

## 1 Allgemeines

Das Protokoll Gastro-IO 1.00 (GIO 1.00) wird in LANs gastronomischer Geräte verwendet.

Für Neuentwicklungen sollte dieses Protokoll verwendet werden. Aus Kompatibilitätsgründen verstehen aber alle Geräte (Dosierer) weiterhin das "alte" Protokoll und antworten entsprechend.

Das Umschalten zwischen "altem" und "neuem" Protokoll ist während des Betriebs möglich (auch Satzweise).

## 2 Schnittstellen

### 2.1 serielle Schnittstelle

Schnittstelle	RS-232
Bits/s	1200,2400,4800,9600 oder 19200
Datenbits	8
Parität	keine
Stoppbits	1

### 2.2 UDP Schnittstelle

Schnittstelle	Ethernet 10/100Mbit
Protokoll	UDP
Port	18500

## 3 Satzformat - Übersicht

Zur Sicherung wird das folgende Satzformat verwendet -

<Header> <Sync> <Bytecount> <Command> <Device> <Nx> <Datensatz> <CKS>  
<CR>

Dabei sind

<Header> Das Zeichen "Z", ASCII-Code 90.

<Sync> Das Zeichen <Null>, ASCII-Code 0.

<Bytecount> Der Bytecount, ein Byte binär kodiert, gibt die Gesamtzahl der Bytes in <Command>, <Device>, <Nx>, <Datensatz> und <Checksum> an.

<Command> Das Commandbyte. Derzeit sind verwendet:

<b>&lt;SI&gt;</b>	Pollen oder Anfordern bestimmter Datensätze
<b>&lt;SO&gt;</b>	Beantwortung oder Senden von Datensätzen

**<DC1>** Anfordern von Konfigurationsdaten

**<DC2>** Beantwortung oder Senden von Konfigurationsdaten

**<SI>/<SO>** wird während des Betriebs zur Übertragung von Buchungen, Bons, Tischkonten etc. verwendet.

**<DC1>/<DC2>** hat ein von <SI>/<SO> verschiedenes Datensatzformat und wird für Entwicklung, Prüfung, Installation, Wartung und zum Übertragen von Produktdaten etc. verwendet.

**<Device>** Gibt das Zielgerät an, d.h. den Slave, der mit dem PC kommuniziert, unabhängig davon, ob der PC oder das Zielgerät sendet. **<Device>** umfasst immer zwei Zeichen; das erste gibt die Geräteart an, das zweite die Nummer.

Die Geräteart ist -

"T" für Terminals

"D" für Dosierer

Die Gerätenummer wird als CHR (48 + Gerätenummer) übertragen

**<Nx>** Die Datensatznummerierung. Sie enthält einen Sendezähler Ns, einen Empfangszähler Nr. Siehe Kap 4.

**<Datensatz>** Der übermittelte Datensatz. Dabei ist zu beachten:

- der Datensatz kann leer sein
- der Datensatz darf nur druckbare Zeichen enthalten (ASCII-Codes 32 bis 255)

**<Checksum>** Dieses Byte ist die negative Summe (2er Komplement) aller Bytes von **<Bytecount>**, **<Command>**, **<Device>**, **<Nx>** und **<Datensatz>** .

(Das LowByte der Summe von **<Bytecount>**, **<Command>**, **<Device>**, **<Nx>**, **<Datensatz>** und **<Checksum>** muss 0 sein )

**<CR>** Das Zeichen **<CR>**, ASCII-Code 0D hex , schließt den Satz ab.

!!!!!! Aus Vereinfachungsgründen verstehen TBT Slaves (Dosierer,Gaby,..) auch Sätze ohne die Datensatznummerierung. Diese Sätze werden dann ohne Numerierung beantwortet.

**CARAT & ORION** pollen die angeschlossenen Slaves **mit** Datensatznummerierung und benötigen diese auch in der Antwort.

DC1/DC2 Sätze sind nicht numeriert.

## 4 Sicherung

### 4.1 Datensatznummerierung

Zur Sicherung der Datensätze und zur Anzeige des Busy-Zustands können Sätze mit einem Nx-Byte versehen werden. Es umfaßt die folgenden Bits:

Ns Nummer des gesendeten Datensatzes, 0 .. 1  
 Nr Nummer des erwarteten Datensatzes, 0 .. 1

Die Bits werden in einem Byte kodiert wie folgt:

```
+-----+
| 0 | 0 | 1 | 1 | 0 | 0 | Ns | Nr |
+-----+
```

Das Nx-Byte wird daran erkannt, dass es im Bereich "0" .. "3" liegt. Kein GIO-Befehl beginnt mit einer Ziffer.

In <Datensatz>-Feld eines Satzes können Daten übertragen werden, die genau einmal vom Empfänger übernommen werden müssen, z.B. Buchungen.

Zur Sicherung werden ein Sendezähler Ns und ein Empfangszähler Nr im <Nx>-Feld des Satzes mitgeschickt. Diese Zähler nummerieren nicht den Satz, sondern den enthaltenen Datensatz.

Innerhalb eines übermittelten Satzes haben die Zähler Ns und Nr die folgende Bedeutung:

Ns Die Nummer des enthaltenen (gesendeten) Datensatzes

Nr Die Nummer des als nächstes erwarteten Datensatzes

Beide Zähler laufen von 0 bis 1, sind also bei fehlerfreier Übertragung abwechselnd 0 und 1. Das bedeutet, dass - wie bei Carat und Gastro-Service üblich - jeder einzelne Satz bestätigt wird. Die sog. Fenstergröße ist Eins, d.h. jedes Gerät muss sich maximal einen gesendeten, aber noch nicht bestätigten Datensatz merken.

Das Ändern eines Zählers von 0 auf 1 oder umgekehrt wird der Einfachheit halber im folgenden als Erhöhen bezeichnet.

Der Zähler Nr des einen Geräts entspricht dem Zähler Ns des anderen Geräts, und umgekehrt.

Bezeichnet man die Zähler eines Geräts A als NsA und NrA und die Zähler eines verbundenen Geräts B als NsB und NrB, dann nummerieren -

NsA und NrB von A gesendete, von B empfangene Datensätze

NsB und NrA von B gesendete, von A empfangene Datensätze

Die beiden Paare laufen prinzipiell unabhängig voneinander: Bei jeder fehlerfreien Übertragung von A an B werden  $Ns_A$  und  $Nr_B$  erhöht; bei jeder fehlerfreien Übertragung von B an A werden  $Ns_B/Nr_A$  erhöht.

### 3.2.2 Verarbeitung der Zähler $Nr$ und $Ns$ durch ein Gerät

Aus der Sicht eines Geräts gibt es insgesamt vier Zähler: Zwei lokal gespeicherte, und zwei mit einem Satz empfangene:

LocalNs - der lokale  $Ns$ , der mit dem letzten Satz gesendet wurde

LocalNr - der lokale  $Nr$ , der mit dem letzten Satz gesendet wurde

OtherNs - der im Satz empfangene  $Ns$  des anderen Geräts

OtherNr - der im Satz empfangene  $Nr$  des anderen Geräts

Im Zuge der Initialisierung werden LocalNs und LocalNr auf Null gesetzt. Vor dem ersten Pollen kann der Master zur Beschleunigung seinen  $Nr$  auf 1 setzen.

Nach dem Empfang eines Satzes geht jedes Gerät vor wie folgt:

#### 1) Vergleich LocalNr - OtherNs:

Bei Gleichheit wurde der erwartete Datensatz empfangen. Deshalb wird -

- der Datensatz verarbeitet
- der eigene Zähler LocalNr erhöht

Bei Ungleichheit hat das andere Gerät die letzte Bestätigung nicht empfangen und deshalb den letzten Datensatz wiederholt. Deshalb wird -

- der Datensatz ignoriert
- der eigene Zähler  $Nr$  unverändert gelassen

#### 2) Vergleich LocalNs - OtherNr:

Bei Ungleichheit ist alles in Ordnung - das andere Gerät fordert einen neuen Datensatz an. Entsprechend wird -

- der eigene Zähler LocalNs erhöht
- der zuletzt gesendete Datensatz gelöscht
- ein neuer Datensatz zum Senden vorbereitet

Bei Gleichheit hat das andere Gerät den letzten Satz nicht verstanden und fordert Wiederholung. Daher wird -

- der eigene Zähler LocalNs unverändert gelassen
- der zuletzt gesendete Datensatz wiederholt

Wenn es keinen zuletzt gesendeten Datensatz gibt, wird ein leerer Datensatz übertragen.

## 3) Senden:

Ein Satz mit LocalNr entsprechend 1) und LocalNs entsprechend 2) sowie einem Datensatz entsprechend 2) wird gesendet. Der gesendete Datensatz muß aufbewahrt werden, damit er ggf. wiederholt werden kann.

## 3.2.5 Beispiele

Die folgenden Beispiele gehen von zwei Geräten A und B aus. Als allgemeinsten Fall wird angenommen, dass in beiden Richtungen zugleich Datensätze übertragen werden.

A überträgt an B die Datensätze A1, A2, A3 ..., von denen keiner verloren gehen oder doppelt übernommen werden darf.

Umgekehrt sendet B an A die Datensätze B1, B2, B3 ... mit denselben Forderungen.

Beispiel 1: Fehlerfreie Kommunikation.

Aktivität	NsA	NrA	A -> B	A <- B	NsB	NrB
Initialisierung	0	1		0	0	
A sendet	0	1	<SO> "1" A1			
B vergleicht NrB-NsA gleich: B verarbeitet Daten und erhöht NrB						1
B vergleicht NsB-NrA ungleich: B erhöht NsB & nimmt neuen Daten B1						1
B sendet				<SO> "3" B1	1	1
A vergleicht NrA-NsB gleich: A verarbeitet Daten und erhöht NrA			0			
A vergleicht NsA-NrB ungleich: A erhöht NsA & nimmt neue Daten A2			1			
A sendet	1	0	<SO> "2" A2			
B vergleicht NrB-NsA gleich: B verarbeitet Daten und erhöht NrB						0
B vergleicht NsB-NrA ungleich: B erhöht NsB & nimmt neuen Daten B1						0
B sendet				<SO> "0" B2	0	0
A vergleicht NrA-NsB gleich: A verarbeitet Daten und erhöht NrA			1			

A vergleicht NsA-NrB ungleich:  
A erhöht NsA & nimmt neuen Daten A3 0

Beispiel 2: B kann A nicht empfangen.

Aktivität	NsA NrA	A -> B	A <- B	NsB NrB
Initialisierung	0 1		0 0	
A sendet	0 1	<SO> "1" A1		
B versteht nicht und schweigt				
A wiederholt nach Timeout	0 1	<SO> "1" A1		
B vergleicht NrB-NsA gleich: B verarbeitet Datensatz und erhöht NrB				
				1
B vergleicht NsB-NrA ungleich: B erhöht NsB & nimmt neuen Datensatz B1				
				1
B sendet		<SO> "3" B1	1 1	
A vergleicht NrA-NsB gleich: A verarbeitet Datensatz und erhöht NrA 0				

Beispiel 3: A kann B nicht empfangen.

Aktivität	NsA NrA	A -> B	A <- B	NsB NrB
Initialisierung	0 1		0 0	
A sendet	0 1	<SO> "1" A1		
B vergleicht NrB-NsA gleich: B verarbeitet Datensatz und erhöht NrB				
				1
B vergleicht NsB-NrA ungleich: B erhöht NsB & nimmt neuen Datensatz B1				
				1
B sendet		<SO> "3" B1	1 1	
A versteht nicht				
A wiederholt nach Timeout	0 1	<SO> "1" A1		
B vergleicht NrB-NsA ungleich: B ignoriert die Daten und beläßt NrB				

B vergleicht NsB-NrA gleich:  
B beläßt NsB und nimmt alte Daten B1

B sendet <SO> "3" B1 1 1

A vergleicht NrA-NsB gleich:  
A verarbeitet Daten und erhöht NrA 0

A vergleicht NsA-NrB ungleich:  
A erhöht NsA & nimmt neuen Daten A2 1

.....

## 5 Vermittlung

### 5.1 Allgemeines

Im Zuge der Vermittlung werden Slaves zyklisch vom PC gepollt und antworten ohne Verzögerung. Slaves übernehmen nie von sich aus die Schnittstelle.

Durch Satznummerierung fallen eigene Bestätigungs-Sätze weg. Weiters fallen Select-Sätze weg, da die Adresse des Slaves grundsätzlich in jedem Satz übermittelt wird.

Die vier verwendeten Befehlsbytes im <Command>-Feld sind -

- <SI> .. Anfordern von Betriebsdaten
- <SO> .. Senden von Betriebsdaten
- <DC1> .. Anfordern von Konfigurationsdaten
- <DC2> .. Senden von Konfigurationsdaten

Alle Befehle können auch mit leerem Datensatz gesendet werden, und haben dann die folgende Bedeutung.

- <SI> <> Pollen (Buchungsabfrage) oder Anfordern des nächsten Elements
- <SO> <> Noch keine Daten (bei zeitlichen Schwierigkeiten)
- <DC1> <> Anfordern des nächsten Elements
- <DC2> <> Keine weiteren Daten

## 5.2 Betriebsdaten

### 5.2.1 Allgemeines

#### 5.2.1.1 Einleitung

Betriebsdaten werden unterschiedlich übertragen:

Erweitertes Format: zweibuchstabige Bezeichner, Delimiter ";"

Für alle anderen Übertragungen wird das erweiterte Format verwendet. Dabei werden Daten mit zweibuchstabigen Bezeichnern identifiziert. Mehrere Datenelemente werden durch Semikolons getrennt.

#### 5.2.1.2 Übersicht

B> Buchung (Bonieren) Produkt : Menge B>1:1,1205:17.50  
 BE Buchung (Entnahme) Produkt [ : Menge ] BE17  
 BF Freigabe (Credit) Produkt [ : Menge ] BF32

C> Buchung (Kanal Boni) Kanal : Menge C>1:1  
 CE Buchung (Kanal Entn) Kanal [ : Menge ] CE1  
 CF Freigabe (Credit) Kanal [ : Menge ] CF32

B! alle Kredite am Zapfkopf löschen

K# Kellner-Nummer Nummer K# 7

T# Tisch-Nummer Tischnummer T#41;

T\* Alle Tische - T\*; TO0

TO Tisch Öffnen Kellnernummer TO7

TO0 Tisch sperren

**!!!!** Argumente werden mit oder ohne führenden Nullen verstanden.

**!!!!** Argumente werden (je nach Geräte- und Softwareversion) mit oder ohne führenden Nullen gesendet.

### 5.2.3 Buchungen

#### 5.2.3.1 Übertragung von Buchungen vom Slave zum PC

Buchungen werden nach dem folgenden Schema vom Slave zum PC übertragen.

Aktivitäten	Master -> Slave	Master <- Slave
Master pollt	<SI>	
Slave antwortet: keine Daten		<SO>
Am Slave werden gebucht: B1, B2		
Master pollt	<SI>	
Slave antwortet B1		<SO> <B1>
Master pollt	<SI>	
Slave antwortet B2		<SO> <B2>
Master pollt	<SI>	
Slave antwortet: keine Daten		<SO>

Anmerkungen:

- 1) Das Schema ist vereinfacht insofern, als der Master im allgemeinen mehrere Slaves pollt.
- 2) Im Schema sind Fehlübertragungen nicht enthalten.

Buchungsformat:

Das Format der Buchung vom Slave zum Master ist -

<Kellner>;<Tisch>;<Buchung>

Dabei sind -

<Kellner> das Zeichen "K#" gefolgt von der Kellnernummer

<Tisch> das Zeichen "T#" gefolgt von der Tischnummer

<Buchung>

Buchungen werden übertragen wie folgt:

Bei Geräten die logische Produktnummern (1..9999) unterstützen  
(PDC 160/24 ab 2.00, F32 ab 2.00, Safe ab 2.00 , Multimax):

B> <Nummer> [ : <Menge> [ : Preis ] ] { , <Nr> [ : Menge [ : Preis ] ] }  
BE <Nummer> [ : <Menge> [ : Preis ] ] { , <Nr> [ : Menge [ : Preis ] ] }

Bei allen anderen Geräten physische Produktnummer (1..160)

C> <Nummer> [ : <Menge> [ : Preis ] ] { , <Nr> [ : Menge [ : Preis ] ] }  
CE <Nummer> [ : <Menge> [ : Preis ] ] { , <Nr> [ : Menge [ : Preis ] ] }

<Nummer> gibt dann die Kanalnummer (1..250) des Geräts an, und der Empfänger errechnet aufgrund einer Tabelle die Produktnummer.

Verwendet werden -

B> und C> bei Bonieren

BE und CE bei Entnahme (Dosieren etc.)

Menge und Preis sind optional.

## 5.2.3.2 Freigabe von Entnahmen auf Zapfköpfen

Das Format der Freigabe vom Master zum Slave ist -

<Kellner> <Freigabe>

Dabei sind -

<Kellner> das Zeichen "K#" gefolgt von der Kellnernummer

<Freigabe>

Freigaben werden übertragen wie folgt:

Bei Geräten die logische Produktnummern unterstützen (160/24 ab 2.00, F-32 ab 2.00, Safe ab 2.00 Multimax): Menge ist optional.

BF <Nummer> [ : <Menge> ] { , <Nr> [ : Menge ] }

Aktivitäten	Master	Slave
-------------	--------	-------

Master sendet	<SO> K#..;BF..	
---------------	----------------	--

Slave bestätigt		<SI>
-----------------	--	------

Bei allen Geräten möglich:

CF <Nummer> [ : <Menge> ] { , <Nr> [ : Menge ] }

<Nummer> gibt dann die Kanalnummer (1..128) des Geräts an  
Menge ist optional.

Aktivitäten	Master	Slave
-------------	--------	-------

Master sendet	<SO> K#..;CF..	
---------------	----------------	--

Slave bestätigt		<SI>
-----------------	--	------

## 5.2.5.9 Tisch Öffnen.

In den aktuellen Schankelektroniken von TBT sind 3 Betriebsarten für die Tischeingabe einstellbar

1- nicht reserviert

2- bei der Eingabe der Tischnummer immer anfragen (damit übernimmt das Kassensystem die Verwaltung der Sperren)

Slave meldet Tischeingabe und wartet auf Antwort

<SO> T#nnnn; TOkk

Master sendet aktuellen Status

<SO> T#nnnn; TOkk

bei Übereinstimmen von Tisch- und Kellnernummer kann gezapft werden

Am Slave wird ein Tisch erst gültig, wenn die Eröffnung vom Master empfangen wurde. Ein Timeout ist nicht vorgesehen; bei Fehlern in der Übertragung muß ggf. der Schlüssel gezogen werden.

3- Blacklist für Tisch 1-999 im WTS (dann muss das Kassensystem den Tisch sperren und beim abrechnen wieder freigeben). Die Tischeingabe ist erfolgreich wenn in der Blacklist 0 oder die eigene Kellnernummer steht.

Master sendet Tisch nnnn belegt von Kellner kk

<SO> T#nnnn; TOkk

Bei Abrechnung des Tisches muß der Master Tisch nnnn frei senden

<SO> T#nnnn; TO0

Initialisierung: Der Master löscht alle Tischöffnungen an den Slaves.

Aktivitäten	Master	Slave
-------------	--------	-------

Master schließt	<SO> T*;	TO0
-----------------	----------	-----

## 6 Beispiele :

Übertragung einer Entnahme mit Kanalnummer:

Pollen des Masters

```
'Z' NUL bc SI 'D' '1' NS CKS CR
```

```
<5A><00><..><0F><44><31><..><..><0D>
      |
      Zapfkopfnummer
```

Antwort des Slaves

```
'Z' NUL bc SO 'D' '1' NS 'K' '#' kkk ';' 'T' '#' ttt ';' ---->
```

```
<5A><00><..><0E><44><31><..><4B><23><...><3B><54><22><...><3B>
      |           |           |
      Zapfkopfnummer |           Tischnummer
                    |
                    Kellnernummer (1-64)
```

```
----> 'C' 'E' ccc CKS CR
      <43><46><...><..><0D>
              |
              Kanalnummer (1..128)
```

Die Argumente für Kellner (kkk), Kanal (ccc) und Tisch (ttt) können mit oder ohne führende Nullen gesendet werden. Bei Entnahmen ist die Anzahl 1 und wird nicht gesendet. Die Zuordnung Kanal-Produkt muß in diesem Fall am Master durchgeführt werden.

Übertragung einer Entnahme mit Produktnummer (160/24, F-32 Version 2):

Pollen des Masters

```
'Z' NUL bc SI 'D' '1' NS CKS CR
```

```
<5A><00><..><0F><44><31><..><..><0D>
      |
      Zapfkopfnummer
```

Antwort des Slaves

'Z' NUL bc SO 'D' '1' NS 'K' '#' kkk ';' 'T' '#' ttt ';' ----->

```

<5A><00><...><0E><44><31><...><4B><23><...><3B><54><23><...><3B>
      |                               |                               |
      Zapfkopfnummer                 Kellnernummer(1-64)
                                     |
                                     Tischnummer
    
```

```

-----> 'B' 'E' ppp CKS CR
          <42><46><...><...><0D>
          |
          Produktnummer (1..1999)
    
```

Die Argumente für Kellner (kkk), Produkt (ppp) und Tisch (ttt) können mit oder ohne führende Nullen gesendet werden. Bei Entnahmen ist die Anzahl 1 und wird nicht gesendet. Die Zuordnung Kanal-Produkt muß in diesem Fall am Zapfkopf (oder über GS-Gastro Service) eingegeben werden.

Freigabe vom Master zum Slave (Kanalnummer) ist -

'Z' NUL bc SO 'D' '1' NS 'K' '#' kkk ';' 'C' 'F' ccc ':' mmm CKS CR

```

<5A><00><...><0E><44><31><...><4B><23><...><3B><43><46><...><3A><...><...><0D>
      |                               |                               |
      Zapfkopfnummer                 Kellnernummer(1-64)
                                     |
                                     Kanalnummer (1-128)
                                     |
                                     Menge
    
```

Freigabe vom Master zum Slave (Produktnummer)ist -

'Z' NUL bc SO 'D' '1' NS 'K' '#' kkk ';' 'B' 'F' ccc ':' mmm CKS CR

```

<5A><00><...><0E><44><31><...><4B><23><...><3B><43><46><...><3A><...><...><0D>
      |                               |                               |
      Zapfkopfnummer                 Kellnernummer(1-64)
                                     |
                                     Kanalnummer (1-128)
                                     |
                                     Menge
    
```

Bestätigung vom Slave

'Z' NUL bc SI 'D' '1' NS CKS CR

```

<5A><00><...><0F><44><31><...><...><0D>
      |
      Zapfkopfnummer
    
```

Die Argumente für Kellner (kkk), Kanal (ccc) und Menge (mmm) können mit oder ohne führende Nullen gesendet werden.

Die Menge kann auch negativ sein

Alle Kredite löschen vom Master zum Slave ist -

**'Z' NUL bc SO 'D' '1' NS 'B' '!' CKS CR**

<5A><00><..><0E><44><31><..><42><21><..><0D>

	Befehl zum löschen
Zapfkopfnummer	

Bestätigung vom Slave

**'Z' NUL bc SI 'D' '1' NS CKS CR**

<5A><00><..><0F><44><31><..><..><0D>

Zapfkopfnummer

Für Die Berechnung von Bytecount, Satznummerierung und Checksumme siehe Kap 4