

Aufgabe 6

Steuerung der Fertigungsanlage

1 Aufgabenstellung

Im Praktikumsraum MI 03.05.012 steht ein Fischertechnik-Modell einer Fertigungsanlage. Auf diesem Modell sollen die über das Eingabeband hinzugefügten Werkstücke identifiziert, und anhand eines bestimmten Arbeitsplans bearbeitet werden. Fertig bearbeitete Werkstücke verlassen das Modell über das Ausgabeband.

Abbildung 1 zeigt die räumliche Anordnung der einzelnen Maschinen und ihrer Anbindung an das Rechnernetz.

- Die Eingabe besitzt entlang ihres Bandes einen Barcodeleser, der die Strichcodes des vorbeilaufenden Werkstücks erkennen kann. Dieser Code wird in der Form von 6 Ziffern dem Steuerprogramm zur Verfügung gestellt.
- Die beiden vierachsigen Roboter können je ein Werkstück aufnehmen. Sie erreichen die an sie angrenzenden Bearbeitungsmaschinen oder Lager. Ihre Arbeitsbereiche überschneiden sich aber, sie können also nicht als völlig unabhängige Transportwerkzeuge betrachtet werden. Es muß deshalb daraufgeachtet werden, daß die Roboter nicht untereinander oder mit den Bearbeitungsmaschinen kollidieren.
- Die drei Werkzeugmaschinen und die Heizzelle besitzen ein bis zwei Ablageplätze, über die der Roboter je nach Zustand der Maschine Werkstücke übergeben oder abholen kann.
- Die Lager sind frei verwendbar.
- Alle aktiven Geräte der Fertigungsanlage werden von Microcontrollern der Serie MC68HC11 bedient. Diese Controller wiederum halten die Verbindung mit dem übergeordneten Leitrechner über den CAN-Bus aufrecht.

Das auf den Controllern laufende Programm ist fest installiert, und muß nicht geladen werden ("Akku-gepuffertes statisches RAM"). Die Befehlsstruktur der einzelnen Maschinen wird in Kapitel 4 "CAN-Objekte" definiert.

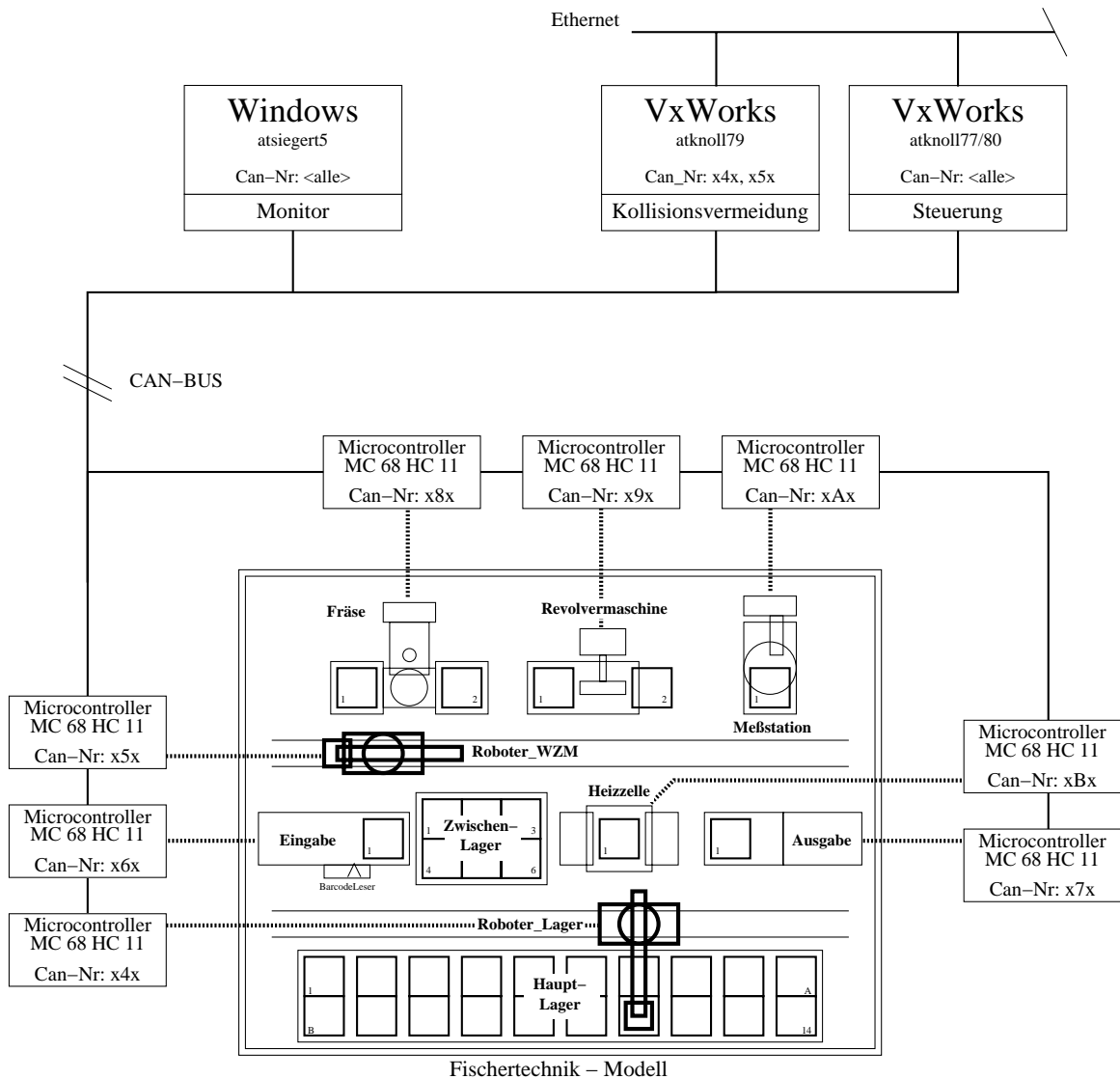


Abbildung 1: Schema der Fertigungsanlage

Ihre Aufgabe ist nun :

1. Definieren Sie Arbeitspläne und Abläufe.
2. Zerteilen Sie das Problem der Behandlung eines Werkstücks in der Fertigungsanlage in kleinere Einzelprobleme (Bearbeitung in Maschinen, Transport durch die Roboter, Bearbeitungsreihenfolge, ...)
3. Teilen Sie die Einzelprobleme eventuell unter den Praktikumsgruppen auf.
4. Erarbeiten Sie gemeinsame Schnittstellen für Synchronisation und Datenaustausch.
5. Implementieren Sie Ihre Lösungen unter Tornado.
6. Fügen Sie alle Teile so zusammen, daß sich ein lauffähiges Programm ergibt, welches vorgeführt werden kann.
7. Geben Sie das kommentierte Programmlisting im DIN A4-Format geheftet ab.

2 WARNUNG

Die Maschinen verfügen zwar größtenteils über hardwaremäßige Endabschalter und bei der Benutzung der Controllerprogramme werden diese Grenzen auch eingehalten, trotzdem können unerwartete Situationen eintreten, die nicht abgefangen werden können und die eine Beschädigung des Modells nach sich ziehen. (z.B. der Roboter versucht in die geschlossene Heizzelle einzudringen und hebt sich dabei selbst aus der Verankerung!)

- *Lassen Sie daher die Fertigungsanlage im Betrieb nie unbeaufsichtigt.*
- *Betätigen Sie in gefährlichen Situationen sofort den NOTAUS.*
- *Versuchen Sie Schäden nicht selbst zu beheben, sondern überlassen Sie dies der Praktikumsaufsicht.*

3 Hinweise zur Implementierung

1. Die CAN-Bus Schnittstelle für die Fertigungsanlage entspricht der des Aufzugs (siehe Kap. 2 der Aufgabe 5).
2. Laden Sie die Dateien aus `atknoll141/Praktikum/PCI200/` in Ihr Verzeichnis und studieren Sie das bereitgestellte Programm `Tx_Rx_Frame_Test.c`. Testen Sie den Prozess durch einfache Steueraufgaben und beobachten Sie die CAN-Bus-Aktionen mit dem Monitor.
3. Der CAN-Bus der Fertigungsanlage ist mit den Rechnern *atsiebert5* und *atknoll77,79,80* verbunden. Zum CAN-Bus der Aufzüge — und damit zu den Rechnern *atknoll74-76* — besteht im allgemeinen keine Verbindung.
4. Die Kollisionsvermeidung soll nicht selbst programmiert werden, sondern das Programm `Kollision` soll verwendet werden. Es wird automatisch auf dem Rechner *atknoll79* gestartet und bedient beide Roboter.

Die Schnittstelle nach oben (zu dem Anwender hin) ist wieder über CAN-Bus-Nachrichten definiert, so daß keine gesonderte Verbindung mit dem Rechner aufgebaut werden muß. Mehr Einzelheiten enthält die Beschreibung der CAN-Bus-Schnittstelle für die gesamte Fertigungsanlage (siehe Kap. 4 CAN-Objekte).

5. Auf dem PC *atsiebert5* (Windows) existiert ein speziell auf die Fertigungsanlage zugeschnittenes Monitorprogramm (`CANspector`). Damit kann sehr einfach beobachtet werden, was sich auf dem CAN-Bus tut. Und über Pull-Down-Menüs sind Sie auch in der Lage, Befehle an die Controller zu schicken.

Sie sollten jedoch beachten, daß der Controller nicht unterscheiden kann, ob eine Nachricht vom / zum Monitor oder Ihrem Programm geht. Eine Quittung die dem Monitorprogramm zugeordnet ist, könnte ein anderes Programm mißverstehen und folglich "falsch" darauf reagieren.

6. Lassen Sie sich durch die Vielzahl der CAN-Objekte nicht verwirren. Verwenden Sie möglichst nur die synchronen Programmbefehle und Quittungen aus Gruppe 4. Reagieren Sie nur auf erwartete Quittungen, verwerfen sie unerwartete, diese könnten die Folge von Aktionen des Monitorprogramms sein (siehe oben).

4 Die CAN-Objekte

Alle Zahlenangaben werden hier hexadezimal oder binär geschrieben.

4.1 Die Aufteilung der Identifier

Der Identifier:

A	9	8	7	6	5	4	3	2	1	0	Identifier-Bits
											Lesen/Schreiben
											je nach Bedarf
											Erweiterung, Zerteilung
											Gerätenummer
											Nachrichtengruppe

Nachrichtengruppen:

Gruppe 0	Nachrichten die alle Geräte sofort betreffen, wie z.B. Nothalt
Gruppe 1	Sofort wirksame Befehle, wie Pause, Weiter, Abbruch
Gruppe 2	direkte Hardwarebeeinflussung
Gruppe 3	Statusmeldungen
Gruppe 4	Programm Befehle, Vollzugsmeldungen der Programme
Gruppe 5 .. 7	nicht verwendet. Kann für höherschichtige Kommunikation verwendet werden, z.B. für Verhandlungen unter Agenten

Gerätenummern:

Gerät 0 .. 3	nicht verwendet
Gerät 4	Roboter Lager
Gerät 5	Roboter WZM (Werkzeugmaschinen)
Gerät 6	Eingabeband
Gerät 7	Ausgabeband
Gerät 8	Fräse
Gerät 9	Mehrspindelmaschine
Gerät A	Meßmaschine
Gerät B	Heizzelle
Gerät C	Zwischenlager (ohne Anschlüsse)
Gerät D	Hauptlager (ohne Anschlüsse)
Gerät E, F	nicht verwendet

Erweiterung, Zerteilung:

Bit	Beschreibung
0	keine
1	Nachrichten betreffen nur einen Teil des Gerätes oder ein Untergerät, das erste Byte der Daten bezeichnet die Teilkomponente

Lesen / Schreiben:

Bit	Beschreibung
0	Nachricht, die an das Gerät geschickt wird
1	Nachricht, die von dem Gerät kommt

4.2 Die Nachrichtenobjekte

4.2.1 Alle Geräte

Identifizier	Daten	Beschreibung
008		Nothalt

4.2.2 Roboter Lager

Die Achsen werden wie folgt nummeriert:

Nummer	Bezeichnung	Beschreibung
0	Unterarm	vertikale Bewegung des Unterarms gegenüber dem Oberarm
1	Oberarm	horizontale Bewegung des Oberarms gegenüber dem Körper
2	Körper	Drehung des Körpers auf dem Wagen
3	Wagen	Bewegung entlang der Schiene

Roboterobjekte:

Identifizier	Daten	Beschreibung
140	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
240	xr	Schalte Achse: $x = [0, 1, 2, 3]$ $r = [0=aus, 1=vor, 2=zurück]$ (Achtung : keine Endabschaltung!)
	4r	Magnet $r = [0=aus, 1=ein]$
	5r	Lampe $r = [0=aus, 1=ein, 2=blinken]$
	F0	alles aus
	FF	Softwarereset (Programm beenden)
R 341		Status anfordern
341	yy, zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7	Taster: Nullstellung Wagen
	6	Taster: Nullstellung Körper
	5	Taster: Nullstellung Oberarm
	4	Taster: Nullstellung Unterarm
	3	Taster: Ein Werkstück befindet sich am Effektor
	2 .. 0	nicht verwendet
zz	7	Wagen bewegt sich
	6	Körper bewegt sich
	5	Oberarm bewegt sich
	4	Unterarm bewegt sich
	3	Roboter macht Pause
	2	Roboter ist nicht vollständig initialisiert
	1	Mindestens eine Achse ist blockiert
	0	Mindestens ein Positionsgeber war zu schnell

Roboterobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
348	0x	Extrastatus der Achse x anfordern
349	0x, yy, zz, ss	Extrastatus der Achse x: Position yy, zz Status ss

Status Byte:

Byte	Bit	Beschreibung
zz	7	Regelprogramm eingeschaltet
	6	Motor ist kraftlos geschaltet
	5	nicht verwendet
	4	Regelung ist aktiv
	3	Achse macht Pause
	2	Die Achse ist nicht initialisiert
	1	Die Achse ist blockiert
	0	Der Positionsgeber war zu schnell

Roboterobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
34A	0x, vv, yy, zz	Parametrisierung der Variable vv von Achse x mit dem Wert yyzz
	00, c0, —, —	alle Parameter laut EEPROM besetzen
	00, c1, 02, 95	alle Parameter ins EEPROM übernehmen
	00, c4, 0g, 0z	Gerätenummer z=[0..F] im EEPROM festlegen (funktioniert nur wenn g die alte Gerätenummer ist und wird erst nach einem Reset wirksam)
34B	0x, vv, yy, zz	Wert der Variable vv der Achse x ist yyzz

Variable	Bezeichnung	Wertebereich
01	Position Minimum (xmin)	-7FFF .. 7FFF
02	Position Maximum (xmax)	-7FFF .. 7FFF
03	Mindestgeschwindigkeit (vmin)	1 .. 1F
04	Höchstgeschwindigkeit (vmax)	vmin .. 1F
05	Totbereich (tot)	00 .. FF
06	Steigung (Rampe)	00 .. FF
07	Beschleunigung (besch)	00 .. FF
08	Verklemmungsgrenze (timeout)	00 .. FF
09	max. Zugabe zu vmin (verkl_plus)	00 .. FF
10	Positionsgeberumpolung	0 , 1
11	ZSyncWert (nur lesen)	-8000 .. 7FFF

Um einer Parametervariablen einen neuen Wert zu geben, muß in ihrer Nummer das oberste Bit gesetzt werden (oder 80hex addieren). Ansonsten wird der Wert ignoriert und nur der augenblickliche Wert in der Nachricht 34B zurückgeschickt.

Roboterobjekte (Fortsetzung):

Identifier	Daten	Beschreibung
448	0x, yy, zz	Achse x Position xxyy= $[\pm 7FFF]$ einnehmen
	1x, —, —	Initialisieren der Achse x (Nullposition suchen)
	14, —, —	Initialisieren aller Achsen (0, 1, 2, 3)
449	0x, 00	Achse x hat Position erreicht
	0x, 10	Positionsgeber x hat sich zu schnell gedreht
	0x, 11	Achse x ist blockiert (Timeout)
	0x, 12	Achse x ist noch nicht initialisiert worden
	0x, 20	Positionsprogramm wurde abgebrochen
	1x, 00	Initialisierung der Achse x beendet

4.2.3 Roboter WZM

Der Roboter WZM funktioniert genauso wie der Roboter Lager, nur trägt er eine andere Gerätenummer (5). Das heißt, daß statt der Ziffer 4 die Ziffer 5 auf die mittlere Stelle des Identifiers zu setzen ist !

Zum Beispiel fordert der Remote-Request R 351 zur Sendung des Status des Roboters WZM auf. Der erwidert die Nachricht mit dem Identifier 351 und dem Status als Daten.

4.2.4 Kollisionsvermeidung

Die Kollisionsvermeidung läuft als VxWorks-Programm auf dem Rechner atknoll79. Sie bietet wie die Geräte selbst Dienstleistungen über den CAN-Bus an.

Nach Erhalten eines Fahrtauftrages für einen der Roboter werden verschiedene Möglichkeiten der Wegbeschreitung geprüft und eine geeignete ausgewählt. Dabei wird auch die Position des anderen Roboters mitberücksichtigt. Wurde ein kollisionsfreier Weg gefunden, so schickt das Programm (wieder über den CAN-Bus) die nötigen Befehle an die Roboter-Controller, welche schließlich das Modell ansteuern.

Der Anwender braucht sich nicht mehr um die Roboter selbst zu kümmern. Er muß nur das „WAS“ und „WO“ auf den CAN-Bus legen. Das „WIE“ erledigt die Kollisionsvermeidung.

Nach dem Start des Programms „*kollision*“ wird auch die Datei „*weltbild.txt*“ geladen. Sie enthält die Abmessungen der Robotergelenke, sowie die Koordinaten und Ausmaße der zu berücksichtigenden Hindernisse. Außerdem sind noch alle Positionen und Nummern der Ablageplätze enthalten.

Wurde ein korrektes Weltbild eingelesen initialisieren sich die Roboter. Es ist darauf zu achten, daß hierbei noch kein Kollisionstest durchgeführt werden kann, da die augenblickliche Stellung der Robotergelenke noch nicht bekannt ist. Sie wird erst durch die Initialisierung festgestellt. Nötigenfalls muß ein Roboter (besonders der **Roboter_WZM**) von Hand an eine frei Stelle geschoben werden.

Erst wenn beide Roboter vollständig initialisiert sind, wird der Auftragsempfänger des CAN-Busses freigeschaltet und steht für den Anwender zur Verfügung.

Kollisionsvermeidungsobjekte:

Identifizier	Daten	Beschreibung
4r0	00, —, —	Roboter parken
	02, gg, pp	Fahre zu Gerät gg über Platz pp
	04, —, —	ein Werkstück aufnehmen
	05, —, —	ein Werkstück ablegen
	06, gg, pp	das Werkstück bei Gerät gg von Platz pp holen
	07, gg, pp	das Werkstück zu Gerät gg auf Platz pp legen
	08, —, —	Prüfen, ob ein Werkstück am Effektor hängt
	10, —, —	Initialisieren
4r1	00	Auftrag erfolgreich ausgeführt
	01	das Werkstück ist nicht / schon vorhanden
	11	der Roboter ist blockiert
	12	der Roboter ist nicht initialisiert
	20	die Bewegung der Achsen wurde abgebrochen
	22	unbekannte Befehlsnummer
	30	Das Gerät oder der Platz ist nicht bekannt
	31	Das Ziel ist nicht in Reichweite dieses Roboters
40	fataler, unerwarteter Fehler	
R 3r3		Status anfordern
3r3	yy	Status

r	Beschreibung
4	Roboter_Lager
5	Roboter_WZM

Byte	Beschreibung
gg	ist die Gerätenummer (wie anfangs beschrieben)
pp	ist die Platznummer, die je nach Gerät einen anderen Umfang besitzen kann: z.B. besitzt das Eingabeband nur einen Platz [01], während es im Hauptlager zwanzig Plätze gibt [01..14].

Status Bytes:

Byte	Bit	Beschreibung
yy	7	Roboter ist geparkt
	6	Roboter wartet auf das Verschwinden des anderen
	5 .. 3	nicht verwendet
	2	Eine Initialisierung ist unbedingt erforderlich
	1	Fehler ist aufgetreten
	0	Programm läuft

4.2.5 Eingabeband

Eingabebandobjekte:

Identifizier	Daten	Beschreibung
160	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
260	00	Band anhalten
	01	Band vorwärts laufen lassen
	02	Band rückwärts laufen lassen
R 263		zuletzt gelesenen Barcode anfordern
263	0y, 0y, 0y, 0y, 0y, 0y	zuletzt gelesener Barcode; y bezeichnet die einzelnen Ziffern
R 361		Status anfordern
361	yy,zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7	Initiator neues Teil auf dem Band belegt
	6	Initiator Abholposition belegt
	5	Bandmotor rückwärts
	4	Bandmotor vorwärts
	3 .. 0	nicht belegt
zz	7 .. 4	nicht belegt
	3	Pause
	2 , 1	Programmphase
	0	Programm läuft

Eingabebandobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
460	01	Programm: Auf ein neues Teil warten
461	00	Teil liegt bereit
	01	Teil liegt bereit, ist aber ohne Barcode
	10	unerwarteter Fehler

4.2.6 Ausgabeband

Ausgabebandobjekte:

Identifizier	Daten	Beschreibung
170	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
270	00	Band anhalten
	01	Band vorwärts laufen lassen
	02	Band rückwärts laufen lassen
R 371		Status anfordern
371	yy,zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7	Initiator neues Teil auf dem Band belegt
	6	Initiator Rutsche oben belegt
	5	Initiator Rutsche unten belegt
	4	Bandmotor rückwärts
	3	Bandmotor vorwärts
	2 .. 0	nicht belegt
zz	7 .. 3	nicht belegt
	2	Pause
	1	Programmphase
	0	Programm läuft

Ausgabebandobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
470	01	Programm: ein Teil ausgegeben
471	00	Teil ausgegeben
	10	Teil fehlte

4.2.7 Fräse (SPS)

Fräseobjekte:

Identifizier	Daten	Beschreibung
180	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzten (nur nach Pause)
280	00bbbbrr (binär)	Lowlevel: bbbb = 0001 Förderband rechts 0010 Förderband mitte 0100 Förderband links 1000 Turm 1001 Werkzeugkopf 1010 Tisch 1100 Werkzeug rr = 00 aus 01 ein / links / vor / rauf 10 / rechts / zurück / runter
R 381		Status anfordern
381	yy, zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7 .. 5	linker Platz Status
	4 .. 2	mittlere Platz Status
	1	Fräskopf unten
	0	Fräskopf dazwischen
zz	7 .. 5	rechter Platz Status
	4	Programm läuft
	3 .. 0	Transport Status

Platz Status (3-bit)	
0	frei
1	Teil ist unbearbeitet
2	Teil wird bearbeitet
3	Teil ist bearbeitet
4	Bearbeitung abgebrochen

Transport Status (4-bit)	
0	kein Transport
1	Teil von rechts nach mitte
2	Teil von links nach mitte
3	Teil von mitte nach rechts
4	Teil von mitte nach links
5	Tisch gedreht
6	Tisch Stellung zur Fräse
7	2 Teile von links nach rechts
8	2 Teile von rechts nach links
A	Teil auf dem Tisch verschoben

Fräseobjekte (Fortsetzung):

Identifer	Daten	Beschreibung
480	00,	Initialisiere Maschine
	01, 0r	Belade Maschine von r = [0=links , 1=rechts]
	02, 0r	Entlade Maschine nach r = [0=links , 1=rechts]
	03, 0r	Be&Entlade Maschine r = [0=links >> rechts, 1=rechts >> links]
	10, yy	Fräsprogramm Nr. yy = [1 .. 6]
481	00, 00	Programm korrekt beendet
	yy, zz	Fehler ist aufgetreten

Fehlerbytes :

Byte	Bit	Beschreibung
yy	7	nicht verwendet
	6	Teil ist noch nicht bearbeitet
	5	Kopf ist nicht oben
	4	Programmnummer ist falsch
	3	Stau rechts
	2	Stau links
	1	Timeout Band rechts
	0	Timeout Band mitte
zz	7	Timeout Band links
	6	Drehtisch schief
	5	Drehtisch Stellung zur Fräse
	4	Wichtiger Platz belegt
	3	Wichtiger Platz leer
	2	Timeout Turmbewegen
	1	Timeout Kopfbewegen
	0	Timeout Tischbewegen

4.2.8 Mehrspindelmaschine (SPS)

Mehrspindelmaschineobjekte:

Identifizier	Daten	Beschreibung
190	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
290	00bbbbrr (binär)	Lowlevel: bbbb = 1001 Werkzeugkopf 1010 Tisch 1100 Werkzeug 1110 Werkzeugwechsel zu Nr. rr (1,2,3) rr = 00 aus 01 ein / links / vor / rauf 10 / rechts / zurück / runter
R 391		Status anfordern
391	yy, zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7 .. 5	linker Platz Status
	4 .. 2	rechter Platz Status
	1	Revolverkopf unten
	0	Revolverkopf dazwischen
zz	7 .. 5	Werkzeug Status
	4	Programm läuft
	3 .. 2	nicht verwendet
	1 .. 0	Tisch Status

Platz Status (3-bit)	
0	frei
1	Teil ist unbearbeitet
2	Teil wird bearbeitet
3	Teil ist bearbeitet
4	Bearbeitung abgebrochen

Werkzeug Status (3-bit)	
0	Werkzeug 1
1	zwischen 1 und 2
2	Werkzeug 2
3	zwischen 2 und 3
4	Werkzeug 3
5	zwischen 3 und 1

Tisch Status (2-bit)	
0	ist links
1	ist rechts
2	links \gg rechts
3	links \ll rechts

Mehrspindelmaschineobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
490	00, —	Initialisiere Maschine
	01, —	Belade Maschine
	02, —	Entlade Maschine
	03, —	Be&Entlade Maschine
	10, yy	Bearbeitungsprogramm Nr. yy = [0 , 1]
491	00, 00	Programm korrekt beendet
	yy, zz	Fehler ist aufgetreten

Fehlerbytes :

Byte	Bit	Beschreibung
yy	7 .. 2	nicht verwendet
	1	Wichtiger Platz belegt
	0	Ungültige Werkzeugnummer
zz	7	Teil noch nicht bearbeitet
	6	Sequence fehlerhaft
	5	Kopf ist nicht oben
	4	Tisch ist dazwischen
	3	Wichtiger Platz ist leer
	2	Timeout Werkzeugwechsel
	1	Timeout Kopfbewegen
	0	Timeout Tischbewegen

4.2.9 Meßmaschine

Meßmaschineobjekte:

Identifizier	Daten	Beschreibung
1A0	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
2A0	00bbbbrr (binär)	Lowlevel: bbbb = 1001 Meßkopf 1111 Lampe rr = 00 halt / aus 01 rauf / ein 10 runter / blinken 11 Position bestimmen
R 2A3		Meßwert anfordern
2A3	yy	Meßwert yy
R 3A1		Status anfordern
3A1	yy, zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7 .. 5	Platz 2 Status
	4 .. 2	Platz 1 Status
	1	Meßkopf unten
	0	Meßkopf dazwischen
zz	7 .. 5	nicht verwendet
	4	Programm läuft
	3 .. 2	nicht verwendet
	1 .. 0	Tisch Status

Platz Status (3-bit)	
0	frei
1	Teil ist unbearbeitet
2	Teil wird bearbeitet
3	Teil ist bearbeitet
4	Bearbeitung abgebrochen

Tisch Status (2-bit)	
0	Platz 2 unter dem Meßkopf
1	Platz 1 unter dem Meßkopf
2	Wechsel zu Platz 2
3	Wechsel zu Platz 1

Meßmaschineobjekte (Fortsetzung):

Identifizier	Daten	Beschreibung
4A0	00, —	Initialisiere Maschine
	01, —	Belade Maschine
	02, —	Entlade Maschine
	03, —	Be&Entlade Maschine
	04, —	Wechsele die Plätze
	10, —	Meßprogramm
4A1	00, 00	Programm korrekt beendet
	00, zz	Fehler ist aufgetreten

Fehlerbytes :

Byte	Bit	Beschreibung
zz	7	nicht verwendet
	6	Wichtiger Platz belegt
	5	Teil noch nicht bearbeitet
	4	Kopf ist nicht oben
	3	Tisch ist dazwischen
	2	Wichtiger Platz leer
	1	Timeout Kopfbewegen
	0	Timeout Tischbewegen

4.2.10 Heizzelle (SPS)

Heizzelleobjekte:

Identifizier	Daten	Beschreibung
1B0	00	Programm abbrechen
	01	Programmpause
	02	Programm fortsetzen (nur nach Pause)
2B0	30	Heizdraht aus
	31	Heizdraht ein
	34	Ventilator links aus
	35	Ventilator links ein
	38	Ventilator rechts aus
	39	Ventilator rechts ein
R 3B1		Status anfordern
3B1	yy, zz	Status

Status Bytes:

Byte	Bit	Beschreibung
yy	7 .. 5	Platz Status
	4 .. 2	nicht verwendet
	1 .. 0	Dach Status
zz	7 .. 5	nicht verwendet
	4	Programm läuft
	3 .. 0	nicht verwendet

Platz Status (3-bit)		Dach Status (2-bit)	
0	frei	0	offen
1	Teil ist unbearbeitet	1	geschlossen
2	Teil wird bearbeitet	2	geht auf
3	Teil ist bearbeitet	3	geht zu
4	Bearbeitung abgebrochen		

Heizzelleobjekte (Fortsetzung):

Identifer	Daten	Beschreibung
4B0	00, —	Initialisiere Maschine
	05, 00	Dach öffnen
	05, 01	Dach schließen
	10, yy	Heizprogramm Nr. yy = [00 .. 02]
4B1	00, 00	Programm korrekt beendet
	00, zz	Fehler ist aufgetreten

Fehlerbytes :

Byte	Bit	Beschreibung
zz	7 .. 5	nicht verwendet
	4	Timeout Heizen
	3	Sequence fehlerhaft
	2	Dach ist nicht geschlossen
	1	Wichtiger Platz leer
	0	Timeout Dachbewegen