Home and Building Automation Systems

An Overview
Summary

- Definitions
- Technology Overview
- Technology Focus
  - MyHome / OpenWebNet
  - KNX
  - Modbus
- Modeling / Designing for Dog
Definitions

What are we talking about?
Home Automation System (HAS)

Home automation is the use of one or more computers to control basic home functions and sometimes features automatically and remotely. An automated home is sometimes called a smart home.

Home automation can include the scheduling and heating automatic operation of water sprinkling, security and air conditioning, window coverings, lighting, and food preparation appliances. Home automation may also allow vital home functions to be controlled remotely from anywhere in the world using a computer connected to the Internet. Besides the functions already mentioned, remote control can be extended to telephones, answering machines, fax machines, amateur radio, robots such as automatic vacuum cleaners. Domotics, Domotica, Smart Home Technology, the Smart House, the Electronic Home or Home Automation (HA) is an integrated automation system that is specific to the requirements of a private residence. It applies automatic techniques (mechanical or electronic) for the comfort, security, entertainment, communications, and information processing needs of its residents. In simplest terms it is control and monitoring of devices and information in your home.

http://whatis.techtarget.com

(DOMUs infOrmaTICS) Information technology in the home (domus is Latin for home). Although remote lighting and appliance control have been used for years (see X10), domotics is another term for the digital home, including the networks and devices that add comfort and convenience as well as security. Controlling heating, air conditioning, food preparation, TVs, stereos, lights, appliances, entrance gates and security systems all fall under the domotics umbrella.

thefreedictionary.com
Home Automation System (HAS)

- **Home automation**
  - Automation of the home, housework or household activity.
  - (remote) control of:
    - lighting
    - HVAC (Heating, Ventilation and Air Conditioning)
    - appliances
    - and other systems

- **Home automation system**
  - An integrated system (computer-based) offering home automation functionalities
  - Integrates electrical devices in a house
    - Through a communication network
    - Possibly includes devices using different communication protocols
Building Automation System (BAS)

The commonly accepted definition of a Building Automation System (BAS) includes the comprehensive and co-coordinated control of one or more major system functions required in a facility, such as heating, ventilating and air conditioning (HVAC) systems, Fire and Life Safety, and Vertical Transportation. As the technology moves forward, there are lot more functionalities that could be forged with the conventional building automation system.

a fully-integrated control system in which building services are monitored and controlled by a computer-based management system.

thefreedictionary.com
Building Automation System (BAS)

- Building automation
  - The advanced functionality provided by the control system of a building
    - E.g., security & access control, fire detection & alarms, HVAC, lighting control, air quality, smoke detection, intrusion detection, environmental control, asset location/management

- Building Automation System (BAS)
  - A computerized, intelligent network of electronic devices designed to monitor and control the mechanical, electronic, and lighting systems in a building
Home vs. Building Automation

- Home Automation is almost a subset of Building Automation
  - Most functionalities are shared
  - Different protocols and technologies
Technology Overview

Which devices? Where? For what?
Technology Overview

Building Automation
- RS-485
- DALI
- CAN
- M-Bus
- Modbus
- Echelon
- LonWorks
- KNX

Home Automation
- Bluetooth
- UPnP
- ZigBee Alliance
- INSTEON
- WiFi
- Dlna

Dario Bonino - Home and Building Automation Systems
Network Technology

- **Bus**
  - MyOpen
  - KNX
  - Modbus
  - Echelon
  - Dali
  - CAN
  - MBus

- **Powerline**
  - Echelon
  - X10
  - Insteon

- **Wireless**
  - ZigBee
  - Z-Wave
  - EnOcean
Application Area

- Automation
  - Real-time Control
    - CAN
  - KNX, MyHome, Insteon, Echelon, Modbus, X10, ZigBee, Z-Wave, EnOcean
- Lighting
  - Dali
  - (all Automation)
- Metering
  - Mbus
  - RS-485
  - KNX, Modbus, Echelon, ZigBee, Z-Wave,…
- Entertainment
  - UPnP
  - DLNA
- General purpose
  - Bluetooth
  - WiFi
Challenges

- Integration
  - Different electrical requirements
  - Different interaction modalities
  - Different behaviors (temporal, etc.)

- Interoperation
  - Different protocols
  - Different interaction modalities
    - Master/slave
    - Peer-to-peer
    - Etc.

- Modeling
  - Different technologies & assumptions
  - A single shared, common description
Technology Focus

Assumptions, Networks, Idiosyncrasies
MyOpen / OpenWebNet

- MyOpen system
  - Initially proposed by Bticino (Legrand group)
  - Proprietary bus (SCS)
  - Proprietary low-level protocol
  - Simple configuration (can be carried by electricians)
  - Accessible via OpenWebNet gateways
OpenWebNet

- Allows external applications to communicate, monitor and control MyHome devices
- Open Specification
- Open protocol designed to work on minimal network requirements
  - E.g., phone connections
DTMF compatible

E.g., *1*1*12## → Light 12, On

Defines 2 types of communication sessions

Command session
- To send commands
- To ask for device states
- To require measure values

Event session
- To monitor all the bus events, asynchronously
OpenWebNet - Messages

- Tag structure
  - \(*tag1*tag2*tag3*...*tagN##

- Tag
  - Allowed characters
    - \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #\}
  - Delimitator
    - \*
  - Message end
    - ##

- Content and structure changes for
  - Commands / State requests
  - Requests of measure values
OpenWebNet – Commands/States

- **3-tag structure**
  - *WHO* *WHAT* *WHERE* ##
- **WHO**
  - The command/request functionality (among a set of pre-defined values)
- **WHAT**
  - The action to perform
  - Possible actions are specified for each WHO value
- **WHERE**
  - Identifies the message destination
    - Single device, device groups, scenarios, zones, etc.
  - WHAT and WHERE can have additional parameters
    - WHAT#PAR1#PAR2…#PARn
    - WHERE#PAR1#PAR2…#PARn

<table>
<thead>
<tr>
<th></th>
<th>Functionality Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Scenarios</td>
</tr>
<tr>
<td>1</td>
<td>Lighting</td>
</tr>
<tr>
<td>2</td>
<td>Actuators</td>
</tr>
<tr>
<td>3</td>
<td>Load control</td>
</tr>
<tr>
<td>4</td>
<td>Temperature control</td>
</tr>
<tr>
<td>5</td>
<td>Anti-burglar systems</td>
</tr>
<tr>
<td>6</td>
<td>Intercom</td>
</tr>
<tr>
<td>7</td>
<td>Multimedia</td>
</tr>
<tr>
<td>13</td>
<td>Gateway management</td>
</tr>
<tr>
<td>15</td>
<td>CEN commands</td>
</tr>
<tr>
<td>16/22</td>
<td>Sound diffusion</td>
</tr>
<tr>
<td>17</td>
<td>Scenarios for MH200N gateways</td>
</tr>
<tr>
<td>18</td>
<td>Energy management</td>
</tr>
<tr>
<td>25</td>
<td>CEN plus/ plus scenarios/ clean contacts</td>
</tr>
<tr>
<td>1001</td>
<td>Automation diagnostics</td>
</tr>
<tr>
<td>1004</td>
<td>Thermal control diagnostics</td>
</tr>
<tr>
<td>1013</td>
<td>Device diagnostics</td>
</tr>
</tbody>
</table>
Switch on the lamp with id=12
  *1*1*12##

Switch on the webcam with id=4000
  *7*0*4000##

Switch off the temperature control for the zone 1
  *4*303*1##

Switch off all lights
  *1*0*0##
Idiosyncrasies / Peculiarities

- Explicit State Notion
  - Same format for State changes and Commands
  - State change events only in monitoring sessions
  - State change events only for “active devices”, buttons and switches do not generate events.

- States and commands only “defined” for “active” devices
  - Buttons and switches do not have an explicit state (independent from the controlled object)
Formal merger of 3 leading systems for Home and Building Automation
- EIB
- EHS
- BatiBus

All devices carry a “bit” of intelligence on-board

2 Operating modes
- S-MODE (System Mode)
  - Requires centralized binding & parameterization (with ETS)
- E-MODE
  - Simple manipulation without PCs
  - Similar to MyOpen
KNX - Architecture

Common Object definitions

Common Logo

System-Mode

PC based Tool

Easy-Mode

Ctrl PB LTE

Configuration

Runtime Interworking

Common Kernel

Common Run Time

Profile 1

Profile 2

Communication

Standard Addressing

TP1 PL110 RF Ethernet

Standard Configuration/Engineering Tool

Network Management
Device implement “Distributed Applications”
- Based on *datapoints*

**Datapoints:**
- Represent process and control variables in the system
- May be
  - inputs
  - Outputs
  - parameters
  - diagnostic data
  - …

**Standardized Datapoint Types, grouped into Functional Blocks**
KNX – Application (2)

- Communication System and Protocol offer a reduced instruction set to
  - Read and Write (set and get) Datapoint values

- Application semantics is mapped to
  - Data format
  - Bindings
    - 3 binding schemes
      - Free
      - Structured
      - Tagged binding
Free binding:
- No a priori prescription on which Datapoints may be linked
- Free addressing
- Customized multicast grouping at the level of individual Datapoints
- Central to S-Mode

Structured binding
- Precise pattern for linking a whole set of Datapoints, usually corresponding to a Functional Block or Channel
- Free-address
  - E.g., Controller and Push-button Modes

Tagged binding
- … too complex …
GroupAddress (multicast access to a datapoint)
KNX – Network technology

- TP 1 (basic medium inherited from EIB)
  - Twisted pair cabling
  - SELV network and supply system
  - Asynchronous character oriented data transfer and half duplex bi-directional communication
  - Transmission rate: 9600 bit/s
  - CSMA/CA collision avoidance
  - All topologies may be used and mixed (line, star, tree, ….)

- PL 110
  - Communication over the mains supply network
  - Spread frequency shift keying signaling
  - Asynchronous transmission of data packets and half duplex bi-directional communication
  - Central frequency 110 kHZ
  - Transmission rate: 1200 bit/s
  - CSMA, compliant to EN 50065-1
KNX – Network technology

- **RF**
  - 868.3 MHz band for Short Range
  - Frequency Shift Keying, maximum duty cycle of 1%
  - 32768 cps (chips per second)
  - Manchester data encoding

- **KNXnet/IP**
  - Standard protocol for KNX devices connected to an IP network
  - IP network as a fast backbone in KNX installations
  - Tunnels KNX Frames over IP
KNX – Network topology

- **Line**
  - Up to 256 devices
  - Connected into Areas via a Main Line

- **Area**
  - Up to 16 lines per area
  - Up to 16 Areas
  - Connected via a Backbone Line

- **Max. Number of devices**
  - 65536
Idiosyncrasies / Peculiarities

- **Implicit State Notion**
  - Only a subset of device states can be queried

- **State Events**
  - State change events can be published by carefully configuring the KNX devices
    - S-MODE
    - Specific group address for notifications
    - Pass-through gateway configuration
  - All devices can publish state changes
  - In-operation events available

- **KNXNet/IP**
  - Tunnel mode → event-based
  - Supports partial device discovery

- **EIBNet/IP**
  - Requires participation to a multicast delivery group:
    - 239.192.39.238
Modbus

- Application layer messaging protocol (level 7 of the OSI model)
- Provides client/server communication between devices
  - different types of buses or networks
- Industry serial de facto standard since 1979
- Request/reply protocol
  - Services specified by function codes
- MODBUS function codes
  - elements of MODBUS request/reply PDUs
Modbus - Protocol

- Defines a simple protocol data unit (PDU)
  - Independent of the underlying communication layers
  - Mapping of MODBUS protocol on specific buses can introduce some additional fields on the application data unit (ADU)
Function Code

- Indicates to the server what kind of action to perform
- Is coded in one byte
- Valid codes are in the range of 1 ... 255 decimal
  - the range 128 – 255 is reserved and used for exception responses
- Function code "0" is not valid
- Sub-function codes are added to some function codes to define multiple actions
# Modbus - Function Codes

<table>
<thead>
<tr>
<th>Bit access</th>
<th>Data access</th>
<th>File record access</th>
<th>Diagnostics</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Discrete Inputs</td>
<td>Read Discrete Inputs</td>
<td>02</td>
<td>02</td>
<td>6.2</td>
</tr>
<tr>
<td>Internal Bits Or Physical coils</td>
<td>Read Coils</td>
<td>01</td>
<td>01</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>Write Single Coil</td>
<td>05</td>
<td>05</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>Write Multiple Coils</td>
<td>15</td>
<td>0F</td>
<td>6.11</td>
</tr>
<tr>
<td>Physical Input Registers</td>
<td>Read Input Register</td>
<td>04</td>
<td>04</td>
<td>6.4</td>
</tr>
<tr>
<td>Internal Registers Or Physical Output Registers</td>
<td>Read Holding Registers</td>
<td>03</td>
<td>03</td>
<td>6.3</td>
</tr>
<tr>
<td></td>
<td>Write Single Register</td>
<td>06</td>
<td>06</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>Write Multiple Registers</td>
<td>16</td>
<td>10</td>
<td>6.12</td>
</tr>
<tr>
<td></td>
<td>Read/Write Multiple Registers</td>
<td>23</td>
<td>17</td>
<td>6.17</td>
</tr>
<tr>
<td></td>
<td>Mask Write Register</td>
<td>22</td>
<td>16</td>
<td>6.16</td>
</tr>
<tr>
<td></td>
<td>Read FIFO queue</td>
<td>24</td>
<td>18</td>
<td>6.18</td>
</tr>
<tr>
<td>Read File record</td>
<td>20</td>
<td>14</td>
<td>6.14</td>
<td></td>
</tr>
<tr>
<td>Write File record</td>
<td>21</td>
<td>15</td>
<td>6.15</td>
<td></td>
</tr>
<tr>
<td>Read Exception status</td>
<td>07</td>
<td>07</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>Diagnostic</td>
<td>08</td>
<td>00-18,20</td>
<td>08</td>
<td>6.8</td>
</tr>
<tr>
<td>Get Com Event counter</td>
<td>11</td>
<td>OB</td>
<td>6.9</td>
<td></td>
</tr>
<tr>
<td>Get Com Event Log</td>
<td>12</td>
<td>0C</td>
<td>6.10</td>
<td></td>
</tr>
<tr>
<td>Report Slave ID</td>
<td>17</td>
<td>11</td>
<td>6.13</td>
<td></td>
</tr>
<tr>
<td>Read device Identification</td>
<td>43</td>
<td>14</td>
<td>2B</td>
<td>6.21</td>
</tr>
<tr>
<td>Encapsulated Interface</td>
<td>43</td>
<td>13,14</td>
<td>2B</td>
<td>6.19</td>
</tr>
<tr>
<td>Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANopen General Reference</td>
<td>43</td>
<td>13</td>
<td>2B</td>
<td>6.20</td>
</tr>
</tbody>
</table>
Modbus – Protocol (3)

Successful Transaction

Client

- Initiate request

Server

- Perform the action
- Initiate the response

Function code | Data Request

Receive the response

Function code | Data Response

Failing Transaction

Client

- Initiate request

Server

- Error detected in the action
- Initiate an error

Function code | Data Request

Receive the response

Exception Function code | Exception code
Modbus – Data Model

- Based on 4 primary tables
  - Distinctions between inputs and outputs, and between bit-addressable and word-addressable data items, do not imply any application behavior
  - All four tables can overlay one another
  - Each table supports up to 65536 data items
  - Read or write of items can span multiple consecutive data blocks up to a data size limit which is dependent on the transaction function code
Modbus – Data Model

<table>
<thead>
<tr>
<th>Primary tables</th>
<th>Object type</th>
<th>Type of</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discretes Input</td>
<td>Single bit</td>
<td>Read-Only</td>
<td>This type of data can be provided by an I/O system.</td>
</tr>
<tr>
<td>Coils</td>
<td>Single bit</td>
<td>Read-Write</td>
<td>This type of data can be alterable by an application program.</td>
</tr>
<tr>
<td>Input Registers</td>
<td>16-bit word</td>
<td>Read-Only</td>
<td>This type of data can be provided by an I/O system.</td>
</tr>
<tr>
<td>Holding Registers</td>
<td>16-bit word</td>
<td>Read-Write</td>
<td>This type of data can be alterable by an application program.</td>
</tr>
</tbody>
</table>

Multiple data blocks

Single data block
Modbus - Addressing

- precise PDU addressing rules
  - Each data block is addressed from 0 to 65535
  - Each element within a data block is numbered from 1 to n
- pre-mapping between the MODBUS data model and the device application is totally vendor device specific
Modbus - Transaction

- Wait for a MB indication
- Validate function code
- Validate data Address
- Validate data value
- Execute MB function
- Send Modbus Exception Response
- Send Modbus Response
Modbus – Network technology

MODBUS APPLICATION LAYER

- Modbus on TCP
- TCP
- IP

- Other
- MODBUS+ / HDLC
  - Physical layer
- Master / Slave
  - EIA/TIA-232 or EIA/TIA-485
- Ethernet II / 802.3
  - Ethernet Physical layer
Modbus – Network technology (2)

- TCP/IP over Ethernet
- Asynchronous serial transmission over a variety of media
  - wire: EIA/TIA-232-E, EIA-422, EIA/TIA-485-A
  - fiber, radio, etc.
- MODBUS PLUS
  - high speed token passing network
Idiosyncrasies / Peculiarities

- **Explicit State Notion**
  - Can only be queried (read)
- **No command notion**
  - Only write operations on registers
- **Client/Server approach**
  - Does not support event-based interaction
    - Polling
- **No explicit device notion**
  - Registers exposed through a Modbus Gateway
  - Devices identified by their slave id
- **No device discovery**
  - Available registers are defined by the vendor
Modeling/Design for Dog

How to conciliate reality with Dog requirements
Dog – Device Abstraction

- Based on DogOnt
  - http://elite.polito.it/ontologies/dogont.owl
- Devices are described in terms of network-independent
  - Functionalities
    - Commands
    - Notifications
  - States
    - State Values
- Event-based interaction
- Network specific idiosyncrasies must be wrapped and reconciled to the Dog abstract model
For each technology

- Network driver (mandatory)
  - Handles networks-specific communication
    - Connection
    - Polling vs. Event Monitoring
    - Connection loss recovery

- Gateway driver (optional)
  - Handles different gateways using the same network protocol
    - E.g., for installations with multiple sub-networks

> Device drivers
  - 1 for each supported device
  - On-line conversions between operations on the abstract model and on real devices
Reconciling to Dog

- Device representation
  - URI
  - Commands
  - States
- Event-based interaction
  - Notifications
- OpenWebNet
  - WHERE tag
  - WHAT tag
  - WHO tag
  - Monitoring session

Dario Bonino - Home and Building Automation Systems 5/30/2012
Reconciling to Dog

Dog

- Device representation
  - URI
  - Commands
  - States

- Event-based interaction
  - Notifications

KNX

- Individual Address
- Group Address
- Datapoints

- State events
  - Only if purposely configured

- Device discovery
Reconciling to Dog

- Device representation
  - URI
  - Commands
  - States

- Register
  - Read
  - Write
  - Slave ID

- Event-based interaction
  - Notifications

- Polling
And the other technologies?

- **Z-Wave**
  - Does not support event-based interaction (partially?)
    - Polling
  - Based on device discovery
    - Device creation at run-time (in Dog)
  - Same implementation constraints

- **Echelon**
  - Almost equal to Modbus
  - Connection based on web services

...
Questions?
Thanks!
License

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3.0)” License.

- You are free:
  - to Share - to copy, distribute and transmit the work
  - to Remix - to adapt the work

- Under the following conditions:
  - Attribution - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
  - Noncommercial - You may not use this work for commercial purposes.
  - Share Alike - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

- To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/