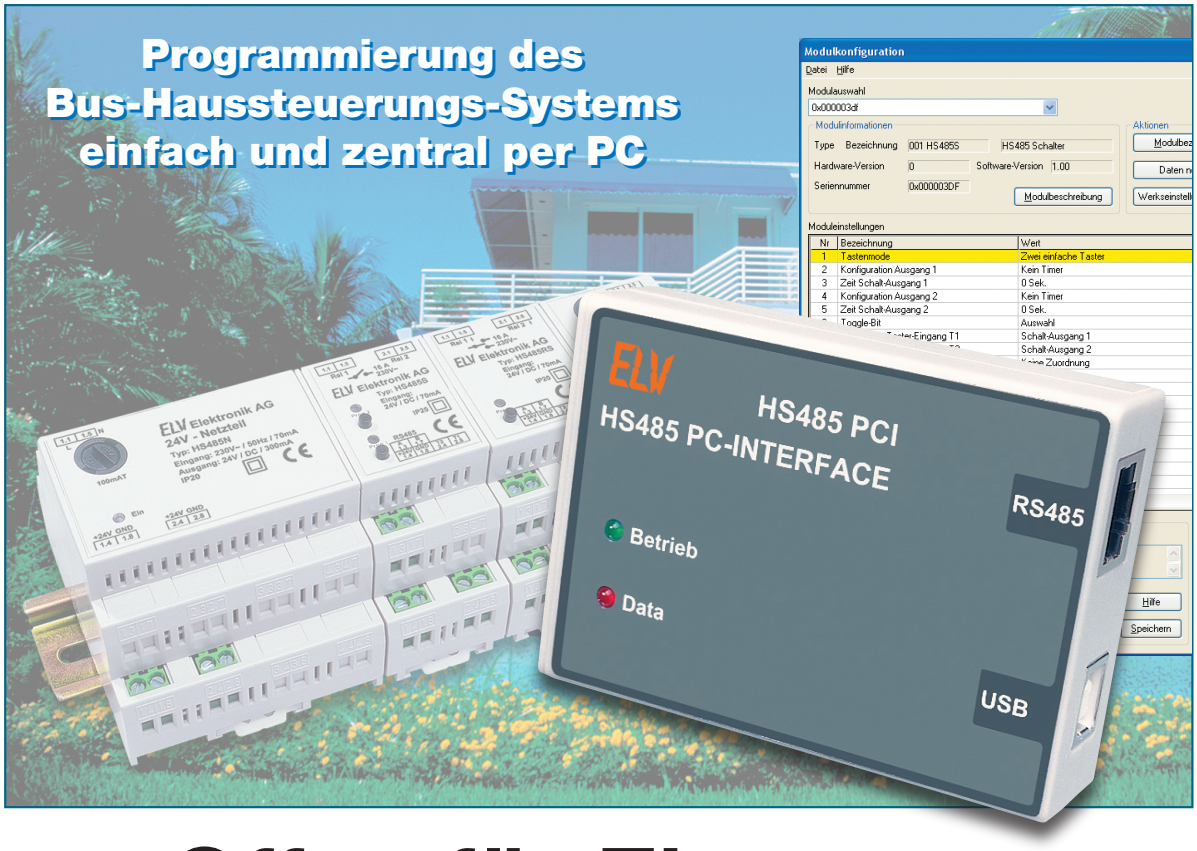

Offen für Eigenes – Protokollbeschreibung HS485

Das busorientierte HS485-Hausschaltsystem entfaltet seine volle Funktionalität erst durch die Programmierbarkeit von einem PC aus. Um Anwendern, die eigene Software-Lösungen hierzu anstreben, auch diesbezügliche Eigenentwicklungen zu ermöglichen, möchten wir Ihnen mit diesem Artikel das Steuerungsprotokoll des HS485-Systems vorstellen.

Offene Schnittstelle

Bereits nach Erscheinen des ELV-Funk-Haussteuerungs-Systems FS20 und des FHT-Heizungssteuerungs-Systems war bei Anwendern und Software-Entwicklern schnell der Ruf nach Offenlegung des Übertragungsprotokolls aufgetaucht. Aus verschiedenen Gründen erfolgte dies von ELV aus ausschließlich an lizenzierte Software-Entwickler. Beim neuen, drahtgebundenen HS485-Haussteuerungs-System gehen wir einen anderen Weg. Wir konzentrieren uns auf die Entwicklung, Fertigung und Erweiterung der System-Hardware und bieten derzeit eine einfache Software zur Konfiguration des Systems an. Natürlich entsteht bei vielen Anwendern sofort der Wunsch nach einem komfortableren PC-Frontend, etwa bei wechselnden Einsatzfällen, Einbindung von Sensoren oder anderen Anwendungswünschen. Um die Entwicklung einer eigenen Software zu erleichtern und eine normgerechte Anbindung der Hardware an die eigene Software zu gewährleisten, beschreiben wir hier das Protokoll der Software-Schnittstelle des Systems. Diese setzt den Einsatz des HS485 PCI voraus, da ein Teil der Kommunikation auf dem Vorhandensein dieses Moduls beruht.

Die Kommunikation im System

Grundlagen

Jedes HS485-Gerät am Bus (ausgenommen Busabschluss und Netzgerät) besitzt eine eindeutige, ab Werk fest vorgegebene 32-Bit-Adresse. Somit ist ausgeschlossen, dass zwei Geräte mit der gleichen Adresse am Bus betrieben werden können. Die Adresse 0xFFFFFFFF ist als so genannte Broadcast-Adresse reserviert und wird nicht an Module vergeben. Nachrichten, die an die Broadcast-Adresse gerichtet sind, werden von jedem Modul verarbeitet. Weiterhin sind die Adressen 0x00000000 und 0x00000001 fest vergeben. Sie sind für Logging- und Steuermodule, wie z. B. einen PC, reserviert.

Um einen sicheren Betrieb zu gewährleisten, ist es erforderlich, dass am Ende des Busses die A-Ader über einen Widerstand mit +5 V und die B-Ader ebenfalls über einen Widerstand mit Masse verbunden werden. Diese Aufgabe kann einer der Busabschluss-Bausteine HS485 BA oder HS485 BAplus übernehmen.

Datenübertragung

Die Datenübertragung erfolgt seriell über den RS485-Bus mit einer Datenübertragungsrate von 19.200 Bit/s mit 8 Datenbit, 1 Stoppbit und gerader Parität.

Jede Nachricht wird von dem angesprochenen Empfängermodul bestätigt. Falls keine Bestätigung kommt, erfolgt eine bis zu zweimalige Wiederholung der Nachricht. Pro Nachricht dürfen maximal 64 Byte an Nutzdaten übertragen werden.

Protokollrahmen

Jede Nachricht wird in einem Protokollrahmen übertragen, innerhalb dessen die Übertragung der Quell- und Zieladressen sowie weiterer Informationen erfolgt. Innerhalb des Rahmens werden die eigentlichen Nutzdaten übertragen. Jede gesendete Nachricht ist entsprechend der in Tabelle 1 gezeigten Struktur aufgebaut.

Startzeichen

Das Startzeichen ist immer 0xFD. Es ist ein Steuerzeichen und darf in den restlichen Daten der Nachricht nicht mehr vorkommen. Neben 0xFD gibt es noch 0xFE als weiteres Steuerzeichen. Dieses spielt bei der normalen Kommunikation keine Rolle, darf aber ebenfalls nicht im Datenstrom enthalten sein. Falls diese Steuerzeichen in den Daten enthalten sind, wird ein weiteres Steuerzeichen (0xFC) eingefügt, das vor ein Steuerzeichen zu setzen ist und selbstverständlich ebenfalls nicht im Datenstrom vorkommen darf. Dieses Steuerzeichen wird als Escape-Zeichen bezeichnet. Muss jetzt der Sender innerhalb der Daten eines dieser drei Steuerzeichen übertragen, wird das Steuerzeichen durch zwei Bytes ersetzt, und zwar durch das Escape-Zeichen, gefolgt von dem zu sendenden Steuerzeichen, bei dem das höchstwertige Bit gelöscht ist.

Beispiel:

Es sollen die Datenbytes 0x05 0xFD und 0xFA übertragen werden. Das erste Byte wird einfach übertragen, das zweite Byte entspricht einem Steuerzeichen und wird entsprechend ersetzt und das dritte Byte wird wieder normal übertragen.

Die Daten: 0x05 0xFD 0xFA werden also gesendet als 0x05 0xFC 0x7D 0xFA

Zieladresse

Die Übertragung der Zieladresse erfolgt im Big-Endian-Format. Das höchstwertige Byte wird als erstes übertragen, das niederwertigste Byte als letztes.

Kontrollzeichen

Das Kontrollzeichen gibt über den Typ der Nachricht Auskunft und enthält außerdem noch nachrichtenspezifische Bits. Der Aufbau des Kontrollbytes ist in der Tabelle 2 dargestellt. Die drei unterschiedlichen Nachrichtentypen werden anhand der markierten Bits unterschieden. Enthält das Kontrollbyte ein gesetztes B-Bit, so wird die 4 Byte große Absenderadresse eingefügt.

Die Bits haben folgende Bedeutung:

S – Sendefolgenummer: Die Sendefolgenummer setzt sich aus zwei Bits zusammen, die die Zahlen 0 bis 3 darstellen. Bei jeder erfolgreich versendeten Nachricht wird diese Nummer um 1 erhöht. Nach der Sendefolgenummer 3 beginnt die Zählung wieder bei 0. Muss die Nachricht erneut gesendet werden, weil Fehler bei der Übertragung aufgetreten sind, so wird die Sendefolgenummer erneut verwendet.

Kommen beim Empfänger zwei Nachrichten mit der gleichen Sendefolgenummer an, so werden beide bestätigt, aber nur eine verarbeitet.

R – Empfangsfolgenummer: Die Empfangsfolgenummer wird zur Bestätigung von Nachrichten benötigt. Erhält ein Modul eine Nachricht, so wird deren Empfang bestätigt. In der Bestätigungsnachricht entspricht die Empfangsfolgenummer der Sendefolgenummer der erhaltenen Nachricht.

Der Sender erkennt daran, dass die Nachricht erfolgreich an den Empfänger übertragen wurde.

Beispiel Protokollrahmen:

In diesem Beispiel wird ein Key-Event (Erklärung siehe Befehlsbeschreibung) von Adresse 0x2DE nach 0x1DA gesendet.

					Framelänge	Aktor		
FD	00 00 01 DA	1A	00 00 02 DE	06	4B	01	00	0C	F9 8E
	Zieladresse		Absenderadresse			Taste			Checksumme
Startzeichen		Kontrollzeichen			Befehl ('k')		Event		

Y – Synchronisationsbit: Wird in einer Nachricht das Synchronisationsbit beim Senden einer Nachricht gesetzt, so wird im Empfänger die Sendefolgenummer zurückgesetzt. Die Empfangsfolgenummer wird auf den Wert der Sendefolgenummer der Nachricht gesetzt.

Danach wird die Nachricht bestätigt und verarbeitet. Dabei ist zu beachten, dass jede Nachricht mit gesetztem Y-Bit verarbeitet wird, also auch wiederholte Nachrichten.

B – Absenderadresse: Ist das B-Bit gesetzt, so wird direkt nach dem Kontrollbyte die Absenderadresse eingefügt.

Die Absenderadresse ist während der Kommunikation immer erforderlich, da zur Bestätigung von Nachrichten die Absenderadresse bekannt sein muss.

Es gibt jedoch einige spezielle Befehle, in denen die Absenderadresse nicht notwendig ist.

F – letztes Paket: Ist ein Datensatz zu groß für eine Nachricht, so kann die Nachricht aufgeteilt werden. Die letzte Nachricht des Datensatzes wird dabei mit einem gesetzten F-Bit gesendet.

Die zur Zeit verfügbaren Module unterstützen keine geteilten Nachrichten. Aus diesem Grund ist das F-Bit immer zu setzen.

M – Adressmaske: Wird eine Discovery-Nachricht versendet, so entsprechen die 5 M-Bit der Adressmaske. Sie gibt an, wie viele Bits (M+1) der Empfängeradresse gewertet werden sollen.

Absenderadresse

Die Übertragung der Absenderadresse erfolgt wieder im Big-Endian-Format. Das höchstwertige Byte wird als erstes übertragen, das niederwertigste Byte zuletzt.

Tabelle 1: Protokollrahmen

Anzahl Bytes	Beschreibung
1	Startzeichen
4	Zieladresse
1	Kontrollzeichen
4	Absenderadresse
1	Framelänge inkl. Checksumme
N	Framedaten
2	CRC16-Checksumme

Framelänge

Danach folgt die Framelänge. Sie enthält die Anzahl der Datenbytes zuzüglich zwei Bytes für die Checksumme.

Framedaten

Nach der Framelänge werden die Framedaten übertragen. Pro Nachricht dürfen nicht mehr als 64 Byte Framedaten übertragen werden. Die Framedaten entsprechen den Nutzdaten der Nachricht.

Checksumme

Schließlich folgt die 2 Byte lange CRC16-

Tabelle 2: Aufbau Kontrollzeichen

Type / Bit	7	6	5	4	3	2	1	0
I-Nachricht	Y	R	R	F	B	S	S	0
ACK-Nachricht	0	R	R	1	B	0	0	1
Discovery	M	M	M	M	M	0	1	1

Beispiel Checksumme:

FE 00 98 00 07 80 xx xx xx xx YY YY
- YY YY: Checksumme
- xx xx xx xx: gefundene Adresse
- 80: Befehl
- 07: Länge
- 00: Absenderadresse
- 98: Kontrollbyte
- 00: Zieladresse
- FE: Startzeichen

Checksumme. Sie wird mit dem Polynom 0x1002 nach dem allgemein bekannten Berechnungsverfahren berechnet.

Nachrichtentypen

Die im Kontrollbyte definierten unterschiedlichen Nachrichtentypen haben folgende Bedeutung:

I-Nachricht

Soll ein Datenaustausch zwischen den Modulen erfolgen, so wird eine I-Nachricht verwendet. Enthält die gesendete Nachricht eine Abfrage an das Modul, so wird mit einer I-Nachricht geantwortet. Diese Antwort enthält bereits die Bestätigung der vorherigen Nachricht.

ACK-Nachricht

Enthält ein Modul eine Nachricht, auf die es keine Antwort senden muss, bestätigt es diese Nachricht mit einer ACK-Nachricht.

Discovery-Nachricht

Um festzustellen, welche Module am Bus angeschlossen sind, sendet der PC Discovery-Nachrichten aus. Alle Module vergleichen mit Hilfe der Adressmaske die Zieladresse mit ihrer eigenen Adresse. Die Adressmaske bestimmt, wie viele Bits gültig sind, beginnend beim höchstwertigen Bit. Stellt der Modul-Controller eine Übereinstimmung fest, dann sendet er ein 0xF8. Der PC erkennt dies und stellt dadurch fest, dass sich noch weitere Module mit dieser Adressmaske am Bus befinden, und passt die Adressmaske und die Zieladresse an. Dies wird so lange durchgeführt, bis alle Module am Bus gefunden sind. Die Discovery-Funktion ist im HS485-PC-Interface implementiert und kann daher einfach aufgerufen werden.

Um unterscheiden zu können, welche Informationen für das HS485 PCI und welche für die Module am Bus sind, wird im HS485 PCI das Startzeichen überprüft. Ist dies 0xFD, so ist die Nachricht für die Module am Bus, beim Startzeichen 0xFE für das PC-Interface.

Der Aufruf zum Starten des Discovery-Mode sieht wie folgt aus:

FE 04 01 AE DO
- FE: Startzeichen
- 04: Länge
- 01: Befehl
- AE DO: Checksumme

Die einzelnen Bytes haben folgende Bedeutung:

Startzeichen: Bei Nachrichten an das HS485 PCI ist das Startzeichen immer 0xFE.

Länge: Die Länge enthält die Anzahl der Datenbytes zuzüglich zwei Bytes für die Checksumme.

Befehl: Der Befehl für den Discovery-Mode ist 0x01.

Checksumme: Die CRC16-Checksumme ist 2 Byte lang. Sie wird wieder mit dem Polynom 0x1002 nach dem allgemein bekannten Berechnungsverfahren berechnet.

Nun fängt das HS485 PCI an, den Bus nach angeschlossenen Modulen zu durchsuchen. Bei jedem gefundenen Modul sendet es eine Nachricht an den PC. Die Antwortnachricht besteht im Prinzip aus einem I-Block, allerdings mit verkürzten Adressen. Sie bestehen nur aus einem Byte.

Bis auf Ziel- und Absenderadresse entsprechen alle Bezeichnungen den bereits beschriebenen. Der Befehl ist hierbei mit 0x80, Ziel- und Absenderadresse mit 0x00 und das Kontrollzeichen mit 0x98 festgelegt. Ist das Durchsuchen des Busses nach neuen Modulen abgeschlossen, wird eine Nachricht mit der Adresse 0xffffffff gesendet.

Tabelle 3: Befehlsübersicht

Befehl	Hex-Code	Beschreibung
K	0x4B	Key-Event
s	0x73	Aktor setzen
S	0x53	Aktorzustand abfragen
h	0x68	Modultyp und Hardware-Version abfragen
v	0x76	Firmware-Version des Gerätes anfragen
!	0x21	führt einen Reset des Moduls durch
C	0x43	Konfiguration des Moduls neu lesen
R	0x52	EEPROM lesen
W	0x57	EEPROM schreiben
q	0x71	Zieladresse hinzufügen
c	0x63	Zieladresse löschen

Befehlssatz

Um die Befehle verstehen zu können, ist es notwendig, einige Grundlagen über die Module zu kennen. Jedes Modul besitzt einen Controller mit integriertem EEPROM-Speicher. In diesem Speicher wird die Konfiguration der Module abgelegt. Jeder Eingang und Ausgang besitzt innerhalb des Moduls eine eindeutige Nummer. Wird ein Taster an einem Modul betätigt, so wird das EEPROM nach möglichen Zielaktoren durchsucht. Sind ein oder mehrere Aktoren gefunden, so wird eine Nachricht an die Aktoren der jeweiligen Module gesendet.

Die Steuerung von Modulen erfolgt mit nur wenigen einfachen Befehlen. Die Tabelle 3 zeigt eine Liste der wichtigen Befehle. Das Byte mit dem Befehl steht immer an der ersten Stelle der Framedaten.

Auf die Beschreibung evtl. vorhandener modulspezifischer weiterer Befehle wird an dieser Stelle verzichtet. Diese sind der jeweiligen Bedienungsanleitung zu entnehmen.

Beschreibung der Befehle

„s“ – Aktor setzen

Diese Funktion setzt den Zustand eines Modul-Ausgangs. Da es unterschiedliche Module gibt, müssen die übertragenen Daten an das Modul angepasst sein. Dies wird im Allgemeinen nur von einem PC durchgeführt.

Für jedes bisher verfügbare Modul gilt folgender Nachrichtenaufbau:

1. Befehlsbyte „s“
2. Nummer des Sensoreingangs
3. Nummer des Zielaktors
4. Aktion

Jeder Aktor beim HS485 S kann den Zustand „Aus“ (0x00) oder „An“ (0x01) annehmen. Mit 0xFF kann man den Zustand wechseln (toggeln). Beim HS485 RS sind die Zustände „Runter“ (0x20), „Hoch“ (0x10), „Aus“ (0xFE) oder „Auf Schlitz fahren“ (0xFF) möglich.

Beim Dimmer HS485 D sind die Werte 0 bis 16 zum direkten Setzen des Helligkeitswertes einzusetzen. Zusätzlich kann heruntergedimmt (0x11), heraufgedimmt (0x12), herauf- bzw. heruntergedimmt (im „Kreis“) (0x13), an- bzw. ausgeschaltet (0x14) oder mit dem alten Helligkeitswert (0x15) angeschaltet werden. Der HS485 IO4 UP verhält sich bei entsprechender Konfiguration wie der Schalter HS485 S.

„S“ – Aktorzustand abfragen

Der Befehl „S“ gefolgt von der Aktornummer sendet den jeweiligen Zustand. Als Antwort wird zunächst die Aktornummer im Datenbyte 0 wiederholt. Im Datenbyte 1 steht der Aktorzustand. Für einige Geräte kann auch der Zustand eines Eingangs abgefragt werden. Dies ist zum Beispiel bei Schalteingängen sinnvoll.

„h“ – Modultyp und Hardware-Version abfragen

Jedes Modul am Bus verfügt über Informationen zu seinem Hardware-Typ und der Hardware-Version. In Tabelle 4 sind die bisher eingesetzten Module mit ihrem Hardware-Typ aufgelistet.

Als Antwort auf einen „h“-Befehl werden die Informationen zu Hardware-Typ und die Hardware-Version gesendet. Hardware-Typ und -Version benötigen jeweils ein Byte.

„v“ – Firmware-Version

Neben der Hardware-Version besitzt jedes Modul auch eine Firmware-Version. Diese setzt sich aus zwei Bytes zusammen. Das erste Byte enthält die Vorkommastelle und das zweite die Nachkommastelle.

„!“ – Modulreset durchführen

Mit dem Befehl „!“ kann man einen Neustart eines Moduls erzwingen. Damit nicht versehentlich ein Modul neu gestartet wird, muss das zweite Byte des Nachrichtenframes ebenfalls ein „!“ enthalten.

„C“ – Konfiguration neu lesen

Jedes Modul lässt sich konfigurieren. Die Konfigurationsparameter werden im EEPROM auf dem Mikrocontroller gespeichert. Da sich nicht alle Änderungen im EEPROM direkt auf die Funktion auswirken, ist in einigen Fällen ein erneutes Auslesen der Konfiguration erforderlich.

„W“ – EEPROM schreiben

Nicht nur das Lesen des EEPROMs ist möglich, sondern auch das Schreiben. Hier kann allerdings eine falsche Parametrierung eine Fehlfunktion des Moduls verursachen. Die Module können jedoch nicht zerstört werden. Dem Befehl folgt eine 2 Byte lange Startadresse und ein Byte für die Anzahl der Datenbytes. Danach folgen die eigentlichen Daten. Da das Schreiben in das EEPROM einige Zeit dauert, sollte die maximale Anzahl an EEPROM-Daten pro Nachricht 32 Byte nicht überschreiten.

Jedes Modul kann auf die Werkseinstellung zurückgesetzt werden, indem das gesamte EEPROM mit 0xFF gefüllt wird. Danach ist ein Modulreset erforderlich. Der Aufbau des EEPROMs der unterschiedlichen HS485-Module kann aus den XML-Files der Konfigurations-Software des HS485 PCI entnommen werden. Sie stehen im Unterverzeichnis „XML-Dokumente“. Dort können Sie alle nötigen Informationen entnehmen, um das EEPROM richtig zu beschreiben.

Tabelle 4: Bisher verfügbare Module mit Hardware-Typ und EEPROM-Größe

Hardware-Typ	Geräte	EEPROM
0	HS485 D – Dimmer	512 Byte
1	HS485 S – Schalter	512 Byte
2	HS485 RS – Rollladenschalter	512 Byte
4	JCU10 TFS – Temperatur-Feuchte-Sensor	512 Byte
5	HS485 IO4 UP – 4fach-I/O-Modul	512 Byte

„R“ – EEPROM lesen

Wie bereits beschrieben, befinden sich die Konfigurationsparameter im EEPROM. Dieses kann mit dem „R“-Befehl ausgelesen werden. Maximal sind 64 Byte an EEPROM-Daten mit einem Befehl auslesbar. Die Größe des EEPROMs der Module ist Tabelle 4 zu entnehmen. Dem „R“-Befehl folgt die 2 Byte lange Startadresse (wieder im Big-Endian-Format) und ein Byte für die Anzahl der Datenbytes. Als Antwort wird dann der EEPROM-Inhalt gesendet.

Befehle ohne PC

Die folgend beschriebenen Befehle werden während der Programmierung ohne PC zwischen den Modulen ausgetauscht. Dazu zunächst einige Informationen über den Ablauf der Programmierung.

An einem Modul wird ein Schaltausgang über die Programmiertaste in den Programmiermodus gebracht. Das Modul lauscht jetzt auf dem Bus nach Key-Events, die an die Broadcast-Adresse gerichtet sind. Wird jetzt an einem beliebigen anderen Modul am Bus eine Taste betätigt, werden Key-Events an alle bisher programmierten Aktoren und danach an die Broadcast-Adresse gesendet.

Das Modul, welches sich im Programmiermodus befindet, empfängt diesen Broadcast und sendet jetzt einen „q“-Befehl, um sich beim sendenden Modul zu registrieren. Danach wird auch der gerade programmierte Aktor bei jedem Tastendruck angesprochen.

Beim Löschen eines Aktors („Trennen“ von einer zugewiesenen Taste) wird der „c“-Befehl gesendet. Dadurch wird die Programmierung aufgehoben.

„K“ – Key-Event

Das Key-Event wird bei jedem Drücken und Loslassen eines an einem Modul angeschlossenen Tasters gesendet. Wird ein Taster länger betätigt, so wird in festen Zeitabständen erneut ein Key-Event übertragen. Nachrichten an die Broadcast-Adresse werden mit dem Zielaktor „0“ versendet.

Es werden folgende Daten gesendet

1. Befehlsbyte „K“
2. Nummer des Sensoreingangs
3. Nummer des Zielaktors
4. Event

Die Bits im Event geben Informationen über das Tasten-Ereignis an:

Bit	0	1	2	3	4	5	6	7
	R	R	Y	Y	T	T	E	E

„E“ entspricht dem Tasten-Event.

0 0	Taste gedrückt
0 1	Taste gehalten
1 0	Taste losgelassen
1 1	Reserve

„T“ wird bei jedem Loslassen der Taste um eins erhöht

„Y“ gibt den Typ der Taste an

0 0	Toggle-Taste
0 1	Hoch/An-Taste
1 0	Runter/Aus-Taste
1 1	Reserve

„R“ ist für zukünftige Anwendungen reserviert und sollte nicht genutzt werden.

„q“ – Zieladresse hinzufügen

Jedes Modul besitzt eine unterschiedliche Anzahl an Eingängen. Um diese Eingänge mit den Ausgängen anderer Module zu verknüpfen, müssen Zieladresse und Zielaktor im Modul gespeichert werden. Dies kann entweder direkt mit EEPROM-Schreibzugriffen durchgeführt werden oder mit dem „q“-Befehl. Dazu wird mit dem „q“-Befehl auch die Nummer des Eingangs und des Aktors, der programmiert werden soll, mitgesendet.

„c“ – Zieladresse löschen

Soll die Zieladresse gelöscht werden, so wird anstatt des „q“ ein „c“-Befehl gesendet. Hier müssen ebenfalls die Nummer des Eingangs und der Aktor, der gelöscht werden soll, mitgesendet werden.

Wie gesagt, mit diesen wenigen Befehlen und der Einhaltung des vorgegebenen Datenprotokolls ist die Kommunikation mit den Modulen des HS485-Systems bzw. der Module untereinander sowie deren Programmierung möglich. Dem Programmierer sind damit alle relevanten Daten für das Erstellen eigener Software zugänglich. Im Internet auf unserer Download-Seite finden Sie ein Demoprogramm mit Quellcode in C.