

## 3 Automatik Extern

### 3.1 Allgemein

Bei verketteten Produktionsstraßen ist es nötig, Roboterprozesse von zentraler Stelle aus starten zu können.

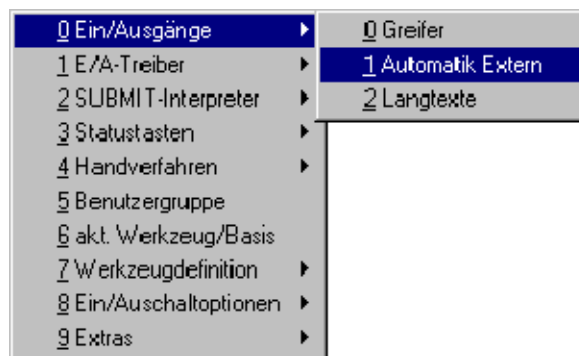
Über die Schnittstelle “Automatik Extern” kann ein Leitrechner mit der Robotersteuerung kommunizieren und verschiedene Roboterprozesse auslösen. Ebenso kann die Robotersteuerung Informationen über Betriebszustände und Störmeldungen an den Leitrechner übermitteln.

Bei der KR C1 wird dies alles durch den automatischen Anlagenanlauf, das technologiespezifische Organisationsprogramm CELL . SRC und durch die Funktionen des Moduls P00 realisiert.

### 3.2 Ein- und Ausgangssignale konfigurieren

**Konfig.**

Den Signalen der Schnittstelle “Automatik Extern” müssen physikalische Ein- und Ausgänge der Robotersteuerung zugeordnet werden. Wählen Sie dazu aus dem Menü “Konfig.” die Option “Ein/Ausgänge” aus.



Es öffnet sich das entsprechende Zustandsfenster, in dessen Eingabefelder die Nummern der Ein- bzw. Ausgänge der Robotersteuerung eingetragen werden, die mit den Signalen der Schnittstelle verknüpft werden.



**Nehmen Sie Änderungen nur im Ausnahmefall vor.**

**AUTO EXTERN - Konfiguration**

Eingänge    Ausgänge

Signal	Wert
PGNO_TYPE	1
PGNO_LENGTH	8
PGNO_FBIT	33
PGNO_PARITY	41
PGNO_VALID	42
EXT_START	1026
MOVE_ENABLE	1025
CONF_MESS	1026
DRIVES_ON	140
DRIVES_OFF	1025

Die Eingänge werden angezeigt

Hier wurde das Signal "DRIVES\_ON" mit dem Eingang "140" verknüpft

Liegt am Eingang der Wert "TRUE" an, wird das entsprechende Signal gesetzt

Signal, welches gesetzt wird, wenn am entsprechenden Eingang "TRUE" anliegt

**AUTO EXTERN - Konfiguration**

Eingänge    Ausgänge

Signal	Wert
STOPMESS	1010
PGNO_REQ	33
APPL_RUN	34
PERI_RDY	1012
ALARM_STOP	1013
USER_SAF	1011
T1	993
T2	994
AUT	995
EXTERN	140
ON_PATH	147
PRO_ACT	1021
IN_HOME	1000
ERR_TO_PLC	35

Die Ausgänge werden angezeigt

Hier wurde das Signal "ON\_PATH" mit dem Ausgang "147" verknüpft

Gewünschter Ausgang, an dem der Wert "TRUE" anliegt, wenn das entsprechende Signal gesetzt wird

Wird dieses Signal ausgegeben, wird der angegebene Ausgang auf "TRUE" gesetzt.

In der Softkeyleiste stehen die folgenden Optionen zur Auswahl:

**Tab +**

Zum Umschalten zwischen Ein- und Ausgängen der Automatik-Extern-Schnittstelle.

**Ändern**

Wollen Sie einem Eintrag eine andere Nummer zuweisen, wählen Sie den gewünschten Ein- bzw. Ausgang mit den Cursor-Tasten an und drücken den Softkey "Ändern". Daraufhin können Sie über den Nummernblock die gewünschte Nummer eingeben.

AUT	995
EXTERN	140
ON_PATH	147
PRO_ACT	1021
IN_HOME	1000

Geben Sie die gewünschte Ein- bzw. Ausgangsnunmer ein

**OK**

Zum Speichern Ihrer Änderungen betätigen Sie den Softkey "OK" oder die Eingabe-Taste, wobei die Werte auf ihre Gültigkeit überprüft werden.



Werden Änderungen in der Datei "\$MACHINE.DAT" vorgenommen, wird zur Datenübertragung der Submit-Interpreter kurzfristig abgewählt und nach der Sicherung automatisch wieder angewählt.

**Abbruch**

"Abbruch" oder die ESC-Taste bricht den Vorgang ohne Sicherung der Daten ab.



Eine Beschreibung der Signale und deren Wertebereiche finden Sie in Abschnitt **3.6**. Siehe auch Abschnitt **3.9** (Beispielkonfiguration).

### 3.3 Automatischer Anlagenanlauf

Ist die E/A-Schnittstelle durch Setzen der Systemvariablen `$I_O_ACT` auf den Wert `TRUE` aktiv geschaltet, wird der Ausgang `$_I_O_ACTCONF` als Rückmeldung ebenfalls auf `TRUE` geschaltet. Sind alle weiteren Startbedingungen erfüllt, so kann durch ein Signal auf der Leitung `$EXT_START` das Programm `CELL.SRC` gestartet werden.

Natürlich kann das Programm `CELL.SRC` zu jeder Zeit auch von der Bedienoberfläche aus gestartet werden.

Für den automatischen Anlagenanlauf muß der Systemvariablen `$PRO_I_O` in der Datei `"C:\KRC\Roboter\KRC\Steu\MaDa\$_CUSTOM.DAT"` folgender Wert zugewiesen werden:

```
CHAR $PRO_I_O[ ]="/R1/SPS( )"
```



Nach dem Hochfahren der Steuerung wird immer versucht das Programm auszuführen, daß in `$PRO_I_O` bezeichnet wurde.

Das Programm `SPS.SUB` wählt `CELL.SRC` an und ist damit beendet. Es liegt zwar auf der Roboterseite, wird aber vom Submit-Interpreter (Steuerungsebene) bearbeitet.



Damit die E/A-Schnittstelle überhaupt aktiviert werden kann, muß der Eingang gesetzt werden, welcher durch Signalvereinbarung der Variablen `$I_O_ACT` zugewiesen wurde.

### 3.4 Technologiespezifisches Organisationsprogramm CELL.SRC

*Anweisung zum Einbinden der benutzerdefinierten, externen Unterprogramme ...*

```
;EXT EXAMPLE1 ( )  
;EXT EXAMPLE2 ( )  
;EXT EXAMPLE3 ( )
```

*Initialisierungssequenz ...*

```
INIT  
BAS INI  
CHECK HOME  
PTP HOME Vel= 100 % DEFAULT  
AUTOEXT INI
```

*Schleifenbeginn ...*

```
LOOP
```

*Aufruf des Moduls P00, um die Programmnummer vom externen Leitrechner abzurufen ...*

```
P00 (#EXT_PGNO,#PGNO_GET,DMY[ ],0 )
```

*Kontrollstruktur in Abhängigkeit von der empfangenen Programmnummer ...*

```
SWITCH PGNO
```

*Wenn Programmnummer PGNO = 1 ...*

```
CASE 1
```

*... dem Leitrechner den Empfang der Programmnummer mitteilen ...*

```
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[ ],0 )
```

*... und benutzerdefiniertes Programm EXAMPLE1 aufrufen*

```
;EXAMPLE1 ( )
```

*Wenn Programmnummer PGNO = 2 ...*

```
CASE 2
```

*... dem Leitrechner den Empfang der Programmnummer mitteilen ...*

```
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[ ],0 )
```

*... und benutzerdefiniertes Programm EXAMPLE2 aufrufen*

**;EXAMPLE2 ( )**

*Wenn Programmnummer PGNO = 3 ...*

**CASE 3**

*... dem Leitrechner den Empfang der Programmnummer mitteilen ...*

**P00 (#EXT\_PGNO,#PGNO\_ACKN,DMY[ ],0 )**

*... und benutzerdefiniertes Programm EXAMPLE3 aufrufen*

**;EXAMPLE3 ( )**

*Wurde für die vom Leitrechner übermittelte Programmnummer kein CASE-Zweig gefunden,*

**DEFAULT**

*erfolgt hier eine Fehlerbehandlung ...*

**P00 (#EXT\_PGNO,#PGNO\_FAULT,DMY[ ],0 )**

*Ende der Kontrollstruktur ...*

**ENDSWITCH**

*Schleifenende ...*

**ENDLOOP**

*Programmende ...*

**END**

### 3.5 Das Modul P00 (AUTOMATIK-EXTERN)

Im Modul P00 befinden sich die Funktionen für die Übermittlung von Programmnummern über einen Leitrechner. In diesem globalen Unterprogramm sind die Funktionen `INIT_EXT`, `EXT_PGNO`, `CHK_HOME` und `EXT_ERR` zusammengefaßt.

#### 3.5.1 Die Funktion `EXT_PGNO`

Diese Funktion übernimmt die komplette Signal-Handhabung für die Übermittlung von Programmnummern über einen Leitrechner.

Sie kann mit einem der drei folgenden Parameter aufgerufen werden:

<code>#PGNO_GET</code>	Anforderung einer Programmnummer
<code>#PGNO_ACKN</code>	Mitteilen des Erhalt einer Programmnummer
<code>#PGNO_FAULT</code>	Fehlerbehandlung

##### 3.5.1.1 Anforderung einer Programmnummer beim Leitrechner

###### `EXT_PGNO (#PGNO_GET)`

Erkennt der Leitrechner eine Programmnummern-Anforderung auf der Leitung `PGNO_REQ`, so legt er die Programmnummer als Binärwert an die dafür vorgesehenen Eingänge der Robotersteuerung.

Zur Erhöhung der Übertragungssicherheit kann vom Leitrechner zusätzlich zur Programmnummer noch ein Paritätsbit, `PGNO_PARITY`, übergeben werden. Stehen die Signalpegel stabil an, so fordert der Leitrechner durch das Setzen der Leitung `PGNO_VALID` oder `EXT_START` die Robotersteuerung auf, die Programmnummer einzulesen. Die Funktion `EXT_PGNO` berechnet nun aus der empfangenen Programmnummer die Parität und vergleicht sie mit dem angelegten Paritätsbit. Bei positivem Ergebnis gibt die Funktion die empfangene Programmnummer als ganzzahligen Wert zurück. Stimmen empfangene und berechnete Parität jedoch nicht überein, so wird die Programmnummer auf den Wert "0" gesetzt. Im Meldungsfenster des KCP wird eine Fehlermeldung ausgegeben.



Da beim Auftreten eines Paritätsfehlers die Programmnummer immer auf den Wert Null gesetzt wird, darf dieser Wert natürlich nicht als gültige Programmnummer in `CELL.SRC` verwendet werden !

##### 3.5.1.2 Mitteilen des Erhalts einer gültigen Programmnummer

###### `EXT_PGNO (#PGNO_ACKN)`

Wurde die Programmnummer korrekt übertragen, so wird in der Kontrollstruktur in `CELL.SRC` versucht, dieser Programmnummer ein Anwenderprogramm zuzuordnen. Gelingt dies, so nimmt die Funktion die Programmnummer-Anforderung selbständig zurück. Sie signalisiert dies dem Leitrechner durch Setzen der Leitung `APPL_RUN`.

Im anderen Fall wird die nachfolgend beschriebene Funktion zur Fehlerbehandlung aufgerufen.

### 3.5.1.3 Fehlerbehandlung

#### **EXT\_PGNO (#PGNO\_FAULT)**

Wurde die Programmnummer nicht korrekt übertragen, d.h.

- (1) die Paritätsprüfung war nicht erfolgreich, oder
- (2) die BCD-Kodierung war falsch, besser gesagt: die Dekodierung führte zu keinem gültigen Ergebnis, oder
- (3) es war dieser Programmnummer kein Anwenderprogramm zugeordnet,

so zeigt die Funktion **EXT\_PGNO** über das Meldungsfenster des KCP einen Übertragungsfehler an. Die Leitung **PGNO\_REQ** wird zurückgesetzt. Dadurch wird dem Leitreechner mitgeteilt, daß die Übertragung fehlerhaft war.



Eine fehlerhafte Übermittlung kann vom Leitreechner durch einen Timeout festgestellt werden. Dieser Timeout wird mit dem Setzen der Leitung **PGNO\_VALID** gestartet. Sollte nach einer festgelegten Zeitdauer ( etwa 200 ms ) die Programmnummer-Anforderung auf der Leitung **PGNO\_REQ** nicht zurückgenommen werden, so muß bei der Übertragung ein Fehler aufgetreten sein. Der Leitreechner kann jetzt auf den Fehler reagieren.



### 3.5.2 Die Funktion EXT\_ERR

Mit dieser Funktion kann über acht festgelegte Ausgänge der Robotersteuerung eine vereinbarte Fehlernummer im Bereich 1 ... 255 zum Leitrechner übertragen werden. Zusätzlich werden die letzten 64 aufgetretenen Fehler im Ringspeicher ERR\_FILE zu einer genaueren Analyse aufbewahrt.

Um die Funktion EXT\_ERR nutzen zu können, müssen Sie die Datei p00.dat wie nachfolgend beschrieben editieren:

```
&ACCESS R
&COMMENT EXTERNAL package
DEFDAT P00
```

```
BOOL PLC_ENABLE = TRUE
```

Setzen Sie diesen Wert auf **TRUE**

```
INT I
```

```
INT F_NO=1
```

```
INT MAXERR_C = 1
```

Tragen Sie hier die **Anzahl** der **Steuerungsfehler** ein, für deren Übertragung Sie Parameter festgelegt haben

```
INT MAXERR_A = 1
```

Tragen Sie hier die **Anzahl** der **Applikationsfehler** ein, für deren Übertragung Sie Parameter festgelegt haben

```
DECL STOPMESS MLD
```

```
SIGNAL ERR $OUT[25] TO $OUT[32]
```

Legen Sie hier fest, über welche **Ausgänge** der Robotersteuerung der Leitrechner die Fehlernummer auslesen soll  
Im Beispiel sind dies die Ausgänge 25 bis 32

```
BOOL FOUND
```

```
STRUC PRESET INT OUT,CHAR PKG[3],INT ERR
```

```
DECL PRESET P[255]
```

Im folgenden Bereich müssen Sie die **Parameter** der Fehler eintragen:

**OUT –**

Fehlernummer, die zum Leitrechner übertragen werden soll

**PKG[ ] –**

Technologiepaket

**ERR –**

Fehlernummer im ausgewählten Technologiepaket

```
P[1]={OUT 2,PKG[ ]"P00",ERR 10}
```

```
...
```

```
P[127]={OUT 27,PKG[ ]"S00",ERR 11} eintragen
```

```
P[128]={OUT 12,PKG[ ]"CTL",ERR 1}
```

```
...
```

```
P[255]={OUT 25,PKG[ ]"CTL",ERR 10} eintragen
```

Im Bereich von **P[1] ... P[127]** können Sie nur **Applikationsfehler**

Im Bereich von **P[128] ... P[255]** können Sie nur **Steuerungsfehler**

```
STRUC ERR_MESS CHAR P[3],INT E
DECL ERR_MESS ERR_FILE[64]
ERR_FILE[1]={P[ ] "XXX",E 0}

...

ERR_FILE[64]={P[ ] "XXX",E 0}
ENDDAT
```

### 3.6 Signalbeschreibungen

Die Signale sind schreibgeschützt, können aber jederzeit gelesen oder in Programmen verwendet werden.

#### 3.6.1 Eingänge

##### 3.6.1.1 PGNO\_TYPE

Dies ist kein Eingang oder Signal, sondern eine Variable. Mit ihrem Wert wird festgelegt, in welchem Format die vom Leitrechner übermittelte Programmnummer eingelesen wird.

PGNO_TYPE	Einlesen als...	Bedeutung	Beispiele
1	Binärzahl	Die Programmnummer wird von der übergeordneten Steuerung als binär codierter <b>Integerwert</b> übergeben	0 0 1 0 0 1 1 1 => PGNO = 39
2	BCD-Wert	Die Programmnummer wird von der übergeordneten Steuerung als Binär Codierter <b>Dezimalwert</b> übergeben	0 0 1 0 0 1 1 1 <div style="display: flex; justify-content: space-around; width: 100%;"> <span>2</span> <span>7</span> </div> => PGNO = 27
3	"1 aus n" *1	Die Programmnummer wird von der übergeordneten Steuerung oder der Peripherie als "1 aus n" codierter Wert übergeben	0 0 0 0 0 0 0 1 => PGNO = 1 0 0 0 0 1 0 0 0 => PGNO = 4
*1 Bei diesem Übergabeformat werden die Werte von PGNO_REQ, PGNO_PARITY sowie PGNO_VALID nicht ausgewertet und sind somit ohne Bedeutung.			

##### 3.6.1.2 PGNO\_LENGTH

Auch dies ist kein Eingang oder Signal, sondern wieder eine Variable. Mit ihrem Wert wird die Bitbreite der vom Leitrechner übermittelten Programmnummer festgelegt.

PGNO\_LENGTH = 1...16

Beispiel:

PGNO\_LENGTH = 6 => die externe Programmnummer ist sechs Bit breit



Während PGNO\_TYPE den Wert 2 besitzt (Programmnummer als BCD-Wert einlesen), sind nur die Bitbreiten 4, 8, 12 und 16 zugelassen.

##### 3.6.1.3 PGNO\_FBIT

Eingang, der das erste Bit der Programmnummer darstellt.

PGNO\_FBIT = 1...1024 (PGNO\_LENGTH)

Beispiel:

PGNO\_FBIT = 5 => die externe Programmnummer beginnt mit \$IN[5]

#### 3.6.1.4 PGNO\_PARITY

Eingang, auf den das Paritätsbit vom Leitrechner übertragen wird.

Eingang	Funktion
negativer Wert	ungerade Parität
0	keine Auswertung
positiver Wert	gerade Parität



Während PGNO\_TYPE den Wert 3 besitzt (Programmnummer als "1 von n"-Wert einlesen), wird PGNO\_PARITY **NICHT** ausgewertet.

#### 3.6.1.5 PGNO\_VALID

Eingang, auf den das Kommando zum Einlesen der Programmnummer vom Leitrechner übertragen wird.

Eingang	Funktion
negativer Wert	Nummer wird mit der abfallenden Flanke des Signals übernommen
0	Nummer wird mit der ansteigenden Flanke des Signals an der Leitung EXT_START übernommen
positiver Wert	Nummer wird mit der ansteigenden Flanke des Signals übernommen



Während PGNO\_TYPE den Wert 3 besitzt (Programmnummer als "1 von n"-Wert einlesen), wird PGNO\_VALID **NICHT** ausgewertet.

#### 3.6.1.6 EXT\_START

Mit dem Setzen dieses Eingangs kann bei aktiver E/A-Schnittstelle ein Programm gestartet, bzw. wieder fortgesetzt werden.



Es wird nur die ansteigende Flanke des Signals ausgewertet.



Im Automatik-Extern-Betrieb gibt es keine SAK-Fahrt und damit auch keinen Programmhalt an der ersten programmierten Position. Dies gilt sowohl nach generatorischem Stop mit Verlassen der Bahn (z.B. Bedienerschutz) als auch beim Verlassen der Bahn von Hand.

Die erste anzufahrende Position ist in diesen Fällen die in \$POS\_RET gespeicherte Position vor der Unterbrechung. Demzufolge muß beim Setzen von EXT\_START vom Bediener sichergestellt werden, daß der Roboter auf dieser Position steht, bzw. diese gefahrlos erreichen kann.

Der erste Bewegungssatz muß ein PTP-Satz mit absoluter Zielpunktangabe sein. Dieser wird immer genau und mit **voller** Geschwindigkeit angefahren, wobei eine programmierte **Überschleifanweisung ignoriert** wird!

## 3.6.1.7 MOVE\_ENABLE

Dieser Eingang wird zur Kontrolle der Roboterantriebe durch den Leitreechner verwendet.

Signal	Funktion
TRUE	Handverfahren und Programmausführung möglich
FALSE	Stillsetzen aller Antriebe und Verriegelung aller aktiven Kommandos



Sind die Antriebe vom Leitreechner stillgesetzt worden, so erscheint im Meldungsfenster des KCP die Meldung "FAHRFREIGABE GESAMT". Das Bewegen des Roboters ist erst nach dem Löschen dieser Meldung und einem erneuten externen Startsignal wieder möglich.



Während der Inbetriebnahme wird die Variable für die Fahrfreigabe "\$MOVE\_ENABLE" häufig auf den Wert "\$IN[1025]" projiziert. Wird danach vergessen, einen anderen Eingang zu projizieren, ist kein externer Start möglich.

## 3.6.1.8 CHCK\_MOVENA

Besitzt die Variable \$CHCK\_MOVENA den Wert "FALSE", kann MOVE\_ENABLE umgangen werden. Der Wert der Variablen kann nur in der Datei "C:\KRC\Roboter\KRC\Steu\MaDa\OPTION.DAT" geändert werden.

Signal	Funktion
TRUE	Überwachung für MOVE-ENABLE ist wirksam
FALSE	Überwachung für MOVE_ENABLE ist deaktiviert



Um die Überwachung für MOVE\_ENABLE verwenden zu können, muß \$MOVE\_ENABLE auf den Eingang "\$IN[1025]" projiziert worden sein. Andernfalls hat "\$CHCK\_MOVENA" keinerlei Wirkung.

## 3.6.1.9 CONF\_MESS

Durch Setzen dieses Eingangs kann der Leitreechner aufgetretene Fehlermeldungen selbst löschen (quittieren).



Es wird nur die ansteigende Flanke des Signals ausgewertet.

Ein Quittieren der Fehlermeldungen ist selbstverständlich nur dann möglich, wenn die Störungsursache beseitigt wurde.

## 3.6.1.10 DRIVES\_ON

Durch einem High-Impuls von mind. 20 ms Dauer an diesem Eingang kann der Leitreechner die Roboterantriebe einschalten.



Ab dem Software-Stand 1.1.7 und bei Verwendung der Powermodule PM6-600, Fertigungsstände A, B oder C und PM0-600 Pro wird das Wiedereinschalten der Antriebe zum Schutz des Antriebsrelais K2 13-18.5 Sekunden lang nach dem letzten Einschalten der Antriebe verhindert.

Eine anstehende positive Flanke von DRIVES\_ON wird am Ende des Zeitfensters (nach 18.5 Sekunden) erkannt und die Antriebe werden verzögert eingeschaltet.

### 3.6.1.11 DRIVES\_OFF

Durch einem Low-Impuls von mind. 20 ms Dauer an diesem Eingang kann der Leitrechner die Roboterantriebe abschalten.

### 3.6.2 Ausgänge

#### 3.6.2.1 STOPMESS

Dieser Ausgang wird von der Robotersteuerung gesetzt, um dem Leitreechner das Auftreten einer Meldung anzuzeigen, die das Anhalten des Roboters erforderlich machte.

(z.B. NOT-AUS, Fahrfreigabe, Bedienerschutz, Sollgeschwindigkeit usw.)

#### 3.6.2.2 PGNO\_REQ

Mit einem Signalwechsel an diesem Ausgang wird der Leitreechner aufgefordert, eine Programmnummer zu übermitteln.



Es werden beide Flanken des Signals ausgewertet.

Während PGNO\_TYPE den Wert 3 besitzt (Programmnummer als "1 von n"-Wert einlesen), wird PGNO\_REQ **NICHT** ausgewertet.

#### 3.6.2.3 APPL\_RUN

Mit dem Setzen dieses Ausgangs teilt die Robotersteuerung dem Leitreechner mit, daß gerade ein Programm abgearbeitet wird.



Der Wert von APPL\_RUN darf nicht kleiner als "0" sein.

#### 3.6.2.4 PERI\_RDY

Mit Setzen dieses Ausgangs teilt die Robotersteuerung dem Leitreechner mit, daß die Roboterantriebe eingeschaltet sind.

#### 3.6.2.5 ALARM\_STOP

Dieser Ausgang wird beim Auftreten eines Not-Aus-Ereignisses zurückgesetzt.

#### 3.6.2.6 USER\_SAF

Dieser Ausgang wird beim Öffnen des Schutzgitter-Abfrageschalters (in der Betriebsart AUTO), bzw. beim Loslassen eines Zustimmungsschalters (in der Betriebsart TEST) zurückgesetzt.

#### 3.6.2.7 T1, T2, AUT, EXTERN

Diese Ausgänge werden gesetzt, wenn die entsprechende Betriebsart angewählt wurde.

#### 3.6.2.8 ON\_PATH

Dieser Ausgang ist gesetzt, solange sich der Roboter auf seiner programmierten Bahn befindet.

Nach der SAK-Fahrt wird der Ausgang ON\_PATH gesetzt. Dieser Ausgang bleibt solange gesetzt, bis der Roboter die Bahn verläßt, das Programm zurückgesetzt wird oder eine Satzanzahl durchgeführt wird. Das Signal ON\_PATH hat aber kein Toleranzfenster; sobald der Roboter die Bahn verläßt, wird dieses Signal zurückgesetzt.

### 3.6.2.9 NEAR\_POSRET

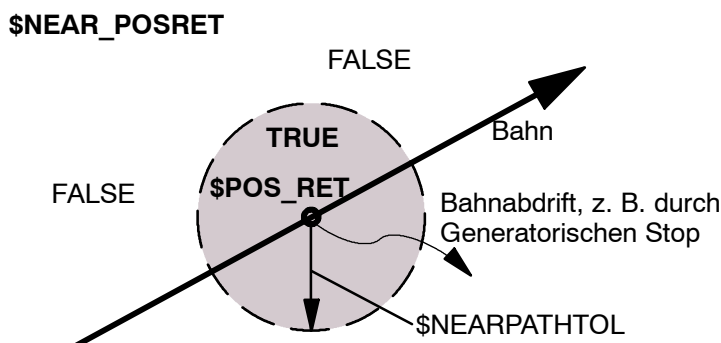
Über ein zweites Signal, **NEAR\_POSRET**, kann der Leitrechner feststellen, ob der Roboter innerhalb einer Kugel um die in **\$POS\_RET** gespeicherte Position steht. Der Radius der Kugel kann vom Anwender in der Datei **\$CUSTOM.DAT** über die Systemvariable **\$NEARPATHTOL** eingestellt werden.



Mit dieser Information kann der Leitrechner entscheiden, ob das Programm wieder gestartet werden darf oder nicht.

Die Rückkehrposition **\$POS\_RET** ist die Position, an welcher der Roboter die Bahn verlassen hat.

Beim Wechsel in die Betriebsart "Automatik Extern" wird überprüft, ob die Variable "**\$NEAR\_POSRET**" auf "TRUE" steht. Ist dies nicht der Fall, wird eine entsprechende Meldung im Meldungsfenster ausgegeben.



Mögliche Zustände von **NEAR\_POSRET**:

#### TRUE:

ON\_PATH ist gesetzt, oder wenn ON\_PATH nicht gesetzt ist: **\$POS\_RET** ist gültig und die Position ist innerhalb der Kugel um **\$POS\_RET**.

#### FALSE:

ON\_PATH ist zurückgesetzt und **\$POS\_RET** ist ungültig oder die Position ist außerhalb der Kugel um **\$POS\_RET**.

Einstellung:

Datei: **\$MACHINE.DAT**  
**SIGNAL \$NEAR\_POSRET \$OUT[xxx]**

### 3.6.2.10 PRO\_ACT

Dieser Ausgang ist immer dann gesetzt, wenn ein Prozeß bzw. die Programmbearbeitung auf Roboterebene aktiv ist.

Sein Signalzustand wird von der Systemvariablen **\$PRO\_STATE1** abgeleitet:

**\$PRO\_STATE1=#P\_ACTIVE** → **\$PRO\_ACT=TRUE**

alle anderen Prozeßzustände → **\$PRO\_ACT=FALSE**

Der Prozeß ist aktiv, solange ein Programm oder ein Interrupt bearbeitet wird. Die Programmbearbeitung am Ende des Programms wird erst dann inaktiv, wenn alle Impulsausgänge und Trigger abgearbeitet sind. Im Falle eines Fehlerstops ist zwischen den drei folgend beschriebenen Möglichkeiten zu unterscheiden:



- Wurden Interrupts aktiviert, aber zum Zeitpunkt des Fehlerstops nicht bearbeitet, so gilt der Prozeß als inaktiv (`PRO_ACT=FALSE`)
- Wurden Interrupts aktiviert und zum Zeitpunkt des Fehlerstops bearbeitet, so gilt der Prozeß solange als aktiv (`PRO_ACT=TRUE`), bis das Interruptprogramm abgearbeitet ist oder auf einen HALT läuft (`PRO_ACT=FALSE`)
- Wurden Interrupts aktiviert und das Anwenderprogramm läuft auf einen HALT, so gilt der Prozeß als inaktiv (`PRO_ACT=FALSE`). Ist nach diesem Zeitpunkt eine Interruptbedingung erfüllt, so gilt der Prozeß solange als aktiv (`PRO_ACT=TRUE`), bis das Interruptprogramm abgearbeitet ist oder auf einen HALT läuft (`PRO_ACT=FALSE`)

#### **3.6.2.11 IN\_HOME**

Dieser Ausgang teilt dem Leitreechner mit, ob sich der Roboter in seiner HOME-Position befindet.

#### **3.6.2.12 ERR\_TO\_PLC**

Durch das Setzen dieses Ausgangs teilt die Robotersteuerung dem Leitreechner mit, daß ein Steuerungs- oder Technologiefehler aufgetreten ist.



Diese Funktion ist nur aktiv, wenn `PLC_ENABLE` den Wert `TRUE` besitzt.

### 3.6.3 Sonstiges Variablen

#### 3.6.3.1 PGNO

In dieser Variablen legt das Programm `EXT_PGNO.SRC` die vom Leitrechner empfangene Programmnummer (unabhängig vom parametrisierten Datentyp) als ganzzahligen Wert ab.

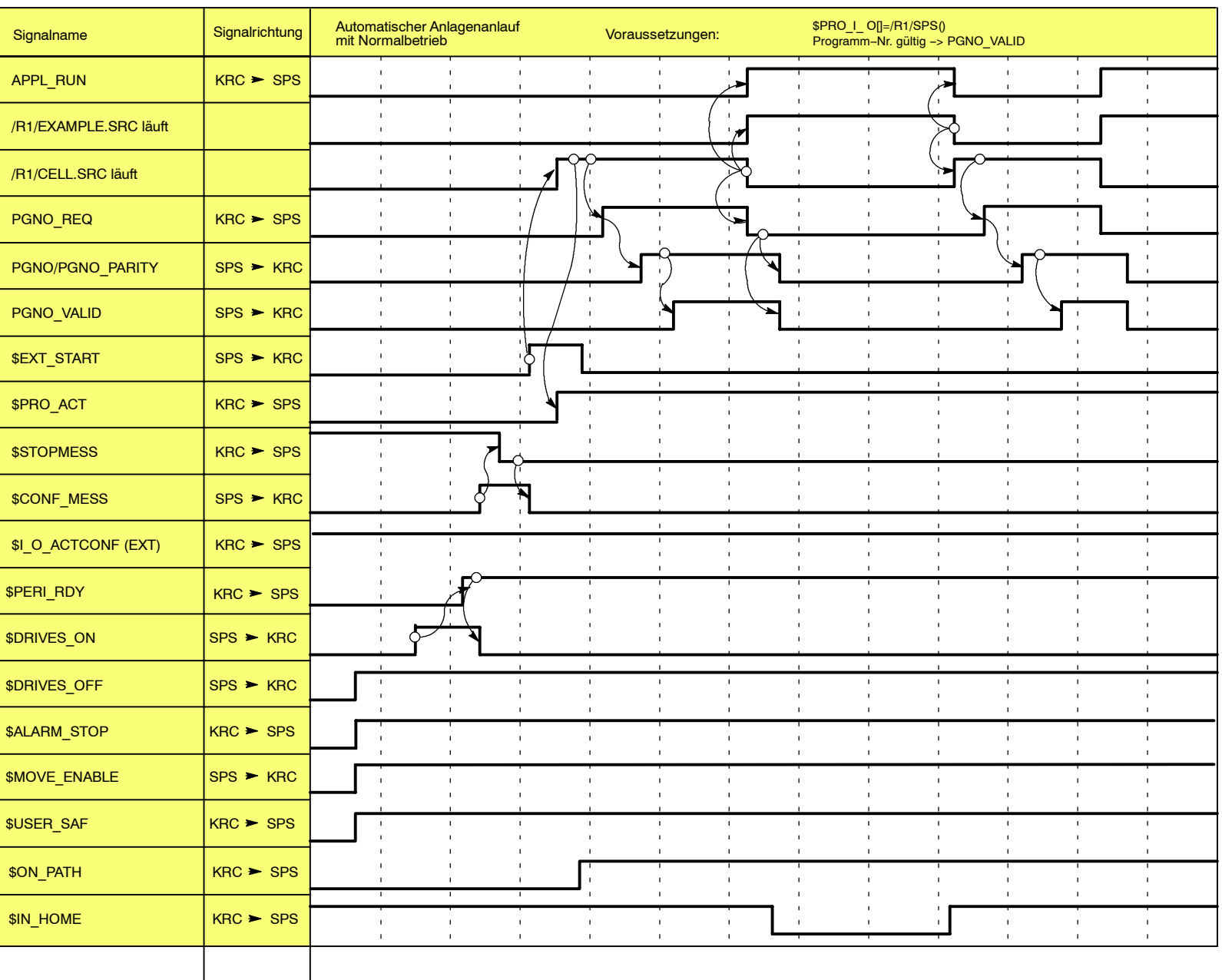
Das technologiespezifische Organisationsprogramm `CELL.SRC` ordnet mittels dieser Variablen der Programmnummer das entsprechende Anwenderprogramm zu.

#### 3.6.3.2 PGNO\_ERROR

Diese Variable dient der internen Fehlerverwaltung des Programms `EXT_PGNO.SRC` und darf nicht verwendet oder beschrieben werden!

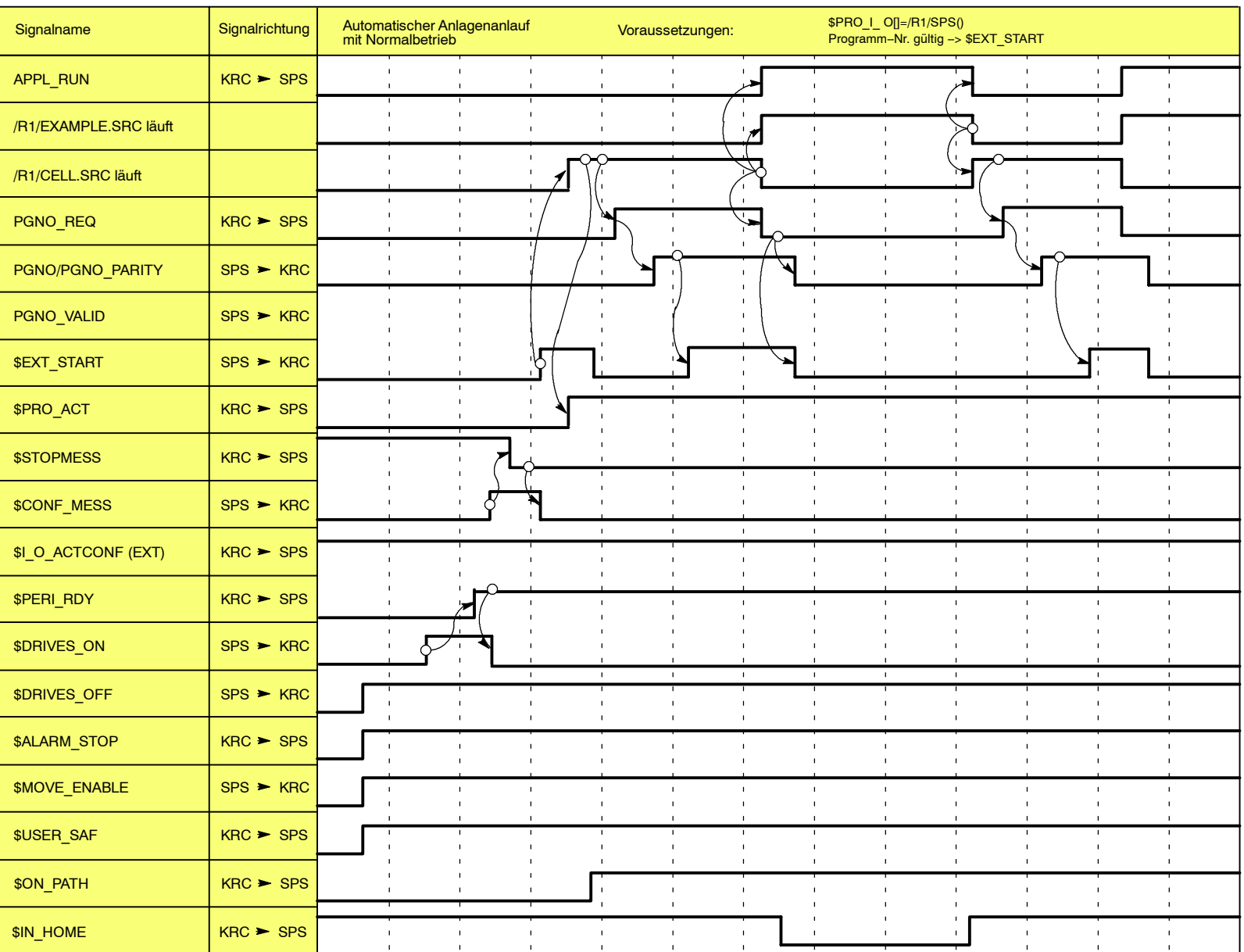
### 3.7      Signaldiagramme

#### 3.7.1    Auto. Anlagenanlauf und Normalbetr. mit Prog.-Nr.-Quitt. durch PGNO\_VALID

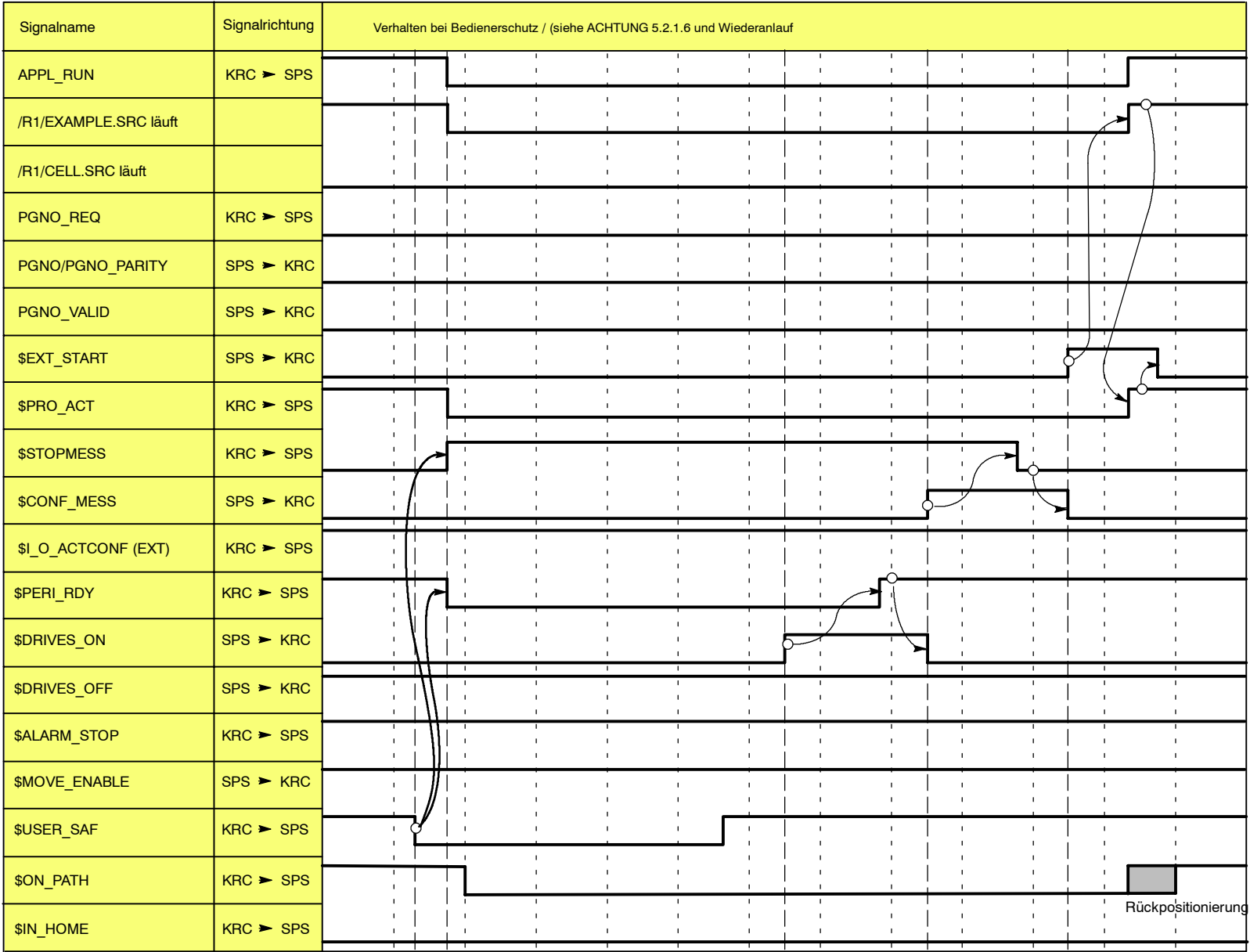




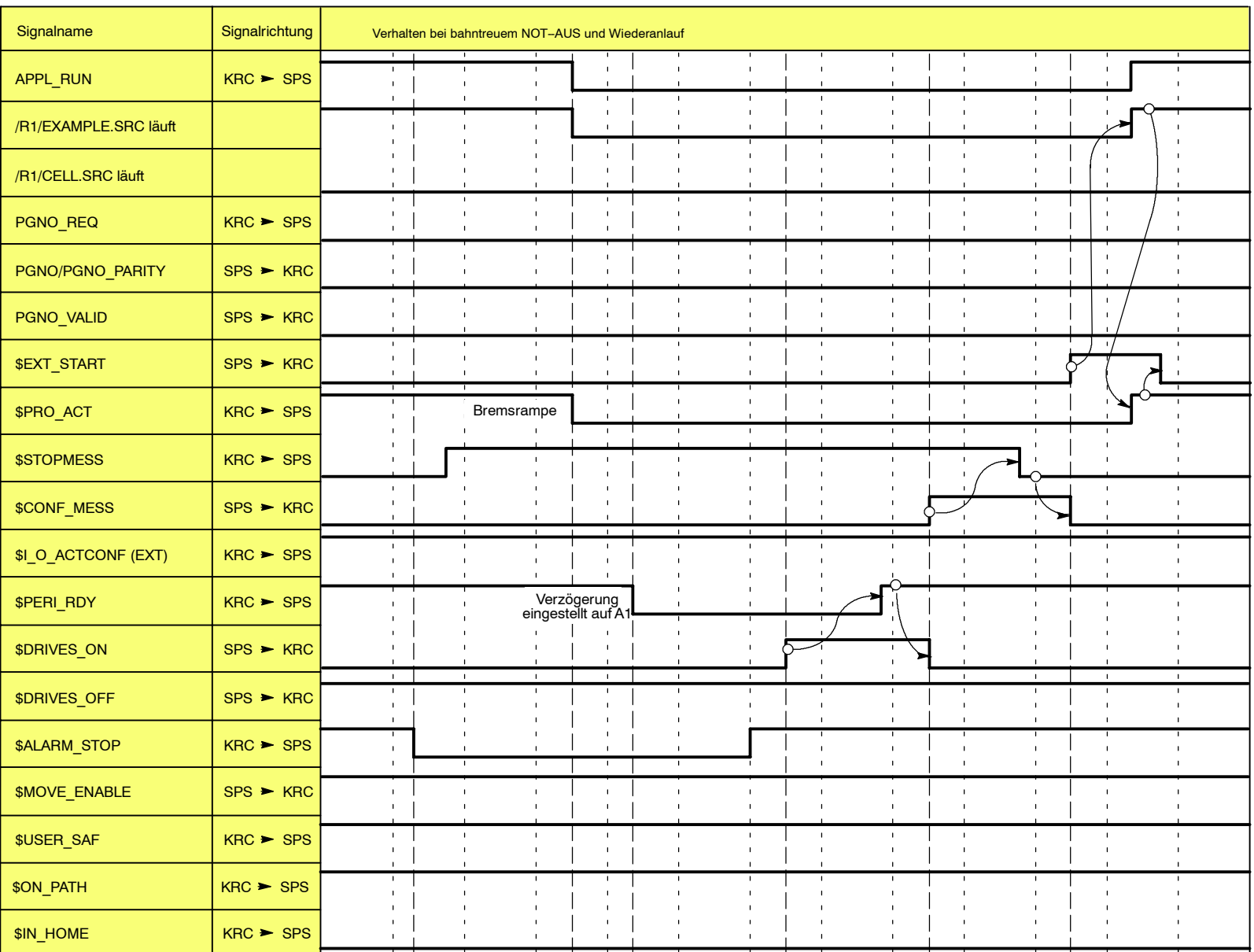
### 3.7.2 Auto. Anlagenanlauf und Normalbetr. mit Prog.-Nr.-Quitt. durch \$EXT\_START



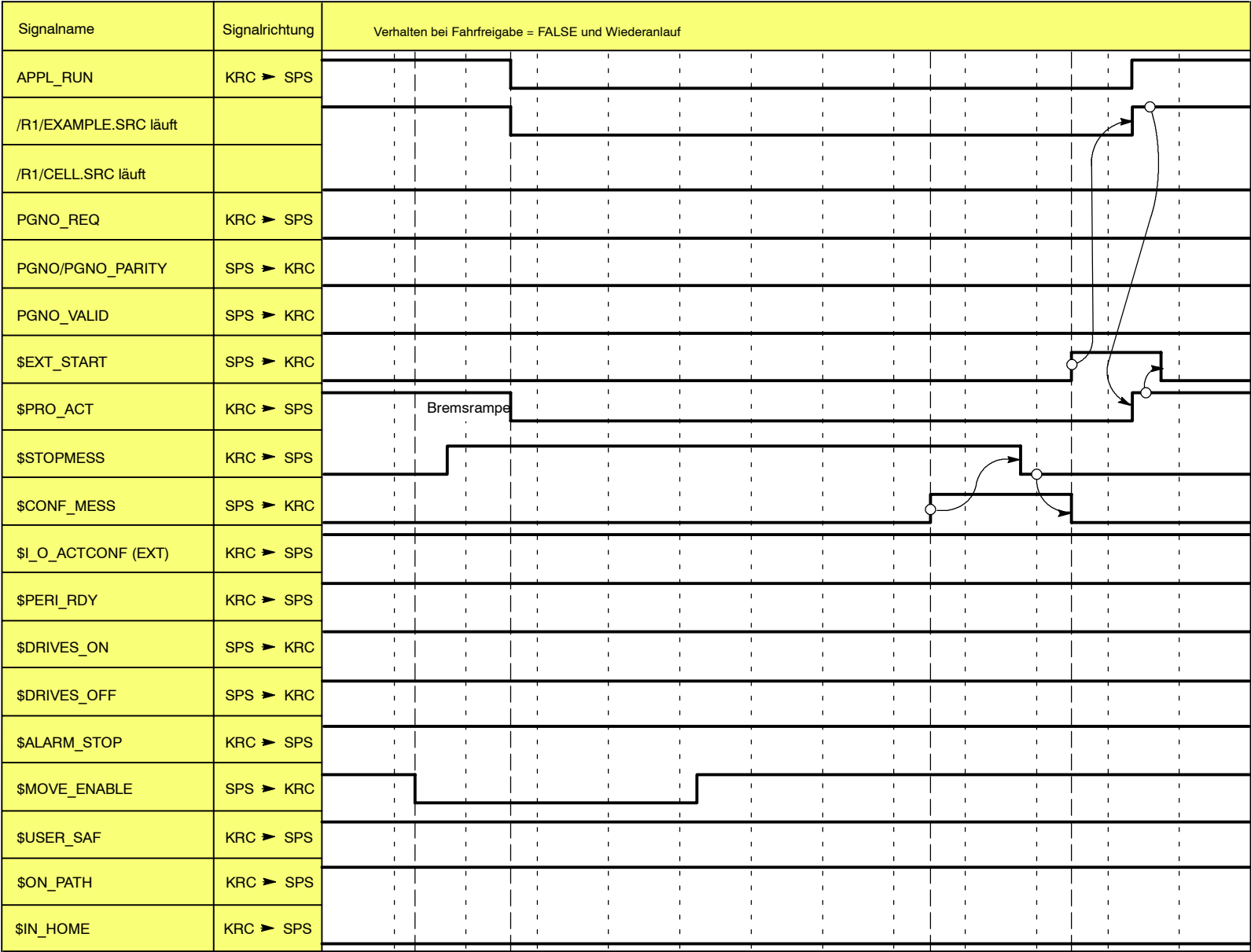
3.7.3      Wiederanlauf nach generat. Stop (Bedienerschutz und Wiederanlauf)



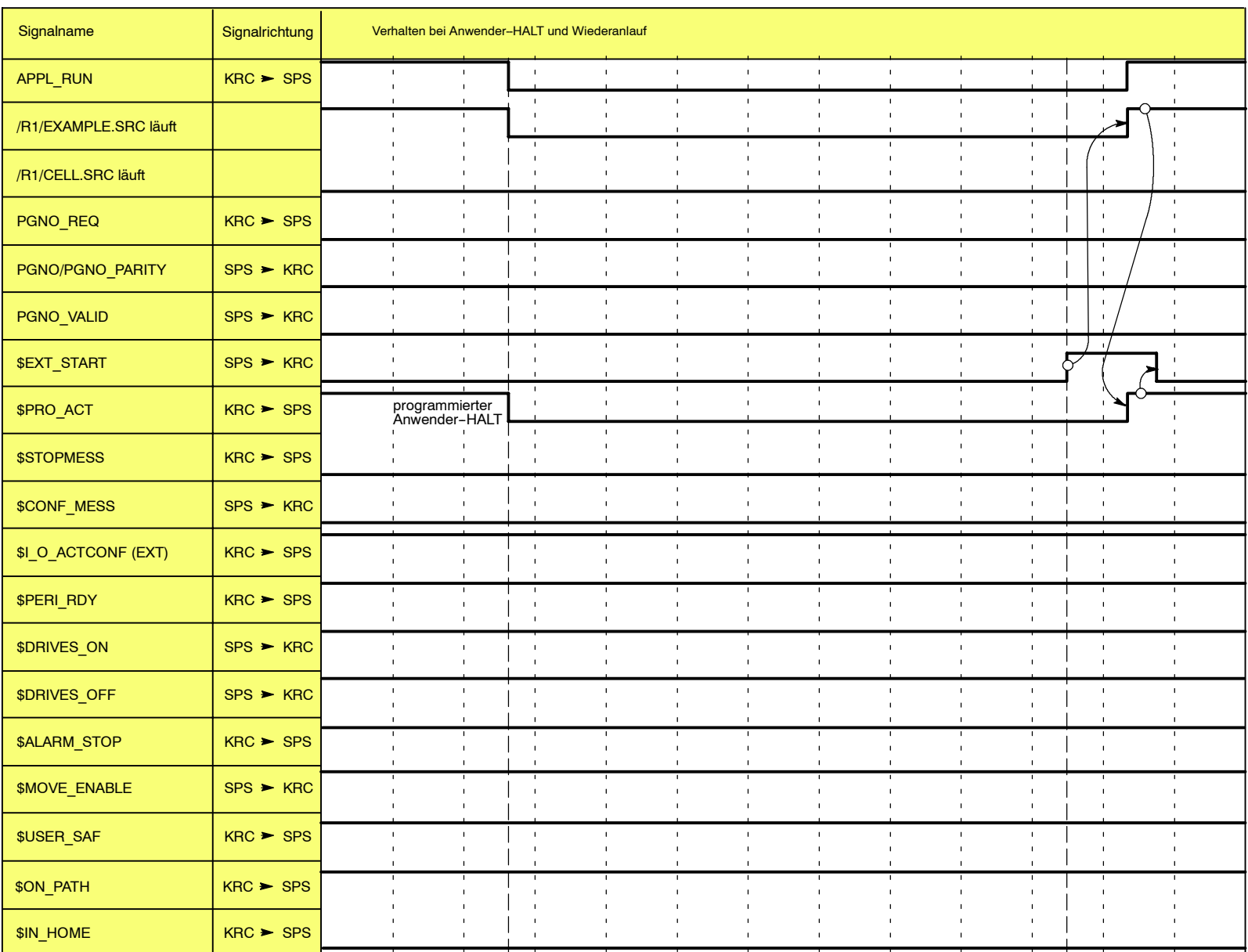
### 3.7.4 Wiederanlauf nach bahntreuem NOT-AUS



3.7.5 Wiederanlauf nach Fahrfreigabe



### 3.7.6 Wiederanlauf nach Anwender-HALT





## 3.8 Sonstiges

### 3.8.1 Wiederanlauf nach passivem Stop

Erfolgt ein passiver Stop vom KCP aus ohne Betriebsartenwechsel, so muß die Fehlermeldung "Q1370 :PASSIVER STOP" am KCP quittiert werden. Danach kann das Programm mit einem externen Start fortgesetzt werden.



Bei passivem Stop vom KCP und Betriebsartenwechsel muß von Hand rückpositioniert werden.

### 3.8.2 Schrittweise Programmausführung

Normalerweise ist in der Betriebsart "Automatik Extern" nur die Programmablaufart "#GO" erlaubt. Für bestimmte Fälle steht die Betriebsart "#MSTEP" zur schrittweisen Ausführung des Programms zur Verfügung.

Hierzu muß in der Datei "C:\KRC\Roboter\Init\Progress.ini" die folgende Zeile geändert werden:

```
[ FEATURES ]
.
.
MSTEP_IN_EXT = TRUE
.
.
```

Schon nach dem nächsten Hochfahren des Systems steht diese Option zur Verfügung. Zum Umschalten drücken Sie einfach den Statuskey für die Programmablaufart.

### 3.8.3 Geschwindigkeit zum Zurückfahren auf die programmierte Bahn

Hat der Roboter den programmierten Pfad verlassen, wird der Punkt an dem er die Bahn verlassen hat, mit reduzierter Geschwindigkeit angefahren. Das gleiche gilt, falls der Roboter eine SAK-Fahrt durchführt. Die Geschwindigkeit, mit der die Positionierung erfolgt, entspricht der Handverfahrgeschwindigkeit.

Als Voraussetzung muß in der Datei "C:\KRC\Roboter\Init\Progress.ini" die nachfolgende Zeile geändert werden:

```
[ FEATURES ]
.
.
SLOW_BCO_EXT = TRUE
.
.
```

Maßgebend für die Reduzierung der Achsverfahrgeschwindigkeit sind die Variablen \$RED\_VEL\_AXC[1]...[12], die nur in der Datei "C:\KRC\Roboter\KRC\R1\MaDa\$Machine.dat" geändert werden können.



Ändern Sie die Variablen "\$RED\_VEL\_AXC[1]...[12]" nach Möglichkeit nicht. **Für Schäden bzw. Ausfallzeiten aufgrund veränderter Variablenwerte wird keine Haftung übernommen.**

## 3.9 Beispielkonfiguration

### 3.9.1 Vereinbarungen

- Die Programmnummer soll als Binärzahl übermittelt werden.
- Die Programmnummer ist 7 Bit breit und wird ab dem Eingang 1 empfangen.
- Das Paritätsbit wird am Eingang 8 empfangen, wobei auf ungerade Parität geprüft wird.
- Die Anforderung einer neuen Programmnummer wird durch die ansteigende Flanke des Signals am Ausgang 1 signalisiert.
- Der Leitreechner meldet eine anliegende Programmnummer mit einer ansteigenden Flanke am Eingang 9.
- Über den Ausgang 2 wird dem Leitreechner mitgeteilt, daß ein Programm läuft.
- Die aktive E/A-Schnittstelle wird dem Leitreechner über den Ausgang 3 signalisiert.
- Ein externer Start vom Leitreechner erfolgt über den Eingang 10.
- Die Meldung eines Sammelfehlers an den Leitreechner erfolgt über den Ausgang 4.
- Vom Leitreechner werden Fehler über den Eingang 11 quittiert.



#### Erforderliche Eingaben in der Datei "C:\KRC\Roboter\KRC\R1\System\\$CONFIG.DAT" (Beispielkonfiguration)

PGNO=0	Vorbelegung der Programmnummer
PGNO_TYPE=1	Datenformat der Programmnummer: Binärzahl
PGNO_FBIT=1	Erstes Bit der Programmnummer: Eingang 1
PGNO_LENGTH=7	Breite der Programmnummer: 7 Bit
PGNO_PARITY=-8	Ungerade Parität, Paritätsbit auf Eingang 8
PGNO_REQ=1	Anforderung einer neuen Programmnummer über Ausgang 1
PGNO_VALID=9	Meldung, das die Programmnummer übertragen wurde kommt auf Eingang 9
APPL_RUN=2	Meldung, daß ein Programm abgearbeitet wird über Setzen des Ausganges 2
PGNO_ERROR=0	Vorbelegung des Fehlermerkers

**Eingaben in der Datei \$MACHINE.DAT (Beispielkonfiguration)**

\$EXT\_START\$IN[10] ; externer Start  
 \$I\_O\_ACTCONF \$OUT[3] ; E/A-Schnittstelle aktiv  
 \$STOPMESS \$OUT[4] ; Stop-Fehler  
 \$CONF\_MESS \$IN[11] ; Sammelquittung

**Schnittstellenbelegung (Beispielkonfiguration)**

Steuerung	Signalname	Leitrechner
\$IN[1]	PGNO Bit 1	A 20.0
\$IN[2]	PGNO Bit 2	A 20.1
\$IN[3]	PGNO Bit 3	A 20.2
\$IN[4]	PGNO Bit 4	A 20.3
\$IN[5]	PGNO Bit 5	A 20.4
\$IN[6]	PGNO Bit 6	A 20.5
\$IN[7]	PGNO Bit 7	A 20.6
\$IN[8]	PGNO_PARITY	A 20.7
\$IN[9]	PGNO_VALID	A 21.0
\$IN[10]	\$EXT_START	A 21.1
\$IN[11]	\$CONF_MESS	A 21.2
\$IN[12]	\$DRIVES_OFF	A 21.3
\$IN[13]	\$DRIVES_ON	A 21.4
\$IN[14]	\$MOVE_ENABLE	A 21.5
\$OUT[1]	PGNO_REQ	E 20.0
\$OUT[2]	APPL_RUN	E 20.1
\$OUT[3]	\$I_O_ACTCONF	E 20.2
\$OUT[4]	\$STOPMESS	E 20.3
\$OUT[5]	\$PERI_RDY	E 20.4
\$OUT[6]	\$PRO_ACT	E 20.5

## 3.10 Meldungen

In diesem Abschnitt werden die im Zusammenhang mit der Schnittstelle "Automatik extern" auftretenden Fehlermeldungen beschrieben.

Meldungsnummer	Meldungstext	Ursache
<b>P00:1</b>	PGNO_TYPE falscher Wert zulässige Werte (1,2,3)	Der Datentyp der Programmnummer wurde falsch angegeben.
<b>P00:2</b>	PGNO_LENGTH falscher Wert Wertebereich $1 \leq \text{PGNO\_LENGTH} \leq 16$	Die Bitbreite der Programmnummer wurde falsch projiziert.
<b>P00:3</b>	PGNO_LENGTH falscher Wert zulässige Werte (4,8,12,16)	Wurde zum Lesen der Programmnummer das BCD-Format gewählt, so muß auch eine entsprechende Bitbreite eingestellt werden
<b>P00:4</b>	PGNO_FBIT falscher Wert liegt nicht im \$IN-Bereich	Für das erste Bit der Programmnummer wurde der Wert "0" oder ein nicht vorhandener Eingang angegeben
<b>P00:7</b>	PGNO_REQ falscher Wert liegt nicht im \$OUT-Bereich	Für die Ausgang, über den die Programmnummer angefordert werden soll, wurde der Wert "0" oder ein nicht vorhandener Ausgang angegeben
<b>P00:10</b>	Übertragungsfehler falsche Parität	Bei der Überprüfung der Parität trat eine Unstimmigkeit auf. Es muß ein Übertragungsfehler aufgetreten sein
<b>P00:11</b>	Übertragungsfehler falsche Programmnummer	Vom Leitrechner wurde eine Programmnummer übermittelt, für die in der Kontrollstruktur von CELL.SRC ( noch ) kein Zweig zur Abarbeitung angelegt wurde
<b>P00:12</b>	Übertragungsfehler falsche BCD-Kodierung	Der Versuch, die Programmnummer im BCD-Format einzulesen, führte zu einem ungültigen Ergebnis
<b>P00:13</b>	Falsche Betriebsart	Die E/A-Schnittstelle ist nicht aktiviert worden, d.h. die Systemvariable \$I_O_ACTCONF hat im Moment den Wert FALSE. Dies kann die folgenden Ursachen haben:  Der Schlüsselschalter steht nicht in der Stellung "Ext."  Das Signal \$I_O_ACT besitzt im Moment den Wert FALSE
<b>P00:14</b>	Home-Position in Betriebsart T1 anfahren	Der Roboter hat die HOME-Position nicht erreicht
<b>P00:15</b>	Programmnummer fehlerhaft	Bei "1 aus n" ist mehr als ein Eingang gesetzt.