



REMOTE ACCESS [V1.07.00]

BENUTZER INFORMATION

Autor: MOTTO:lean!
Stand: 28.02.2017
Version: v1.03.00

Inhaltsverzeichnis

1. Dokumenten-Historie	4
2. Einleitung.....	5
2.1. Grundlegende Anforderungen.....	5
3. Der RemoteAccess	6
3.1. Verfügbarkeit und Einsatz.....	6
3.2. Schnittstelle	6
3.3. „Answer On Request“-Prinzip	6
3.3.1. „Answer On Event“-Prinzip	6
3.4. Verarbeitung der Kommandos.....	7
4. Die R-iZO-Einheiten	8
4.1. Eigenschaften und Funktionen des Basismoduls.....	8
4.2. Neu im Basismodul	8
4.2.1. Betriebsarten.....	8
4.2.2. Verwaltung der CAN-ID.....	9
4.3. Eigenschaften und Funktionen des Leuchtmoduls	9
4.4. Neu im Leuchtmodul.....	9
4.5. Verbindungsparameter der seriellen Schnittstelle.....	9
5. Die Kommando-Gruppen und -Übermittlung.....	11
5.1. Simultaner Einsatz von RemoteAccess- und Barcode-Kommandos.....	11
5.2. CAN-Bus und Message-ID.....	12
5.2.1. Bestehende CAN-ID Zuordnung.....	12
5.2.2. CAN-ID Nummernkreise.....	13
5.2.3. Dynamische CAN-ID Vergabe durch das Basismodul.....	13
5.2.4. Die CAN-ARP Tabelle.....	14
5.2.5. Anfragen durch Benutzer.....	15
5.3. Kommando-Typen.....	16
5.3.1. Globale Kommandos (GLOBALS)	16
5.3.2. Individuelle Kommandos (UNIQUE).....	17
5.3.3. Aufbau des UID.....	17
6. Prinzip und Aufbau des Datenaustauschs.....	18
6.1. Die Memory-Map.....	18
6.2. Aufbau eines Kommando-Pakets.....	18
6.3. Binäre Datenübertragung ohne Terminierungszeichen.....	18
6.4. Die Kommandos	21

6.4.1. GWRITE - global write.....	21
6.4.2. UWRITE - unique write.....	22
6.4.3. UREAD - unique read.....	24
6.4.4. Fehlerhafte Syntax.....	26
6.5. Aufbau der Memory-Map.....	27
6.5.1. Modulabhängige Speicherbereiche.....	27
6.6. Zugriffsrechte.....	27
6.6.1. Reaktion bei Lesen lese-geschützter Bytes.....	28
6.6.2. Reaktion bei Schreiben schreib-geschützter Bytes.....	28
7. Die Memory-Map.....	29
7.1. R-izo Varianten: basic, advanced und x-pert.....	29
7.1.1. Schreiben in nicht verfügbaren Adressbereich.....	29
7.1.2. Lesen aus nicht verfügbarem Adressbereich.....	29
7.2. Polling.....	38
8. Ereignis gesteuerte Kommandos	39
8.1. GEVENT - global event.....	39
8.2. Globale Ereignisse.....	40
8.2.1. Erläuterung zu UserInfos.....	42
9. Module identifizieren.....	43
9.1. Identify Mode verwenden.....	43
9.2. Identify Mode beenden.....	44
10. Anhang.....	45
10.1. Schnellübersicht der Kommandos.....	45

6.4. Die Kommandos

Im Folgenden werden die Basiskommandos für den RemoteAccess beschrieben.

6.4.1. GWRITE - global write

Beschreibung:	Das Kommando GWRITE steht für die Abkürzung „global write“ und erlaubt das Schreiben von Daten und Parametern in einen definierten Adressbereich aller im System befindlicher Module. So können z.B. mit Hilfe dieses Kommandos die RGB-Anzeigen aller Leuchtmodule einheitlich deaktiviert oder auf „rot“ gesetzt werden.
Kommando-Syntax: (abstrakt)	GWRITE <COUNT> <ADDR> <DATA>
Kommando-Byte:	Zeichen 'G' ASCII hex: 0x47h
Kommando-Parameter:	<COUNT> 1-Byte breiter Wert zur Definition der Anzahl folgender Datenbytes (inkl. Adressbytes). Wertebereich 0..30 <ADDR> 2-Byte breite Speicheradresse entsprechend Memory-Map, an welche der Inhalt der Datenbytes <DATA> geschrieben werden soll. Es wird stets das MSB zuerst übertragen. Wertebereich 0..65535. <DATA> Bytes, welche die zu schreibenden Daten beinhalten.
Antwort-Syntax: (abstrakt)	<i>keine Rückmeldung</i>
Antwort-Byte:	<i>keine Rückmeldung</i>
Antwort-Parameter:	<i>keine Rückmeldung</i>
Bemerkung:	-
Beispiel:	Ab der Speicheradresse 12d = 0x0Ch sollen 3 Bytes geschrieben werden. Somit ergibt sich für die Speicheradressen folgende Wertzuweisungen: Adr. 0x0Ch → 49d = 0x31h Adr. 0x0Dh → 50d = 0x32h Adr. 0x0Eh → 51d = 0x33h Das abstrakte Kommando hierfür lautet: GWRITE 0x05 0x00 0x0C 0x31 0x32 0x33 Das Datenpaket ist somit wie folgt aufgebaut: 0x47 0x05 0x00 0x0C 0x31 0x32 0x33

Hierfür wurde somit insgesamt **7 Bytes** zur Datenübermittlung benötigt.

6.4.2. UWRITE - unique write

Beschreibung: Das Kommando UWRITE steht für die Abkürzung „unique write“ und erlaubt das Schreiben von Daten und Parametern in einen definierten Adressbereich eines spezifischen im System befindlichen Moduls. So kann z.B. mit Hilfe dieses Kommandos die RGB-Anzeigen nur dieses einen Leuchtmoduls deaktiviert oder auf „rot“ gesetzt werden. Für dieses Kommando ist die Angabe des UID des entsprechenden Moduls erforderlich.

Kommando-Syntax:
(abstrakt)

UWRITE <COUNT> <UID> <ADDR> <DATA>

Kommando-Byte:

Zeichen **'W'**
ASCII hex: 0x57h

Kommando-Parameter:

<COUNT>
1-Byte breiter Wert zur Definition der Anzahl folgender Datenbytes (inkl. UID und Adressbytes). Wertebereich 0..30

<UID>
8-Byte breiter UID zur Identifikation des gewünschten Moduls.

<ADDR>
2-Byte breite Speicheradresse entsprechend Memory-Map, an welche der Inhalt der Datenbytes <DATA> geschrieben werden soll. Es wird stets das MSB zuerst übertragen. Wertebereich 0..65535.

<DATA>
Bytes, welche die zu schreibenden Daten beinhalten.

Antwort-Syntax:
(abstrakt)

UwRDONE <COUNT> <UID> <ADDR> <WRITTENBYTES>

Antwort-Byte:

Zeichen **'w'**
ASCII hex: 0x77h

Antwort-Parameter:

<COUNT>
1-Byte breiter Wert zur Definition der Anzahl folgender Datenbytes (inkl. UID und Adressbytes). Wertebereich 0..30

<UID>
8-Byte breiter UID zur Identifikation des antwortenden Moduls.

<ADDR>
2-Byte breite Speicheradresse entsprechend Memory-Map, an welche der Inhalt der Datenbytes <DATA> geschrieben wurde. Es wird stets das MSB zuerst übertragen. Wertebereich 0..65535.

<WRITTENBYTES>

1-Byte breiter Wert, welcher darüber informiert, wie viele Bytes tatsächlich geschrieben wurden.

Ein Wert von 0 kann z.B. darauf hinweisen, dass eine unzulässige Adressierung (Adresse außerhalb des zulässigen Adressbereichs) stattgefunden hat. Ebenso wurde ggf. versucht, in einen schreib-geschützten Bereich zu schreiben. Bei Schreiben mehrerer Bytes hintereinander werden die schreib-geschützten Bytes „übersprungen“ und von dem WrittenBytes-Wert abgezogen.

Sollten so z.B. 3 Bytes geschrieben werden, jedoch eines von diesen in der Memory-Map als schreib-geschützt definiert ist, erhält die Rückmeldung einen WrittenByte-Wert von nur 2.

Bemerkung: Die Antwort enthält wiederum den UID mit der Speicheradresse und der Anzahl geschriebener Datenbytes. Somit kann das externe System über diese Parameter den Erfolg des Schreibauftrags rückverfolgen. Die übertragenen Datenbytes selbst werden zur Reduzierung des Datenaufkommens nicht wieder zurückgesendet.

Beispiel: Ab der Speicheradresse 12d = 0x0Ch sollen 2 Bytes geschrieben werden. Das Kommando ist nur an ein Leuchtmodul mit spezifischer UID gerichtet.

Für die Speicheradressen ergibt sich hieraus folgende Wertzuweisungen:

Adr. 0x0Ch → 49d = 0x31h

Adr. 0x0Dh → 50d = 0x32h

Die UID des Leuchtmoduls lautet 0x12 0x23 0x34 0xFF 0xA5 0x88 0x45 0x9D

Das abstrakte Kommando mit UID hierfür lautet:

UWRITE 0x0C 0x12 0x23 0x34 0xFF 0xA5 0x88 0x45 0x9D 0x00 0x0C 0x31
0x32

Das Datenpaket ist somit wie folgt aufgebaut:

0x57 0x0C 0x12 0x23 0x34 0xFF 0xA5 0x88 0x45 0x9D 0x00 0x0C 0x31
0x32

Hierfür wurde somit insgesamt **14 Byte** zur Datenübermittlung benötigt.

Wird das Kommando empfangen und erfolgreich verarbeitet, wird folgende Antwort zurückgesendet:

UWDONE 0x0B 0x12 0x23 0x34 0xFF 0xA5 0x88 0x45 0x9D 0x00 0x0C 0x02
bzw.
0x77 0x0B 0x12 0x23 0x34 0xFF 0xA5 0x88 0x45 0x9D 0x00 0x0C 0x02

Bezeichnung	Verfügbar ab FW-Version	Startadresse	Länge [Byte]	Schreib-Lese- Zugriff ¹	Verfügbar in Lizenz ²			Beschreibung
					basic	advanced	x-pert	
Leuchtmodul [0x0000]								
Modultyp	v2.11.00	0x0000	1	R	X	X	X	Auslesen des Modultyps 0x01 = Leuchtmodul 0x02 = Basismodul
Firmware-Version		0x0001	3	R	X	X	X	Auslesen der Firmware-Version Byte 1 (0x0001) = FwVersion Main, 0..255, z.B. 2 Byte 2 (0x0002) = FwVersion Sub, 0..255, z.B. 11 Byte 3 (0x0003) = FwVersion Fix, 0..255, z.B. 0 → ergibt Firmware-Version v2.11.00
Alternierende RGB-LED- Anzeige Sekundärer Farbwert		0x0004	3	R/W	X	X	X	Definiert die Helligkeitswerte für den zweiten Farbwert bei Wechselblinken (Alternieren). Wechselblinken ist aktiv, wenn beide Werte für die Blinkzeiten (siehe unten) gesetzt und ungleich 0 sind. Byte 1 (0x0004) = Rot-Wert, Wertebereich 0..40 Byte 2 (0x0005) = Grün-Wert, Wertebereich 0..40 Byte 3 (0x0006) = Blau-Wert, Wertebereich 0..40 Die Helligkeitswerte werden stets auf einen Wert von max. 40 (=100%) limitiert.
Alternierende RGB-LED- Anzeige Blinkzeiten		0x0007	2	R/W	X	X	X	Anzeigedauer für Wechselblinken (Alternieren). Die RGB-LED-Anzeige wechselt entsprechend den hier definierten Zeiten ihre Anzeige zwischen den sekundären und primären Farbwerten. Durch Setzen z.B. der sekundären Farbwerte auf 0 kann ein „einfarbiges“ Blinken erzeugt werden. Byte 1 (0x0007) = Anzeigedauer des primären Farbwerts. Byte 2 (0x0008) = Anzeigedauer des sekundären Farbwerts. Werte sind in x 10ms anzugeben, z.B. Wert 22 = 220 ms. Es müssen beide Werte gesetzt sein, um das Wechselblinken zu aktivieren. Wird nur einer der beiden Werte auf 0 gesetzt,

¹ Schreib-Lese-Zugriff: R = readable (lesbar), W = writeable (beschreibbar)

² Verfügbar in Lizenz: X = verfügbar, - = nicht verfügbar

7.2. Polling

Um nun definierte Zustände eines Leuchtmoduls wie z.B. den Sensor-Status dauerhaft zu überwachen, ist der Einsatz des Daten-Pollings möglich. Hierbei werden alle Leuchtmodule zyklisch bezüglich ihres Sensor-Status abgefragt und ausgewertet. Da es sich bei den meisten R-iZO Applikationen nicht um zeitintensive Anwendungen handelt, ist das Polling hierfür ein vertretbares Verfahren.

MOTTO: LEAN!

IHR IMPULSGEBER FÜR DIE OPTIMIERTE PRODUKTIONSLANDSCHAFT

Karlsruher Str.21
78048 Villingen-Schwenningen

Tel.: 07721 / 94429 -42

info@mottolean.de

www.mottolean.de