb7.pin_p);  
connect(mb7.pin_p1, mb8.pin_p);  
connect(mb7.pin_n1, mb8.pin_n);  
connect(netzpower1.pin_p, cb1.pin_p);  
connect(netzpower1.pin_n1, cb1.pin_n);  
connect(ground1.p, netzpower1.pin_n1);  
connect(cb1.pin_n, cb1.pin_n1);  
connect(cb1.pin_n1, mb1.pin_n1);  
connect(mb1.pin_n1, mb2.pin_n1);  
connect(mb2.pin_n1, mb3.pin_n1);  
connect(mb3.pin_n1, mb4.pin_n1);  
connect(mb4.pin_n1, mb5.pin_n1);  
connect(mb5.pin_n1, mb7.pin_n);  
connect(mb6.pin_n1, mb7.pin_n1);  
connect(mb7.pin_n1, mb8.pin_n1);  
connect(mb8.pin_n1, mb9.pin_n);  
connect(mb8.pin_p1, mb9.pin_p);
```

```
connect(mb9.pin_p1, mb10.pin_p);
connect(mb9.pin_n1, mb10.pin_n);
connect(mb10.pin_p1, mb11.pin_p);
connect(mb10.pin_n1, mb11.pin_n);
connect(mb11.pin_p1, mb12.pin_p);
connect(mb11.pin_n1, mb12.pin_n);
connect(mb12.pin_p1, mb13.pin_p);
connect(mb12.pin_n1, mb13.pin_n);
connect(mb13.pin_p1, mb14.pin_p);
connect(mb13.pin_n1, mb14.pin_n);
connect(mb14.pin_p1, mb15.pin_p);
connect(mb14.pin_n1, mb15.pin_n);
connect(mb9.pin_n, mb9.pin_n1);
connect(mb9.pin_n1, mb10.pin_n1);
connect(mb11.pin_n, mb11.pin_n1);
connect(mb12.pin_n, mb12.pin_n1);
connect(mb13.pin_n, mb13.pin_n1);
connect(mb14.pin_n, mb14.pin_n1);
connect(mb14.pin_n1, mb15.pin_n1);
connect(mb15.pin_p1, mb16.pin_p);
connect(mb15.pin_n1, mb16.pin_n);
connect(mb16.pin_n1, mb17.pin_n);
connect(mb16.pin_p1, mb17.pin_p);
connect(mb17.pin_n1, mb18.pin_n);
connect(mb17.pin_p1, mb18.pin_p);
connect(mb16.pin_n, mb16.pin_n1);
connect(mb16.pin_n1, mb17.pin_n1);
connect(mb18.pin_p1, mb19.pin_p);
connect(mb18.pin_n1, mb19.pin_n);
connect(mb19.pin_p1, mb20.pin_p);
connect(mb19.pin_n1, mb20.pin_n);
connect(mb20.pin_n1, mb21.pin_n);
connect(mb20.pin_p1, mb21.pin_p);
connect(mb21.pin_p1, mb22.pin_p);
connect(mb21.pin_n1, mb22.pin_n);
connect(mb22.pin_n1, mb23.pin_n);
connect(mb22.pin_p1, mb23.pin_p);
connect(mb23.pin_n1, mb24.pin_n);
connect(mb23.pin_p1, mb24.pin_p);
connect(mb24.pin_p1, mb25.pin_p);
connect(mb24.pin_n1, mb25.pin_n);
```

Librería DOMOSYS

```
connect(mb25.pin_p1, mb26.pin_p);
connect(mb25.pin_n1, mb26.pin_n);
connect(mb26.pin_p1, mb27.pin_p);
connect(mb26.pin_n1, mb27.pin_n);
connect(mb18.pin_n, mb18.pin_n1);
connect(mb18.pin_n1, mb19.pin_n1);
connect(mb20.pin_n, mb20.pin_n1);
connect(mb20.pin_n1, mb21.pin_n1);
connect(mb21.pin_n1, mb22.pin_n1);
connect(mb22.pin_n1, mb23.pin_n1);
connect(mb23.pin_n1, mb24.pin_n1);
connect(mb25.pin_n, mb25.pin_n1);
connect(mb25.pin_n1, mb26.pin_n1);
connect(mb26.pin_n1, mb27.pin_n1);
connect(mb27.pin_n1, mb28.pin_n);
connect(mb27.pin_p1, mb28.pin_p);
connect(mb28.pin_p1, mb29.pin_p);
connect(mb28.pin_n1, mb29.pin_n);
connect(mb28.pin_n, mb28.pin_n1);
connect(mb28.pin_n1, mb29.pin_n1);
connect(mb29.pin_p1, mb30.pin_p);
connect(mb29.pin_n1, mb30.pin_n);
connect(mb29.pin_n1, mb30.pin_n1);
connect(mb30.pin_p1, mb31.pin_p);
connect(mb30.pin_n1, mb31.pin_n);
connect(mb31.pin_n1, mb32.pin_n);
connect(mb31.pin_p1, mb32.pin_p);
connect(mb32.pin_p1, mb33.pin_p);
connect(mb32.pin_n1, mb33.pin_n);
connect(mb33.pin_p1, mb34.pin_p);
connect(mb33.pin_n1, mb34.pin_n);
connect(mb34.pin_p1, mb35.pin_p);
connect(mb34.pin_n1, mb35.pin_n);
connect(mb35.pin_p1, mb36.pin_p);
connect(mb35.pin_n1, mb36.pin_n);
connect(mb36.pin_p1, mb37.pin_p);
connect(mb36.pin_n1, mb37.pin_n);
connect(mb37.pin_p1, mb38.pin_p);
connect(mb37.pin_n1, mb38.pin_n);
connect(mb30.pin_n1, mb31.pin_n1);
connect(mb31.pin_n1, mb32.pin_n1);
```

```
connect(mb32.pin_n1, mb33.pin_n1);
connect(mb33.pin_n1, mb34.pin_n1);
connect(mb34.pin_n1, mb35.pin_n1);
connect(mb35.pin_n1, mb36.pin_n1);
connect(mb36.pin_n1, mb37.pin_n1);
connect(mb37.pin_n1, mb38.pin_n1);
connect(mb38.pin_p1, mb39.pin_p);
connect(mb38.pin_n1, mb39.pin_n);
connect(mb39.pin_n1, mb40.pin_n);
connect(mb39.pin_p1, mb40.pin_p);
connect(mb39.pin_n, mb39.pin_n1);
connect(mb40.pin_n1, mb41.pin_n);
connect(mb40.pin_p1, mb41.pin_p);
connect(mb39.pin_n1, mb40.pin_n1);
connect(mb41.pin_p1, mb42.pin_p);
connect(mb41.pin_n1, mb42.pin_n);
connect(mb42.pin_n1, mb43.pin_n);
connect(mb42.pin_p1, mb43.pin_p);
connect(mb43.pin_n1, mb44.pin_n);
connect(mb43.pin_p1, mb44.pin_p);
connect(mb44.pin_n1, mb45.pin_n);
connect(mb44.pin_p1, mb45.pin_p);
connect(mb45.pin_p1, mb46.pin_p);
connect(mb45.pin_n1, mb46.pin_n);
connect(mb46.pin_p1, mb47.pin_p);
connect(mb46.pin_n1, mb47.pin_n);
connect(mb47.pin_p1, mb48.pin_p);
connect(mb47.pin_n1, mb48.pin_n);
connect(mb48.pin_p1, mb49.pin_p);
connect(mb48.pin_n1, mb49.pin_n);
connect(mb49.pin_n1, mb50.pin_n);
connect(mb49.pin_p1, mb50.pin_p);
connect(mb50.pin_n1, mb51.pin_n);
connect(mb50.pin_p1, mb51.pin_p);
connect(mb51.pin_n1, mb52.pin_n);
connect(mb51.pin_p1, mb52.pin_p);
connect(mb52.pin_p1, mb53.pin_p);
connect(mb52.pin_n1, mb53.pin_n);
connect(mb53.pin_p1, mb54.pin_p);
connect(mb53.pin_n1, mb54.pin_n);
connect(mb54.pin_p1, mb55.pin_p);
```

Librería DOMOSYS

```
connect(mb54.pin_n1, mb55.pin_n);
connect(mb41.pin_n, mb41.pin_n1);
connect(mb41.pin_n1, mb42.pin_n1);
connect(mb42.pin_n1, mb43.pin_n1);
connect(mb43.pin_n1, mb44.pin_n1);
connect(mb44.pin_n1, mb45.pin_n1);
connect(mb45.pin_n1, mb46.pin_n1);
connect(mb46.pin_n1, mb48.pin_n);
connect(mb48.pin_n, mb48.pin_n1);
connect(mb48.pin_n1, mb49.pin_n1);
connect(mb49.pin_n1, mb50.pin_n1);
connect(mb50.pin_n1, mb51.pin_n1);
connect(mb51.pin_n1, mb52.pin_n1);
connect(mb52.pin_n1, mb53.pin_n1);
connect(mb54.pin_n, mb54.pin_n1);
connect(mb54.pin_n1, mb55.pin_n1);
connect(mb55.pin_n1, mb56.pin_n);
connect(mb55.pin_p1, mb56.pin_p);
connect(mb56.pin_p1, mb57.pin_p);
connect(mb56.pin_n1, mb57.pin_n);
connect(mb56.pin_n, mb56.pin_n1);
connect(mb57.pin_p1, mb58.pin_p);
connect(mb57.pin_n1, mb58.pin_n);
connect(mb58.pin_p1, mb59.pin_p);
connect(mb58.pin_n1, mb59.pin_n);
connect(mb56.pin_n1, mb57.pin_n1);
connect(mb57.pin_n1, mb58.pin_n1);
connect(mb58.pin_n1, mb59.pin_n1);
connect(mb59.pin_n1, mb60.pin_n);
connect(mb59.pin_p1, mb60.pin_p);
connect(mb60.pin_p1, mb61.pin_p);
connect(mb60.pin_n1, mb61.pin_n);
connect(mb61.pin_p1, mb62.pin_p);
connect(mb61.pin_n1, mb62.pin_n);
connect(mb62.pin_p1, mb63.pin_p);
connect(mb62.pin_n1, mb63.pin_n);
connect(mb63.pin_p1, mb64.pin_p);
connect(mb63.pin_n1, mb64.pin_n);
connect(mb64.pin_p1, mb65.pin_p);
connect(mb64.pin_n1, mb65.pin_n);
connect(mb65.pin_p1, mb66.pin_p);
```

```
connect(mb65.pin_n1, mb66.pin_n);
connect(mb66.pin_p1, mb67.pin_p);
connect(mb66.pin_n1, mb67.pin_n);
connect(mb67.pin_p1, mb68.pin_p);
connect(mb67.pin_n1, mb68.pin_n);
connect(mb68.pin_p1, mb69.pin_p);
connect(mb68.pin_n1, mb69.pin_n);
connect(mb69.pin_p1, mb70.pin_p);
connect(mb69.pin_n1, mb70.pin_n);
connect(mb60.pin_n, mb60.pin_n1);
connect(mb60.pin_n1, mb61.pin_n1);
connect(mb62.pin_n, mb62.pin_n1);
connect(mb62.pin_n1, mb63.pin_n1);
connect(mb63.pin_n1, mb64.pin_n1);
connect(mb65.pin_n, mb65.pin_n1);
connect(mb65.pin_n1, mb66.pin_n1);
connect(mb66.pin_n1, mb67.pin_n1);
connect(mb67.pin_n1, mb68.pin_n1);
connect(mb68.pin_n1, mb69.pin_n1);
connect(mb69.pin_n1, mb70.pin_n1);
connect(mb70.pin_n1, mb71.pin_n);
connect(mb70.pin_p1, mb71.pin_p);
connect(mb71.pin_p1, mb72.pin_p);
connect(mb71.pin_n1, mb72.pin_n);
connect(mb72.pin_p1, mb73.pin_p);
connect(mb72.pin_n1, mb73.pin_n);
connect(mb73.pin_p1, mb74.pin_p);
connect(mb73.pin_n1, mb74.pin_n);
connect(mb74.pin_p1, mb75.pin_p);
connect(mb74.pin_n1, mb75.pin_n);
connect(mb75.pin_p1, mb76.pin_p);
connect(mb75.pin_n1, mb76.pin_n);
connect(mb76.pin_p1, mf1.pin_p);
connect(mb76.pin_n1, mf1.pin_n);
connect(mb71.pin_n, mb71.pin_n1);
connect(mb71.pin_n1, mb72.pin_n1);
connect(mb72.pin_n1, mb73.pin_n1);
connect(mb74.pin_n, mb74.pin_n1);
connect(mb74.pin_n1, mb75.pin_n1);
connect(mb75.pin_n1, mb76.pin_n1);
end viviendawago;
```

HTML-documentation generated by [Dymola](#) Mon Aug 31 23:28:57 2015.