



Transponderleser

LEGIC



Protokoll-Dokumentation

Version 3.0

Stand: 02 / 2007

Inhaltsverzeichnis:

Kapitel	Seite
1. Physikalischer Anschluss	3
2. Übertragungsparameter	3
3. Kommandos	4
3.1. Aufbau der Kommandostruktur	4
3.2. Übersicht Kommandocodes	5
3.3. Übersicht Fehlercodes	6
4. Detaillierte Kommandoübersicht	7
4.1. Standard-Kommandos	7
4.1.1. Kommando <31h> Read Data, unsegmentierte Medien	7
4.1.2. Kommando <32h> Read Data, segmentierte Medien	9
4.1.3. Kommando <33h> Write Data, unsegmentierte Medien	11
4.1.4. Kommando <34h> Write Data, segmentierte Medien	13
4.1.5. Kommando <36h> Launch oder De-Launch	15
4.1.6. Kommando <50h> Activate LED	16
4.1.7. Kommando <55h> Read Unique-ID	17
4.2. Kommandos für "Continuous Modus"	18
4.2.1. Kommando <60h> Leser parametrieren	18
4.2.2. Ausgabeparameter für Kommando <60h>	19
4.2.3. Kommando <61h> Parameter auslesen	21
4.2.4. Kommando <65h> Firmwarestand auslesen	22
4.2.5. Beschreibung Ausgabeformate	23
4.3. Der KABA-Group-Header	26
4.3.1. Kommando <40h> Build KGH, unsegmentierte Medien	26
4.3.2. Kommando <41h> Build KGH, segmentierte Medien	27
4.3.3. Kommando <42h> Build CRC, unsegmentierte Medien	28
4.3.4. Kommando <43h> Build CRC, segmentierte Medien	29
4.3.5. Kommando <44h> Read KGH, unsegmentierte Medien	31
4.3.6. Kommando <45h> Read KGH, segmentierte Medien	32
4.3.7. Kommando <46h> Read CRC, unsegmentierte Medien	34
4.3.8. Kommando <47h> Read CRC, segmentierte Medien	36
5. Impressum	38

1.) Physikalischer Anschluss

Der Transponderleser "LEGIC" ist in zwei Schnittstellenvarianten lieferbar. Er wird mit RS232-Schnittstelle oder mit USB-Schnittstelle geliefert.

USB-Variante:

Bei dieser Variante wird der Leser über den USB-Port des IPC oder Ihres Standard-PC's mit Spannung versorgt.

Zur Kommunikation mit dem Leser wird ein spezieller Treiber benötigt, der schon fertig installiert ist oder im technischen Download-Bereich der noax-Homepage (www.noax.com) erhältlich ist.

Er bindet den Leser per sog. "virtuellem COM-Port" ein, d.h. der Transponderleser kann so genutzt werden, als ob er an einem RS232-Anschluß des PCs angeschlossen wäre.

Dies ist ab dem Betriebssystem Windows 98 SE oder neuer möglich. Windows NT4 besitzt keine USB-Funktionalität und wird daher nicht unterstützt. Auch Linux wird unterstützt

RS232-Variante an der Steckerabdeckung:

In dieser Variante wird der Leser über eine spezielle eingestellte RS232-Schnittstelle des IPC mit Spannung versorgt. An Pin 4 der Schnittstelle liegen +5V DC an. Dieser Transponderleser kann daher nicht ohne ein Anschluss-Set an einem Standard-PC betrieben werden!

RS232-Variante als Tischgerät:

Zum Betrieb des RS232-Tischgerätes wird ein Anschluss-Set benötigt. (Art. 11873) Diese Anschluss-Set wird an eine serielle Schnittstelle und an einen USB-Anschluss des Standard-PC angeschlossen. Über den USB-Anschluss wird der Leser mit Spannung versorgt.

2.) Übertragungsparameter

Die Befehlsübertragung zwischen dem noax IPC oder Ihrem Standard-PC und dem Transponderleser basiert auf einem binären Protokoll.

Die serielle Schnittstelle muss auf folgende Parameter eingestellt werden:

Baudrate:	9600 baud
Parität:	none
Datenlänge:	8 bit
Stopbits:	1 bit

Die Befehlsdaten werden als hexadezimale Werte übertragen und sind mit einer Checksumme abgeschlossen. Die **Checksumme** wird gebildet, in dem alle **Datenbytes mit EXOR verknüpft** werden. Wenn nach einer maximalen Zeit von 1,5 Sekunden nach einem Kommando an den Leser keine Reaktion erfolgt, tritt ein Timeout-Fehler auf. Die Reaktionszeit ist abhängig vom Kommando, dem Transponder und der Datenmenge. Bei den Kommandos <Launch> und <De-Launch> beträgt die Timeout-Zeit 15 Sekunden.

Über den Befehl <Leser parametrieren> kann der Leser in den "Continuous Modus" versetzt werden und es können verschiedene Ausgabeformate eingestellt werden. Siehe dazu Punkt 4.2. ff

3.) Kommandos

3.1.) Aufbau der binären Kommandostruktur

Die Kommandostruktur basiert auf einem binären Protokoll.
Alle Werte werden generell als hexadezimale Zeichen übermittelt.

Das Kommando an den Leser setzt sich wie folgt zusammen:

<i>Kommando zum Leser</i>	<i>nb cc pp cs</i>
---------------------------	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nb</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>cc</i>	1 Byte	<i>Kommandocode</i>
<i>pp</i>	variabel	<i>Parameter des Kommandos</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR der gesendeten Bytes (ohne cs) ermittelt

<i>Antwort vom Leser</i>	<i>nb cc dd cs</i>
--------------------------	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nb</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>cc</i>	1 Byte	<i>Wiederholung des Kommandocodes oder Kommandocode für Fehler</i>
<i>dd</i>	variabel	<i>Datenbytes oder Fehlercode (siehe Tabelle Seite 6)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR der übertragenen Bytes (ohne cs) ermittelt

Beispiele für die Checksummen-Ermittlung:

Kommando zum Leser:

Lesen der Unique-ID

Checksummenbildung:

Byte 1: 02h → 0000 0010 binär
 Byte 2: 55h → 0101 0101 binär
 EXOR-Ergebnis: → 0101 0111 binär → 57h

025557h

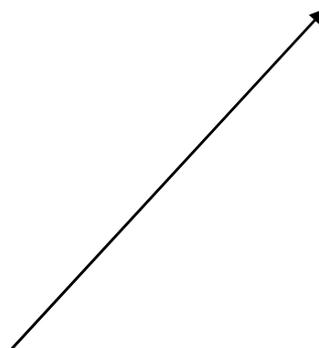


Antwort vom Leser:

Checksummenbildung:

Byte 1: 06h → 0000 0110 binär
 Byte 2: 55h → 0101 0101
 EXOR-Ergebnis: → 0101 0011
 Byte 3: 24h → 0010 0100
 EXOR-Ergebnis: → 0111 0111
 Byte 4: F4h → 1111 0100
 EXOR-Ergebnis: → 1000 0011
 Byte 5: 18h → 0001 1000
 EXOR-Ergebnis: → 1001 1011
 Byte 6: 3Bh → 0011 1011
 EXOR-Ergebnis: → 1010 0000 → A0h

065524F4183BA0h



3.2.) Übersicht der Kommandocodes:

Kommando	Funktion
30h	<i>Error (nur als Antwort vom Leser)</i> Ein Fehler ist aufgetreten. Fehlercodes siehe Tabelle unter 3.3.
31h	<i>Read Data</i> bei unsegmentierten LEGIC-Transpondern
32h	<i>Read Data</i> bei segmentierten LEGIC-Transpondern
33h	<i>Write Data</i> bei unsegmentierten LEGIC-Transpondern
34h	<i>Write Data</i> bei segmentierten LEGIC-Transpondern
36h	<i>Launch</i> oder <i>De-Launch</i> Funktion abhängig vom aktuellen Status des Lesers
50h	<i>Activate LED</i> Ansteuern der angeschlossenen LED
55h	<i>Read Unique ID</i> Lesen der Unique-ID des Transponders
60h	<i>Leser parametrieren</i> Lesers für "Continuous Modus" einstellen
61h	<i>Parameter auslesen</i> Die Parametereinstellung des Lesers auslesen
65h	<i>Firmwarestand auslesen</i> Den Firmwarestand des Lesers auslesen
40h	<i>Build KGH</i> bei unsegmentierten LEGIC-Transpondern Erzeugen eines KABA-Group-Headers
41h	<i>Build KGH</i> bei segmentierten LEGIC-Transpondern Erzeugen eines KABA-Group-Headers
42h	<i>Build CRC</i> bei unsegmentierten LEGIC-Transpondern Erzeugen einer Checksumme über einen bestimmten Bereich
43h	<i>Build CRC</i> bei segmentierten LEGIC-Transpondern Erzeugen einer Checksumme über einen bestimmten Bereich
44h	<i>Read KGH</i> bei unsegmentierten LEGIC-Transpondern Lesen des KABA-Group-Headers
45h	<i>Read KGH</i> bei segmentierten LEGIC-Transpondern Lesen des KABA-Group-Headers
46h	<i>Read CRC Block</i> bei unsegmentierten LEGIC-Transpondern Lesen der Checksumme eines Blocks
47h	<i>Read CRC Block</i> bei segmentierten LEGIC-Transpondern Lesen der Checksumme eines Blocks

3.3.) Übersicht der Fehlercodes:

Kommando	Funktion
31h	<i>Unbekanntes Kommando</i>
32h	<i>Checksumme falsch</i> Es ist ein Übertragungsfehler aufgetreten
33h	<i>SSC nicht gefunden</i> Byte 9 bei unsegmentierten LEGIC-Transpondern Byte 6 im Segment bei segmentierten LEGIC-Transpondern
34h	<i>OEM-Nummer nicht gefunden</i> Byte 10 bei unsegmentierten LEGIC-Transpondern Byte 7 im Segment bei segmentierten LEGIC-Transpondern
35h	<i>Segment nicht gefunden</i> Nur bei segmentierten LEGIC-Transpondern
36h	<i>Kartenfehler</i>
37h	<i>Transponder nicht verfügbar</i> Es befindet sich kein Transponder im RF-Feld des Lesers
39h	<i>Unsegmentierter Transponder wird benötigt</i>
40h	<i>Timeout LEGIC-Modul (ca. nach 1 Sekunde)</i>
41h	<i>Launch oder De-Launch ausgeführt</i>
42h	<i>Kein Launch- oder De-Lauch-Transponder</i>
43h	<i>Timeout des Interface</i>
50h	<i>Fehler beim Erzeugen des KGH-Headers</i> Bei unsegmentierten LEGIC-Transpondern
51h	<i>Fehler beim Erzeugen des KGH-Headers</i> Bei segmentierten LEGIC-Transpondern
52h	<i>Fehler beim Erzeugen der CRC</i> Bei unsegmentierten LEGIC-Transpondern
53h	<i>Fehler beim Erzeugen der CRC</i> Bei segmentierten LEGIC-Transpondern
54h	<i>Fehler beim Lesen des KGH-Headers</i> Bei unsegmentierten LEGIC-Transpondern
55h	<i>Fehler beim Lesen des KGH-Headers</i> Bei segmentierten LEGIC-Transpondern
56h	<i>Fehler beim Lesen des CRC-Blocks</i> Bei unsegmentierten LEGIC-Transpondern
57h	<i>Fehler beim Lesen des CRC-Blocks</i> Bei segmentierten LEGIC-Transpondern
60h	<i>Fehler bei Parametrierung</i>
61h	<i>Ungültige Parameternummer</i>
62h	<i>Falscher Parameterwert</i> Bei Einschränkungen eines bestimmten Parameters

4.) Detaillierte Kommandobeschreibung

4.1.) Standard-Kommandos

4.1.1.) Kommando <31h> - Read Data, unsegmentierte Medien

Es wird ab der angegebenen Startadresse im Anwenderdatenspeicher des LEGIC-Transponders die angegebene Anzahl von Bytes (max. 32 Byte) gelesen. Vor dem Lesevorgang wird, wenn gefordert, eine Überprüfung des SSC (System-Supplier-Code) und / oder der OEM-Nummer durchgeführt.

<i>Kommando zum Leser</i>	<i>07 31 p1 p2 p3 p4 p5 cs</i>
---------------------------	--------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der zu lesenden Datenbytes</i> maximal 32 Bytes → mögliche Werte von 01h bis 20h
<i>p4</i>	1 Byte	SSC (SSC = Byte 9) angegebener Wert wird mit SSC-Byte verglichen bei FFh → keine Überprüfung
<i>p5</i>	1 Byte	OEM-Nummer (OEM-Nummer = Byte 10) angegebener Wert wird mit OEM-Nummer verglichen bei FFh → keine Überprüfung
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>nn 31 pd cs</i>
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>pd</i>	variabel	<i>Datenbytes</i>
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

Beispiele: siehe nächste Seite

Beispiele zum Befehl <31h>- Read Data, unsegmentierte Medien:

Beispiel 1:

Kommando zum Leser: **0731000105FFFF32h**

Lese 5 Bytes vom Anfang des Datenspeichers
ohne SCC-Check
ohne OEM-Nummer-Check

Antwort vom Leser bei Erfolg: **073108154711007Dh**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 07h
Kommandowiederholung: 31h
Gelesene Datenbytes: 0815471100h
Checksumme: 7Dh

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 03h
Kommando: 30h → Fehler
Fehlercode: 37h → Kein Transponder
Checksumme: 04h

Beispiel 2:

Kommando zum Leser: **0731000A032C5546h**

Lese 3 Bytes ab Adresse 0Ah des Datenspeichers
SCC-Check auf phg SSC (= 2Ch)
OEM-Nummer-Check mit Nummer 55h

Antwort vom Leser bei Erfolg: **053101020334h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 05h
Kommandowiederholung: 31h
Gelesene Datenbytes: 010203h
Checksumme: 34h

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 03h
Kommando: 30h → Fehler
Fehlercode: 37h → Kein Transponder
Checksumme: 04h

4.1.2.) Kommando <32h> - Read Data, segmentierte Medien

Es wird in einem angegebenen Segment des LEGIC-Transponders ab der angegebenen Startadresse die angegebene Anzahl von Bytes (max. 32 Byte) gelesen oder es wird nach dem angegebenen SSC-Byte und / oder der angegebenen OEM-Nummer gesucht und von dort ab die angegebene Anzahl von Bytes gelesen.

<i>Kommando zum Leser</i>	08 32 p1 p2 p3 p4 p5 p6 cs
---------------------------	-----------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der zu lesenden Datenbytes</i> maximal 32 Bytes → mögliche Werte von 01h bis 20h
<i>p4</i>	1 Byte	<i>Segment-Nummer</i> ab dieser Segmentnummer wird nach SSC und/oder OEM-Nummer gesucht. Wenn SSC = FFh und/oder OEM-Nummer = FFh wird im angegebenen Segment gelesen
<i>p5</i>	1 Byte	SSC (SSC = Byte 6 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>p6</i>	1 Byte	OEM-Nummer (OEM-Nummer = Byte 7 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	nn 32 pd cs
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>pd</i>	variabel	<i>Datenbytes vom Leser</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiele: siehe nächste Seite

Beispiele zum Befehl <32h>- Read Data, segmentierte Medien:

Beispiel 1:

Kommando zum Leser: **083200010503FFFF3Dh**
Lese 5 Bytes vom Anfang des Segments 3
ohne SCC-Check
ohne OEM-Nummer-Check

Antwort vom Leser bei Erfolg: **073208154711007Eh**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 07h
Kommandowiederholung: 32h
Gelesene Datenbytes: 0815471100h
Checksumme: 7Eh

Antwort vom Leser bei Fehler: **03303704h**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 03h
Kommando: 30h → Fehler
Fehlercode: 37h → Kein Transponder
Checksumme: 04h

Beispiel 2:

Kommando zum Leser: **0832000A03042C554Eh**
Suche ab Segment 4 in den folgenden Segmenten nach nach phg SSC (= 2Ch) und OEM-Nummer 55h. Bei Erfolg lese 3 Bytes ab Adresse 0Ah des gefundenen Segments.

Antwort vom Leser bei Erfolg: **053201020337h**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 05h
Kommandowiederholung: 32h
Gelesene Datenbytes: 010203h
Checksumme: 37h

Antwort vom Leser bei Fehler: **03303704h**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 03h
Kommando: 30h → Fehler
Fehlercode: 37h → Kein Transponder
Checksumme: 04h

4.1.3.) Kommando <33h> - Write Data, unsegmentierte Medien

Es wird ab der angegebenen Startadresse im Anwenderdaten-speicher des LEGIC-Transponders die angegebene Anzahl von Bytes (max. 32 Byte) geschrieben. Vor dem Schreibvorgang wird, wenn gefordert, eine Überprüfung des SSC (System-Supplier-Code) und / oder der OEM-Nummer durchgeführt.

<i>Kommando zum Leser</i>	<i>nn 33 p1 p2 p3 p4 p5 pd cs</i>
---------------------------	-----------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der zu schreibenden Datenbytes</i> maximal 32 Bytes → mögliche Werte von 01h bis 20h
<i>p4</i>	1 Byte	SSC (SSC = Byte 9) angegebener Wert wird mit SSC-Byte verglichen bei FFh → keine Überprüfung
<i>p5</i>	1 Byte	OEM-Nummer (OEM-Nummer = Byte 10) angegebener Wert wird mit OEM-Nummer verglichen bei FFh → keine Überprüfung
<i>pd</i>	variabel	<i>Datenbytes</i>
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 33 31</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

Beispiele: siehe nächste Seite

Beispiele zum Befehl <33h>- Write Data, unsegmentierte Medien:

Beispiel 1:

Kommando zum Leser: **0C33000105FFFF0815FF47118Fh**

Schreibe 5 Bytes am Anfang des Datenspeichers
ohne SCC-Check, ohne OEM-Nummer-Check
Daten: 0815FF4711h

Antwort vom Leser bei Erfolg: **023331h**

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

37h → Kein Transponder

Checksumme:

04h

Beispiel 2:

Kommando zum Leser: **0A330010032C550815FFB1h**

Schreibe 3 Bytes ab Adresse 10h des Datenspeichers
SCC-Check auf phg SSC (= 2Ch)
OEM-Nummer-Check mit Nummer 55h
Daten: 0815FFh

Antwort vom Leser bei Erfolg: **023331h**

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

37h → Kein Transponder

Checksumme:

04h

4.1.4.) Kommando <34h> - Write Data, segmentierte Medien

Es wird in einem angegebenen Segment des LEGIC-Transponders ab der angegebenen Startadresse die angegebene Anzahl von Bytes (max. 32 Byte) geschrieben oder es wird nach dem angegebenen SSC-Byte und / oder der angegebenen OEM-Nummer gesucht und von dort ab die angegebene Anzahl von Bytes geschrieben.

<i>Kommando zum Leser</i>	<i>nn 34 p1 p2 p3 p4 p5 p6 pd cs</i>
---------------------------	--------------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der zu schreibenden Datenbytes</i> maximal 32 Bytes → mögliche Werte von 01h bis 20h
<i>p4</i>	1 Byte	<i>Segment-Nummer</i> ab dieser Segmentnummer wird nach SSC und/oder OEM-Nummer gesucht. Wenn SSC = FFh und/oder OEM-Nummer = FFh wird im angegebenen Segment geschr.
<i>p5</i>	1 Byte	SSC (SSC = Byte 6 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>p6</i>	1 Byte	OEM-Nummer (OEM-Nummer = Byte 7 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>pd</i>	variabel	<i>Datenbytes</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 34 36</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiele: siehe nächste Seite

Beispiele zum Befehl <34h>- Write Data, segmentierte Medien:

Beispiel 1:

Kommando zum Leser: **0D3400010503FFFF0815FF47118Ah**

Schreibe 5 Bytes ab Anfang des Segments 3
ohne SCC-Check, ohne OEM-Nummer-Check
Daten: 0815FF4711h

Antwort vom Leser bei Erfolg: **023436h**

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

37h → Kein Transponder

Checksumme:

04h

Beispiel 2:

Kommando zum Leser: **0B34001003042C550815FFB3h**

Suche ab Segment 4 in den folgenden Seg-
menten nach nach phg SSC (= 2Ch) und
OEM-Nummer 55h. Bei Erfolg schreibe 3 Bytes ab
Adresse 10h des gefundenen Segments.
Daten: 0815FFh

Antwort vom Leser bei Erfolg: **023436h**

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

37h → Kein Transponder

Checksumme:

04h

4.1.5.) Kommando <36h> - Launch oder De-Launch

LEGIC-Transponder können mit einer sog. "write condition" geschützt werden. Zu diesem Zweck muss der Leser einen sog. "Launch" mit Hilfe eines speziellen "Launch-Transponders" (SAM63) durchführen. Dabei muss sich der "Launch-Transponder" im RF-Feld des Lesers befinden und das Kommando <36h> ausgeführt werden. Der Leser kann nun nur Karten mit dieser "write condition" beschreiben.

Um dem Leser das Beschreiben anderer Karten wieder zu ermöglichen, muss er wiederum einen sog. "De-Launch" mit Hilfe eines speziellen "De-Launch-Transponders" (SAM64) durchführen. Dabei muss sich der "De-Launch-Transponder" im RF-Feld des Lesers befinden und das Kommando <36h> ausgeführt werden.

<i>Kommando zum Leser</i>	02 36 34
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	03 30 41 72
---	--------------------

<i>Antwort vom Leser im Fehlerfall</i>	03 30 42 71
--	--------------------

Beispiel:

Kommando zum Leser: Launch oder De-Launch	023634h
Antwort vom Leser bei Erfolg: <u>Aufschlüsselung der Antwort:</u> Anzahl der übertragenen Bytes: Kommando: Fehlercode: Checksumme:	03304172h 03h 30h → Status 41h → "Launch" oder "De-Launch" durchgeführt 72h
Antwort vom Leser bei Fehler: <u>Aufschlüsselung der Antwort:</u> Anzahl der übertragenen Bytes: Kommando: Fehlercode: Checksumme:	03304271h 03h 30h → Fehler 42h → Kein "Launch" oder "De-Launch"-Medium 71h

4.1.6.) Kommando <50h> - Activate LED

Steuert die angeschlossene LED an.

<i>Kommando zum Leser</i>	06 50 00 00 p1 00 cs
---------------------------	-----------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>LED-Status</i> 00h = Aus, 01h = Ein, 02 = Impuls, 03h = Blinken
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	02 50 52
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiele:

Kommando zum Leser: **06500000010057h**
 LED einschalten

Kommando zum Leser: **06500000020054h**
 LED kurz einschalten (Impuls)

Kommando zum Leser: **06500000030055h**
 LED blinken lassen

Kommando zum Leser: **06500000000056h**
 LED ausschalten

Antwort vom Leser bei Erfolg: **025052h**

Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 02h
 Kommandowiederholung: 50h
 Checksumme: F2h

Antwort vom Leser bei Fehler: **03303102h**

Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 31h → Unbekanntes Kommando
 Checksumme: 02h

4.1.7.) Kommando <55h> - Read Unique-ID

Liest die Unique-ID des Transponders

<i>Kommando zum Leser</i>	02 55 57
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	06 55 pd cs
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>pd</i>	4 Byte	<i>Unique-ID des Transponders</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	Fehlercode (siehe Tabelle unter 3.3.)
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **025557h**
 Lies Unique-ID vom Transponder

Antwort vom Leser bei Erfolg: **065526329900DEh**

Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 06h
 Kommandowiederholung: 55h
 Unique-ID: 26329900h
 Checksumme: DEh

Antwort vom Leser bei Fehler: **03303704h**

Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 37h → Kein Transponder im Feld
 Checksumme: 04h

4.2.) Kommandos für "Continuous Modus"

4.2.1.) Kommando <60h> - Leser parametrieren

Mit diesem Kommando können im Leser bestimmte Funktionen bzw. Verhaltensweisen eingestellt werden. Diese Parameter sind nichtflüchtig gespeichert und sind damit auch nach einem Spannungsverlust noch gültig.

<i>Kommando zum Leser</i>	<i>nn 60 ap p1 p2 cs</i>
---------------------------	--------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>ap</i>	1 Byte	<i>Ausgabeparameter (siehe Tabelle unter 4.2.2.)</i>
<i>p1</i>	1 Byte	<i>Kommandoparameter 1 (siehe Tabelle unter 4.2.2.)</i>
<i>p2</i>	1 Byte	<i>Kommandoparameter 2 (siehe Tabelle unter 4.2.2.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 60 62</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiele zum Befehl <60h>- Leser parametrieren:

Beispiel:

Kommando zum Leser: **0460010065h**
 Beendet den "Continuous Modus"

Antwort vom Leser bei Erfolg: **026062h**

Antwort vom Leser bei Fehler: **03306251h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 62h → Falscher Parameterwert
 Checksumme: 51h

Weiter Beispiele nach Parameternummern-Tabelle

4.2.2.) Ausgabeparameter für Kommando <60h> - Leser parametrieren

Parameternr.	Default	Länge	Bedeutung
01h	01h	1 Byte	<i>Ausgabetyyp im "Continuous Modus":</i> 00h → "Continuous Modus" aus 01h → "Unique-ID hex" 02h → "Unique-ID dezimal" 03h → "KGH mit SSC=02h" 04h → "KGH benutzerdefiniert" 05h → "KGH benutzerdefiniert – mpdv" 06h → "Daten aus <u>nicht</u> segmentiertem Transponder " 07h → "Daten aus segmentiertem Transponder" 08h → "Daten aus <u>nicht</u> segmentiertem Transponder – Zeit AG"
02h	FFFFh	2 Byte	<i>SSC/OEM für Ausgabetyyp</i> 1. Byte = SSC; 2. Byte = OEM = 2. Byte (bei Parameter 01 = 04h, 06h, 07h oder 08h)
03h	01h	1 Byte	<i>LED-Verhalten</i> 00h → Keine LED-Signalisierung 01h → LED bei gültiger Lesung kurz ein
04h	00h	1 Byte	<i>Ausgabeverhalten</i> 00h → Transponder einmal lesen, solange er im Feld ist 01h → Transponder ständig lesen, solange er im Feld ist
05h	03h	1 Byte	<i>Ausgabe <CR><LF></i> 00h → kein <CR><LF> ausgeben 01h → nur <CR> ausgeben 02h → nur <LF> ausgeben 03h → <CR> und <LF> ausgeben
06h	00h	1 Byte	<i>Startadresse Low-Byte</i> mögliche Werte: 00h bis FFh (nur wirksam bei Parameter 01 = 06h, 07h oder 08h)
07h	0Ch	1 Byte	<i>Anzahl gelesene Bytes</i> mögliche Werte: 01h bis 20h (nur wirksam bei Parameter 01 = 06h, 07h oder 08h)
08h	01h	1 Byte	<i>Startsegmentnummer</i> mögliche Werte: 01h bis FFh (nur wirksam bei Parameter 01 = 07h oder 08h)

ACHTUNG:

Die farbig markierten Ausgabetyypen bzw. Parameter sind erst bei Legic-Lesern ab Lieferdatum 02/2007 vorhanden.

Defaulteinstellung der Parameter:

Ausgabetyyp:	01h	Continuous Modus Unique-ID hex
SSC / OEM:	FFFFh	SSC = FFh – OEM = FFh
LED-Verhalten:	01h	LED bei gültiger Lesung kurz ein
Ausgabeverhalten:	00h	Transponder einmalig lesen
Ausgabe <CR><LF>:	03h	<CR> und <LF> ausgeben
Startadresse Low-Byte:	00h	
Anzahl gelesene Bytes:	0Ch	
Startsegmentnummer:	01h	

Beschreibung der Ausgabeformate: siehe Punkt 4.2.5.

Weitere Beispiele zum Befehl <60h>:

Kommando zum Leser: **0460010164h**
Stelle Ausgabetyt auf "Continuous Modus Unique-ID hex" ein

Kommando zum Leser: **0460010267h**
Stelle Ausgabetyt auf "Continuous Modus Unique-ID dezimal" ein

Kommando zum Leser: **0460010366h**
Stelle Ausgabetyt auf "Continuous Modus KGH hex" ein

Kommando zum Leser: **0460010461h**
Stelle Ausgabetyt auf "Continuous Modus KGH benutzerdefiniert" ein
Hinweis: Vor diesem Kommando muss Parameter p2 eingestellt werden

Kommando zum Leser: **0460010560h**
Stelle Ausgabetyt auf "Continuous Modus KGH benutzerdefiniert mpdv" ein
Hinweis: Vor diesem Kommando muss Parameter p2 eingestellt werden

Kommando zum Leser: **0560022C2D66h**
SSC/OEM für Ausgabetyt p1=4 und p1=5 einstellen.
1.Byte = SSC = 2Ch ; 2.Byte = OEM = 2D

Kommando zum Leser: **0460030067h**
Schaltet die LED-Signalisierung aus

Kommando zum Leser: **0460030166h**
Schaltet die LED-Signalisierung bei gültiger Lesung ein

Kommando zum Leser: **0460040060h**
Stellt den Leser so ein, dass der Ausgabetyt einmalig ausgegeben wird, solange der Transponder sich im Feld des Lesers befindet.

Kommando zum Leser: **0460040161h**
Stellt den Leser so ein, dass der Ausgabetyt kontinuierlich ausgegeben wird, solange der Transponder sich im Feld des Lesers befindet.

Kommando zum Leser: **0460050061h**
Es wird kein <CR><LF> ausgegeben

Kommando zum Leser: **0460050362h**
Es kein <CR><LF> ausgegeben

4.2.3.) Kommando <61h> - Parameter auslesen

Mit diesem Kommando können die, im Leser gespeicherten, Parameter ausgelesen werden.

<i>Kommando zum Leser</i>	02 61 63
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgr. Ausführung</i>	0B 61 lp cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>lp</i>	9 Byte	<i>Einstellungen der Ausgabeparameter (siehe Tabelle unter 4.2.2.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **026163h**
 Parametrierung des Lesers auslesen

Antwort vom Leser bei Erfolg: **0B61082C3C010103000C017Ch**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 0Bh
 Kommando: 61h → Kommando
 Ausgabety: 05h → Continuous Modus KGH mpdv
 SSC/OEM: 2C2Dh → SSC = 2Ch ; OEM = 2Dh
 LED-Verhalten: 01h → LED kurz ein bei gültiger Lesung
 Ausgabe <CR><LF>: 03h → <CR> und <LF> ausgeben
 Startadresse Low-Byte: 00h → Ab Byte 00h lesen
 Anzahl gelesener Bytes: 0Ch → 12 Bytes lesen
 Startsegment: 01h → Bei Segment 1 starten
 Checksumme: 7Ch

4.2.4.) Kommando <65h> - Firmwarestand auslesen

Mit diesem Kommando kann der Firmwarestand des Lesemoduls ausgelesen werden.

ACHTUNG:

Erst bei Legic-Lesern ab Lieferdatum 02/2007 möglich.

<i>Kommando zum Leser</i>	02 65 63
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgr. Ausführung</i>	0C 65 fw cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fw</i>	10 Byte	<i>Firmwarebezeichnung, 10 ASCII-Zeichen (siehe Tabelle unter 4.1.8.1)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **026563h**
 Parametrierung des Lesers auslesen

Antwort vom Leser bei Erfolg: **0C65363936313843202020201Ah**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 0Ch

Kommando: 65h → Kommando

Firmwarebezeichnung: "69618C " (36 39 36 31 38 43 20 20 20 20)

Checksumme: 1Ah

4.2.5.) Beschreibung der Ausgabeformate im "Continuous Modus"

Hinweis:

Die Daten werden als ASCII-Zeichen ausgegeben. Zum Beispiel wird der Hexwert 02h als die zwei ASCII-Zeichen "0" und "2" ausgegeben.

Format "Unique-ID hex"

<i>Antwort vom Leser</i>	<i>pd <CR> <LF></i>	
<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>pd</i>	8 Zeichen	<i>Unique-ID des Transponders</i> hexadezimal
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "Unique-ID dezimal"

<i>Antwort vom Leser</i>	<i>pd <CR> <LF></i>	
<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>pd</i>	10 Zeichen	<i>Unique-ID des Transponders</i> dezimal mit führenden Nullen
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "KGH mit SSC=02h"

<i>Antwort vom Leser</i>	<i>p1 pd <CR> <LF></i>	
<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	2 Zeichen	<i>SSC</i> hexadezimal
<i>pd</i>	12 Zeichen	<i>Daten</i>
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "KGH benutzerdefiniert"

<i>Antwort vom Leser</i>	<i>p1 p2 pd <CR> <LF></i>	
<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	2 Zeichen	<i>SSC</i> hexadezimal
<i>p2</i>	2 Zeichen	<i>OEM-Nummer</i> hexadezimal
<i>pd</i>	10 Zeichen	<i>Daten</i>
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "KGH benutzerdefiniert – mpdv"

<i>Antwort vom Leser</i>	<i>ps p1 p2 p3 p4 p5 pe cs</i>
--------------------------	--------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>ps</i>	1 Zeichen	<i>Startzeichen</i> ASCII-Zeichen mit dem festen Wert 02h
<i>p1</i>	5 Zeichen	<i>Suffix</i> ASCII-String "n110A" (hex: 6Eh 31h 31h 30h 41h)
<i>p2</i>	2 Zeichen	<i>SSC</i> hexadezimal
<i>p3</i>	2 Zeichen	<i>OEM-Nummer</i> hexadezimal
<i>p4</i>	4 Zeichen	<i>Firmennummer</i>
<i>p5</i>	6 Zeichen	<i>Kartenummer</i>
<i>pe</i>	1 Zeichen	<i>Endezeichen</i> ASCII-Zeichen mit dem festen Wert 03h
<i>cs</i>	1 Zeichen	<i>BCC</i> Checksumme über die Daten

Format "Daten aus nicht segmentiertem Transponder"

<i>Antwort vom Leser</i>	<i>p1 p2 pd <CR> <LF></i>
--------------------------	---------------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	2 Zeichen	<i>SSC</i> hexadezimal
<i>p2</i>	2 Zeichen	<i>OEM-Nummer</i> hexadezimal
<i>pd</i>	n Zeichen	<i>Daten</i> Länge abhängig von Parameter 07h aus Kommando <60h>
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "Daten aus segmentiertem Transponder"

<i>Antwort vom Leser</i>	<i>p1 p2 pd <CR> <LF></i>
--------------------------	---------------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	2 Zeichen	<i>SSC</i> hexadezimal
<i>p2</i>	2 Zeichen	<i>OEM-Nummer</i> hexadezimal
<i>pd</i>	n Zeichen	<i>Daten</i> Länge abhängig von Parameter 07h aus Kommando <60h>
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> abhängig von Parameter 05h aus Kommando <60h>

Format "Daten aus segmentiertem Transponder – Zeit AG"

<i>Antwort vom Leser</i>	<i>00 00 00 00 00 pd<CR> <LF></i>
--------------------------	---

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fest</i>	10 Zeichen	<i>Zeichen "0" (ASCII 30h)</i>
<i>pd</i>	6 Zeichen	<i>Daten</i>
<i><CR><LF></i>	2 Zeichen	<i>Formatierungszeichen</i> <i>abhängig von Parameter 05h aus Kommando <60h></i>

4.3.) Der KABA-Group-Header (KGH)

Der sogenannten KABA-Group-Header, kurz KGH, ist eine Vereinbarung zwischen LEGIC-Lizenznehmern und der KABA-Group und spezifiziert die Daten-Organisation auf LEGIC-Transpondern.

Die Vereinbarung beinhaltet auch die Bildung einer Checksumme (CRC) über relevante Daten. Der KGH definiert genau, über welchen Bereich der Daten die Checksumme gebildet wird.

Bei **unsegmentierten** LEGIC-Medien ist eine Checksumme mit 8 Bit Länge möglich.

Bei **segmentierten** LEGIC-Medien sind Checksummen mit 8 Bit oder 16 Bit Länge möglich.

4.3.1.) Kommando <40h> - Build KGH, unsegmentierte Medien

Erzeugt den KABA-Group-Header bei unsegmentierten Medien.

<i>Kommando zum Leser</i>	02 40 42
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	02 40 42
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **024042h**
 Erzeuge KGH bei unsegmentiertem Medium

Antwort vom Leser bei Erfolg: **024042h**

Antwort vom Leser bei Fehler: **03305063h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 50h → Fehler beim Erzeugen
 Checksumme: 63h

4.3.2.) Kommando <41h> - Build KGH, segmentierte Medien

Erzeugt den KABA-Group-Header bei segmentierten Medien.

<i>Kommando zum Leser</i>	<i>05 41 p1 p2 p3 cs</i>
---------------------------	--------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	Segment-Nummer ab dieser Segmentnummer wird nach S_SSC und/oder S_OEM-Nummer gesucht. Wenn S_SSC = FFh und/oder S_OEM-Nummer = FFh wird der KGH im angegebenen Segment erzeugt
<i>p2</i>	1 Byte	S_SSC (SSC = Byte 6 im Segment) angegebener Wert wird ab gegebenem Segment gesucht, bei FFh → keine Suche
<i>p3</i>	1 Byte	S_OEM-Nummer (OEM-Nummer = Byte 7 im Segment) angegebener Wert wird ab gegebenem Segment gesucht, bei FFh → keine Suche
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 41 43</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	Fehlercode (siehe Tabelle unter 3.3.)
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **0541012CFF96h**
 Erzeuge KGH bei segmentiertem Medium
 S_SSC = 2Ch, S_OEM-Nummer = FFh

Antwort vom Leser bei Erfolg: **024143h**

Antwort vom Leser bei Fehler: **03305162h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 51h → Fehler beim Erzeugen
 Checksumme: 63h

4.3.3.) Kommando <42h> - Build CRC, unsegmentierte Medien

Erzeugt eine Checksumme (CRC) beginnend bei der angegebenen Startadresse über die angegebene Anzahl von Datenbytes und speichert die CRC in der angegebenen Speicheradresse

<i>Kommando zum Leser</i>	<i>07 42 p1 p2 p3 p4 p5 cs</i>
---------------------------	--------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der Datenbytes für die CRC</i>
<i>p4</i>	1 Byte	<i>High-Byte von Speicheradresse für CRC</i>
<i>p5</i>	1 Byte	<i>Low-Byte von Speicheradresse für CRC</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 42 40</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel:

Kommando zum Leser: **074200000A000B44h**
 Erzeuge CRC bei unsegmentiertem Medium
 von Startadresse 0000h über 10 Bytes und
 speichere CRC im Byte 000Bh

Antwort vom Leser bei Erfolg: **024240h**

Antwort vom Leser bei Fehler: **03305261h**

Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 52h → Fehler beim Erzeugen
 Checksumme: 61h

4.3.4.) Kommando <43h> - Build CRC, segmentierte Medien

Erzeugt eine Checksumme (CRC) beginnend bei der angegebenen Startadresse im entsprechenden Segment über die angegebene Anzahl von Datenbytes und speichert die CRC in der angegebenen Speicheradresse im Segment

<i>Kommando zum Leser</i>	<i>0B 43 p1 p2 p3 p4 p5 p6 p7 p8 p9 cs</i>
---------------------------	--

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der Datenbytes für die CRC</i>
<i>p4</i>	1 Byte	<i>High-Byte von Speicheradresse für CRC</i>
<i>p5</i>	1 Byte	<i>Low-Byte von Speicheradresse für CRC</i>
<i>p6</i>	1 Byte	<i>CRC-Algorithmus</i> 00h = 8-Bit-CRC, 01h = 16-Bit-CRC
<i>p7</i>	1 Byte	<i>Segment-Nummer</i> ab dieser Segmentnummer wird nach S_SSC und/oder S_OEM-Nummer gesucht. Wenn S_SSC = FFh und/oder S_OEM-Nummer = FFh wird der KGH im angegebenen Segment erzeugt
<i>p8</i>	1 Byte	S_SSC (SSC = Byte 6 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>p9</i>	1 Byte	S_OEM-Nummer (OEM-Nummer = Byte 7 im Segment) angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>02 43 40</i>
---	-----------------

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel: siehe nächste Seite

Beispiel zum Befehl <43h>- Build CRC, segmentierte Medien:

Kommando zum Leser: **0B4300000F0010010120FF88h**

Erzeuge CRC bei segmentiertem Medium
von Startadresse 0000h über 15 Bytes und
speichere CRC im Byte 0010h, 16-Bit-CRC
Segment 1, S_SSC = 20h, Keine S_OEM-Nummer

Antwort vom Leser bei Erfolg: **024240h**

Antwort vom Leser bei Fehler: **03305360h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

53h → Fehler beim Erzeugen

Checksumme:

60h

4.3.5.) Kommando <44h> - Read KGH, unsegmentierte Medien

Liest den KABA-Group-Header (KGH) von unsegmentierten Medien

<i>Kommando zum Leser</i>	02 44 42
---------------------------	-----------------

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	0A 44 pd cs
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>pd</i>	8 Byte	<i>Daten des KGH</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	03 30 fc cs
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel

Kommando zum Leser: **024446h**
 Lies KGH

Antwort vom Leser bei Erfolg: **0A44482C00081512345647h**
Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 0Ah
 Kommandowiederholung: 44h
 Daten des KGH: 482C000815123456h
 Checksumme: 47h

Antwort vom Leser bei Fehler: **03305467h**
Aufschlüsselung der Antwort:
 Anzahl der übertragenen Bytes: 03h
 Kommando: 30h → Fehler
 Fehlercode: 54h → Fehler beim Lesen
 Checksumme: 67h

4.3.6.) Kommando <45h> - Read KGH, segmentierte Medien

Liest den KABA-Group-Header (KGH) von segmentierten Medien aus dem angegebenen Segment

<i>Kommando zum Leser</i>	<i>05 45 p1 p2 p3 cs</i>
---------------------------	--------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	Segment-Nummer ab dieser Segmentnummer wird nach S_SSC und/oder S_OEM-Nummer gesucht. Wenn S_SSC = FFh und/oder S_OEM-Nummer = FFh wird der KGH aus dem angegebenen Segment gelesen
<i>p2</i>	1 Byte	S_SSC (SSC = Byte 6 im Segment) angegebener Wert wird ab gegebenem Segment gesucht, bei FFh → keine Suche
<i>p3</i>	1 Byte	S_OEM-Nummer (OEM-Nummer = Byte 7 im Segment) angegebener Wert wird ab gegebenem Segment gesucht, bei FFh → keine Suche
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>0D 45 pd cs</i>
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>pd</i>	11 Byte	Daten des KGH
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	Fehlercode (siehe Tabelle unter 3.3.)
<i>cs</i>	1 Byte	Checksumme Durch EXOR aller Bytes ermittelt

Beispiel: siehe nächste Seite

Beispiel zum Befehl <45h>- Read KGH, segmentierte Medien:

Kommando zum Leser: **054501FFFF41h**
Lies KGH von Segment 1
S_SSC = FFh, S_OEM-Nummer = FFh

Antwort vom Leser bei Erfolg: **0D45080004002C00081512345605h**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 0Dh
Kommandowiederholung: 45h
Daten des KGH: 080004002C000815123456h
Checksumme: 05h

Antwort vom Leser bei Fehler: **03305566h**
Aufschlüsselung der Antwort:
Anzahl der übertragenen Bytes: 03h
Kommando: 30h → Fehler
Fehlercode: 55h → Fehler beim Lesen
Checksumme: 67h

4.3.7.) Kommando <46h> - Read CRC, unsegmentierte Medien

Liest den Daten-Block für die CRC, beginnend bei der angegebenen Startadresse über die angegebene Anzahl von Datenbytes, und das CRC aus der angegebenen Speicheradresse

<i>Kommando zum Leser</i>	<i>07 46 p1 p2 p3 p4 p5 cs</i>
---------------------------	--------------------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der Datenbytes für die CRC</i>
<i>p4</i>	1 Byte	<i>High-Byte von Speicheradresse für CRC</i>
<i>p5</i>	1 Byte	<i>Low-Byte von Speicheradresse für CRC</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>nn 46 pd sc</i>
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>pd</i>	variabel	<i>Daten für die CRC und CRC-Byte</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel: siehe nächste Seite

Beispiel zum Befehl <46h>- Read CRC, unsegmentierte Medien:

Kommando zum Leser: **074600000A000B40h**

Lese Daten für CRC von Startadresse 0000h
über 10 Bytes und CRC-Byte aus Adresse 000Bh

Antwort vom Leser bei Erfolg: **0C462C004711004974208956F2h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

0Ch

Kommandowiederholung:

46h

Daten:

2C004711004974208956h

Checksumme:

F2h

Antwort vom Leser bei Fehler: **03305665h**

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:

03h

Kommando:

30h → Fehler

Fehlercode:

52h → Fehler beim Lesen

Checksumme:

61h

4.3.8.) Kommando <47h> - Read CRC, segmentierte Medien

Liest die Daten für eine Checksumme (CRC), beginnend bei der angegebenen Startadresse im entsprechenden Segment über die angegebene Anzahl von Datenbytes, und liest das CRC-Byte aus der angegebenen Speicheradresse im Segment

<i>Kommando zum Leser</i>	<i>0B 47 p1 p2 p3 p4 p5 p6 p7 p8 p9 cs</i>
---------------------------	--

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>p1</i>	1 Byte	<i>High-Byte von Startadresse</i>
<i>p2</i>	1 Byte	<i>Low-Byte von Startadresse</i>
<i>p3</i>	1 Byte	<i>Anzahl der Datenbytes für die CRC</i>
<i>p4</i>	1 Byte	<i>High-Byte von Speicheradresse für CRC</i>
<i>p5</i>	1 Byte	<i>Low-Byte von Speicheradresse für CRC</i>
<i>p6</i>	1 Byte	<i>CRC-Algorithmus</i> 00h = 8-Bit-CRC, 01h = 16-Bit-CRC
<i>p7</i>	1 Byte	<i>Segment-Nummer</i> ab dieser Segmentnummer wird nach S_SSC und/oder S_OEM-Nummer gesucht. Wenn S_SSC = FFh und/oder S_OEM-Nummer = FFh werden die Daten für die CRC aus dem angegebenen Segment gelesen
<i>p8</i>	1 Byte	<i>S_SSC-Check (SSC = Byte 6 im Segment)</i> angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>p9</i>	1 Byte	<i>S_OEM-Nummern-Check (OEM-Nummer = Byte 7 im Segment)</i> angegebener Wert wird ab angegebenem Segment gesucht, bei FFh → keine Suche
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser bei erfolgreicher Ausführung</i>	<i>nn 47 pd sc</i>
---	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>nn</i>	1 Byte	<i>Anzahl der übertragenen Bytes</i>
<i>pd</i>	variabel	<i>Daten für die CRC und CRC-Byte</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

<i>Antwort vom Leser im Fehlerfall</i>	<i>03 30 fc cs</i>
--	--------------------

<i>Parameter</i>	<i>Länge</i>	<i>Bedeutung</i>
<i>fc</i>	1 Byte	<i>Fehlercode (siehe Tabelle unter 3.3.)</i>
<i>cs</i>	1 Byte	<i>Checksumme</i> Durch EXOR aller Bytes ermittelt

Beispiel: siehe nächste Seite

Beispiel zum Befehl <47h>- Read CRC, segmentierte Medien:

Kommando zum Leser: **0B4700000F0010010120FF88h**

Lese CRC bei segmentiertem Medium
von Startadresse 0000h über 15 Bytes und
lese CRC-Byte aus Adresse 0010h, 16-Bit-CRC
Segment 1, S_SSC = 20h, Keine S_OEM-Nummer

Antwort vom Leser bei Erfolg:

114720000102030405060708090A0B0C0D21h

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:	11h
Kommandowiederholung:	47h
Daten:	20000102030405060708090A0B0C0Dh
Checksumme:	21h

Antwort vom Leser bei Fehler:

03305764h

Aufschlüsselung der Antwort:

Anzahl der übertragenen Bytes:	03h
Kommando:	30h → Fehler
Fehlercode:	57h → Fehler beim Lesen
Checksumme:	60h

5.) Impressum

Sie erreichen uns unter folgender Adresse:

noax Technologies AG
Am Forst 6
85560 Ebersberg

Fon: +49 (0) 8092 / 8536 – 0
Fax: +49 (0) 8092 / 8536 – 55

E-Mail: hotline@noax.com
Internet: www.noax.com

Gültig für folgende Firmware-Versionen:

- 69618a – LEGIC_RW01_Noax - für Legic-Module bis LT 12 / 2006
- 69618b – LEGIC_RW01_Noax - nur für Legic-Module ab LT 12 / 2006
- 69618c – LEGIC_RW01_Noax - nur für Legic-Module ab LT 02 / 2007

Dokument erstellt am: 12.02.2007

Erstellt durch: ps

Die jeweils aktuelle Version dieses Dokuments und der notwendigen Treiber finden Sie im Download-Bereich unserer Homepage.

Technische Änderungen vorbehalten !!