Home and Building Automation Systems

A SHORT INTRODUCTION

A brief overview on home and building automation systems, with a particular focus on technologies, protocols and plant issues.
What?

WHAT ARE THESE SYSTEMS?
DEFINITIONS

Building Automation Systems (BAS)

A computerized, intelligent network of electronic devices designed to monitor and control the mechanical, electronic, and lighting systems in a building.

Home Automation Systems (HAS)

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

WHY ARE THESE SYSTEMS NEEDED?

AREAS / DOMAINS

- COMFORT
- LIGHTING
- SECURITY
- REMOTE
- DEVICES
- ENTERTAINMENT
- NETWORKS
- FEATURES
- FUNCTIONS
- CONVENIENCE
- AUTOMATION

HEATING AND AIR CONDITIONING

10/3/2013

Home and Building Automation - Dario Bonino
How?

HOW TO GET IT?
PLANTS / ARCHITECTURES / CONTROL / INTELLIGENCE

INTELLIGENCE

NETWORKS

DEVICES
The current state of the art of (commercial) of home and building automation: available protocols, wired vs wireless, pros and cons of most diffused systems
Wired (BUS)

- Typically based on a BUS
  - Pair (twisted)
  - Multipolar (e.g. Can?)

- Cabling
  - Dedicated bus wires
  - Typically semi-rigid
    - due to shielding and single-wire conductors

- Exception: Powerline
  - Uses existing wiring as a BUS
Wireless

• Based on radio connection
  – Different carrier frequencies
    • 2.4GHz, 868MHz, 433MHz
  – Different security
  – Different modulation

• Mesh networks
  – Dynamically reconfigurable topology
  – (Almost) Every node may act as a router

• Exception: WiFi
Which One?
MOST DIFFUSED NETWORKS

WIRELESS
- ZigBee Alliance
- Green, Smart, Wireless
- enocean
- Bluetooth
- dlna
- UPnP

WIRED
- DALI
- Echelon
- LonWorks
- KNX
- M-Bus
- CAN
- Modbus

10/3/2013
Home and Building Automation - Dario Bonino
Technologies in SUNSLICE

ALREADY DEFINED / REQUIRED

• KNX (Schneider)
  – Main automation
    • Plugs
    • Shutter / Shades
    • Metering?
    • What else? (TODO)

• Modbus (Daikin)
  – HVAC
  – Metering?

• ZigBee (Energy@Home)
  – Appliances
  – Metering?
What else?

- **Goal**
  - Cover all the needs
    - HVAC
    - Lighting
    - Metering
    - Control
    - Low Power

- **Constraints**
  - Keep complexity low
  - #Technologies ≤ 3 (Already defined)
KNX

THE EUROPEAN DE-FACTO STANDARD (ALMOST)

KNX is a standardized (EN 50090, ISO/IEC 14543), OSI-based network communications protocol for intelligent buildings. KNX is the successor to, and convergence of, three previous standards: the European Home Systems Protocol (EHS), BâtiBUS, and the European Installation Bus (EIB or Instabus). The KNX standard is administered by the KNX Association.
A bit of history

EHS → BATIBUS → KNX → EIB
Basic Principles

• Based on the idea of distributed applications
• Every device carries a BIT of «intelligence» on board
• 2 Main operating modes

S-MODE (System Mode)

E-MODE
Logic Architecture

S-MODE  E-MODE

Common Object definitions
Common Logo

Configuration
Application
Link

Standard Configuration/Engineering Tool
Configuration
Runtime Interworking
Common Kernel
Media Coupler between Media

TOOL
NETWORK MANAGEMENT
Profile 1
Profile 2
Common Run Time
Communication
Standard Addressing

PC based Tool
Crl
PB
LTE

TP1
PL110
RF
Ethernet

10/3/2013
Home and Building Automation - Dario Bonino
Application

• Device implement “Distributed Applications”
  – Based on Datapoints

• Distributed Applications = Datapoint Binding

• Datapoints:
  – Represent process and control variables in the system
  – May be inputs, outputs, parameters, diagnostic data,…
  – Standardized Datapoint types
  – Grouped into Functional Blocks
## Functional blocks: Lighting

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch OnOff (SOO)</td>
<td>Info On Off (IOO)</td>
</tr>
<tr>
<td>Timed Start Stop (TSS)</td>
<td></td>
</tr>
<tr>
<td>Forced (FO)</td>
<td></td>
</tr>
<tr>
<td>Lock Device (LD)</td>
<td></td>
</tr>
<tr>
<td>Scene Number (SN)</td>
<td></td>
</tr>
<tr>
<td>Scene Control (SC)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>additional I/Os</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Delay (OND)</td>
</tr>
<tr>
<td>Off Delay (OFFD)</td>
</tr>
<tr>
<td>Timed On Duration (TOD)</td>
</tr>
<tr>
<td>Prewarning Duration (PWD)</td>
</tr>
<tr>
<td>Timed On Retrigger Function (TRF)</td>
</tr>
<tr>
<td>Manual Off Enable (MOE)</td>
</tr>
</tbody>
</table>
# Datapoints: Lighting

<table>
<thead>
<tr>
<th>Datapoint</th>
<th>Description/Remarks</th>
<th>Datapoint Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch On Off</td>
<td>Binary control of the set value.</td>
<td>1.001 DPT_Switch</td>
</tr>
<tr>
<td>Timed StartStop</td>
<td>Activation of an autonomous switch off function.</td>
<td>1.010 DPT_Start</td>
</tr>
<tr>
<td>Scene Number</td>
<td>Recall the output state related to the encoded scene number.</td>
<td>17.001 DPT_SceneNumber</td>
</tr>
<tr>
<td>Scene Control</td>
<td>Recall or learn the output state related to the encoded scene number.</td>
<td>18.001 DPT_SceneControl</td>
</tr>
<tr>
<td>Lock Device</td>
<td>Setting of a parameterized value in a lock state of the device.</td>
<td>1.003 DPT_Enable</td>
</tr>
<tr>
<td>Forced</td>
<td>Forces value dependent high priority on or off state.</td>
<td>2.001 DPT_Switch_Control</td>
</tr>
<tr>
<td><strong>Outputs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info OnOff</td>
<td>Reflects the binary state of the actuator</td>
<td>1.001 DPT_Switch</td>
</tr>
</tbody>
</table>

* Excerpt from 07_20_02 Lighting Actuators v1.4 AS – The KNX 2.0 specification
# Datapoint specification: Blinds

## 3.3.2 DPT_Control_Blinds

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Move up or down.</td>
<td>See 1.008&lt;br&gt;0 = Up&lt;br&gt;1 = Down</td>
</tr>
<tr>
<td>StepCode</td>
<td>The amount of intervals into which the range of 0% ... 100% is subdivided, or the break indication.</td>
<td>001b...111b: Step&lt;br&gt;Number of intervals = 2^(stepcode-1)&lt;br&gt;- 000b: Break</td>
</tr>
</tbody>
</table>
Application = Datapoints Binding

GroupAddress
(multicast access to a datapoint)
Link - 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
<table>
<thead>
<tr>
<th>Link - Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TP 1</strong></td>
</tr>
<tr>
<td>Twisted pair cabling</td>
</tr>
<tr>
<td>SELV network and supply system</td>
</tr>
<tr>
<td>Asynchronous character oriented data transfer and half duplex bi-directional communication</td>
</tr>
<tr>
<td>Transmission rate: <strong>9600 bit/s</strong></td>
</tr>
<tr>
<td>CSMA/CA collision avoidance</td>
</tr>
<tr>
<td>All topologies may be used and mixed (line, star, tree, ....)</td>
</tr>
</tbody>
</table>

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

| **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 |

| **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 |
Modbus is a serial communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). Simple and robust, it has since become a *de facto* standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices.
Basic Principles

- Application layer messaging protocol (level 7 of the OSI model)
  - 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
Logic Architecture

MODBUS APPLICATION LAYER

Other

MODBUS+ / HDLC
Physical layer

Master / Slave
EIA/TIA-232 or EIA/TIA-485

Ethernet II /802.3
Ethernet Physical layer

Modbus on TCP

TCP

IP
Request/reply protocol

SUCCESSFUL TRANSACTION

Client

Initiate request

Function code

Data Request

Server

Perform the action

Initiate the response

Function code

Data Response

Receive the response

FAILING TRANSACTION

Client

Initiate request

Function code

Data Request

Server

Error detected in the action

Initiate an error

Exception Function code

Exception code

Receive the response
Data Model

- 4 primary tables
  - up to 65536 data items, each
- Distinctions between
  - inputs and outputs
  - bit-addressable and word-addressable data items
- Read or write of items can span multiple consecutive data blocks

<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>

10/3/2013
Home and Building Automation - Dario Bonino
Function codes (1/2)

- Simple protocol data unit (PDU)
  - Independent of the underlying communication layers
  - Specific buses can introduce additional fields on the application data unit (ADU)
- Function Code
  - Kind of action to perform, is coded in one byte
    - Valid codes are in the range of 1 ... 255
    - the range 128 – 255 is reserved and used for exception responses
Function Codes (2/2)

<table>
<thead>
<tr>
<th>Bit access</th>
<th>Physical Discrete Inputs</th>
<th>Read Discrete Inputs</th>
<th>02</th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Internal Bits Or Physical coils</td>
<td>Read Coils</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write Single Coil</td>
<td>05</td>
<td>05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write Multiple Coils</td>
<td>15</td>
<td>0F</td>
</tr>
<tr>
<td>Data Access</td>
<td>Physical Input Registers</td>
<td>Read Input Register</td>
<td>04</td>
<td>04</td>
</tr>
<tr>
<td>16 bits access</td>
<td>Internal Registers Or Physical Output Registers</td>
<td>Read Holding Registers</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write Single Register</td>
<td>06</td>
<td>06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write Multiple Registers</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read/Write Multiple Registers</td>
<td>23</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Write Register</td>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read FIFO queue</td>
<td>24</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read File record</td>
<td>20</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write File record</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>Diagnostics</td>
<td></td>
<td>Read Exception status</td>
<td>07</td>
<td>07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagnostic</td>
<td>08</td>
<td>00-18,20,08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Com event counter</td>
<td>11</td>
<td>OB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Com Event Log</td>
<td>12</td>
<td>0C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report Slave ID</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read device Identification</td>
<td>43</td>
<td>14,2B</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td>Encapsulated Interface Transport</td>
<td>43</td>
<td>13,14,2B</td>
</tr>
</tbody>
</table>
Link - Topology

- Depends on the link technology
  - RS485
    - serial (daisy chain)
    - 1 master, many slaves (up to 255)
  - Modbus TCP/IP
    - Multiple masters
    - Multiple slaves
Link - Connection

**RS485**

- Serial cable (1 or 2 pairs)
- Variable transmission speed from 9600 to 115200 baud

**Modbus TCP/IP**

- Ethernet cable (Fiber for longer distances)
- Transmission speed depends on the underlying hardware / network
Modbus on Stage
**ZigBee**

**WIRELESS CONTROL THAT SIMPLY WORKS (?)**

ZigBee is a specification for a suite of high level communication protocols used to create personal area networks built from small, low-power digital radios. ZigBee is based on an IEEE 802.15 standard. Though low-powered, ZigBee devices often transmit data over longer distances by passing data through intermediate devices to reach more distant ones, creating a mesh network; i.e., a network with no centralized control or high-power transmitter/receiver able to reach all of the networked devices.
Basic Principles

- Low cost, very low power consumption, two way, wireless communications standard
  - Built on top of IEEE 802.15.4
- Secure
  - Messages are encrypted
- Defines several profiles
  - Standard: Home Automation, Smart Energy, etc.
  - Custom
- Each profile includes one or more ZCL (ZigBee Cluster Library) specification
Basic principles

ZigBee Profiles

Group of devices / functionalities pertaining a given application domain, e.g., Home Automation (ZigBee HA), Smart Energy (ZigBee SE), etc.

ZigBee Cluster Libraries

Device functionalities described in terms of client-server interactions
Logic Architecture
ZDO – ZigBee Device Objects

Initialize the application support sub-layer (APS), the network layer (NWK), and the Security Service Provider.

Assemble configuration information from the end applications to determine and implement discovery, security management, network management, and binding management.
Agreements for messages, message formats, and processing actions that enable developers to create an interoperable, distributed application employing application entities that reside on separate devices. These application profiles enable applications to send commands, request data, and process commands and requests.

Clusters are identified by a cluster identifier, which is associated with data flowing out of, or into, the device. Cluster identifiers are unique within the scope of a particular application profile.
Profiles

SMART ENERGY

This profile defines device descriptions and standard practices for Demand Response and Load Management “Smart Energy” applications needed in a Smart Energy based residential or light commercial environment. Installation scenarios range from a single room to an entire home up to 20,000 square feet (approximately 1850m²). The key application domains included in this initial version are lighting, HVAC, window shades and security.

ENERGY@HOME

The E@H features extend the HA and SE ZigBee profiles in order to build a new class of devices, i.e. White Goods for Energy@Home.

HOME AUTOMATION

This profile defines device descriptions and standard practices for applications needed in a residential or light commercial environment. Installation scenarios range from a single room to an entire home up to 20,000 square feet (approximately 1850m²). The key application domains included in this initial version are lighting, HVAC, window shades and security.
Clusters

• **Cluster**
  – Collection of attributes and commands
  – Defines a communications interface between two devices
  – Devices implement server and client sides of the interface

• **Client**
  – A cluster interface which is listed in the output cluster list of the simple descriptor on an endpoint.
  – Sends commands that manipulate the attributes on the corresponding server cluster.

• **Server**
  – A cluster interface which is listed in the input cluster list of the simple descriptor on an endpoint.
  – Typically this interface supports all or most of the attributes of the cluster.
## Cluster Definition: OnOffSwitch

<table>
<thead>
<tr>
<th>Server Side</th>
<th>Client Side</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mandatory</strong></td>
<td></td>
</tr>
<tr>
<td>On/Off Switch Configuration</td>
<td>On/Off (subject to binding)</td>
</tr>
<tr>
<td></td>
<td>Scenes</td>
</tr>
<tr>
<td></td>
<td>Groups</td>
</tr>
<tr>
<td></td>
<td>Identify</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
Cluster Definition: OnOffOutput

<table>
<thead>
<tr>
<th>Server Side</th>
<th>Client Side</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Mandatory</strong></td>
</tr>
<tr>
<td>On/Off</td>
<td>None</td>
</tr>
<tr>
<td>Scenes</td>
<td></td>
</tr>
<tr>
<td>Groups</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Optional</strong></td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
Link - Topology

★ STAR

- The network is controlled by one single device called the ZigBee coordinator.
- The ZigBee coordinator is responsible for initiating and maintaining the devices on the network.

Tree

- Coordinator is responsible for starting the network and for choosing certain key network parameters.

Mesh

- Full peer-to-peer communication.
- Routers in mesh networks do not currently emit regular IEEE 802.15.4-2003 beacons.
- The network may be extended through the use of ZigBee routers.

- Hierarchical routing strategy.
- Beacon-oriented communication.
Questions?

HOW TO HANDLE ALL THESE DIFFERENCES?