

## **Gebäudeautomation - Software: Parametrierung, Programmierung und Visualisierung**

Mit den vorangegangenen Erörterungen sollte deutlich geworden sein, dass sich eine nahezu umfassende Gebäudeautomation bereits mit wenigen Standardtypen von Bus-Komponenten realisieren lässt, wenn das Gesamtsystem sorgfältig und differenziert geplant wurde. Dennoch scheitern viele Fachplaner an dieser Aufgabe, weil sie nicht einmal die wesentlichen Bus-Komponenten und deren Eigenschaften kennen, nicht das gesamte Gebäude mit allen haustechnischen Geräten in eine ganzheitliche Planung einbeziehen und/oder sich nicht der Mühe unterziehen, die Bedürfnisse und Lebensgewohnheiten der zukünftigen Bewohner ernsthaft zu reflektieren. Dies führt häufig bereits zu einer Verweigerungshaltung der Planer gegenüber Bus-Elektroinstallationen und damit letztlich zu einer erheblichen Komfort- und Sicherheitseinbuße für den Bauherrn, da die betreffenden Eigenschaften nicht oder nur mit großem Aufwand auf Basis einer konventionellen Elektroinstallation abzubilden sind.

Als noch folgenreicher sowohl hinsichtlich des Kosten-Nutzen-Verhältnisses als auch der späteren Funktion erweist sich allerdings die inkompetente und/oder wenig sorgfältige Planung eines Bus-Systems. Deshalb sind Bus-Installationen in Wohnhäusern nicht nur selten, sondern meist völlig unangemessen als teure Abbildung einer konventionellen Installation anzutreffen. Obgleich ein großer Teil der Funktionalität hier über die reine Hardware hinaus von der Software und Parametrierung bestimmt wird, lassen sich konzeptionelle Fehler leider nicht oder nur eingeschränkt Software-seitig ausbügeln. Umgekehrt setzen Komponentenauswahl und Verdrahtungsplanung fundierte Kenntnisse in der Programmierung und Parametrierung voraus. Die Planer können und dürfen sich diesbezüglich eben nicht aus der Verantwortung stellen, indem sie diese Aufgaben einfach auf den ausführenden Elektriker deligieren.

Zur langfristigen Sicherstellung von Service, Wartung und Erweiterung der komplexen technischen Anlagen - insbesondere der Bus-Installation - nach der Gewährleistungsfrist bzw. auch nach Ausscheiden des Planungspartners und/oder Installationsbetriebes z.B. durch Insolvenz, müssen adäquate Maßnahmen vorbeugend getroffen werden. Dazu gehören detaillierte Pläne und Dokumentationen, exakte Beschriftung der Anlagenkomponenten mit Aktualisierung bei jeder Änderung (auch reversionssichere Aktualisierung der Schalt- und Leitungspläne) sowie die selten stattfindende ausführliche Einweisung der Bewohner in die (noch) ungewöhnliche Technik des Hauses mit hilfreichen Verhaltensregeln.

### **Standard-Software-Werkzeuge**

Wünschenswert ist ein Systemadministrator unter den Bewohnern, der selbst in der Lage ist, in die Parametrierung des Bus-Systems einzugreifen. Nur so lässt sich der Vorteil, funktionale Änderungen ohne Modifikation der Hardware umsetzen und die Steuerung des Gebäudes fortlaufend auf die Lebensgewohnheiten und Betriebsbedingungen abstimmen zu können, ohne regelmäßige Anforderung des Service realisieren.

Bedauerlicherweise muss man feststellen, dass die für EIB-Systeme verfügbare Software zur Programmierung bzw. Parametrierung in keiner Weise befriedigen kann. Das gilt sowohl aus der Sicht des IT- und EIB-Profis als auch erst recht aus der Sicht des Bewohners, also in der Regel eines absoluten Laien, der mit dem System lebt. Es grenzt an Anmaßung, wenn Experten die Bewohner entmündigen, indem sie ihnen grundsätzlich keinen Eingriff in das System zutrauen oder gar gestatten.

Obgleich der Autor die Grundprogrammierung eines solchen Systems allein einem IT- und EIB-Experten (und auch nicht jedem Elektriker) anvertrauen würde, hält er es für erforderlich, dass die Bewohner zumindest in die Lage versetzt werden, in die Feinparametrierung einzugreifen. In der Tat ist es eine Zumutung für die Bewohner und gleichzeitig eine Pervertierung der Vorzüge eines Bus-Systems im Hinblick auf funktionale Änderungen ohne Eingriff in die Hardware oder Verdrahtung, wenn geringfügige Modifikationen häufig kosten- und zeitintensive externe Dienstleistungen erfordern.

Deshalb wäre dringend zu wünschen, dass eine Standard-Software zur Programmierung und Parametrierung über einen Experten- und einen Laien-Modus verfügt, mit dem z.B. Ein- und Ausschaltverzögerungen oder Einschaltdauer gefahrlos für das Gesamtsystem an den täglichen Bedarf zu adaptieren sind. Es gibt wirklich keinen vernünftigen Grund, warum die Bewohner die Nachlaufzeit einer Automatikbeleuchtung bei Bedarf nicht selbst von einer auf zwei Minuten oder die Haltedauer eines Türöffners nicht von drei auf fünf Sekunden verlängern dürften.

Derart einfache Maßnahmen gestalten sich jedoch unter der ETS (EIB Tool Software) für den Laien ausgesprochen umständlich: Software starten, in der Gebäudeansicht das Gerät und dann den Kanal des entsprechenden Aktors suchen, zurück zum Geräteeintrag, über das Kontextmenü den Parameter-Dialog öffnen, dort wieder den gewünschten Kanal und dessen Funktionsseite "Zeiten" suchen, den betreffenden Parameter einstellen, Dialog bestätigen und schließen, über das Kontextmenü einen weiteren Dialog öffnen und die Übertragung der Applikation via Bus in den Speicher der Komponente auslösen. Für den Laien ist dies kaum praktikabel, sollte er überhaupt über die ETS verfügen, die ja schließlich zusätzlich und kostenpflichtig nicht vom Installationsunternehmen, sondern nur von der EIBA direkt zu erwerben ist.

Aber auch Profis finden in der ETS selbst nach umfassendem Facelifting der Version 3 nicht das adäquate Werkzeug. Das liegt nicht nur an den immer noch zahlreich vorhandenen Programmfehlern, z.B. dem Löschen von Gruppenadressen nach Umparametrierung trotz Abbrechen in der Rückfrage, sondern auch an den mangelhaften Komfortfunktionen, z.B. die Nichteinsortierung neu erfasster Gruppenadressen ohne Schließung des Strukturbaumes der Liste oder die fehlende Verzweigungsmöglichkeit zwischen Geräte- und Gruppenansicht. Zudem leidet die Übersichtlichkeit bei komplexen Installationen aufgrund der ausgeprägten "Atomisierung" von Detailinformationen in zahlreichen kleinen Dialogen mit unzähligen Registriertarten und irritierendem Layout. Dafür gibt es zahlreiche Beispiele, von denen wir nur eines herausgreifen möchten:

Nach der Installation und Grundprogrammierung des Systems kommen häufig reine Parameteränderungen im Rahmen der Feinabstimmung sowie bei Anpassung an geänderte Lebensbedingungen zum Zuge. In Anlehnung an das obige Beispiel nehmen wir an, eine Leuchte soll verzögert ausschalten. Nach dem Suchspiel der betreffenden Bus-Komponente in der Gebäudeansicht, öffnet man den Parameter-Dialog über das Kontextmenü. Dieser zeigt eine Liste mit massenhaften Einträgen wie "Kanal A Allgemein, Kanal A Zeiten, Kanal A Verknüpfung/Sperrung, Kanal B Allgemein, Kanal B Zeiten, Kanal B Verknüpfung/Sperrung etc.". Bei einem 12-fach-Aktor zwingen ggf. bis zu 60 Einträge bereits zum Scrollen der Liste und erzeugen ein Informationschaos.

Sollte man nun nicht wissen, welcher Kanal betroffen ist, und auch keinen Schaltplan zur Hand haben, muss man den Dialog abbrechen, den Knoten des Gerätes in der Liste öffnen und die einzelnen Objekte in der Ebene darunter durchwandern, um an den dabei jeweils angezeigten Gruppenadressnamen den betroffenen Kanal erkennen zu können. Nun geht es zurück auf den Geräteeintrag, gefolgt von erneutem Öffnen des Parameter-Dialoges über das Kontextmenü und Suchen des Kanaleintrages. Die direkte Auswahl der Kanalseite "Zeiten" hilft nun nicht weiter, weil die betreffenden Steuerelemente ausgeblendet sind. Also zurück zur Seite Allgemein und Ausschaltverzögerung einschalten, danach auf der Seite "Zeiten" die Zeitbasis und den Faktor definieren. Anschließend Dialog bestätigen und schließen, neuen Programmier-Dialog via Kontextmenü öffnen, "Applikation" auswählen und bestätigen. In dieser Zeit hätte man sich "einen Knopf an die Backe nähern" können.

Sinnvoll wäre natürlich eine Benennungsmöglichkeit der Kanäle, ein Öffnen des Parameter-Dialoges auf jedem Objekt des Gerätes mit automatischer Positionierung auf den betreffenden Kanal des zuletzt aktiven Objektes, eine Darstellung von Kanälen und Funktionsseiten (hier Allgemein, Zeiten, Verknüpfung und Sperrung) in einer Strukturliste oder besser eine Zusammenfassung dieser Funktionsseiten, z.B. ein Steuerelement im Zeitformat "hh:nn:ss" (h=Stunden und n=Minuten optional, 0=aus) anstelle von drei Elementen zur Aktivierung der Ausschaltverzögerung, Auswahl einer Zeitbasis und Eingabe des Faktors auf zwei Bildschirmseiten des Dialoges. Im Dialog fehlt dann nur noch eine Schaltfläche zum direkten Download der Applikation in die betreffende Komponente zusammen mit dem Speichern und Schließen des Parameter-Dialoges.

Eine Reduktion des Programmier-Dialoges wäre nicht nur komfortabler, sondern auch weniger fehlerträchtig. Schließlich werden in diesem Dialog allerlei weitere, höchst selten gebrauchte Funktionen angeboten, wie z.B. das Programmieren der physikalischen Adresse, welches im Gegensatz zur relativ häufigen Änderung von Gruppenzuordnungen und insbesondere Parametern regulär nur einmalig nach Installation einer Komponente verwendet wird.

Ebenso ließe sich auch die Einstellung von Verknüpfung/Sperrung erheblich übersichtlicher direkt auf einer Kanalseite steuern. Die Auswahl eines übergeordneten Objektes (keines, Verknüpfung oder Sperrung) auf der Allgemein-Seite und die Auswahl der Art auf einer weiteren Kanalseite wäre naheliegend in einer Auswahlliste zusammenzufassen, nämlich "- (keines), UND-Verknüpfung, ODER-Verknüpfung, Sperrung bei Sperrojekt=0, Sperrung bei Sperrojekt=1". Hinzu käme nur die Einstellung des Verhaltens bei Busspannungsverlust und -wiederkehr, für die normalerweise die Parameterwerte "keine Änderung" (ohne Lastabwurf) und "aus" (sicherer und energieärmerer Zustand) zum Zuge kommen.

Bedauerlicherweise fehlt hier - wie auch für andere Parameter - die Möglichkeit, Standardwerte (Defaults) vorzudefinieren und komplexe Parametrierungen von einem Kanal auf einen oder alle anderen, ggf. sogar auf ein anderes Gerät gleichen Typs, übernehmen zu können. Wer hunderte von Aktor-Ausgängen in einem Projekt zu konfigurieren hat, würde die bei anderen IT-Applikationen üblichen Komfortmerkmale jedenfalls zu schätzen wissen.

Ärgerlich ist auch, dass die ETS kaum und auf Kommunikationsobjekt-Ebene überhaupt keine ausführlichen Bemerkungen oder Kommentierungen zulässt. Wichtige Anmerkungen, Erläuterungen und Begründungen zu Einstellungen unterbleiben deshalb oft sogar bei komplexen Verknüpfungen und Parametrierungen oder sie werden in einer eigenen Software bzw. auf Papier geführt, was weder ihrer Aktualität noch der Übersichtlichkeit zugute kommt.

Selbst ohne weitere Vertiefung von Unzulänglichkeiten der ETS stellt sich die Frage, warum man ausgerechnet diese Software favorisieren sollte. Man soll es nicht, sondern man muss, weil überhaupt keine Alternative zur Verfügung steht. Der Autor hat es zunächst nicht für möglich gehalten, dass hier ein einziges Produkt konkurrenzlos den Markt beherrscht, zumal dies nicht auf die Hardware zutrifft, die von vielen Herstellern entwickelt (sofern kein OEM-Produkt) und vermarktet wird. Ein offener Industriestandard verlangt geradezu nach einer breiten Beteiligung von Entwicklern auch im Bereich der Software. Die marktwidrige Alleinstellung dieses Produktes bedarf dringend einer Korrektur im Sinne eines angemessenen Wettbewerbes.

**Nachträgliche Anmerkung des Verfassers im Spätsommer 2006:** Inzwischen wurde das ETS-"Kartell" gesprengt und durch T4 EIB der Firma TETRA eine Alternative zur Planung, Programmierung und Pflege einer EIB-Installation bereichert (siehe [www.tetra-software.de](http://www.tetra-software.de)). Neben einem erheblich erweiterten Funktionsumfang bis zur integrierten Visualisierung, wurden viele der o.g. Schwächen behoben. Einige Probleme sind jedoch systembedingt auf unzureichende Definitionen des Standards sowie mangelhafte Parametrierungsstrukturen in den Hersteller-Produktdatenbanken zurückzuführen, die deshalb leider nicht vollständig durch die neue Software optimiert werden konnten.

## **Proprietäre Tools**

So sehr man sich über die ETS auch aufregen mag, schaffen es die Gerätehersteller problemlos, diese Software durch eigene proprietäre Spezialapplikationen für bestimmte Komponenten in puncto mangelndem Komfort zu toppen. Zunächst einmal durchbrechen sie damit den durchaus sinnvollen Ansatz, ein Programmierwerkzeug Geräte- und Hersteller-neutral zur Verfügung zu stellen.

Es nervt ungemein, für diverse Geräte eines Bus-Systems mehrere Applikationen gleichzeitig einsetzen zu müssen, insbesondere wenn zusammenhängende Funktionen programmiert oder geändert werden, keine Schnittstellen zwischen diesen Werkzeugen existieren und Gruppenadressen mehrfach zu deklarieren sind. Dabei sieht man in einer Software nicht die anderen Geräte, welche mit der gleichen Gruppenadresse verknüpft wurden. Die Gruppenadressen fremdprogrammierter Komponenten sind zusätzlich in Form von Dummy-Geräten bekannt zu machen, weil sie ansonsten beim Laden der Filtertabellen in die Linienkoppler keine Berücksichtigung finden.

Als wäre dies nicht genug der überflüssigen Belastung, schaffen es die Gerätehersteller bzw. deren im Outsourcing tätigen Software-Entwickler, wirklich lieblos implementierte Applikationen abseits aller üblichen Standards zu codieren. Da mutieren z.B. Menüs zu Schaltflächen, die eine einzige Funktion auslösen oder gar eine weitere Software starten, wie z.B. der Loader aus der Funktionsmodul-Software mehrerer Anbieter.

Ein Software-"Highlight" bietet beispielsweise eine sonst so respektable und seriöse Firma im Zusammenhang mit ihren relativ neuen Bedienpanels. Nachdem der erste Schock, dass fast 1.000 € teure Produkte mit einem miserabel lesbaren Display im technischen Standard von 1990 ausgestattet sind, überwunden wurde, zeigten sich die Software-mäßigen Differenzen zwischen der kleinen 8-Tasten-Version und den ansonsten völlig gleich anmutenden mittleren und großen Versionen mit Display und Menü. Erstere ist als Mehrfachtaster einfach über die ETS zu programmieren, während für Letztere eine eigene Software mit den o.g. Problemen zum Einsatz kommt, obgleich aufgrund der Funktionalität sachlich nichts gegen eine Vollintegration in den Standard gesprochen hätte.

Als völlig widersinnig nimmt man dann zur Kenntnis, dass wichtige Funktionen der kleinen Version in den teuren großen Versionen fehlen, z.B. die Möglichkeit, mehrere Objekte bei steigender und fallender Flanke eines Tastendruckes getrennt anzusteuern oder Tasten über Gruppenadressen dynamisch zu

sperrt. Die willkürliche, nicht einstellbare Gruppierung der Funktionen und Menüs überrascht dann kaum noch. So gibt es keine Gruppe "Sicherheit", weshalb ein Türöffner beispielsweise in der Gruppe "Beleuchtung" landet. Unter "Heizung" stehen keine reinen Schaltfunktionen (1 Bit) zur Verfügung und müssen folglich auch unter "Beleuchtung" ihr Unwesen treiben. Schließlich reichen die verfügbaren Objekte insbesondere bei Verwendung von Rückobjekten für Display und LEDs überhaupt nicht aus, um die Tasten vollständig zu belegen und schon gar nicht, um zusätzlich selten benutzte Befehle allein im Menü unterzubringen. Was immer - wenn überhaupt - sich die Entwickler dabei gedacht haben mögen?

Das Hinzufügen, Ändern und Entfernen von Gruppenzuordnungen gestaltet sich als Geduldsspiel, weil die Entwickler entgegen allen Software-Standards nicht einmal fundamentale Tastaturbedienungen vorgesehen haben. Man müsste den Hersteller für die Arthrose im "Mausklickfinger" verantwortlich machen, die sicher dann entsteht, wenn jemand häufig zahlreiche komplexe Programmierungen solcher Geräte vornehmen muss. Allein die Suche nach der Funktion zu einer bestimmten Gruppenadresse grenzt an "Körperverletzung". Die Strukturliste der Funktionsübersicht zeigt nämlich keine Adressen an. Also jeden Eintrag per Doppelklick öffnen, im Dialog nach der Gruppenadresse schauen und Dialog per Mausklick auf die Abbrechen-Taste schließen - als wäre es eine softwaretechnische Herausforderung, wenigstens diese Schaltfläche durch Drücken der Esc-Taste zu emulieren. Leider wird man bei dieser Tortour nicht einmal mit liebevoll gestalteten Dialogen entschädigt. Die Entwickler fanden es offensichtlich ausreichend, die Steuerelemente irgendwie dort hineinzustreuen anstatt sie übersichtlich und optisch ansprechend zu positionieren. Ohne die weiteren Unzulänglichkeiten zu vertiefen, wäre hier unzweifelhaft die Vergabe einer "Silberne Zitrone" angemessen.

Da alle Software-Produkte auf die gleiche Falcon-Entwicklungsumgebung (DLLs etc.) der EIBA zugreifen, sollte bei der Installation proprietärer Software kein Fehler unterlaufen oder diese gar deinstalliert werden müssen. Ansonsten könnte damit die Funktionsfähigkeit aller anderen Produkte inklusive der ETS selbst in Frage gestellt werden. Die "vernagelte" Basisinstallation lässt sich dann nicht einmal durch das Setup-Programm korrigieren.

Bei differenzierter Betrachtung und Anwendung dieser dilettantischen Werkzeuge reibt man sich verwundert die Augen und mag es kaum fassen, dass diese von den Anwendern (zumeist Elektroinstallateuren) so kritiklos akzeptiert und die Hersteller nicht mit massiven Mängelrügen überzogen werden. Hier stellt sich die Frage, ob die Entwickler es wirklich nicht besser können, oder die Software zu Bus-Produkten als lästige Nebenwirkung des Hardware-Geschäftes angesehen wird, die sich keinesfalls ertragsreduzierend auf die ohnehin überhöhten Preise dieser Produkte auswirken darf. Deshalb wäre es endlich an der Zeit, dieser - gelinde ausgedrückt - wenig kundenorientierten Haltung der Anbieter auf breiter Front mit der gebotenen Härte entgegenzutreten.

Das gilt auch für die vielfach anzutreffende unsinnige Update-Praxis der Hersteller. Ohne wirklich erkennbaren Grund erscheinen Nachfolgemodelle von Komponenten ohne Abwärtskompatibilität der zugehörigen Applikationen. Bei funktional nahezu identischen Produkten ist dies überhaupt nicht nachvollziehbar. Selbst erweiterte Funktionen und Parameter sprechen keineswegs gegen eine Abwärtskompatibilität, wie sie in der Software-Branche weitgehend üblich ist. Stattdessen muten die Hersteller dem Anwender eine komplette Neuparametrierung/-programmierung beim Zwangsaustausch einer fehlerhaften Komponente durch ein Nachfolgemodell zu. Im Falle eines 12-fach-Aktors musste der Autor 36 Objekte neu zuordnen und über 200 Parameter einstellen - ein geradezu grotesker Zustand.

## **Transaktionsprobleme**

Es wäre noch auf ein grundsätzliches Problem beim Programmieren von Bus-Komponenten hinzuweisen, das nach Auffassung des Autors zugleich auch den gravierendsten Nachteil dezentraler Steuerungssysteme vor Augen führt. Es gibt nämlich keine Transaktionssicherheit über alle BCUs einer Steuerungskette hinweg und keine definierten Zustände nach dem Upload von Applikationsänderungen einzelner Komponenten. Wenn eine Bus-Nachricht nur bei Änderung des zugrundeliegenden Wertes gesendet wird, geht die aktuelle Zustandsinformation bei einer neu programmierten Komponente verloren. Ein 12-fach-Aktor kann z.B. danach seine Ausgänge nur auf die parametrisierten Default-Zustände setzen. Sind die Schalt- oder Sperrobjekte z.B. von einem Helligkeitsgrenzwert abhängig, der nur zweimal täglich bei Einbruch der Dunkelheit und der Morgendämmerung gesendet wird, schalten alle Ausgänge entweder bei Programmierung am Tage oder bei Dunkelheit falsch.

Bei Sensoren und Binäreingängen lässt sich dieser Effekt teilweise dadurch abmildern, dass sie den Wert auch ohne Änderung engmaschig zyklisch senden, die Aktoren also innerhalb weniger Sekunden oder Minuten wieder die korrekte Zustandsinformation erhalten. Bei Nachrichten, welche auf Rückobjekten von

Aktoren basieren ist dies jedoch ohne Lesen-Flag und ReadRequest nicht möglich. Das wirkt sich besonders fatal aus, wenn es sich um fundamentale Zustandsinformationen über Anwesenheit, Aktivität etc. oder komplexe Abhängigkeiten handelt. So kann sich die Programmierung eines Funktionsmoduls zu einem echten Abenteuer entwickeln, wobei sämtliche Automaten "aus dem Ruder laufen" und Sicherheitsfunktionen sogar Alarm auslösen.

Diese Problematik wäre gänzlich vermeidbar, wenn die Entwickler der Bus-Komponenten sie nur ernst nehmen würden. Warum sind sie in der Lage, während des Uploads einer Applikation alle bisherigen Zustände beizubehalten, um erst nach dessen Abschluss einen definierten Reset mit den neuen Gruppenadressen und/oder Parametern durchzuführen, ohne die Eingangswerte fortbestehender Gruppenadressen zu übernehmen und danach initial adäquat zu schalten? Die immer wieder bemühte Argumentation mit dem eingeschränkten Speicherplatz war jedenfalls vor 10 Jahren schon eine faule Ausrede und ist aktuell angesichts minimaler Größe und Kosten der Speicher-Chips sowie erweiterter Adressräume geradezu grotesk.

Verschärft wird die Problematik übrigens noch dadurch, dass die Komponenten nur schleppend eine Differenzierung von Reset nach Bus-Spannungsausfall und nach Programmierung erhalten. Anwesenheits- oder Aktivitäts-abhängige Aktoren müssen sonst z.B. zwangsläufig auf "aus" nach Bus-Spannungswiederkehr parametrisiert werden, damit sich im Fehlerfall die angeschlossenen Geräte nicht automatisch einschalten. Beim Programmieren ist man hingegen zwangsläufig anwesend und aktiv, so dass man natürlich "begeistert" das Abschalten von Licht und Steckdosen inklusive Monitor miterlebt.

### **Strukturierung der Bus-Installation**

Nach dieser leider überfälligen Herstellerschelte, kommen wir zurück zu den grundlegenden Regeln zur Software-seitigen Implementierung eines EIB-Systems. Auch bei der Software-Strukturierung des Bus-Systems gilt die Prämisse "keep it simple", d.h. die Gliederung darf im Interesse der Übersichtlichkeit nicht überzogen werden. Für die Hardware-Seite genügt in der Regel eine einzige Sicht mit folgenden Grundsätzen:

- Verzicht auf Gewerk-Sicht (meist überflüssig, s.o.)
- möglichst deckungsgleiche Gebäude- und Topologie-Sicht
- nicht für jeden Raum mit einem Gerät ein eigener Knoten (zusammenfassen)
- je (Unter-)Verteilung mit vielen Geräten ein Knoten
- möglichst sinnvolle Vergabe der physikalischen Adressen, die dann nicht nur die Gliederung in Bereiche und Linien, sondern auch in Etagen, Räume und Funktionsbereiche widerspiegeln

Die funktionale Sicht kommt in Ein- oder Zweifamilienhäusern mit zweistufigen Gruppenadressen aus, die folgende Hauptgruppen ohne Mittelgruppen aufweisen sollten:

- Zentralfunktionen (anwesend, aktiv, Zugangskontrolle, elektrische Türöffner, Riegelkontakte, Alarm etc.)
- Beleuchtung
- Jalousien
- Klima (Lüftung und Heizung/Kühlung)
- besondere technische Anlagen (z.B. Schwimmbad)
- Multimedia (IT, AV/UE etc.)

**... to be continued**