Networked Appliances
Generic CAL Device Addressing and Control

Generic CAL Device Addressing

- Device Addresses are either Physical or Logical
  - Physical Addresses may be hardcoded during manufacturing process
  - Logical addresses can exist at either Layer 3 or Layer 2:
    - Layer 3 Network Service Access points
    - Layer 2 Medium Access Control
- Device address may be Short or Long
  - Short device address
    - 2 x 16-bit words = 4 bytes
    - Backward compatible with EIA 600 (CEBus)
  - Long device address
    - 8 x 16-bit words = 16 bytes
    - Compatible with other protocols (e.g. IPv6)
- Node Control object (in Universal Context) is responsible for the management of addresses.
Generic CAL Device Addressing

Short Device Address

- 16 bits
- 16 bits

Area Address

Unit Address

- Each logical area that shares a Medium (power line, radio) is given a unique Area Address. Corresponds to House Code in EIA-600.
- Devices sharing a logical network within an Area are assigned unique Unit addresses. Corresponds to MAC address in EIA-600.

- Devices can determine their Unit and Area addresses either Statically or Dynamically.
  - Dynamic: Device asks other devices within the home or logical area for their Area Address. Unit address is determined by selecting an address not currently used under the Area Address.
  - Static: Addresses are allocated statically by another means.

MAC Addresses (CEBus)

- Individual MAC addresses identify particular nodes within the CEBus network.
- CEBus nodes must recognise: its individual MAC address, Broadcast address. Recognition of Group addresses is optional.
- Addresses are persistent (through power interruptions).
- Method for acquiring house code is not specified by EIA-600:
  - One allowable method is to use mechanical switches or factory preset addresses. (Static allocation)
  - Another method is to use the Resource Allocation function to allow a node to dynamically select and acquire its own individual address and group address.
**Generic CAL Device Addressing**

**MAC Addresses (CEBus)**

<table>
<thead>
<tr>
<th>Address range</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>F000 – FFFF</td>
<td>Reserved</td>
</tr>
<tr>
<td>8000 – EFFF</td>
<td>Individual</td>
</tr>
<tr>
<td>8000</td>
<td>Reserved</td>
</tr>
<tr>
<td>1001 – 7FFF</td>
<td>Individual</td>
</tr>
<tr>
<td>1000</td>
<td>Reserved</td>
</tr>
<tr>
<td>0101 – 0FFF</td>
<td>Group</td>
</tr>
<tr>
<td>0100</td>
<td>Reserved</td>
</tr>
<tr>
<td>00FE – 00FF</td>
<td>Reserved for Routers</td>
</tr>
<tr>
<td>0001 – 00FD</td>
<td>Individual</td>
</tr>
<tr>
<td>0000</td>
<td>Broadcast Address</td>
</tr>
</tbody>
</table>

---

**House Codes (CEBus)**

- House codes are used in the Medium Access Control (MAC) layer to identify particular nodes in a CEBus network.
- House codes are the equivalent of CAL Area Addresses.
- House codes can be either:
  - **House system address**: used with a group of devices which span multiple physical media
  - Or **House zone address**: only associated with a group of devices which reside on a single physical medium
- House code 0x0000 is the broadcast house code
- Method for acquiring house code is **not specified** by EIA-600:
  - One allowable method is to use mechanical switches or factory preset addresses. (Static allocation)
  - Another method is to use Resource Allocation function to allow a node to dynamically select its own individual address and group address.
### Generic CAL Device Addressing

#### House Codes (CEBus)

<table>
<thead>
<tr>
<th>Address range</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>F000 – FFFF</td>
<td>Reserved</td>
</tr>
<tr>
<td>8000 – EFFF</td>
<td>House System Addresses</td>
</tr>
<tr>
<td>8000</td>
<td>Reserved</td>
</tr>
<tr>
<td>0200 – 7FFF</td>
<td>House Zone Addresses</td>
</tr>
<tr>
<td>0100 – 01FF</td>
<td>House System Addresses</td>
</tr>
<tr>
<td>0001 – 00FF</td>
<td>House Zone Addresses</td>
</tr>
<tr>
<td>0000</td>
<td>Broadcast House Code / Global Network Address</td>
</tr>
</tbody>
</table>

### Generic CAL Device Object Hierarchy

- **Device**
  - **Universal Context**
    - **Object 1**
      - **Instance Variable 1 (IV)**
      - **Instance Variable m (IV)**
    - **Object n**
  - **Other Contexts**
Generic CAL Device Object Hierarchy

TV example

- **TV**
  - **00 Universal**
  - **10 Audio Amp**
  - **12 Tuner**

**Context Class**

- **01 Context Control (01)**
- **02 Context Control (02)**

**Object Class**

- **01 Node Control (01)**
- **02 Source Switch (09)**
- **03 Gain Control (07)**
- **02 Source Switch (09)**
- **03 Channel Tuning (09)**

**IV's include:** units_of_measure, step_size, step_rate, min_value, max_value, default_value, current_value, reporting_condition, dest_address

Generic CAL Contexts

- Different Context for each control on a device...
- Universal
- User Interface
- Operation Group
  - Audio
  - Video Display
  - Tuning
  - Security Zone
  - Environmental
  - (plus others to be added…)
**Generic CAL Objects**

**Overview**

- Object model implemented by Instance Variables (IV’s)
  - Change device characteristics by changing IV’s
  - IV’s are identified by:
    - Name, Function, Data type, Read-write, “Secured”
- “Reporting” objects
  - Report on conditions, events
  - Employ reporting_condition IV with dest_address IV
  - Use “report” method
- Node Control Object
  - Product_class IV set by manufacturer
  - Includes Power, On_offline, Reset IV’s

**Listing**

- 01 NODE CONTROL
- CONTEXT CONTROL
- 03 DATA CH RCVR
- 04 DATA CH XMTR
- 05 BINARY SWITCH
- 06 BINARY SENSOR
- 07 ANALOG CONTROL
- 08 ANALOG SENSOR
- 09 MULTI_STATE SWITCH
- 0A MULTI_STATE SENSOR
- 0B MATRIX SWITCH
- 0C MULTIPLANE SWITCH
- 0D GANGED ANALOG CTRL
- 0F METER
- 10 DISPLAY
- 11 TRANSPORT
- 12 DIALER
- 14 KEYPAD
- 15 LIST MEMORY
- 16 DATA MEMORY
- 17 MOTOR
- 19 SYNTHESIZER/TUNER
- 1A TONE GENERATOR
- 1C COUNTER
- 1D CLOCK
**Generic CAL Object Methods**

**Listing**

- 40 NOP
- 41 SETOFF
- 42 SETON
- 43 GETVALUE
- 44 GETARRAY
- 45 SETVALUE
- 46 SETARRAY
- 47 ADD
- 48 INCREMENT
- 49 SUBTRACT
- 4A DECREMENT
- 4B COMPARE
- 4E SWAP
- 4F REPORT
- 52 EXIT

- 53 ALIAS
- 54 INHERIT
- 55 DISINHERIT
- 56* IF
- 57* DO
- 58* WHILE
- 59* REPEAT
- 5A* BUILD
- 5B* COPYVALUE

* indicates a “complex” method

**Generic CAL Object Responses**

**Listing**

- FE COMPLETED
- FD ERROR
- FC FALSE EVALUATION
**Generic CAL Device Addressing**

**Conclusions**

- The Generic CAL (CEBus) addressing scheme operates on the Medium Access Control (MAC) layer (i.e. Layer 2).
- Uses hierarchical address (Area, Unit) and potentially supports IPv6 addressing (by using Long address).
- EIA-600 provides information on Area & Unit Address ranges.
- Does not specify how these addresses are acquired (which is what we are really interested in).
- So Generic CAL has not solved our device addressing problem, and we will have to develop our own way to allocate addresses...
- (Home Plug 'n' Play extends Generic CAL with concept of “Sub-systems”. Further investigation is required into this.)