

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 Leitreechner mit der Robotersteuerung kommunizieren und verschiedene Roboterprozesse auslösen. Ebenso kann die Robotersteuerung Informationen über Betriebszustände und Störmeldungen an den Leitreechner 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 "Konfigurieren" die Option "Ein/Ausgänge" aus.



Es öffnet sich ein Zustandsfenster.

Konfigurieren AUTO EXTERN Eingänge			
PGNO	Signal	Eingang	Signal
1	PGNO_TYPE	E	1026 EXT_START
8	PGNO_LENGTH	E	1025 MOVE_ENABLE
E 33	PGNO_FBIT	E	1026 CONF_MESS
E 41	PGNO_PARITY	E	140 DRIVES_ON
E 42	PGNO_VALID	E	1025 DRIVES_OFF

In den Eingabefeldern werden die Nummern der Eingänge der Robotersteuerung eingetragen, die mit den nachstehenden Signalen der Schnittstelle verknüpft werden sollen.

Im Beispiel wurde der Eingang 140 mit dem Signal "DRIVES_ON" verbunden.

Ändern

Zum Speichern Ihrer Eingaben drücken Sie den Softkey "Ändern".

Ausgänge

Um zum Zustandsfenster zur Konfiguration der Ausgänge zu kommen, betätigen Sie bitte den Softkey "Ausgänge".

Konfigurieren AUTO EXTERN Ausgänge					
Ausgang	Signal	Ausgang	Signal		
A	<input type="text" value="1070"/>	STOPMESS	A	<input type="text" value="994"/>	T2
A	<input type="text" value="33"/>	PGNO_REQ	A	<input type="text" value="995"/>	AUT—
A	<input type="text" value="34"/>	APPL_RUN	A	<input type="text" value="140"/>	EXTERN
A	<input type="text" value="1012"/>	PERI_RDY	A	<input type="text" value="147"/>	ON_PATH
A	<input type="text" value="1013"/>	ALARM_STOP	A	<input type="text" value="1021"/>	PRO_ACT
A	<input type="text" value="1011"/>	USER_SAF	A	<input type="text" value="1000"/>	IN_HOME
A	<input type="text" value="993"/>	T1	A	<input type="text" value="35"/>	ERR_TO_PLC

In den Eingabefeldern werden die Nummern der Ausgänge der Robotersteuerung eingetragen, die mit den nachstehenden Signalen der Schnittstelle verknüpft werden sollen.

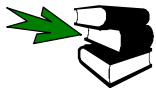
Im Beispiel wurde der Ausgang 147 mit dem Signal "ON_PATH" verbunden.

Ändern

Zum Speichern Ihrer Eingaben müssen Sie auch hier den Softkey "Ändern" drücken.

Eingänge

Um wieder zum Zustandsfenster zur Konfiguration der Eingänge zu kommen, betätigen Sie bitte den Softkey "Eingänge".



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 `$I_0_ACTCONF` auf den Wert `TRUE` aktiv geschaltet und sind alle weiteren Startbedingungen erfüllt, so kann durch ein Signal auf der Leitung `$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_0` in der Datei `c:\programme\krc\mada\steu\SCUSTOM.DAT` folgender Wert zugewiesen werden:

```
CHAR $PRO_I_0[ ] = " /R1/SPS() "
```



Nach dem Hochfahren der Steuerung wird immer versucht das Programm auszuführen, daß in `$PRO_I_0` 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.

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 Leitreechner abzurufen ...

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

Kontrollstruktur in Abhängigkeit von der empfangenen Programmnummer ...

```
SWITCH PGNO
```

Wenn Programmnummer PGNO = 1 ...

```
CASE 1
```

... dem Leitreechner den Empfang der Programmnummer mitteilen ...

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

... und benutzerdefiniertes Programm EXAMPLE1 aufrufen

```
; EXAMPLE1 ( )
```

Wenn Programmnummer PGNO = 2 ...

```
CASE 2
```

... dem Leitreechner den Empfang der Programmnummer mitteilen ...

```
P00 (#EXT_PGNO, #PGNO_ACKN, DM[ ], 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, DMX[], 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, DMX[], 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:

#PGNO_GET	Anforderung einer Programmnummer
#PGNO_ACKN	Mitteilen des Erhalt einer Programmnummer
#PGNO_FAULT	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 dem 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 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** bleibt gesetzt. Dadurch wird dem Leitrechner mitgeteilt, daß die Übertragung fehlerhaft war.



Eine fehlerhafte Übermittlung kann vom Leitrechner 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 Leitrechner 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 Leitreechner 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 Leitreechner 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 Leitreechner übertragen werden soll

PKG[] -

Technologiepaket

ERR -

Fehlernummer im ausgewählten Technologiepaket

```
P[1]={OUT2, PKG[ ] "P00", ERR10}
```

```
...
```

```
P[127]={OUT27, PKG[ ] "S00", ERR11}
```

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

```
P[128]={OUT12, PKG[ ] "CTL", ERR1}
```

```
...
```

```
P[255]={OUT25, PKG[ ] "CTL", ERR10}
```

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


```
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 Integerwert übergeben	0 0 1 0 0 1 1 1 => PGNO = 39
2	BCD-Wert	Die Programmnummer wird von der übergeordneten Steuerung als Binär Codierter Dezimalwert übergeben	0 0 1 0 0 1 1 1 <div style="display: flex; justify-content: space-around; width: 100%;"> 2 7 </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 Leitrechner verwendet.

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



Sind die Antriebe vom Leitrechner 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.

3.6.1.8 CONF_MESS

Durch Setzen dieses Eingangs kann der Leitrechner 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.

3.6.1.9 DRIVES_ON

Durch einem High-Impuls von mind. 20 ms Dauer an diesem Eingang kann der Leitrechner 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.

3.6.1.10 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 Ausganges teilt die Robotersteuerung dem Leitreechner mit, daß gerade ein Programm abgearbeitet wird.

3.6.2.4 PERI_RDY

Mit Setzen dieses Ausganges 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.

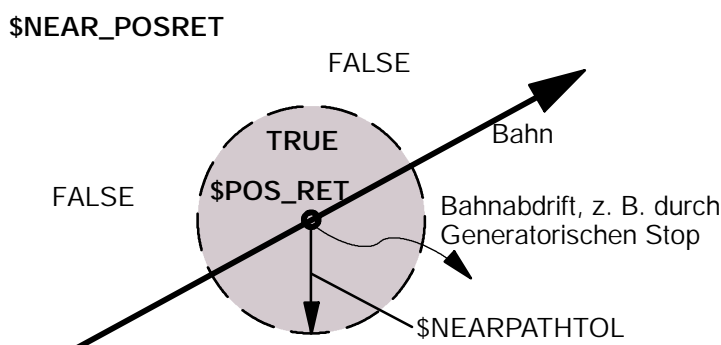
3.6.2.9 NEAR_POSRET

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.



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_STATE** abgeleitet:

\$PRO_STATE=#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:

- G Wurden Interrupts aktiviert, aber zum Zeitpunkt des Fehlerstops nicht bearbeitet, so gilt der Prozeß als inaktiv (**PRO_ACT=FALSE**)
- G 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**)

- G 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 Leitrechner 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 Leitrechner mit, daß ein Steuerungs- oder Technologiefehler aufgetreten ist.



Diese Funktion ist nur aktiv, wenn **PLC_ENABLE** den Wert **FALSE** 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.

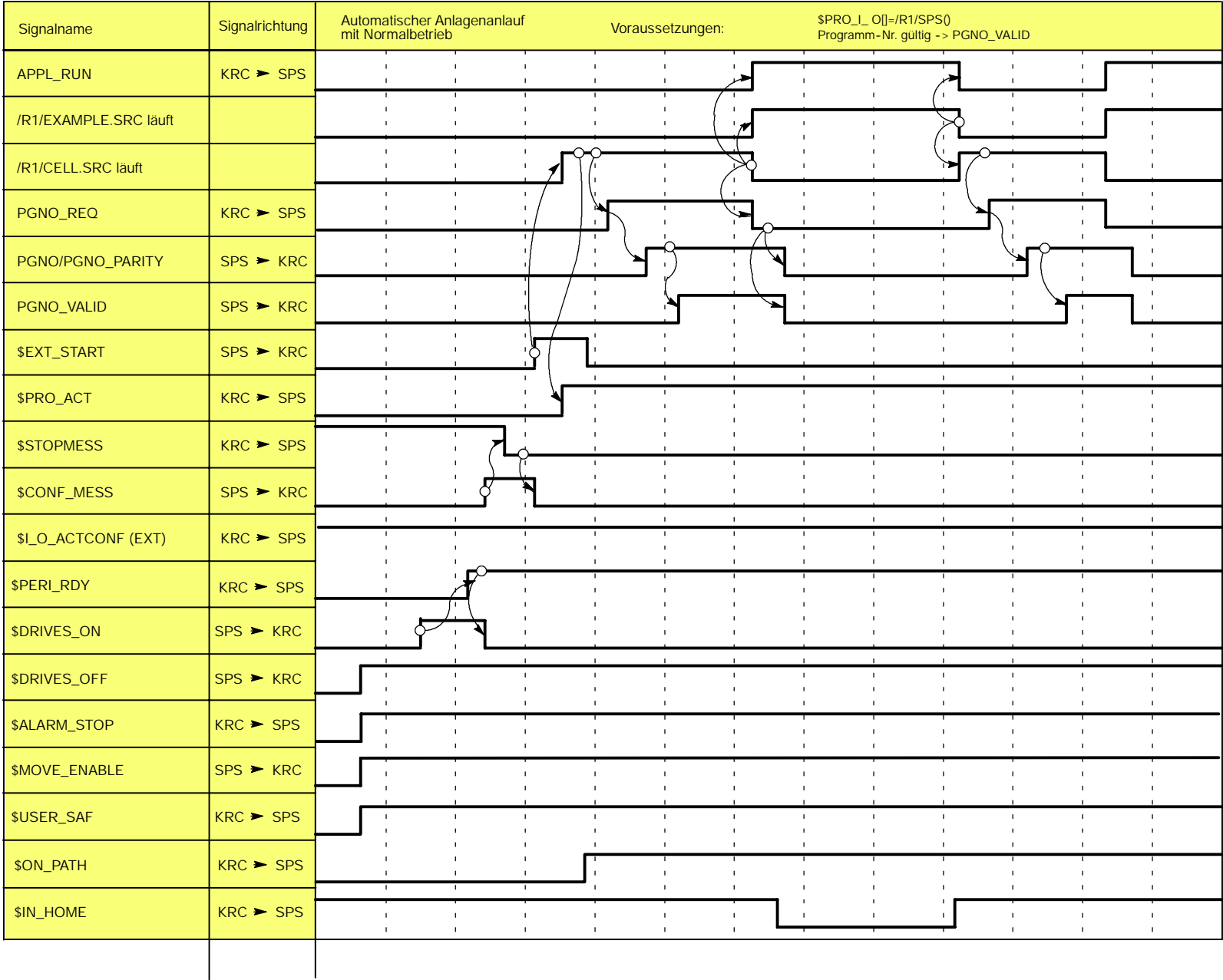
Das technologiespezifische Organisationsprogramm **CELL.SRC** ordnet mittels dieser Variablen der Programmnummer das entsprechende Anwenderprogramm zu.

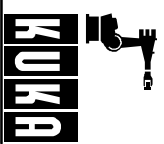
3.6.3.2 PGNO_ERROR

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

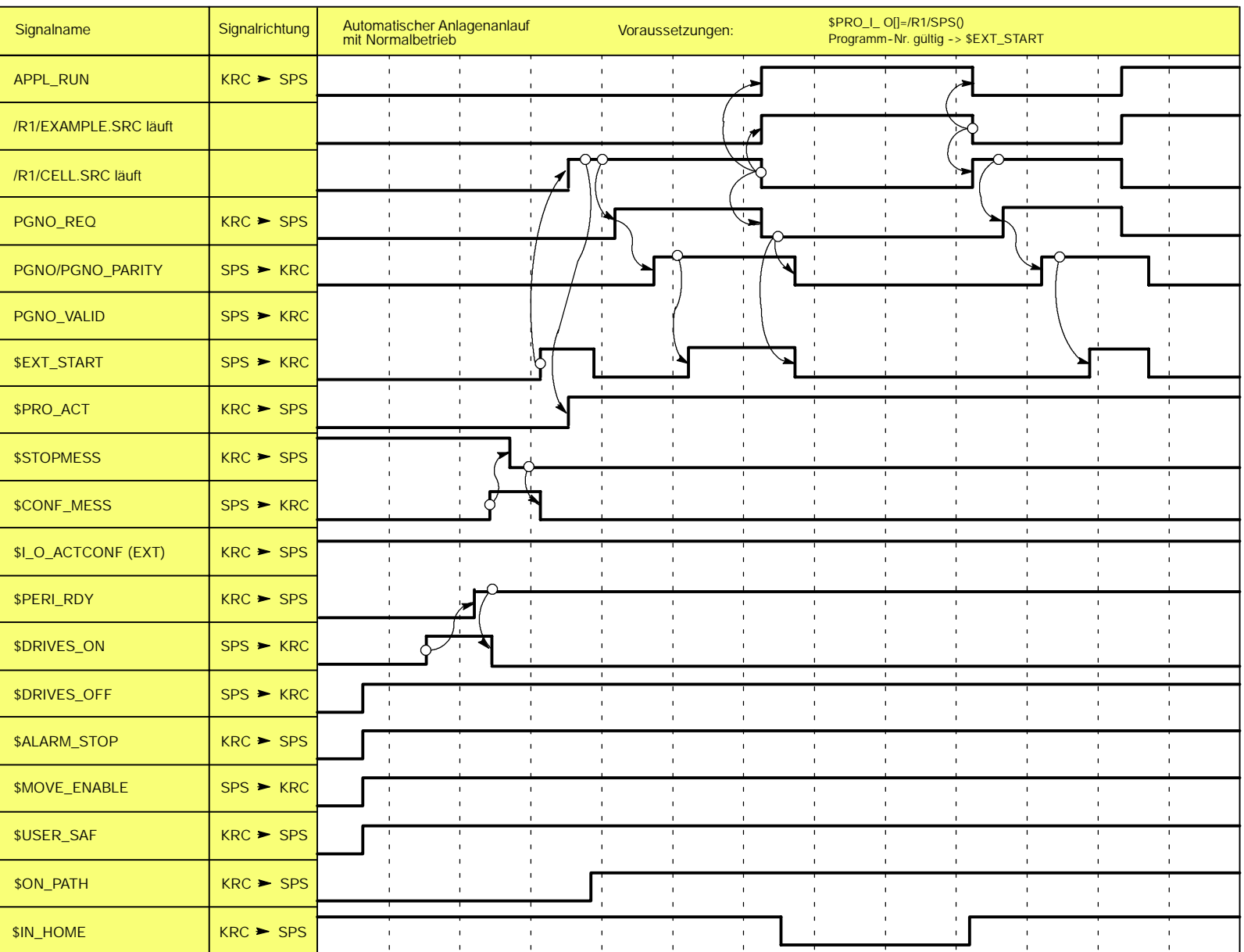
3.7 Signaldiagramme

3.7.1 Auto. Anlagenanlauf und Normalbetr. mit Prog.-Nr.-Quitt. durch PGNO_VALID

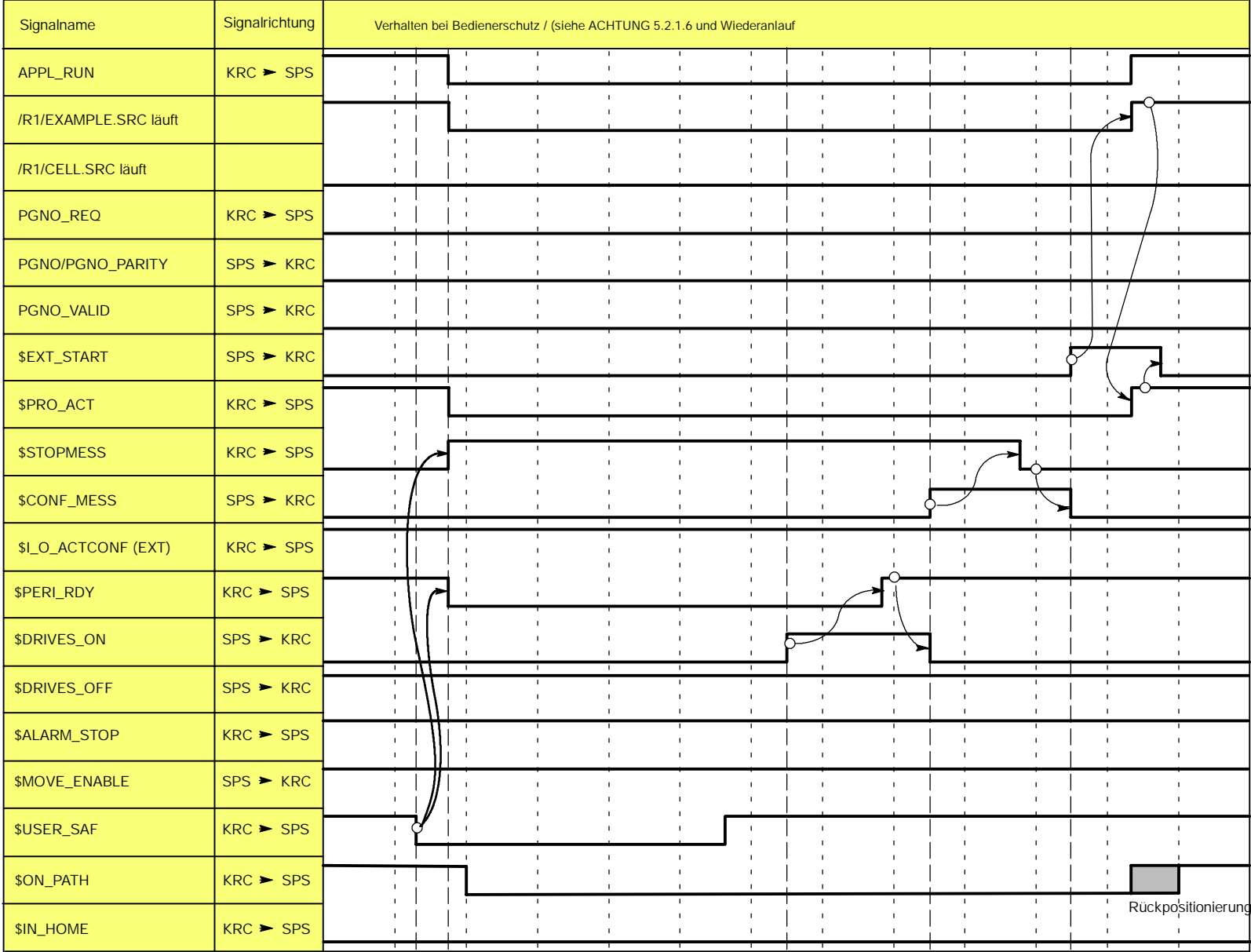




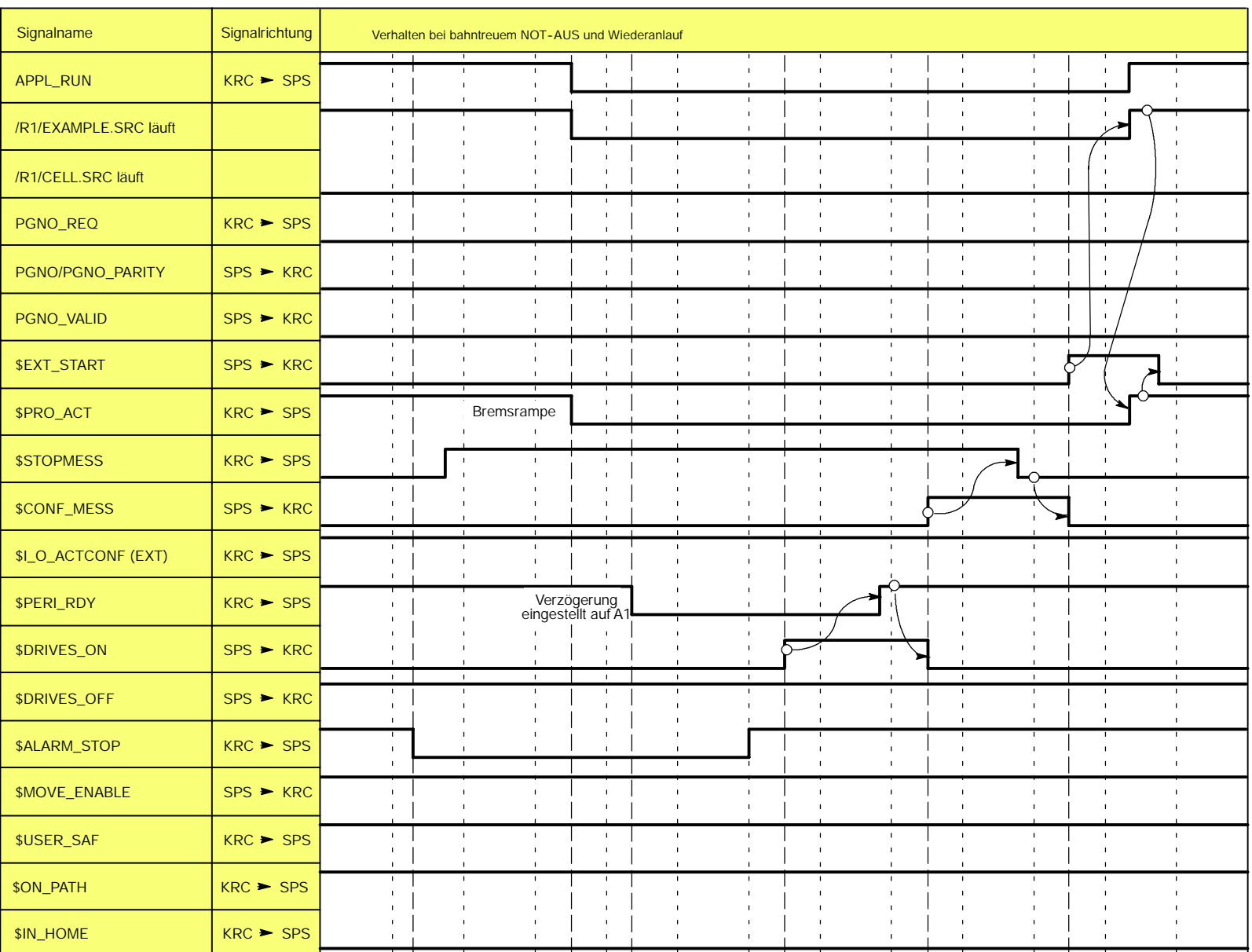
3.7.2 Auto. Anlagenanlauf und Normalbetr. mit Prog.-Nr.-Quitt. durch \$EXT_START



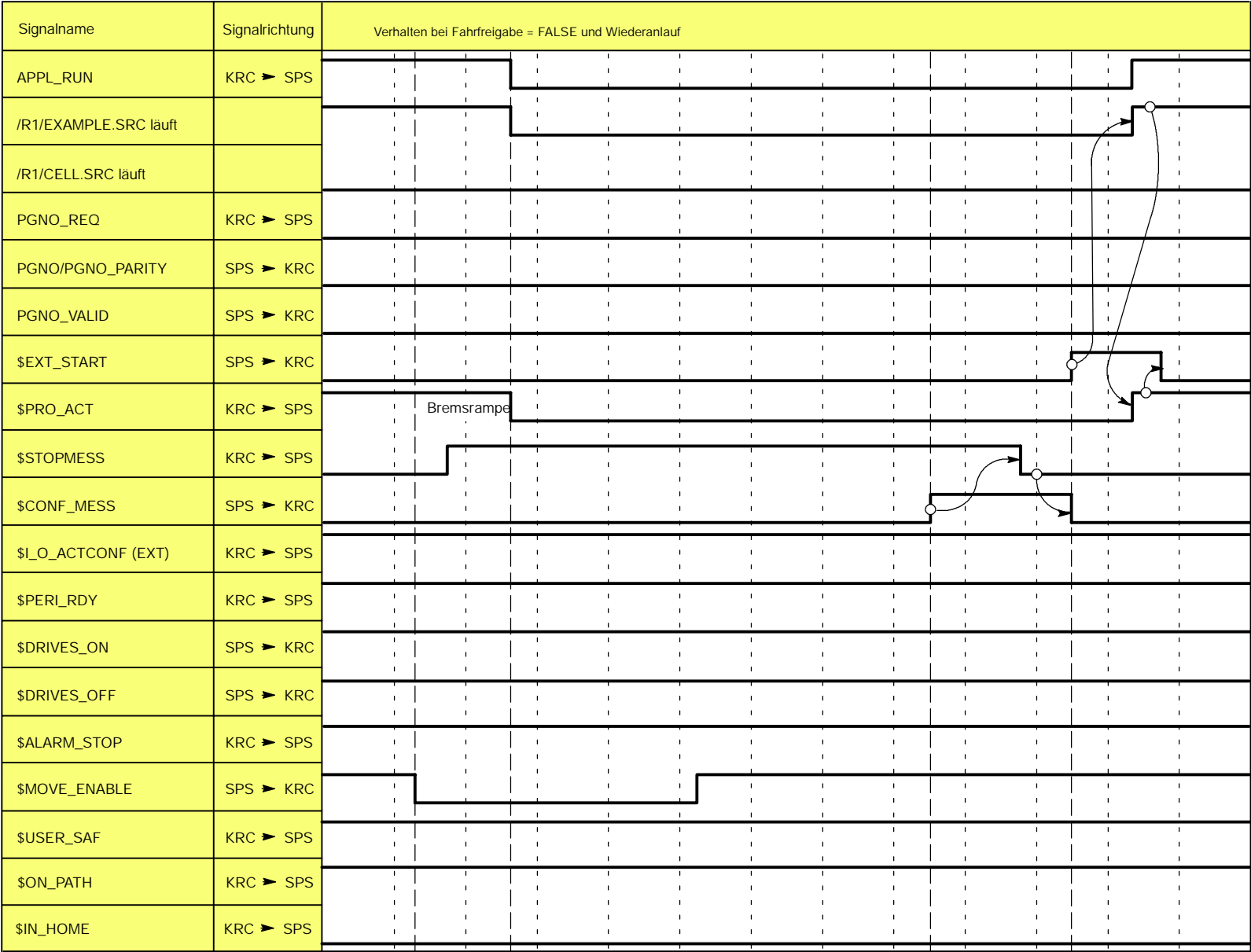
3.7.3 Wiederanlauf nach generat. Stop (Bedienerschutz und Wiederanlauf)



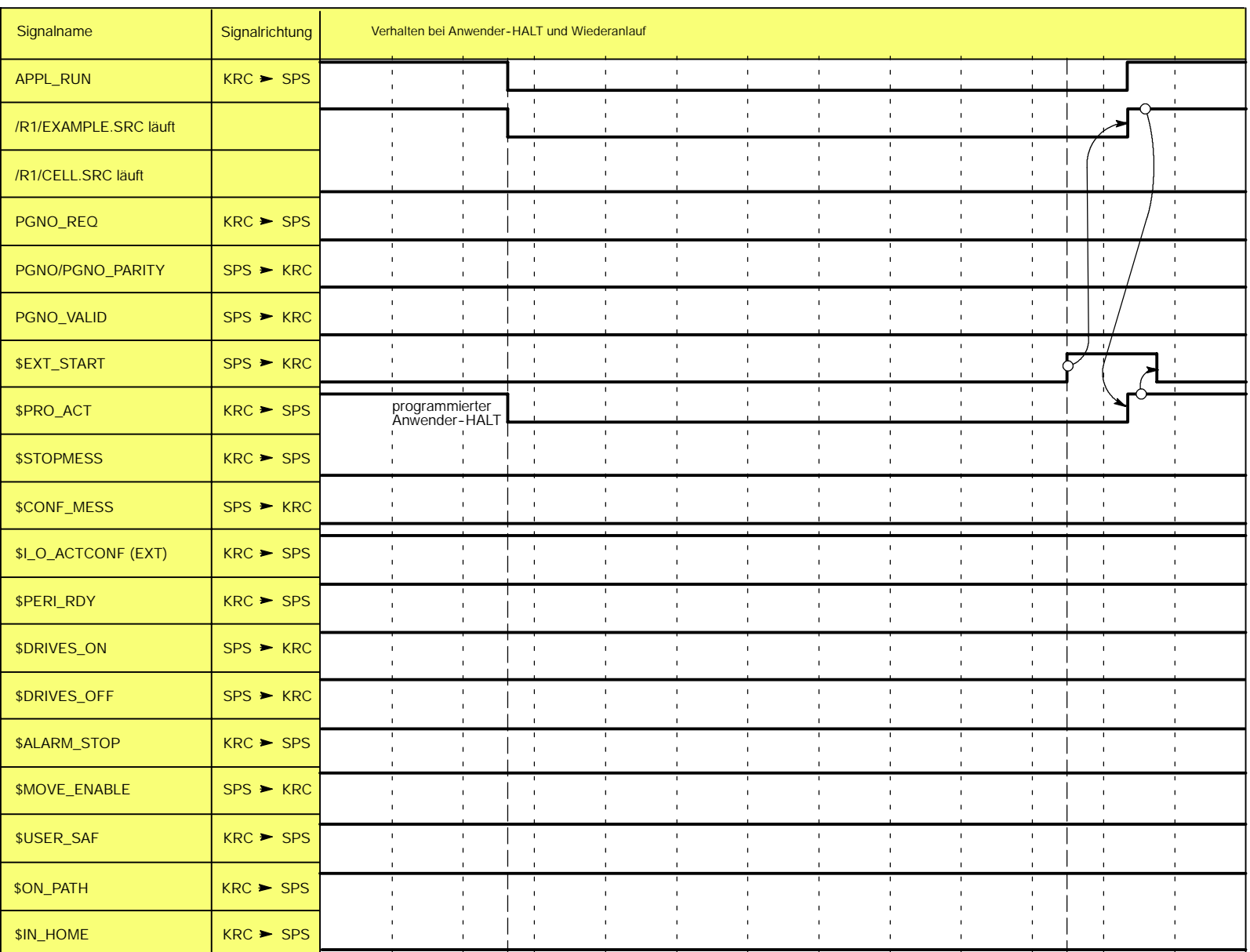
3.7.4 Wiederanlauf nach bahntreuem NOT-AUS



3.7.5 Wiederanlauf nach Fahrfreigabe



3.7.6 Wiederanlauf nach Anwender-HALT



3.8 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.

3.9 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 Leitrechner meldet eine anliegende Programmnummer mit einer ansteigenden Flanke am Eingang 9.
- Über den Ausgang 2 wird dem Leitrechner mitgeteilt, daß ein Programm läuft.
- Die aktive E/A-Schnittstelle wird dem Leitrechner über den Ausgang 3 signalisiert.
- Ein externer Start vom Leitrechner erfolgt über den Eingang 10.
- Die Meldung eines Sammelfehlers an den Leitrechner erfolgt über den Ausgang 4.
- Vom Leitrechner werden Fehler über den Eingang 11 quittiert.



Erforderliche Eingaben in der Datei ...R1\$CONFIG.DAT (Beispielkonfiguration)

PGNO=0	Vorbelegung der Programmnummer
PGNO_TYPE=1	Datenformat der Programmnummer: Binärzahl
PGNO_FBI T=1	Erstes Bit der Programmnummer: Eingang 1
PGNO_LENGTH=7	Breite der Programmnummer: 7 Bit
PGNO_PARI TY=- 8	Ungerade Parität, Paritätsbit auf Eingang 8
PGNO_REQ=1	Anforderung einer neuen Programmnummer über Ausgang 1
PGNO_VALI D=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
P00: 1	PGNO_TYPE falscher Wert zulässige Werte (1,2,3)	Der Datentyp der Programmnummer wurde falsch angegeben.
P00: 2	PGNO_LENGTH falscher Wert Wert zu groß (max. 16 Bit)	Die Bitbreite der Programmnummer wurde zu groß gewählt.
P00: 3	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
P00: 4	PGNO_FBI T falscher Wert liegt nicht im \$IN-Bereich	Für das erste Bit der Programmnummer wurde der Wert "0" oder ein nicht vorhandener Eingang angegeben
P00: 5	PGNO falscher Wert liegt nicht im \$IN-Bereich	Für das erste Bit der Programmnummer wurde ein so hoher Wert angegeben, daß zusammen mit der angegebenen Kanalbreite der E/A-Bereich der Steuerung überschritten wurde
P00: 6	PGNO_PARI TY falscher Wert liegt nicht im \$IN-Bereich	Für den Eingang, der dem Paritätsbit zugeordnet werden soll, wurde der Wert "0" oder ein nicht vorhandener Eingang angegeben
P00: 7	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
P00: 8	PGNO_VALID falscher Wert liegt nicht im \$IN-Bereich	Für den Eingang, an den die Aufforderung zum Einlesen der Programmnummer gesendet wird, wurde der Wert "0" oder ein nicht vorhandener Eingang angegeben
P00: 9	APPL_RUN falscher Wert liegt nicht im \$OUT-Bereich	Für den Ausgang, an dem das Signal "Programm läuft" anliegen soll, wurde der Wert "0" oder ein nicht vorhandener Ausgang angegeben
P00: 10	Übertragungsfehler falsche Parität	Bei der Überprüfung der Parität trat eine Unstimmigkeit auf. Es muß ein Übertragungsfehler aufgetreten sein
P00: 11	Übertragungsfehler falsche Programmnummer	Vom Leitreechner wurde eine Programmnummer übermittelt, für die in der Kontrollstruktur von CELL. SRC (noch) kein Zweig zur Abarbeitung angelegt wurde
P00: 12	Übertragungsfehler falsche BCD-Kodierung	Der Versuch, die Programmnummer im BCD-Format einzulesen, führte zu einem ungültigen Ergebnis

P00: 13	KCP passiv schalten Schlüsselschalter EXT	<p>Die E/A-Schnittstelle ist nicht aktiviert worden, d.h. die Systemvariable \$I_0_ACTCONF hat im Moment den Wert FALSE. Dies kann die folgenden Ursachen haben:</p> <p>Das KCP wurde nicht passiv geschaltet</p> <p>Der Schlüsselschalter steht nicht in der Stellung "Ext."</p> <p>Das Signal \$I_0_ACT besitzt im Moment den Wert FALSE</p> <p>Ein anderer externer Kommunikationspartner hat den Zugriff auf die Steuerung \$DEVI CE=#ACTIVE für sich aktiv geschrieben z.B. SINEC H1</p>
P00: 14	FEHLER Roboter nicht in HOME-Position!!	Der Roboter hat die HOME-Position nicht erreicht
P00: 15	FEHLER DATA_OK	Kommando erfolgreich, es stehen Daten zum Lesen bereit
P00: 16	FEHLER CMD_ABORT	Ausführung CWRITE wurde abgebrochen
P00: 17	FEHLER CMD_REJ	Schreiben auf diesem Kanal nicht möglich oder Kommando abgelehnt
P00: 18	FEHLER CMD_PART	Kommando wurde nur teilweise ausgeführt
P00: 19	FEHLER CMD_SYN	Syntaxfehler im Kommando
P00: 20	FEHLER FMT_ERR	Fehlerhafte Formatangabe der Variablen