



Hello world!!!



# LINUX-Control-Unit LCU 1

## Der Interfaceprozess hs485d

Im vorangegangenen Artikel wurden die Logiksteuerung mit immediateC und der Zugriff auf die Ein- und Ausgänge der LCU 1 beschrieben. In diesem Teil der Artikelserie wird die Software der LCU 1 nochmals erweitert. Es kommt der Interfaceprozess hs485d hinzu, über den sich Komponenten der Gerätefamilie HomeMatic Wired an den RS485-Anschluss der LCU 1 anschließen lassen. Als praktisches Einsatzbeispiel stellen wir ein mit HomeMatic Wired realisiertes Türschloss mit Pin-Eingabe-Zugang vor.

### Kontakt nach draußen

Der Interfaceprozess hs485d „spricht“ das bereits von der HomeMatic-Zentrale bekannte XML-RPC-Protokoll wie der im vorangegangenen Artikel vorgestellte Plattformprozess pfmd. Dadurch lassen sich die Ein- und Ausgänge der HomeMatic-Wired-Komponenten auf die gleiche Art und Weise wie die Anschlüsse der LCU 1 in einem Logikprogramm verwenden.

Um dies zu demonstrieren, wollen wir das Logikprogramm aus dem letzten Artikel um die Funktionalität eines Türschlosses mit Pin-Eingabe erweitern. Die externe Hardware zu dieser Applikation wird an ein HomeMatic-Wired-Modul vom Typ HMW-IO12-SW14-DR (ELV-Art.-Nr. JM-09 20 11) angeschlossen.

Zur Konfiguration und Bedienung der neuen Applikation wird die Oberflächenapplikation um einige

Seiten erweitert und implementiert zusätzlich eine Logfunktion. Hierüber können die letzten 50 Ereignisse, die mit dem Türschloss im Zusammenhang stehen, nebst Zeitstempeln über das TFT-Display abgerufen werden.

Die Einbindung eines HomeMatic-Wired-Gerätes in eine Logikapplikation erfordert gegenüber der Verwendung der Anschlüsse der LCU 1 einen weiteren Konfigurationsschritt. Es muss einem in der Konfiguration des Logik-Switches (siehe Artikel im ELVjournal 6/2011) definierten logischen Gerät ein bestimmtes physikalisches Gerät zugeordnet werden. Für diese Zuordnung und einige weitere Funktionen wurden Webseiten auf dem Webserver der LCU 1 abgelegt. Damit kann diese Konfiguration über einen normalen Webbrowser erfolgen.

```
<interfaces>
  <interface id=»RS485» url=»bin://127.0.0.1:2000»/>
  <interface id=»SYSTEM» url=»bin://127.0.0.1:2002»/>
</interfaces>
```

Bild 17: Die Definition des Interfaceprozesses in der Konfigurationsdatei des Logik-Switches

## Der Interfaceprozess hs485d

Der Interfaceprozess hs485d greift über den Treiber hss\_comm auf den RS485-Anschluss der LCU 1 zu. Der Treiber implementiert das HomeMatic-Wired-Protokoll. Er befindet sich im Quellcodeverzeichnis `drivers/communication`, wird als Kernelmodul kompiliert und beim Hochfahren über das Startscript `S90hs485d` geladen. Im gleichen Startscript erfolgt auch der Start des hs485d. Auf der anderen Seite stellt der hs485d über XML-RPC den Zugriff auf die Busgeräte bereit. Die Definitionsdateien für die unterstützten Gerätetypen befinden sich im Quellcode im Verzeichnis `devicetypes/hs485types`. In der fertigen Firmware liegen diese im Verzeichnis `/firmware/hs485types`. Die angelernten Geräte werden vom hs485d im Verzeichnis `/etc/config/hs485d` verwaltet.

Das XML-RPC-API. Entsprechend wird in der Konfigurationsdatei des Logik-Switches `icserver_system.conf` der Interfaceprozess mit der Id `RS485` wie in **Bild 17** gezeigt definiert.

## Die Applikation „Türschloss“

### Die Hardware

Die Grafik in **Bild 18** zeigt den Schaltplan der für unsere Türschlossapplikation nötigen externen Hardware. Zentrales Element ist ein HomeMatic-Wired-Hutschienenmodul vom Typ `HMW-IO12-SW14-DR`. Dieses Modul hat 12 digitale Eingänge, 6 Relaisausgänge und 8 Open-Collector-Transistorausgänge.

Die Ein- und Ausgänge bieten noch weitere Funktionen zum Umgang mit Analogsignalen oder Frequenzsignalen, die aber nicht Gegenstand dieses Artikels sind.

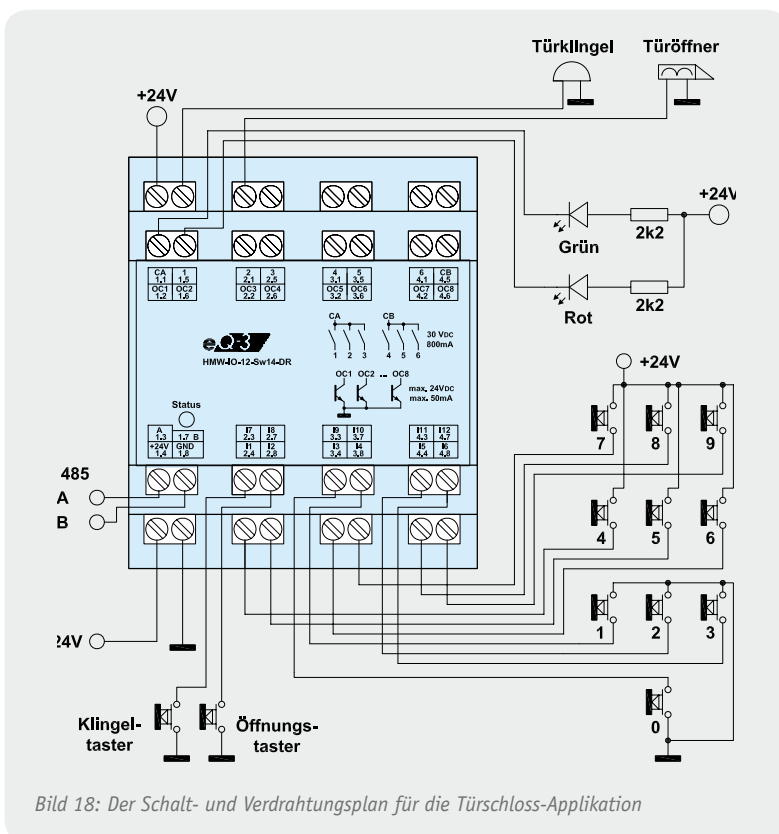


Bild 18: Der Schalt- und Verdrahtungsplan für die Türschloss-Applikation

Das Tastenfeld zur Pin-Eingabe besteht aus 10 einzelnen Tastern für die Ziffern 0 bis 9. Diese sind an zehn verschiedene „Digitaleingänge“ des Moduls angeschlossen. Aufgrund der modulinternen Realisierung und der Zweitfunktion der Eingänge schalten die Taster 0 bis 3 gegen GND, während die Taster 4 bis 9 gegen +24 V schalten.

Darüber hinaus sind ein Klingeltaster und ein Öffnungstaster vorgesehen. Diese werden analog zu den Tasten 3–3 gegen GND geschaltet.

Für das Tastenfeld zur Pin-Eingabe sind je eine rote und eine grüne LED vorgesehen. Diese signalisieren während der Pin-Eingabe den Betriebszustand.

Eine Türklingel für innen sowie der Türöffner werden an je einen Relaisausgang angeschlossen. Die Spannungsversorgung erfolgt mit +24 V am gemeinsamen Kontakt der beiden Relais.

Der Anschluss an die LCU 1 erfolgt über die beiden Anschlüsse A und B des RS485-Busses.

### Die Oberfläche

Die Oberfläche der Beispielapplikation besteht aus drei neuen Oberflächenseiten, die wir in der Folge genauer vorstellen.

### Übersichtsseite

Die Übersichtsseite für den Türöffner (**Bild 19**) wird über die entsprechend benannte neue Schaltfläche auf der Hauptseite erreicht. Diese Seite ist in der Datei `dooroverview.xml` implementiert.

Die obere Schaltfläche löst den Türöffner aus. Die Schaltfläche „Pin setzen“ öffnet die bereits bekannte Seite `numpad.xml` für die Eingabe der Pin. Wenn die Checkbox „Klingeltaster öffnet Tür“ aktiv ist, löst ein Betätigen des im Außenbereich angebrachten Klingeltasters automatisch den Türöffner aus. Diese Funktion lässt sich natürlich auch gut mit einer Zeitsteuerung kombinieren. Aus Gründen der Übersichtlichkeit wurde jedoch darauf verzichtet. In Verbindung mit der im letzten Artikel vorgestellten Implementierung einer Zeitschaltuhr kann diese Funktion allerdings leicht nachgerüstet werden.

Die Schaltfläche „Ereignisliste“ zeigt die letzten 50 im Zusammenhang mit dem Türöffner aufgetretenen Ereignisse in einer Liste an. Die zugehörige Oberflächenseite ist weiter unten beschrieben.

Die Schaltfläche „Einstellungen“ öffnet eine neue Seite mit den zum Türöffner gehörenden Konfigurationsoptionen.

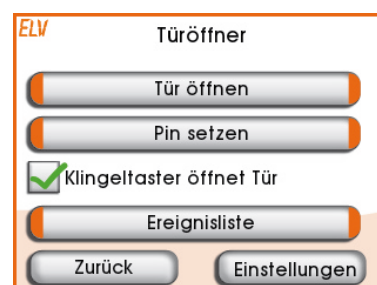


Bild 19: Die in die Datei `dooroverview.xml` implementierte Übersichtsseite für den Türöffner

## Einstellungsseite

Diese Seite (Bild 20) ist in der Datei `doorconfig.xml` implementiert.



Bild 20: Die Einstellungsseite für den Türöffner mit den zugehörigen Konfigurationsoptionen

Auf dieser Seite werden die Konfigurationsoptionen für den Türöffner gesetzt. „Einschaltdauer“ gibt die Zeit in Sekunden an, die das Relais für den Türöffner aktiviert.

Der „Pintimeout“ ist die Zeit, die maximal zwischen der Eingabe von zwei Ziffern der Pin vergehen darf. „Pinlänge“ ist die Anzahl von Stellen, die die Pin hat.

## Ereignisliste

Diese Seite (Bild 21) ist in der Datei `doorlogging.xml` implementiert.



Bild 21: Die in der Datei `doorlogging.xml` implementierte Ereignisliste

Die Seite für die Logfunktion zeigt die jeweils 50 letzten im Zusammenhang mit dem Türöffner aufgetretenen Ereignisse mit Datum und Zeit in einer Liste an. Die Liste wird unterteilt in Seiten zu jeweils 10 Einträgen. Es stehen zwei Schaltflächen bereit, um durch die Seiten zu blättern. Diese Schaltflächen werden nach Bedarf ein- und ausgeblendet. Die Schaltfläche „Löschen“ leert die Ereignisliste.

## Implementierung der Ereignisliste

Das Logikprogramm stellt das Signal `intUiDoorLogEvent` (siehe Abschnitt „Signale für die Oberfläche“) bereit. Im Ruhezustand hat dieses Signal den Wert 0. Bei jedem Ereignis ändert es sich von 0 auf einen anderen Wert, der das Ereignis angibt. Es sind folgende Ereignisse definiert:

- 1 = Pin akzeptiert
- 2 = geöffnet durch Klingeltaster
- 3 = geöffnet durch Freigabetaster
- 4 = geöffnet über den Touchscreen
- 5 = falsche Pin wurde eingegeben
- 6 = Time-out bei der Eingabe der Pin

Die Oberflächenseite definiert für dieses Signal einen globalen Trigger, siehe Bild 22. Globale Trigger sind durch das Attribut `global="true"` gekennzeichnet. Im Gegensatz zu einem regulären Trigger wird ein globaler Trigger auch aufgerufen, wenn die entsprechende Oberflächenseite nicht angezeigt wird. Dieser Trigger sorgt dafür, dass bei jeder Änderung des Signals `intUiDoorLogEvent` die TCL-Methode `OnDoorLogEvent` (Bild 23) aufgerufen wird.

Diese Methode speichert die eingehenden Ereignisse in der globalen Liste `events`. Es werden immer die neuesten 50 Ereignisse vorgehalten. Die Variable `pageShown` ist 1, wenn die Ereignisliste aktuell angezeigt wird. In diesem Fall wird durch Aufruf der Methode

```
<triggers>
  <trigger value=>intUiDoorLogEvent<> method=>OnDoorLogEvent<> global=>true</>
</triggers>
```

Bild 22: Die Trigger-Definition für das Signal „intUiDoorLogEvent“

```
proc OnDoorLogEvent { valueId eventCode } {
  variable events
  variable NB_LOGENTRIES
  variable pageShown
  if { $eventCode } {
    set events [linsert $events 0 [list [clock seconds] $eventCode]]
    if { [llength $events] > $NB_LOGENTRIES } {
      set events [lrange $events 0 [expr $NB_LOGENTRIES - 1]]
    }
    if { $pageShown } {
      UpdateDisplay
    }
    SaveEvents
  }
}
```

Bild 23: Bei jeder Änderung des Signals „intUiDoorLogEvent“ wird die Methode „OnDoorLogEvent“ aufgerufen.

UpdateDisplay die Anzeige aktualisiert.

Die Methode SaveEvents speichert die aktualisierte Ereignisliste in der Datei /etc/config/doorevents. Aus dieser Datei wird beim Starten die Ereignisliste initialisiert. Dadurch bleibt die Ereignisliste über einen Reset hinaus erhalten.

Die komplette Implementierung der Ereignisliste befindet sich in der Datei doorlogging.xml.

## Die Logik

Die Logik der Beispielapplikation ist in der Datei iclogic/doorlock.ica implementiert. Da die Programmierung mit immediateC bereits im letzten Artikel besprochen wurde, verzichten wir hier auf eine detaillierte Beschreibung der Implementierung. Die o. g. Datei ist ausführlich kommentiert, so dass die Funktionsweise ohne größere Probleme nachvollzogen werden kann.

## Logiksignale

Die zur Applikation gehörenden Logiksignale sind in der Konfigurationsdatei icserver\_system.conf des Logik-Switches definiert. Es handelt sich um drei Gruppen von Signalen:

- Physikalische Ein- und Ausgänge des Hutschienenmoduls
- Von der TFT-Oberfläche verwendete Signale
- Logiksignale für die in der Oberfläche vorhandenen Konfigurationsoptionen

## Physikalische Ein- und Ausgänge

Der Programm-Ausschnitt in [Bild 24](#) zeigt die Logiksignale, die den Ein- und Ausgängen im Schaltplan entsprechen.

Die unter display\_name angegebenen beschreibenden Texte werden auf der Webseite für den Verdrahtungsplan angezeigt:

```
<!-- Door lock IOs -->
<value name=»IX100.0« alias=»binInKeyBell«
  display_name=»Key mounted outdoor to activate the door bell«/>
<value name=»IX100.1« alias=»binInKeyOpener«
  display_name=»Key mounted indoor for activating the door opener«/>
<value name=»QX100.0« alias=»binOutDoorBell«
  display_name=»Connection for the door bell«/>
<value name=»QX100.1« alias=»binOutDoorOpener«
  display_name=»Connection for the door opener«/>
<value name=»QX100.2« alias=»binOutDoorLedRed«
  display_name=»Red LED for Keypad«/>
<value name=»QX100.3« alias=»binOutDoorLedGreen«
  display_name=»Green LED for Keypad«/>
<loop start=»0« count=»10«>
  <value name=»IX[111+n/8].[n%8]« alias=»binInKey_[n]«
    display_name=»Keypad [n]«/>
</loop>
```

Bild 24: Die Definition der Logiksignale, die im Verdrahtungsplan angezeigt werden

```
<!-- Door lock UI values -->
<value name=»QB100« alias=»intUiDoorLogEvent« default_value=»0«/>
<value name=»IX100.2« alias=»binUiDoorOpener« default_value=»0«/>
```

Bild 25: Die Signale für die Oberfläche

```
<!-- Door lock configuration values -->
<value name=»IL100« alias=»intUiDoorPin« default_value=»0000«
  persistent=»true«/>
<value name=»IL101« alias=»intUiDoorOpenerTime« default_value=»3« unit=»s«
  persistent=»true«/>
<value name=»IL102« alias=»intUiDoorPinTimeout« default_value=»5« unit=»s«
  persistent=»true«/>
<value name=»IB100« alias=»intUiDoorPinDigits« default_value=»4«
  persistent=»true«/>
<value name=»IX101.0« alias=»binUiDoorAutoopen« default_value=»0«
  persistent=»true«/>
```

Bild 26: Die Signale für die über die TFT-Oberfläche zugänglichen Konfigurationseinstellungen



```

<device id=»DOORLOCK» type=»HMW-IO-12-Sw14-DR» display_name=»Door Lock Module»>
  <channel index=»1» terminal=»1.5»>
    <value id=»STATE» source=»binOutDoorBell»/>
  </channel>
  <channel index=»2» terminal=»2.1»>
    <value id=»STATE» source=»binOutDoorOpener»/>
  </channel>
  <channel index=»7» terminal=»1.2»>
    <value id=»STATE» source=»binOutDoorLedGreen»/>
    <configuration>
      <parameter name=»BEHAVIOUR» value=»DIGITAL_OUTPUT»/>
    </configuration>
  </channel>
[... ]
  <channel index=»15» terminal=»2.3»>
    <value id=»STATE» target=»binInKeyBell» bool_invert=»true»/>
    <configuration>
      <parameter name=»BEHAVIOUR» value=»DIGITAL_INPUT»/>
    </configuration>
  </channel>
[... ]

```

Bild 27: Auszug aus der Konfiguration der Ein- und Ausgänge des Hutschienenmoduls

`http://<IP der LCU>/wireplan.cgi`

Näheres dazu folgt im nächsten Artikel. Die Binäreingänge für die 10 Zifferntasten werden in der Schleife ganz unten erzeugt, um Schreibarbeit zu sparen.

### Signale für die Oberfläche (Bild 25)

Das Signal `binUiDoorOpener` wird von der Schaltfläche „Tür öffnen“ der Oberfläche angesteuert. Für das Signal `intUiDoorLogEvent` ist in der Oberfläche ein globaler Trigger gesetzt, über den der Logmechanismus implementiert wird. Die Implementierung des Logmechanismus wurde ja bereits im Abschnitt „Implementierung der Ereignisliste“ erläutert.

### Konfigurationssignale

Die in Bild 26 aufgeführten Signale entsprechen den über die TFT-Oberfläche zugänglichen Konfigurationseinstellungen.

- `intUiDoorPin` ist die Pin, bei der die Tür geöffnet wird. Diese ist als Dezimalzahl codiert.
- `intUiDoorOpenerTime` entspricht der Konfigurationseinstellung „Einschaltdauer“. Es ist die Zeit in Sekunden, für die der Türöffner angesteuert wird.
- `intUiDoorPinTimeout` entspricht der Konfigurationseinstellung „Pintimeout“. Es ist die Zeit in Sekunden, die bei der Pin-Eingabe zwischen zwei Tastendrücker maximal vergehen darf, bevor der Zustand **ERROR** eingenommen wird. Die gleiche Zeit wird verwendet für das Zurückschalten in den Zustand **IDLE** aus den Zuständen **ERROR** und **SUCCESS**.
- `intUiDoorPinDigits` entspricht der Konfigurationseinstellung „Pinlänge“ und gibt die Anzahl der Dezimalstellen der Pin an.
- `binUiDoorAutoopen` entspricht dem Zustand

der Checkbox „Klingeltaster öffnet Tür“. Ist diese gesetzt, wird beim Betätigen des Klingeltasters automatisch die Tür geöffnet.

### Die Definition des Schnittstellenmoduls

Ebenfalls in der bereits erwähnten Datei `icserver_system.conf` wird definiert, auf welchen Ein- und Ausgängen des Hutschienenmoduls die o. g. physikalischen Signale liegen. Diese Zuordnung geschieht im Abschnitt `<devices>` der o. g. Datei. Die Textbox in Bild 27 zeigt einen Auszug aus dieser Konfiguration.

So wird z. B. das Logiksignal `binOutDoorBell` dem Wert `STATE` des XML-RPC-Kanals 1 des Hutschienenmoduls zugeordnet. Dieser Kanal entspricht dem ersten Relaisausgang. Das Attribut `terminal` gibt die Bezeichnung des Anschlusses an, wie sie später auf der Oberflächenseite für den Verdrahtungsplan angezeigt wird. Da es sich um einen Ausgang handelt, wird der Name des Logiksignals im Attribut `source` angegeben. Beim Eingang für den Klingeltaster (Logiksignal `binInKeyBell`) hingegen handelt es sich um einen Eingang und daher ist das Logiksignal im Attribut `target` angegeben. Das Attribut `bool_invert="true"` an diesem Eingang sorgt für eine Invertierung des Signals. Das ist nötig, weil der Eingang des Hutschienenmoduls einen internen Pull-up-Widerstand hat und der Eingang in der von uns gewählten Beschaltung (siehe Schaltplan) bei der Tasterbetätigung auf „low“ gezogen wird.

Interessant sind noch die bei mehreren Werten vorhandenen Abschnitte `<configuration>`. Hier werden die angegebenen Kanalkonfigurationsparameter des entsprechenden XML-RPC-Kanals auf die angegebenen Werte gesetzt. Der Kanal 7 z. B. kann als Digitalausgang oder als Frequenzausgang konfiguriert werden. Durch die Angabe des Wertes `DIGI-`

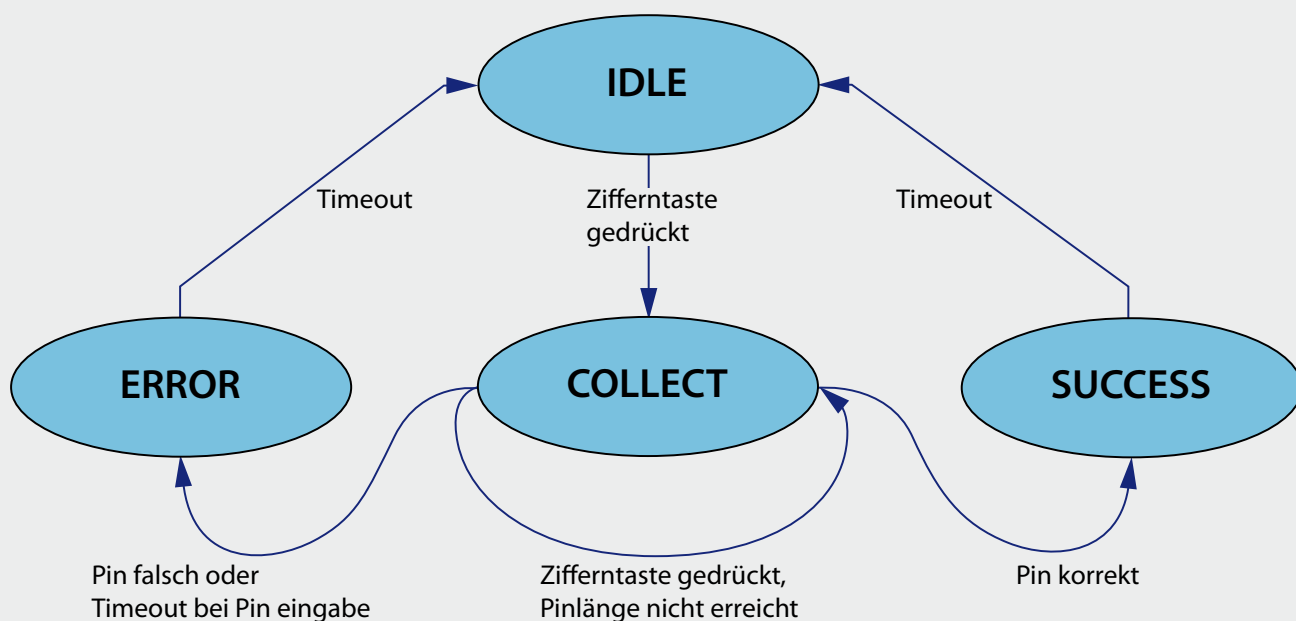


Bild 28: Das Zustands- und Ablaufdiagramm der Pin-Eingabe

TAL\_OUTPUT für den Kanalparameter BEHAVIOUR wird sichergestellt, dass dieser Kanal, wie für unsere Applikation nötig, als Digitalausgang konfiguriert ist. Details dazu finden sich in der Beschreibung des XML-RPC-Protokolls [1], [2] und [3], in der Beschreibungsdatei des entsprechenden Gerätes (in diesem Fall z. B. devicetypes/hs485types/hmw\_io12\_sw14\_dr.xml). Hilfreich ist auch die Konfigurationsoberfläche für XML-RPC-Geräte (siehe [1]).

Zu beachten ist die Tatsache, dass am obigen <device>-Tag gegenüber dem SYSTEM-Gerät (siehe vorangegangener Artikel) die Attribute interface und address fehlen. Diese Attribute können zu diesem Zeitpunkt noch nicht angegeben werden, weil die Adresse (Seriennummer) des später verwendeten Hutschienenmoduls noch nicht bekannt ist. Die Zuordnung zu einem bestimmten Modul erfolgt später über die Konfigurationswebseite:

`http://<IP der LCU>/busdevices.cgi`

### Pin-Eingabe

Die Grafik in Bild 28 zeigt das Zustandsdiagramm der Pin-Eingabe. Startzustand ist IDLE. Die Betätigung einer der Zifferntasten wechselt in den Zustand COLLECT, in dem die Tastendrucke für die eingegebene Pin gesammelt werden. Im Fehlerfall (zu lange Zeit zwischen zwei Tastendrucke oder falsche Pin) wird

in den Zustand ERROR gewechselt. Nach einer Wartezeit wird automatisch wieder der Zustand IDLE angenommen. Wurde die Pin korrekt eingegeben, wird in den Zustand SUCCESS und nach einer Wartezeit automatisch wieder in den Zustand IDLE gewechselt.

### LED-Signalisierung

Im Zustand IDLE sind beide LEDs aus. Im Zustand COLLECT blinken beide LEDs abwechselnd. Der Zustand SUCCESS wird durch Dauerleuchten der grünen LED angezeigt, der Zustand ERROR durch Dauerleuchten der roten LED.

### Auslösen des Türöffners

Der Türöffner kann durch vier verschiedene Ereignisse ausgelöst werden:

- Der Zustandsautomat der Pin-Eingabe wechselt in den Zustand SUCCESS
- Der interne Öffnungstaster (an 2.7) wird gedrückt
- Der Klingeltaster (an 2.3) wird gedrückt und in der Konfigurationsoberfläche ist die automatische Öffnung aktiviert
- In der Oberfläche wurde die Schaltfläche „Tür öffnen“ betätigt.

Im nächsten Teil der Artikelserie der LCU 1 wenden wir uns der Weboberfläche zu (für die zuständige Konfiguration). **ELV**



Weitere Infos:

- [1] ELVjournal 4/2010, S. 6 ff., Via Netzwerk auf HomeMatic® zugreifen – XML-RPC-Schnittstelle/HomeMatic®
- [2] <http://xmlrpc.scripting.com>
- [3] [www.homematic.com/index.php?id=156](http://www.homematic.com/index.php?id=156)