

ER-CIM

User's Manual

Part 2

Appendix F

Catalog #100037

ESHED ROBOTEC 

CIM-ACL System Program Listing

The ER CIM cell includes all necessary ACL programs for CIM production and control.

Following is a listing of the various CIM-ACL programs. This listing is divided into three sections:

- Central controller system programs.
- Station controller system programs.
- ASRS specific system programs.

These ACL programs are for reference only, and in general should not be changed by the user.

Central Controller Program Listing

```
#          <<<<<<  TABLE  >>>>>>

#COMPRESSED PALLET INFORMATION
# 4 bits for each location
GLOBAL D1
GLOBAL D2
GLOBAL D3
GLOBAL D4
GLOBAL D5
GLOBAL D6
GLOBAL D7
GLOBAL D8
GLOBAL D9
GLOBAL D10

GLOBAL STS          ;number of stations
# ----- STATIONS TABLE -----
# CONVEYORS DIVIDED INTO 32 SECTORS
#STATION LOCATION SECTOR NUMBER
DI MG S[ 8]          ;up to 8 stations (S[1] is section number station 1)
DI MG SREF[ 8]       ;previous station location when making changes
DI MG NM[ 6]         ;number of machine in station

DI MG SSTAT[ 6]      ;station status
                    ; 0 : not active
                    ; 1 : i/o problem
                    ; 2 : not responding
                    ; 3 : o.k.
                    ; 4 : error in station

DI MG PLTST[ 6]      ;pallet number in station

#FOR RELEASE
DI MG EPART[ 6]       ;part to release
DI MG ERAW[ 6]        ;raw material to release
DI MG ERAWN[ 6]       ;raw material number to release (for assembly)
DI MG ELINE[ 6]       ;line number in ORDER
DI MG RELES[ 6]       ;flag :
                    ; 1: for release
                    ; 2: after release command is sent
                    ; 0: ready for next

DI MG EASW[ 6]        ;send the command 'W' to start assembly
DI MG ALINE[ 6]       ;line number of assembly part
DI MG APART[ 6]       ;part number for assembly command

DI MG NTMPL[ 6]       ;getting ask for template for part number
DI MG ATMPL[ 6]       ;answere for asking template
                    ; 1 : o.k.
                    ; 2 : missing

DI MG RESET[ 6]       ;=1 send reset command
DI MG RSTAT[ 6]       ;=1 for communication check
                    ; 2 after starting check
                    ; 0 o.k.

#FOR GETTING INFORMATION ON PART
DI MG QPART[ 8]       ;queue for call by communication programs
```

```

DI MG CPART[ 6]      ; part to search for
DI MG CRAW[ 6]      ; raw material to search for
DI MG CRAWN[ 6]     ; raw material number to search for
DI MG CSUBP[ 6]     ; subprocess to search for
DI MG RST[ 6]       ; station number found
DI MG RMA[ 6]       ; machine number found
DI MG RTASK[ 6]     ; task number found
DI MG RCLN[ 6]     ; line found in ORDER table

# ----- WORK -----
#workt[ n] = ..tttmmmssss
#ssss : 4 bit station number
#mmm : 4 bit machine number
#..tt : 24 bit task number
DI MG WORK[ 500]

# ----- PALLETS -----
#PALLET CONTENTS (up to 15 pallets)
DI MG PTMPL[ 15]    ; template type, =0 when pallet empty
DI MG PPART[ 15]    ; finished part number (of part on template)
DI MG PRAW[ 15]     ; raw material
DI MG PRAWN[ 15]    ; raw material number
DI MG PSUBP[ 15]    ; next subprocess
DI MG PINFQ[ 15]    ; information on part (o.k. or not)
DI MG PNEXT[ 15]    ; next station
DI MG PPLAQ[ 15]    ; current location of pallet (1 - 32)
DI MG PSTOP[ 15]    ; =1 when stopped in station, 0 when moving

# ----- ORDER -----
DI MG OTBLP[ 100]   ; priority table
DI MG CPART[ 100]   ; part to make
DI MG CRAW[ 100]    ; raw material
DI MG CRAWN[ 100]   ; raw material number (for assembly)
DI MG OVLN[ 100]    ; line no. in WORKT of first subprocess for this part
DI MG CSUBP[ 100]   ; number of subprocess
DI MG CQTY[ 100]    ; quantity
DI MG CPTS[ 100]    ; number of raw parts in assembly, 0 if none
DI MG CAQ[ 100]     ; quantity of pieces for 1 part (>1 for assembly)
DI MG OASM[ 100]    ; assembly command sent
DI MG CRDL[ 100]    ; ready to release
DI MG OWRK[ 100]    ; in process
DI MG CFIN[ 100]    ; number of finished parts
DI MG CRLS[ 100]    ; >0 number of raw parts left to release
                    ; =0 not releasing
                    ; -1 finished releasing
                    ; -2 missing raw material
                    ; -3 end
                    ; -4 end but missing raw material

GLOBAL CLEN        ; lines in ORDER table

#CONVEYOR SECTORS
DI MG CPLT[ 32]     ; pallet number at location, 0 if none
DI MG CST[ 32]      ; station number at location, 0 if none

# ----- BUFFERS -----
#buffers in stations
DI MG STBTM[ 48]    ; template in buffer
DI MG STBPR[ 48]    ; part
DI MG STBRV[ 48]    ; raw material
DI MG STBRN[ 48]    ; raw material number (in assembly)
DI MG STBSB[ 48]    ; subprocess
DI MG STBLF[ 48]    ; info

```

```

DI MG STBTS[ 48] ; task number
DI MG STBST[ 48] ; status
; 0 : empty
; 1 : contain part
; 2 : part in machine
; 3 : part ready
; 4 : empty template

DI MG RCBST[ 6] ; robot status in station
; 0 : resting
; 10 : get template from pallet
; 11 : pot in buffer
; 20 : get template from buffer
; 21 : pot in pallet
; 30 : get template from buffer
; 31 : pot in second buffer
; 40 : get part from buffer
; 41 : pot in the machine
; 50 : get part from machine
; 51 : pot in buffer
; 60 : get part from buffer
; 61 : pot in storage
; 70 : assemble

DI MG SP1[ 6] ; first parameter
DI MG SP2[ 6] ; second parameter
DI MG SP3[ 6] ; third parameter

DI MG STMCH[ 24] ; machine status
; 0 : empty
; 1 : working
; 2 : loading
; 3 : flipping

DI MG MTIME[ 24] ; time when machine start
DI MG STMCB[ 24] ; buffer number of part in machine
DI MG STMBF[ 24] ; number of buffers for each machine
SET STMBF[ 1] = 2
SET STMBF[ 2] = 2
SET STMBF[ 5] = 2
SET STMBF[ 6] = 2
SET STMBF[ 9] = 2
SET STMBF[ 10] = 2
SET STMBF[ 13] = 2
SET STMBF[ 14] = 2
SET STMBF[ 17] = 2
SET STMBF[ 18] = 2
SET STMBF[ 21] = 2
SET STMBF[ 22] = 2
# <<<<<< VARS >>>>>>

#VECTOR FOR CHARACTERS RECEIVED
DI MG GCOM[ 8]

#pallet location
DI MG STPLT[ 15] ; >0 when pallet is in station
; pallet location
GLOBAL PLT ; pallet number
GLOBAL NEVP ; new location

DI MG QTPL[ 8] ; queue for station want template
GLOBAL STPL ; station number need template
SET STPL = 0 ; 0: no one need template

```

```

#TIME BETWEEN CONVEYOR LOCATIONS
GLOBAL TDLY
SET TDLY = 200 ;for standard 4 station configuration

# for debugging
GLOBAL PRNT
SET PRNT = 0 ;=1, print char received

#station no. of ASRS
GLOBAL ASRS
SET ASRS = 1 ;default: station 1 is ASRS

# number of pallets
GLOBAL MPLT
SET MPLT = 12 ;maximum using

#work status 0: no work, 1: in work, 2: finished work
GLOBAL INWRK
SET INWRK = 0

#station ID (=0 in center)
GLOBAL STNUM
SET STNUM = 0;central controller is always station 0

GLOBAL STNXT ;next reported station
SET STNXT = 1

#do not change station number when printing it
GLOBAL STPND
SET STPND = 1

#flag for adding order, 0: no, 1: replace, 2: add
GLOBAL REPL
SET REPL = 0

#for report
GLOBAL CREP ;flag for no mixing reports
SET CREP = 1
GLOBAL REPN ;report number
GLOBAL REPS ;station number
GLOBAL REPP1 ;parameter 1
GLOBAL REPP2 ;parameter 2
GLOBAL REPP3 ;parameter 3
GLOBAL FREP ;flag 0: no,
; 1: internal (ats) no pallets
; 2: internal (ats) no pallets selected stations
; 6: internal (ats) include pallets
; 7: internal (ats) include pallets selected st.
; 11: external (computer)
DIMG FREPS[6] ; selected station = 1

# direction
GLOBAL DIRCT
SET DIRCT = 1 ; forward direction, -1 backward direction

#max
GLOBAL MAPIX
SET MAPIX = 0 ; for suspend report when getting new order

GLOBAL SUREP
SET SUREP = 0

GLOBAL STUPD ; =1 when update station information

```

```

SET      STUPD = 0

#      <<<<<< REPTB >>>>>

PROGRAM REPTB
DEFIN E  I J K
DEFIN E  WLEN
DEFIN E  ST
DEFIN E  MACH
DEFIN E  TASK

PEND    I FROM CREP

PRI NTLN "%JOA " CLEN
IF CLEN = 0
    GOTO 19
ENDIF

FOR I = 1 TO CLEN
    PRI NTLN "%JOB " CPART[I] CRAW[I] OVLN[I]
    PRI NTLN "%JOC " CSUBP[I] CQTY[I] CPTS[I]
    PRI NTLN "%JOD " CAQ[I] OASM[I] ORDRL[I]
    PRI NTLN "%JOE " CWORK[I] CFIN[I] CRLS[I]
ENDFOR

PRI NTLN "%JE"
PRI NTLN
POST    1 TO CREP

PEND    I FROM CREP

SET     WLEN = OVLN[CLEN] + CSUBP[CLEN] ;first unused work line number
SET     WLEN = WLEN - 1

PRI NTLN "%JVA " WLEN

FOR     I = 1 TO WLEN
    SET     ST = WORKT[I] MOD 16
    SET     MACH = WORKT[I] / 16
    SET     MACH = MACH MOD 16
    SET     TASK = WORKT[I] / 256

    PRI NTLN "%JVVB " ST MACH TASK
ENDFOR

LABEL 19
PRI NTLN "%JE"
PRI NTLN
POST    1 TO CREP

PEND    I FROM CREP
PRI NTLN "%JPA" DRECT
FOR     I = 1 TO 15
    IF     PTMPL[I] > 0
        PRI NTLN "%JPB " I
        PRI NTLN "%JPC " PTMPL[I] PPART[I] PRAW[I]
        PRI NTLN "%JPD " PRAW[I] PSUBP[I] PINFQ[I]
        PRI NTLN "%JPE " PNEXT[I] PPLAQ[I] PSTOP[I]
    ENDIF
ENDFOR
PRI NTLN "%JE"
POST    1 TO CREP

```



```

PEND      I FROM CREP
PRI NTLN  "%TBA"
FOR      I = 1 TO 48
  IF      STBST[I] > 0
    PRI NTLN "%TBB " I
    PRI NTLN "%TBC " STBST[I] STBTM[I] STBPR[I]
    PRI NTLN "%TBD " STBF[I] STBRM[I] STBRN[I]
    PRI NTLN "%TBE " STBSB[I] STBTS[I]
  ENDF
ENDFOR
PRI NTLN  "%TE"
POST     1 TO CREP

PEND      I FROM CREP
PRI NTLN  "%TSA " STS ASRS
SET      J = 1
FOR      I = 1 TO STS
  PRI NTLN "%TSB " S[I]      SSTAT[I] RELES[I]
  PRI NTLN "%TSC " ELINE[I] EASW[I] ALINE[I]
  PRI NTLN "%TSD " PLTST[I] ROBST[I] NTMPL[I]
  PRI NTLN "%TSE " SP1[I]   SP2[I]   SP3[I]
  PRI NTLN "%TSF " NM[I]

  PRI NTLN "%TSG " STMBF[J] STMCH[J] STMCB[J]
  SET      J = J + 1
  PRI NTLN "%TSH " STMBF[J] STMCH[J] STMCB[J]
  SET      J = J + 1
  PRI NTLN "%TSI " STMBF[J] STMCH[J] STMCB[J]
  SET      J = J + 1
  PRI NTLN "%TSJ " STMBF[J] STMCH[J] STMCB[J]
  SET      J = J + 1
ENDFOR
PRI NTLN  "%TE"

POST     1 TO CREP

END
#      <<<<<< REPORT >>>>>

#i ncl ude REPORT.DMC

PROGRAM $REPI

DEFI NE BF
DEFI NE M
DEFI NE B

IF      FREP = 2
  ORIF  FREP = 7
  ANDIF REPS > 0
  IF      FREPS[REPS] = 0
    GOTO 99
  ENDF
ENDIF

IF      REPN >= 3
  ANDIF REPN <= 4
  ANDIF FREP <= 5
  GOTO 99
ENDIF

IF      REPS = 0

```

```

    PRINTLN "CENTRAL : "
ELSE
    IF REPS = ASRS
        PRINTLN "ASRS : "
    ELSE
        PRINTLN "STATION REPS ": "
    ENDIF
ENDIF

```

```

IF REPN > 30
    GOTO 30
ENDIF
IF REPN > 20
    GOTO 20
ENDIF
IF REPN > 10
    GOTO 10
ENDIF

```

```

IF REPN = 1
    PRINT "LOCATION " REPP1
ENDIF

```

```

IF REPN = 2
    PRINT "LOCATION " REPP1
    PRINT ", STATUS: "
    IF REPP2 = 0
        PRINT "NOT ACTIVE"
    ENDIF
    IF REPP2 = 1
        PRINT "I/O PROBLEM"
    ENDIF
    IF REPP2 = 2
        PRINT "NOT RESPONDING"
    ENDIF
    IF REPP2 = 3
        PRINT "COMMUNICATION O.K."
    ENDIF
ENDIF

```

```

IF REPN = 3
    PRINT "PALLET #" REPP1 " ENTER"
ENDIF

```

```

IF REPN = 4
    PRINT "PALLET #" REPP1 " LIVE"
ENDIF

```

```

IF REPN = 5
    PRINT "TEMPLATE #" REPP1
    PRINT " IN ASRS"
ENDIF

```

```

IF REPN = 6
    PRINT "RESET"
ENDIF

```

```

IF REPN = 7
    PRINT "RESET"
ENDIF

```

```

IF REPN = 8
    PRINT "ERROR #" REPP1
ENDIF

```

```

GOTO 99
LABEL 10

IF REPN = 11
  PRINT "GET COMMAND TO RELEASE RAW MATERIAL #"
  PRINT REPP2 " FOR PART #" REPP1
ENDIF

IF REPN = 12
  PRINT "RAW MATERIAL #" REPP2
  PRINT " FOR PART #" REPP1
  PRINT " RELEASED"
ENDIF

IF REPN = 13
  PRINT "CANNOT RELEASED RAW MATERIAL #"
  PRINT REPP2 " FOR PART #" REPP1
ENDIF

GOTO 99
LABEL 20

IF REPN = 21
  PRINT " NEAD TEMPLATE FOR PART #" REPP1
ENDIF

IF REPN = 22
  PRINT "TEMPLATE FOR PART #" REPP1
  PRINT " MOVING TO STATION " REPP2
ENDIF

IF REPN = 23
  PRINT "MISSING TEMPLATE FOR PART #" REPP1
  PRINT " FOR STATION " REPP2
ENDIF

IF REPN = 24
  PRINT "GET COMMAND TO ASSEMBLAE PART #" REPP1
ENDIF

IF REPN = 25
  PRINT "FINISH ASSEMBLE PART #" REPP1
ENDIF

IF REPN = 26
  PRINT "CANNOT ASSEMBLE PART #" REPP1
ENDIF

IF REPN = 27
  PRINT "PART #" REPP1 " ASSEMBLE"
ENDIF

IF REPN = 28
  PRINT "PART #" REPP1 " READY"
ENDIF

GOTO 99
LABEL 30

SET BF = REPP1 MOD 8 ;station buffer
SET M = BF / 2 ;station machine
SET M = M + 1
SET B = BF MOD 2 ;machine buffer

```

```

SET B = B + 1

IF REPN = 31
  PRINT "PART #" REPP2
  PRINT "FINISH SUBPROCESS #" REPP3
ENDIF

IF REPN = 32
  PRINT "LOADING MACHINE #" M
  PRINT "WITH PART #" REPP2
ENDIF

IF REPN = 33
  PRINT "UNLOADING MACHINE #" M
  PRINT "PART #" REPP2
ENDIF

IF REPN = 34
  PRINT "MOVING PART #" REPP2
  PRINT "TO STORAGE IN QUEUE"
ENDIF

IF REPN = 35
  PRINT "MOVING TEMPLATE"
  IF REPP2 = 0
    PRINT "(EMPTY)"
  ELSE
    PRINT "HOLDING PART #" REPP2
  ENDIF
  PRINT "FROM PALLET TO "
  IF REPS = ASRS
    ANDIF BF = 2
    PRINT "ASRS"
  ELSE
    PRINT "BUFFER #" BF
  ENDIF
ENDIF

IF REPN = 36
  PRINT "NEW PART IN BUFFER, RAW MATERIAL #"
  PRINT REPP3 "FOR PART #" REPP2
  PRINT "IN BUFFER #" BF
ENDIF

IF REPN = 37
  PRINT "MOVING TEMPLATE"
  PRINT "HOLDING PART #" REPP3
  PRINT "FROM BUFFER #" BF
  PRINT "TO BUFFER #" REPP2
ENDIF

IF REPN = 38
  PRINT "MOVING TEMPLATE "
  IF REPP2 = 0
    PRINT "(EMPTY)"
  ELSE
    PRINT "HOLDING PART #" REPP2
  ENDIF
  PRINT "FROM BUFFER #" BF
  PRINT "TO PALLET"
# REPP3 = INFO
ENDIF

```

LABEL 99

END

GLOBAL \$REP ;report o.k.
GLOBAL \$REPC ;report error count
GLOBAL \$REPA ;report aknol age

PROGRAM \$REPX
END

PROGRAM \$VREP ;wait 1 minute, cancell the transmtion
DELAY 6000
SET \$REP = 0
STOP \$REPX
SET \$REPC = \$REPC + 1
IF \$REPC > 20
SET FREP = 0
ENDIF
END

PROGRAM \$REPX /Y
WAIT SUREP = 0
RUN \$VREP
PRINTLN "%RA" REPN REPS REPP1
PRINTLN "%RB" REPP2 REPP3
SET \$REP = 0
STOP \$VREP
SET \$REPC = 0
SET \$REPA = \$REPA + 1
;IF \$REPA > 50
; SET FREP = 0
;ENDIF
END

PROGRAM \$REP

IF FREP > 0
ANDIF FREP < 100
GOSUB \$REPI
ENDIF

IF FREP >= 100
ANDIF FREP < 200
SET \$REP = 1
RUN \$REPX
WAIT \$REP = 0
ENDIF
END

PROGRAM \$REP2
SET REPP3=0
GOSUB \$REP
END

PROGRAM \$REP1
SET REPP3=0
SET REPP2=0

```
GOSUB $REP
END
```

```
PROGRAM $REP0
SET REPP3=0
SET REPP2=0
SET REPP1=0
GOSUB $REP
END
```

```
PROGRAM REPT
DEFIN E I C
LABEL 1
FOR I = 1 TO 6
  IF SREF[I] > 0
    SET SSTAT[I] = IN[I] * 2
    SET RSTAT[I] = 1
  ENDF
ENDFOR
DELAY 200
FOR I = 1 TO 6
  IF RSTAT[I] > 0
    .REP1 2 I SSTAT[I]
  ENDF
ENDFOR
DELAY 30000
GOTO 1
END
```

```
PROGRAM STREP
```

```
STOP REPT
DELAY 20
SET FREP = 100
RUN REPTB
RUN REPT
END
```

```
PROGRAM ENREP
```

```
STOP REPT
SET FREP = 0
POST 1 TO GREP
END
```

```
# <<<<< PTCI >>>>>
```

```
#This program gets information on a part for use by communication programs
#to call this program
#set CPART[ <com number> ] = PART to search for
#set CRAVN[ <com number> ] = RAW MATERIAL to search for
#set CRAVN[ <com number> ] = RAW MATERIAL number to search for (assembly)
#set RCLN[ <com number> ] = 0
#qsend <com number> to CPART
#wait until RCLN[ <..> ] > 0
#the return code is :
#RST[ <com number> ] = station
#RMA[ <com number> ] = machine
#RTASK[ <com number> ] = task
#RCLN[ <com number> ] = line in ORDER
```

```
PROGRAM PTCI
```

```

DEFINE CCM          ;port number
DEFINE LINE         ;order line number
DEFINE WLINE       ;work table line number
DEFINE I
DEFINE PRT          ;part to search
DEFINE RW           ;raw to search

LABEL 1
QPEND CCM FROM CPART          ;port question

IF CPART[CCM] = 0             ;empty template
SET RST[CCM] = ASRS          ;target station is ASRS
SET RMA[CCM] = 1             ;machine # 1 is always asrs
SET RTASK[CCM] = 999         ;task = 999 for unloading
SET ROLN[CCM] = CLEN + 1     ;point not in order table
GOTO 1
ENDIF

IF CPART[CCM] < 0            ;template for part
SET PRT = -CPART[CCM]
SET RW = 0
ELSE
SET PRT = CPART[CCM]
SET RW = CRAW[CCM]
ENDIF

SET LINE = 0                 ;initialize LINE
IF CRAW[CCM] > 0              ;if assembly, search the main part
SET RW = 0
ENDIF
FOR I = 1 TO CLEN             ;loop through entire order
IF CPART[I] = PRT             ;if line is found with the same part
ANDIF CRAW[I] = RW            ;and the same raw material
SET LINE = I + CRAW[CCM]      ;tag the line
                                ;added (+ CRAW[CCM]) in 28/7/92
IF OWRK[LINE] > 0             ;if some part is in process
SET I = CLEN + 1              ;don't search another line
ENDIF
ENDIF
ENDFOR

SET WLINE = OWIN[LINE] + CSUBP[CCM] ;line number in work table

SET I = WORK[WLINE]           ;get value from work table
SET RST[CCM] = I MOD 16        ;station number
SET I = I / 16
SET RMA[CCM] = I MOD 16        ;machine number
SET RTASK[CCM] = I / 16        ;task number
SET ROLN[CCM] = LINE          ;order line number

GOTO 1

END
# <<<<<<  RELEASE  >>>>>>

#include REPORT.DMC

GLOBAL PTOST
SET PTOST = 2 ;number of pallets going can go to a station

PROGRAM RLS1 ;control of free template
LABEL 1

```

```

QPEND  ST MPL FROM QT MPL ;next station number need template
WAIT   ST MPL = 0 ;released
GOTO   1
END

#program checks parts to be released for production
#find station to do it that is not busy (previous release has finished)
# set EPART[ <station>] = PART to release
# set ERAV[ <station>] = RAW MATERIAL to release
# set ELINE[ <station>] = line in ORDER
# set RELES[ <station>] = 1 flag to release, =0 when release is finished

# release rules:
# 1. release only if no pallet waits for station
#    where first subprocess has to execute
# 2. after second release, wait until first subprocess is finished
#    before trying to release next one
# 3. assembly part: wait until all subparts are finished
#    before release

#program that starts the manufacturing process
PROGRAM MFCT
PRI NTLN "mf c"
PRI NTLN
#( PROGRAM RELES)
DEFINE IP ;index in priority table
DEFINE I ;index in order table
DEFINE JK ;help variables
DEFINE MN ;number of assembly sets ready
DEFINE I1 ;first order line of subparts
DEFINE I2 ;last order line of subparts
DEFINE ST0 ;station for release
DEFINE ST1 ;station for first subprocess
DEFINE FLAG ;=1 in end loop when finished
DEFINE CLEN1 ;contains CLEN-1
DEFINE C

IF MAPIX = 1
  WAIT CLEN > 0
ELSE
  IF CLEN = 0 ;empty order
    GOTO 9
  ENDF
ENDF

STOP RLS1
DELAY 20

RUN RLS1

SET INWRK = 1 ;indicates start of manufacturing
.REPO 17 0
LABEL 1
SET CLEN1 = CLEN - 1
SET FLAG = 0 ;flag specifying that all release requests
; have been completed

IF ST MPL > 0 ;station need template
ANDIF RELES[ ASRS] = 0 ;ASRS ready to get release command
SET EPART[ ASRS] = 0 ;part
SET ERAV[ ASRS] = -NT MPL[ ST MPL] ;raw material (empty template)

```



```

SET     ERAWV[ASRS] = 0
SET     ELINE[ASRS] = 0           ; dummy order line number
SET     RELES[ASRS] = 1         ; station received release command
SET     FLAG = 1
ENDIF

FOR     IP = 1 TO QLEN
  LABEL 2
  SET     I = OTBIP[IP]         ; get order line number from priority table
  IF     ORLS[I] > -3           ; not finished
    SET     FLAG = 1
  ENDIF
  IF     ORLS[I] > 0             ; can release
    SET     J = OVRK[I] + CFIN[I]
    IF     J >= CQTY[I]
      SET     ORLS[I] = -1       ; release is finished
    ELSE
      IF     OPTS[I] > 0         ; release assembly part
        ANDIF OASM[I] >= 0      ; = -1 if cannot assembly
          SET     MN = 999999    ; only if all subparts ready
          SET     I1 = I + 1     ; first subpart
          SET     I2 = I + OPTS[I] ; last subpart
          FOR     J = I1 TO I2
            SET     K = CFIN[J] / OAJ[J] ; number of possible assemblies
                                           ; with current subpart quantity
            IF     K < MN
              SET     MN = K
            ENDIF
          ENDFOR
          SET     MN = MN - CFIN[I] ; calculate number of parts needed to do
          SET     MN = MN - OVRK[I]
          SET     MN = MN - CRDL[I]
          SET     MN = MN - OASM[I]
          SET     J = OVLN[I]
          SET     ST0 = WORTK[J] MOD 16
          IF     MN > 0
            ANDIF EASWV[ST0] = 0
              SET     EASWV[ST0] = 1
              SET     APART[ST0] = OPART[I]
              SET     ALINE[ST0] = I
              SET     OASM[I] = OASM[I] + 1
            ENDIF
          ENDIF
          IF     CRDL[I] > 0
            SET     J = OVLN[I] ; line number in work table
            SET     ST0 = WORTK[J] MOD 16 ; station where released
            SET     J = J + 1 ; line of subprocess 1
            SET     ST1 = WORTK[J] MOD 16 ; station where first executed
            IF     ST1 <> ASRS ; unlimited place in asrs
              SET     K = 0
              FOR     J = 1 TO 15 ; check if another part
                ; is on the way to that station
                IF     PNEXT[J] = ST1 ; next station pallet
                  SET     K = K + 1
                ENDIF
              ENDFOR
              IF     K >= PTOST
                SET     ST1 = 0
              ENDIF
            ENDIF
            IF     ST1 > 0
              ANDIF RELES[ST0] = 0 ; >0 if release command was sent
                ; already in station
              SET     CRDL[I] = CRDL[I] - 1 ;
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDFOR

```

```

SET      EPART[ STO] = CPART[ I] ; part
SET      ERAV[ STO] = CRAV[ I] ; raw material
SET      ERAVW[ STO] = CRAVW[ I] ; raw material number (assemb.)
SET      ELINE[ STO] = I ; order line number
SET      CRLS[ I] = CRLS[ I] - 1 ; one part released
SET      OVRK[ I] = OVRK[ I] + 1 ; one more part in process
SET      RELES[ STO] = 1 ; station received release command
IF      IP < CLEN ; if current part is not last priority
  FOR    J = IP TO CLEN ; move current part to last priority
    SET   K = J + 1
    SET   OTBP[ J] = OTBP[ K]
  ENDFOR
  ENDF
  SET   OTBP[ CLEN] = I
  GOTO  2
  ENDF
ENDF
ENDF
ELSE
  IF    CRLS[ I] = -1 ; finish release
  ORIF CRLS[ I] = -2
  IF    OVRK[ I] = 0 ; finish working
    SET CRLS[ I] = CRLS[ I] - 2 ; end
  ENDF
ENDF
ENDF
ENDFOR
DELAY  20
IF    FLAG = 1 ; if =0 then program ends
  GOTO 1
ENDF
LABEL 9
IF    MAPIX = 1
  GOTO 1
ENDF
STOP  RLS1
SET   INWRK = 2 ; indicates entire order is ready
.REPO 19 0
END

```

```
#          <<<<<  PALLETS  >>>>>
```

```
#include REPORT.DMC
```

```
PROGRAM NEWP
```

```
#change PALLET place
# parameters are:
# PLT pallet number
# NEWP new place
```

```
DEFINE OLDP ; contains old pallet location
DEFINE I ; for loop on all pallets
```

```
SET OLDP = PPLAQ[ PLT] ; save old location before changing it
IF OLDP <> NEWP ; check if need to change
  SET PPLAQ[ PLT] = NEWP ; update location
  IF CPLT[ NEWP] = 0 ; if new location is tagged as empty,
    ANDIF PPAR[ PLT] > 0 ; and pallet contains a part,
```

```

    SET      CPLT[NEVP] = PLT      ; tag it (location contains part)
  ENDF
  IF      CLDP > 0                ; (=0 only for the first time)
    ANDIF  CPLT[CLDP] = PLT      ; if old location tagged
          ; with that pallet,
    SET      CPLT[CLDP] = 0      ; untag it
    FOR    I = 15 TO 1          ; checks if another nonempty pallet
      IF    PPLAQ[I] = CLDP      ; is in old location
        ANDIF PPART[I] > 0      ; and tags if there is
          SET      CPLT[CLDP] = I
      ENDF
    ENDFOR
  ENDF
END

```

PROGRAM CPPLT

compresses PALLET information into 8 vars

```

DEFINE I      ; compressed var (1 - 8)
DEFINE J      ; part of compressed var (1 - 4)
DEFINE K      ; location number to compress (1 - 32)
DEFINE TMP     ; help var to compute compressed var
DEFINE P      ; part on location
DIM TT[10]    ; save compressed vars

SET      K = 1                ; location number
FOR      I = 1 TO 8          ; compressed var number
  SET      TMP = 0            ; computing value
  FOR    J = 1 TO 4          ; part of compressed var
    SET      TMP = TMP * 16    ; shift previous value 4 bits left
    SET      P = CPLT[K]      ; pallet number in location
    IF      P > 0            ; pallet exists
      SET      P = PPART[P]    ; part on pallet
      IF      P > 0
        SET      P = P - 1    ; convert to (1 - 15) values
        SET      P = P MOD 15
        SET      P = P + 1
      ENDF
    ENDF
    SET      TMP = TMP + P    ; add new value
    SET      K = K + 1        ; next location
  ENDFOR
  SET      TT[I] = TMP        ; save compressed value
ENDFOR

SET      D1 = TT[1]          ; set compressed vars
SET      D2 = TT[2]
SET      D3 = TT[3]
SET      D4 = TT[4]
SET      D5 = TT[5]
SET      D6 = TT[6]
SET      D7 = TT[7]
SET      D8 = TT[8]

END

```

```

#macro NEXTC
SET .1 = .2 + DIRECT
IF DIRECT = 1
  IF .1 > 32

```

```

    SET      .1 = 1
  ENDF
ELSE
  IF      .1 < 1
    SET      .1 = 32
  ENDF
ENDF
#endm

```

```
PROGRAM PLTS
```

```
#track pallets
```

```
#update pallet location on moving conveyor
```

```
#update pallet location using station information
```

```

DEFINE T1      ;time (for pallet location update intervals)
DEFINE A      ;old location of pallet
DEFINE P      ;next location; check if station defined there
DEFINE C      ;used for report

SET      T1 = TIME      ;get starting time
LABEL    1
FOR      PLT = 1 TO 12      ;track 12 pallets
  IF      STPLT[PLT] > 0      ;if station reported that pallet passed by
    SET      A = STPLT[PLT]      ;update location according to the report
    SET      NEVP = S[A]
    GOSUB    NEVP
    SET      STPLT[PLT] = 0      ;clear the report
  ELSE
    IF      PSTOP[PLT] = 0      ;0: pallet stopped 1: pallet moving
      SET      A = PPLAQ[PLT] ;old location
      .NEXTC NEVP A
      SET      P = CPLT[NEVP]      ;pallet number at new location or zero
      IF      P = 0      ;advance pallet to empty place
        ORIF  PPART[P] = 0      ; or empty pallet
        GOSUB    NEVP
        .NEXTC P NEVP
        IF      CST[P] > 0      ;station number in location or zero
          SET      PSTOP[PLT] = 1 ;stop pallet if station defined
                    ; at next sector
        ENDF
      ELSE
        IF      A > 0      ;if old location is a station
          ANDIF  CST[A] > 0      ;always move the pallet
          GOSUB    NEVP
          .NEXTC P NEVP
          IF      CST[P] > 0      ;station number in location or zero
            SET      PSTOP[PLT] = 1 ;stop pallet if station defined
                    ; at next sector
          ENDF
        ENDF
      ENDF
    ENDF
  ENDF
ENDFOR
GOSUB    CPPLT      ;compressed location information
SET      T1 = T1 + TDLY      ;time for next check

LABEL    2      ;when waiting for delay between checks
FOR      PLT = 1 TO 12      ; follow station report
  IF      STPLT[PLT] > 0      ;station report
    SET      A = STPLT[PLT]      ;update pallet location by station report

```

```

        SET      NEVP = S[A]
        GOSUB   NEVP
        SET      STPLT[PLT] = 0 ; clear report
        GOSUB   CPPLT          ; compress location information
    ENDF
ENDFOR

DELAY 20          ; delay to reduce load on system
IF TIME < T1     ; time delay before next movement of pallets
    GOTO 2        ; continue tracking stations
ENDIF
.REPO 90
GOTO 1
END

```

```
# <<<< COM >>>>
```

```
GLOBAL COMN
GLOBAL COMPR
```

```

PROGRAM RDCOM
IF COMN = 1
    READCOM 1, COMPR
ENDIF
IF COMN = 2
    READCOM 2, COMPR
ENDIF
IF COMN = 3
    READCOM 3, COMPR
ENDIF
IF COMN = 4
    READCOM 4, COMPR
ENDIF
IF COMN = 5
    READCOM 5, COMPR
ENDIF
IF COMN = 6
    READCOM 6, COMPR
ENDIF
END

```

```

PROGRAM PROCOM
IF COMN = 1
    PRLNCOM 1, COMPR
ENDIF
IF COMN = 2
    PRLNCOM 2, COMPR
ENDIF
IF COMN = 3
    PRLNCOM 3, COMPR
ENDIF
IF COMN = 4
    PRLNCOM 4, COMPR
ENDIF
IF COMN = 5
    PRLNCOM 5, COMPR
ENDIF
IF COMN = 6
    PRLNCOM 6, COMPR
ENDIF
END

```

```

PROGRAM SNOOM
IF      COVN = 1
  SENCOM 1, CCMPR
ENDIF
IF      COVN = 2
  SENCOM 2, CCMPR
ENDIF
IF      COVN = 3
  SENCOM 3, CCMPR
ENDIF
IF      COVN = 4
  SENCOM 4, CCMPR
ENDIF
IF      COVN = 5
  SENCOM 5, CCMPR
ENDIF
IF      COVN = 6
  SENCOM 6, CCMPR
ENDIF
END

```

wait until byte is received on rs232

```

PROGRAM WAI T1
LABEL 1
WAI T  GCOM[ 1] = 0
GETCOM 1, GCOM[ 1]
GOTO 1
END

```

```

PROGRAM WAI T2
LABEL 1
WAI T  GCOM[ 2] = 0
GETCOM 2, GCOM[ 2]
GOTO 1
END

```

```

PROGRAM WAI T3
LABEL 1
WAI T  GCOM[ 3] = 0
GETCOM 3, GCOM[ 3]
GOTO 1
END

```

```

PROGRAM WAI T4
LABEL 1
WAI T  GCOM[ 4] = 0
GETCOM 4, GCOM[ 4]
GOTO 1
END

```

```

PROGRAM WAI T5
LABEL 1
WAI T  GCOM[ 5] = 0
GETCOM 5, GCOM[ 5]
GOTO 1
END

```

```

PROGRAM WAI T6
LABEL 1
WAI T  GCOM[ 6] = 0
GETCOM 6, GCOM[ 6]
GOTO 1

```

```

END
#          <<<<<<  CCM  >>>>>>

#i ncl ude  REPORT.DMC

#i ncl ude  CCM.DMC

PROGRAM  CCM

#check r ecei ved rs232 byt e and act accordi ngly

DEFI NE  PALET
DEFI NE  TMLT
DEFI NE  LBFR
DEFI NE  BFR
DEFI NE  MACH
DEFI NE  ERR
DEFI NE  I J
DEFI NE  C      ;for report
DEFI NE  CHAR

STOP    VAI T1      ;if al ready runni ng
STOP    VAI T2      ;if al ready runni ng
STOP    VAI T3      ;if al ready runni ng
STOP    VAI T4      ;if al ready runni ng
STOP    VAI T5      ;if al ready runni ng
STOP    VAI T6      ;if al ready runni ng
DELAY  20

SET     GCOM[ 1] = 0
SET     GCOM[ 2] = 0
SET     GCOM[ 3] = 0
SET     GCOM[ 4] = 0
SET     GCOM[ 5] = 0
SET     GCOM[ 6] = 0
RUN     VAI T1
RUN     VAI T2
RUN     VAI T3
RUN     VAI T4
RUN     VAI T5
RUN     VAI T6

LABEL   1

FOR     CCM = 1 TO 6
  IF    GCOM[ CCM] > 0

    SET   CHAR = GCOM[ CCM]

    IF    CHAR = ' P'      ;pal let was in station
      CR I F  CHAR = ' Q
      CR I F  CHAR = ' W
      CR I F  CHAR = ' I '
      CR I F  CHAR = ' O
      CR I F  CHAR = ' Y
      . READCCM PALET      ;get pal let number
      IF    PALET = 0      ;il legal pal let number
        CR I F  PALET > 15
        SET    PALET = 15
      ENDI F
      SET    PLTST[ CCM] = PALET
    ENDI F

```

```

IF      CHAR = ' O      ; pal l et l oaded
  CRIF  CHAR = ' W
  CRIF  CHAR = ' F
  CRIF  CHAR = ' L'
  CRIF  CHAR = ' U
  CRIF  CHAR = ' Z
  CRIF  CHAR = ' C
  CRIF  CHAR = ' N
  . READCOM LBFR
  SET   BFR = COMN - 1
  SET   BFR = BFR * 8
  SET   BFR = BFR + LBFR
ENDIF

```

```

IF      CHAR = ' M
  CRIF  CHAR = ' m
  CRIF  CHAR = ' X
  CRIF  CHAR = ' x'
  CRIF  CHAR = ' J'
  CRIF  CHAR = ' j'
  . READCOM J
  SET   MACH = COMN - 1
  SET   MACH = MACH * 4
  SET   MACH = MACH + J
ENDIF

```

```

IF      CHAR = ' A'
  SET   I = ALINE[ COMN]
  SET   EASW[ COMN] = 0
  SET   CRDRL[ I] = CRDRL[ I] + 1
  IF    O'ASM[ I] > 0
    SET   O'ASM[ I] = O'ASM[ I] - 1
  ENDIF
  . REPO 25 COMN
ENDIF

```

```

IF      CHAR = ' a'
  SET   ROBST[ COMN] = 110
  . REPO 71 COMN
ENDIF

```

```

IF      CHAR = ' t'
  SET   ROBST[ COMN] = 120
  . REPO 72 COMN
ENDIF

```

```

IF      CHAR = ' f'
  SET   ROBST[ COMN] = 130
  . REPO 73 COMN
ENDIF

```

```

IF      CHAR = ' e'
  SET   ROBST[ COMN] = 140
  . REPO 74 COMN
ENDIF

```

```

IF      CHAR = ' n'
  SET   ROBST[ COMN] = ROBST[ COMN] + 1
  . REPO 81 COMN
ENDIF

```



```

IF      CHAR = 'B'          ;rs232 communication o.k.
SET     RSTAT[ COMN] = 0
SET     SSTAT[ COMN] = I N[ COMN] * 2
SET     SSTAT[ COMN] = SSTAT[ COMN] + 1
.REP1 2 COMN SSTAT[ COMN]
ENDIF

IF      CHAR = 'C'
.READCOM I
.READCOM SP3[ COMN]        ;info
SET     J = COMN - 1
SET     J = J * 8
SET     I = I + J
SET     ROBST[ COMN] = 30
SET     SP1[ COMN] = BFR
SET     SP2[ COMN] = I
.REP 36 COMN BFR I SP3[ COMN]
ENDIF

IF      CHAR = 'c'
IF      ROBST[ COMN] = 30
SET     ROBST[ COMN] = 31
.REPO 46 COMN
ELSE
SET     ROBST[ COMN] = 0
SET     BFR = SP1[ COMN]
SET     I = SP2[ COMN]
SET     STBTM[ I] = STBTM BFR
SET     STBPR[ I] = STBPR BFR
SET     STBRV[ I] = STBRV BFR
SET     STBRN[ I] = STBRN BFR
SET     STBSB[ I] = STBSB BFR
SET     STBTS[ I] = STBTS BFR
SET     STBFI[ I] = SP3[ COMN]
SET     STBST[ I] = 1
SET     STBST[ BFR] = 0
.REPO 56 COMN
ENDIF
ENDIF

IF      CHAR = 'D'
SET     I = ALI NE[ COMN]
SET     EASM[ COMN] = 0
SET     CRLS[ I] = -2
; set all subpart CRLS[ I] = -1 (if > -1)
;
SET     O/ASM[ I] = -1
.REPO 26 COMN
ENDIF

IF      CHAR = 'E'          ;answer on rel.: missing material
IF      ERAV[ COMN] < 0    ;template needed
.REPO 23 STMPL
SET     ATMPL[ STMPL] = 2      ;send 'K' to answare "no template"
SET     NTMPL[ STMPL] = 0
SET     STMPL = 0
ELSE
SET     I = ELI NE[ COMN]
SET     CRLS[ I] = -2          ;finish release (no more material)
SET     O/CRK[ I] = O/CRK[ I] - 1 ;the unreleased one is not in process

```

```

    .REPO 13 COMN
  ENDF
  SET    RELES[COMN] = 0          ; can release next
ENDF

IF     CHAR = 'F' ; finished task
SET    CPART[COMN] = STBPR[BFR] ; part
SET    CRAV[COMN] = STBRV[BFR]  ; raw material
SET    CRAVN[COMN] = STBRN[BFR] ; raw material number
SET    CSUBP[COMN] = STBSB[BFR] ; subprocess number finished
SET    STBSB[BFR] = STBSB[BFR] + 1 ; increment subprocess number
IF     CPART[COMN] <> 0
  .REP1 31 COMN BFR
ELSE
  .REP1 5 COMN STBTM[BFR]
ENDF
IF     CRAV[COMN] >= 0
  SET    RCLN[COMN] = 0
  QPOST COMN TO QPART          ; getting part information
  WAIT   RCLN[COMN] > 0
  SET    I = RCLN[COMN]        ; line number in order
  IF     CSUBP[COMN] = 1       ; if first subprocess finished,
    ANDIF ORLS[I] >= 0        ; can release one more
    SET   ORLS[I] = ORLS[I] + 1
  ENDF
  IF     RTASK[COMN] >= 900    ; unloading
  IF     I <= QLEN            ; if valid order line number
    SET   CFIN[I] = CFIN[I] + 1 ; one more part is ready
    IF     OVKR[I] > 0        ; one part finished work
      SET   OVKR[I] = OVKR[I] - 1
    ENDF
  ENDF
  SET    STBPR[BFR] = 0
  SET    STBRV[BFR] = 0
SET    STBRN[BFR] = 0
SET    STBSB[BFR] = 0
  .SENCOM 'F'                  ; finish
  .PRLNCOM LBFR
  IF     CPART[COMN] <> 0
    .REP2 29 COMN BFR I
  ENDF
ELSE
  SET    CSUBP[COMN] = CSUBP[COMN] + 1 ; next subprocess
  SET    RCLN[COMN] = 0
  QPOST COMN TO QPART          ; part information
  WAIT   RCLN[COMN] > 0
  IF     RST[COMN] = COMN      ; if next station=current station
    .SENCOM 'C'                ; continue:
    .PRLNCOM RMA[COMN]         ; machine
    .PRLNCOM RTASK[COMN]       ; task
    .PRLNCOM LBFR
  ELSE
    .SENCOM 'M'                ; put part on the conveyor
    .PRLNCOM LBFR
  ENDF
ENDF
ELSE
  .SENCOM 'M'
  .PRLNCOM LBFR
ENDF
ENDF

```

```

IF CHAR = 'I' ; station released the pallet
  SET PSTOP[ PALET] = 0 ; pallet leaves the station
  . REP1 4 COMN PALET
ENDIF

IF CHAR = 'J'
  SET STMCH[ MACH] = 3
  . REP1 65 COMN MACH
ENDIF

IF CHAR = 'j'
  IF STMCH[ MACH] = 3
    SET STMCH[ MACH] = 0
  ENDIF
  . REP1 66 COMN MACH
ENDIF

IF CHAR = 'K'
  . READCOM CPART[ COMN]
  SET ROBST[ COMN] = 70
  SET SP1[ COMN] = CPART[ COMN]
  . REP1 27 COMN CPART[ COMN]
ENDIF

IF CHAR = 'k'
  SET ROBST[ COMN] = 0
  SET CPART[ COMN] = SP1[ COMN]
  . REP1 28 COMN SP1[ COMN]
ENDIF

IF CHAR = 'L'
  SET ROBST[ COMN] = 40
  SET SP1[ COMN] = BFR
  . REP1 32 COMN BFR
ENDIF

IF CHAR = 'I'
  IF ROBST[ COMN] = 40
    SET ROBST[ COMN] = 41
    . REPO 42 COMN
  ELSE
    SET BFR = SP1[ COMN]
    SET ROBST[ COMN] = 0
    SET STBST[ BFR] = 2
    SET MACH = BFR + 1
    SET MACH = MACH / 2
    SET STMCH[ MACH] = BFR ; save buffer number in machine
    . REPO 52 COMN
  ENDIF
ENDIF

IF CHAR = 'M'
  SET STMCH[ MACH] = 1
  SET TIME[ MACH] = TIME
  . REP1 61 COMN MACH
ENDIF

IF CHAR = 'm'
  SET STMCH[ MACH] = 0
  SET I = TIME - TIME[ MACH]

```

```

SET      J = I MOD 100
SET      I = I / 100
.REP 62 COMN MACH I J
ENDIF

```

```

IF      CHAR = 'N'
.READCOM STBTM[BFR]
.READCOM STBPR[BFR]
.READCOM STBRV[BFR]
.READCOM STBRN[BFR]
SET      STBSB[BFR] = 0
SET      STBTS[BFR] = 0
SET      STBI F[BFR] = 0
SET      STBST[BFR] = 3
SET      CPART[COMN] = STBPR[BFR]
SET      CRAV[COMN] = STBRV[BFR]
SET      CRAVN[COMN] = STBRN[BFR]
SET      CSUBP[COMN] = 0
SET      RCLN[COMN] = 0
IF      ROBST[COMN] > 100
    SET      ROBST[COMN] = 0
ENDIF
QPOST   COMN TO CPART
WAIT    RCLN[COMN] > 0
.REP 39 COMN BFR STBTM[BFR] RCLN[COMN]
ENDIF

```

```

IF      CHAR = 'O' ; pallet loaded
SET      ROBST[COMN] = 20
SET      SP1[COMN] = BFR
SET      SP2[COMN] = PALET
.READCOM SP3[COMN] ; information on part
.REP2 37 COMN BFR SP3[COMN]
ENDIF

```

```

IF      CHAR = 'o' ; pallet loaded
IF      ROBST[COMN] = 20
    SET      ROBST[COMN] = 21
    .REPO    47 COMN
ELSE
    SET      ROBST[COMN] = 0
    SET      BFR = SP1[COMN]
    SET      PALET = SP2[COMN]
    SET      STPLT[PALET] = COMN
    SET      PSTOP[PALET] = 0
    SET      PINFQ[PALET] = SP3[COMN] ; information on part
    .REPO 57 COMN
    .REP1 4 COMN PALET
    SET      TMLPT = STBTM[BFR] ; template
    SET      CPART[COMN] = STBPR[BFR] ; part
    SET      CRAV[COMN] = STBRV[BFR] ; raw material
    SET      CRAVN[COMN] = STBRN[BFR] ; raw material number
    SET      CSUBP[COMN] = STBSB[BFR] ; next subprocess
    SET      STBST[BFR] = 0 ; tag buffer empty
    SET      STBTM[BFR] = 0
    SET      STBPR[BFR] = 0
    SET      STBRV[BFR] = 0
    SET      STBRN[BFR] = 0
    SET      STBSB[BFR] = 0 ; tag buffer empty
    SET      STBTS[BFR] = 0
    IF      CRAV[COMN] >= 0
        SET      RCLN[COMN] = 0
    ENDIF
ENDIF

```

```

        QPOST    COMN TO QPART          ; need to know next station number
        WAIT    RCLN[ COMN] > 0
    ELSE
        SET      RST[ COMN] = STMPL
    ENDIF
    SET      PTMPL[ PALET] = TMPLT      ; update information about the pallet
    SET      PPART[ PALET] = CPART[ COMN]
    SET      PRAW[ PALET] = CRAV[ COMN]
    SET      PRAWN[ PALET] = CRAVN[ COMN]
    SET      PSUBP[ PALET] = CSUBP[ COMN]
    SET      PNEXT[ PALET] = RST[ COMN]
ENDIF
ENDIF

IF      CHAR = 'P' ; pallet was in station
    SET      STPLT[ PALET] = COMN
    SET      PSTOP[ PALET] = 0
    .REP1 4 COMN PALET
ENDIF

IF      CHAR = 'Q'          ; pallet in station, what is on the pallet ?
    SET      STPLT[ PALET] = COMN
    SET      PSTOP[ PALET] = 1          ; pallet stopped in station
    .REP1 3 COMN PALET
    IF      PTMPL[ PALET] = 0
        .SENCOM 'E'          ; empty pallet
    ELSE
        IF      PNEXT[ PALET] = COMN    ; next subprocess is in current station
            .SENCOM 'Y'
            SET      CPART[ COMN] = PPART[ PALET]
            SET      CRAV[ COMN] = PRAW[ PALET]
            SET      CRAVN[ COMN] = PRAWN[ PALET]
            SET      CSUBP[ COMN] = PSUBP[ PALET]
            SET      RCLN[ COMN] = 0
            QPOST    COMN TO QPART          ; return machine number and task
            WAIT    RCLN[ COMN] > 0
            .PRLNCOM RMA[ COMN]          ; machine number
            .PRLNCOM RTASK[ COMN]       ; task
            .PRLNCOM PTMPL[ PALET]     ; template
            .PRLNCOM CPART[ COMN]       ; part
            .PRLNCOM CRAV[ COMN]       ; raw material
            .PRLNCOM CRAVN[ COMN]      ; raw material number
            .PRLNCOM CSUBP[ COMN]      ; subprocess
            .PRLNCOM PINFO[ PALET]     ; information about part
        ELSE
            .SENCOM 'N'          ; part for other station is on pallet
        ENDIF
    ENDIF
ENDIF
ENDIF

IF      CHAR = 'R'
    .READCOM ERR
    .READCOM I
    .READCOM J
    .REP 8 COMN ERR I J
ENDIF

IF      CHAR = 'S'          ; station performed reset
    SET      RESET[ COMN] = 0
    .REPO 6 COMN

```

```

ENDIF

IF      CHAR = 'T'           ; asking for template
  .READCOM NTMPL[ COMN]     ; part number template for
  .REP1 21 COMN NTMPL[ COMN]
  QPOST COMN TO QTMP
ENDIF

IF      CHAR = 'U'
  SET    ROBST[ COMN] = 50
  SET    SP1[ COMN] = BFR
  .REP1 33 COMN BFR
ENDIF

IF      CHAR = 'u'
  IF      ROBST[ COMN] = 50
    SET    ROBST[ COMN] = 51
    .REPO 43 COMN
  ELSE
    SET    ROBST[ COMN] = 0
    SET    BFR = SP1[ COMN]
    SET    STBST[ BFR] = 3
    .REPO 53 COMN
  ENDIF
ENDIF

IF      CHAR = 'V'           ; answer on release command: o.k.
  IF      ERAV[ COMN] < 0
    SET    ATMPL[ STMPL] = 1     ; send 'V' to answer "template ready"
    SET    NTMPL[ STMPL] = 0
    .REPO 22 STMPL
    SET    STMPL = 0
  ELSE
    .REPO 12 COMN
  ENDIF
  SET    RELES[ COMN] = 0       ; finish release, can release next
ENDIF

IF      CHAR = 'W'; station took part and released the pallet
  SET    ROBST[ COMN] = 10
  SET    SP1[ COMN] = BFR
  SET    SP2[ COMN] = PALET
  .REP1 35 COMN BFR
ENDIF

IF      CHAR = 'w'; station took part and released the pallet
  IF      ROBST[ COMN] = 11
    SET    ROBST[ COMN] = 0
    .REPO 55 COMN
  ELSE
    SET    ROBST[ COMN] = 11
    SET    BFR = SP1[ COMN]
    SET    PALET = SP2[ COMN]
    SET    PSTOP[ PALET] = 0 ; pallet leaves the station
    SET    STBTM[ BFR] = PTMPL[ PALET]
    SET    STBPR[ BFR] = PPART[ PALET]
    SET    STBRV[ BFR] = PRAV[ PALET]
    SET    STBRN[ BFR] = PRAW[ PALET]
    SET    STBSB[ BFR] = PSUBP[ PALET]
    SET    STBFI[ BFR] = PINFQ[ PALET]
  ENDIF

```

```

SET      STBST[ BFR] = 1          ; contains part

SET      PTMPL[ PALET] = 0
SET      PPART[ PALET] = 0
SET      PRAW[ PALET] = 0
SET      PRAWN[ PALET] = 0
SET      PSUBP[ PALET] = 0
SET      PNEXT[ PALET] = 0
.REPO 45 COMN
.REP1 4 COMN PALET
ENDIF
ENDIF

IF      CHAR = 'X'
SET      STMCH[ MACH] = 2
SET      ROBST[ COMN] = 80
.REP1 63 COMN MACH
ENDIF

IF      CHAR = 'x'
SET      STMCH[ MACH] = 1
SET      ROBST[ COMN] = 0
.REP1 64 COMN MACH
ENDIF

IF      CHAR = 'Y' ; station took part and released the pallet
SET      ROBST[ COMN] = 150
SET      BFR = COMN - 1
SET      BFR = BFR * 8
SET      BFR = BFR + 2
SET      SP1[ COMN] = BFR
SET      SP2[ COMN] = PALET
.REPO 75 COMN
ENDIF

IF      CHAR = 'y' ; station took part and released the pallet
IF      ROBST[ COMN] = 151
SET      ROBST[ COMN] = 0
.REPO 95 COMN
ELSE
SET      ROBST[ COMN] = 151
SET      BFR = SP1[ COMN]
SET      PALET = SP2[ COMN]

SET      QPART[ COMN] = PPART[ PALET] ; part
SET      CRAV[ COMN] = PRAW[ PALET] ; raw material
SET      CRAWN[ COMN] = PRAWN[ PALET] ; raw material number
SET      CSUBP[ COMN] = PSUBP[ PALET] ; subprocess number finished
IF      QPART[ COMN] = 0
.REP1 5 COMN PTMPL[ PALET]
ENDIF
IF      CRAV[ COMN] >= 0
SET      RCLN[ COMN] = 0
QPOST COMN TO QPART          ; getting part information
WAIT RCLN[ COMN] > 0
SET      I = RCLN[ COMN]      ; line number in order
IF      CSUBP[ COMN] = 1      ; if first subprocess finished,
ANDIF CRLS[ I] >= 0          ; can release one more
SET      CRLS[ I] = CRLS[ I] + 1
ENDIF
IF      RTASK[ COMN] >= 900  ; unloading
IF      I <= CLEN          ; if valid order line number

```

```

        SET      CFIN[I] = CFIN[I] + 1 ; one more part is ready
        IF      OVRK[I] > 0           ; one part finished work
            SET      OVRK[I] = OVRK[I] - 1
        ENDIF
    ENDIF
    IF      CPART[COM] <= 0
        .REP2 29 COM 0 1
    ENDIF
    ENDIF ; IF RTASK[COM] >= 900
    ENDIF ; IF RAW[COM] >= 0

    SET      PSTOP[PALET] = 0 ; pallet leaves the station
    SET      PTMPL[PALET] = 0
    SET      PPART[PALET] = 0
    SET      PRAV[PALET] = 0
    SET      PRAWN[PALET] = 0
    SET      PSUBP[PALET] = 0
    SET      PNEXT[PALET] = 0
    .REPO 85 COM
    .REP1 4 COM PALET
    ENDIF
ENDIF

    IF      CHAR = 'Z'
        SET      ROBST[COM] = 60
        SET      SP1[COM] = BFR
        .REP1 34 COM BFR
    ENDIF

    IF      CHAR = 'z'
        IF      ROBST[COM] = 60
            SET      ROBST[COM] = 61
            .REPO 44 COM
        ELSE
            SET      ROBST[COM] = 0
            SET      BFR = SP1[COM]
            .REPO 54 COM
        ENDIF
    ENDIF
ENDIF

    SET      GCOM[COM] = 0           ; can receive next message
    ENDIF

    IF      RSTAT[COM] = 1           ; communication checks
        .SENCOM 'A'
        SET      RSTAT[COM] = 2
    ENDIF

    IF      RELES[COM] = 1           ; need to release
        .SENCOM 'R'                 ; release
        .PRLNCOM EPART[COM]         ; part
        .PRLNCOM ERAV[COM]         ; raw material
        .PRLNCOM ERAVN[COM]        ; raw material number (for assembly)
        SET      RELES[COM] = 2     ; the release command was sent,
        .REP1 11 COM ELI NE[COM]    ;
    ENDIF ; can't release more now in this station

    IF      RESET[COM] = 1           ; reset in center -> reset in all stations

```



```

.SENCOM 'S' ;reset command
SET RESET[COM] = 2 ;reset command was sent
ENDIF

IF ATMPL[COM] > 0
IF ATMPL[COM] = 1
.SENCOM 'V'
ELSE
.SENCOM 'K'
ENDIF
SET ATMPL[COM] = 0
ENDIF

IF EASW[COM] = 1 ;need to start assrble
.SENCOM 'W' ;start assemble
.PRLNCOM APART[COM] ;part
.REP1 24 COM ALI NE[COM]
SET EASW[COM] = 2 ;the command finished
ENDIF

ENDFOR
DELAY 10
GOTO 1

END
# <<<< COM - 8 >>>>

#information to WZCOM
PROGRAM COM8
DEFINE MODE ;question type
DEFINE CNT ;index of order lines for report
DEFINE IS ;index of stations
DEFINE TMPS ;station * 64 + location
DEFINE STN ;station number to change
DEFINE STP ;new location of station[stn]
DEFINE CR ;release flag + 5 (ensure > 0)

SET IS = 0 ;starting station number
SET CNT = 0 ;starting order lines
SET MODE = 0 ; != 0 after question
LABEL 1

GETCOM 8, COM[8]

IF COM[8] = 13
IF MODE = 0
SENCOM 8,13 ; carriage return
SENCOM 8,10 ; line feed
SENCOM 8,'>' ; prompt
ENDIF
IF MODE = 5
PROCOM 8,D1 ; locations 1 - 4
PROCOM 8,D2 ; locations 5 - 8
PROCOM 8,D3 ; locations 9 - 12
PROCOM 8,D4 ; locations 13 - 16
PROCOM 8,D5 ; locations 17 - 20
PRLNCOM 8
ENDIF
IF MODE = 7
PROCOM 8,D6 ; locations 21 - 24
PROCOM 8,D7 ; locations 25 - 28

```

```

PROCOM 8, D8 ; locations 29 - 32
PROCOM 8, TMPS ; station number and location
PRLNCOM 8
ENDIF
IF MODE = 6
IF CNT = 0 ; flag 'start of report'
PROCOM 8, "0 "
PROCOM 8, "0 "
PROCOM 8, "0 "
PROCOM 8, "0 "
PROCOM 8, "0 "
ELSE
PROCOM 8, CPART[ CNT] ; part
PROCOM 8, CQTY[ CNT] ; quantity
PROCOM 8, CMCRK[ CNT] ; in process
PROCOM 8, CFIN[ CNT] ; finished
SET CR = CRLS[ CNT] + 5 ; add 5 to ensure > 0
PROCOM 8, CR ; release flag
ENDIF
; send carriage return
ENDIF
SET MODE = 0
ENDIF

IF GCOM[ 8] = '5' ; first 20 locations
SET MODE = 5
ENDIF

IF GCOM[ 8] = '7' ; locations 21 - 32, one station location
SET MODE = 7
PEND TMPS FROM STPND
SET IS = STNXT ; current station number
SET STNXT = STNXT + 1 ; next
IF STNXT > 8 ; cyclic
SET STNXT = 1
ENDIF
POST 1 TO STPND
SET TMPS = S[ IS] * 64 ; station location
SET TMPS = TMPS + IS ; station number
ENDIF

IF GCOM[ 8] = '6' ; for report
SET MODE = 6
IF CNT >= CLEN ; cyclic
SET CNT = 0
ELSE
SET CNT = CNT + 1 ; next order line
ENDIF
ENDIF

IF GCOM[ 8] = 'A' ; request / change station location
GETCOM 8, STN
SET STN = STN - '0'
LABEL 2
GETCOM 8, GCOM[ 8]
IF GCOM[ 8] = '='
READCOM 8, STP ; new location
SET S[ STN] = STP ; update
SENCOM 8, 13 ; carriage return
SENCOM 8, 10 ; line feed
SENCOM 8, '>' ; prompt
GOTO 1
ENDIF
IF GCOM[ 8] = 13

```

```

        PROCM 8, S[STN]      ; request
        SENCOM 8, 13        ; carriage return
        SENCOM 8, 10        ; line feed
        SENCOM 8, '>'      ; prompt
        GOTO 1
    ENDF
    IF      GCOM[8] = ' '
        GOTO 2              ; continue
    ENDF
ENDF
GOTO 1
END

#          <<<<<  STATIONS  >>>>

#include REPORT.DMC

#stations active / nonactive
PROGRAM ST
DEFINE I          ; for loop on all stations
DEFINE J
DEFINE K
DEFINE C

LABEL 1

WAIT SUREP = 0    ; suspend when update setup

;FOR I = 1 TO 6
; IF S[I] <> SREF[I]
;   ANDIF SREF[I] > 0
;   SET SREF[I] = -1
;   ENDF
; ENDFOR

FOR I = 1 TO 6    ; all stations

    IF S[I] <> SREF[I] ; station definition was changed
        SET J = S[I] ; check new location
        IF J > 0
            ANDIF CST[J] > 0
                SET S[I] = SREF[I] ; can't define two stations in the same location
            ELSE
                PEND K FROM STPND ; don't change stations when reporting on it
                SET STNXT = I ; next report is the changed station
                POST 1 TO STPND

                .REP1 1 I S[I]

                IF SREF[I] = 0 ; new station defined
                    SET J = I + 4
                    SET OUT[J] = 1 ; station and turn on output for the station
                ENDF

                IF S[I] = 0 ; exits station undefined
                    SET J = I + 4
                    SET OUT[J] = 0 ; turn off the output for the station
                ENDF

                IF SREF[I] > 0 ; station change location
                    SET J = SREF[I]
                    SET CST[J] = 0 ; tag old location empty
                    IF CPLT[J] > 0 ; pallet in old location
                        SET K = CPLT[J] ; get pallet number
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDFOR

```

```

        SET      PSTCP[K] = 0      ; free pallet moving in old station position
      ENDF
    ENDF
  SET      J = S[I]      ; new location
  IF      J > 0
    SET      CST[J] = 1      ; tag location containing station
  ENDF
  SET      SREF[I] = J      ; set reference for next check
ENDF
ENDF
ENDFOR

DELAY 50
GOTO 1
END

```

```
# <<<<<< MAIN >>>>>>
```

```
#include REPORT.DMC
```

```
#Program that runs all necessary CIM CELL system programs
# at central controller
```

```
PROGRAM AUTO
DEFINE I
```

```
IF      OTBI P[1] = 0
  FOR    I = 1 TO 100      ; init priority table
    SET   OTBI P[I] = 1
  ENDFOR
ENDF

```

```
RUN     ENREP      ; cancel report
```

```
RUN     PLTS      ; track pallets
RUN     PTC       ; give part information to com programs
RUN     COM

```

```
IF      SREF[1] > 0      ; start at defined station
  SET    CUT[5] = 1      ; signal to station "you are working"
ENDF

```

```
IF      SREF[2] > 0
  SET    CUT[6] = 1
ENDF

```

```
IF      SREF[3] > 0
  SET    CUT[7] = 1
ENDF

```

```
IF      SREF[4] > 0
  SET    CUT[8] = 1
ENDF

```

```
IF      SREF[5] > 0
  SET    CUT[9] = 1
ENDF

```

```
IF      SREF[6] > 0
  SET    CUT[10] = 1
ENDF

```

```
FOR    I = 1 TO 6
  SET    RSTAT[I] = 0      ; clear status checking
ENDFOR

```

```
POST   1 TO STPND
RUN     COM8      ; information to wizcon
RUN     ST        ; track def. and undef. stations
SET     INWRK = 0  ; don't work when startup

```

```

SET      SUREP = 0
IF      MAPIX = 1
  RUN    MFCT
ENDIF
END

```

```

#program that suspends the manufacturing process
PROGRAM STOPM
DEFINE C
PRI NTLN "st m"
PRI NTLN
STOP    MFCT
STOP    RLS1
.REPO 18 0
DELAY  20
END

```

```

#program that resets all CIM CELL variables
PROGRAM RESET
DEFINE I
DEFINE C

```

```

STOP    MFCT
STOP    RLS1
DELAY  20

```

```

SET      INWRK = 0           ; stop working

```

```

POST     1 TO STPND
POST     1 TO CREP         ; reset report flag

```

```

.REPO 7 0

```

```

#clear pallets

```

```

FOR      I = 1 TO 15
  SET     PTMPL[I] = 0      ; template number
  SET     PPART[I] = 0      ; part number
  SET     PRAW[I] = 0       ; raw material number
  SET     PSUBP[I] = 0      ; subprocess number
  SET     PNEXT[I] = 0      ; next station number
  SET     PSTOP[I] = 0      ; pallet status = MOVING
  SET     PINFC[I] = 0      ; info. value
ENDFOR

```

```

;IF      QLEN > 0          ; QLD only restart order

```

```

; FOR    I = 1 TO QLEN
;   SET   OWRK[I] = 0
;   SET   CFIN[I] = 0
;   SET   CRLS[I] = 2
; ENDFOR
; ENDF

```

```

#clear order

```

```

FOR      I = 1 TO 100
  SET     CPART[I] = 0
  SET     CRAW[I] = 0
  SET     CRAWN[I] = 0
  SET     CWIN[I] = 0
  SET     CSUBP[I] = 0
  SET     CQTY[I] = 0
  SET     CPTS[I] = 0
  SET     CAQ[I] = 0

```

```

SET      O/ASM[I] = 0
SET      CRDRL[I] = 0
SET      O/WRK[I] = 0
SET      CFIN[I] = 0
SET      CRLS[I] = 0
SET      OTBIP[I] = 1      ;init priority table
ENDFOR
SET      QLEN = 0          ;empty order

FOR      I = 1 TO 6
  SET      RSTAT[I] = 0      ;don't check station status
  SET      RELES[I] = 0      ;don't send release command
  SET      EASMW[I] = 0      ;not in assembly
  SET      ROBST[I] = 0      ;robot in station

  IF      S[I] > 0
    SET      RESET[I] = 1      ;send reset command to station
  ELSE
    SET      RESET[I] = 0      ;in undefined station don't send reset command
  ENDIF
ENDFOR

FOR      I = 1 TO 24          ;machines
  SET      STMCH[I] = 0
ENDFOR

FOR      I = 1 TO 48
  SET      STBTM[I] = 0
  SET      STBPR[I] = 0
  SET      STBRW[I] = 0
  SET      STBRN[I] = 0
  SET      STBSB[I] = 0
  SET      STBIF[I] = 0
  SET      STBTS[I] = 0
  SET      STBST[I] = 0
ENDFOR

SET      STMPL = 0
SET      CREP = 1          ;report flag
SET      SUREP = 0
IF      MAPIX = 1
  RUM      MFCT
ELSE
  RUN      ENREP          ;cancel report
ENDIF

END

#      <<<<<   CRDIN   >>>>>

#getting order
PROGRAM CRDIN
DEFINE  I J              ;loop vars
DEFINE  I DXWR          ;pointer to work table
DEFINE  PART            ;part number
DEFINE  QTYA            ;quantity of parts to do
DEFINE  QTYB            ;quantity of raw materials for one part
DEFINE  QRAW            ;number of raw materials for one part (for assembly)
DEFINE  LEN             ;number of subprocess
DEFINE  RAW             ;raw material number
DEFINE  VST            ;station number
DEFINE  VMA            ;machine number
DEFINE  VTASK          ;task

```

```

DEFIN E  CLEN1          ; pointer to order table

IF      MAPIX = 1
  SET    SUREP = 1
ELSE
  RUN    ENREP          ; cancel report
ENDIF

GET     I              ; used for sync with external program

IF      INWRK = 2      ; last order finished
  SET    CLEN = 0      ; clear order
  SET    INWRK = 0     ; reset order status
  SET    IDXWR = 1     ; init work line number
  SET    REPL = 0      ; reset command
  FOR    I = 1 TO 100  ; init priority table
    SET  OTBLP[I] = 1
  ENDFOR
ENDIF

IF      CLEN > 0      ; order exists
  ANDIF  REPL = 0      ; first try
  PRI NTLN  "psy"      ; "order already exists"
  PRI NTLN
  GOTO    9
ELSE
  PRI NTLN  "orx"      ; "can get order"
  PRI NTLN
ENDIF

IF      CLEN > 0      ; order exists but
  ANDIF  INWRK = 0     ; has not started yet
  ANDIF  REPL = 1      ; and a command to replace it is received
  SET    CLEN = 0      ; then clear order before getting
  SET    IDXWR = 1     ; init work line number
  SET    REPL = 0      ; reset the command
  FOR    I = 1 TO 100  ; init priority table
    SET  OTBLP[I] = 1
  ENDFOR
ENDIF

IF      REPL = 1      ; 1: replace, 2: add
  STOP   MNFCT         ; in replace don't release any more
  FOR    I = 1 TO CLEN ; tag all lines in old order as finished release
    IF    CRLS[I] >= 0
      SET  CRLS[I] = -1
    ENDIF
    IF    CRLS[I] >= 0 ; change value twice because
      SET  CRLS[I] = -1 ; other program can change it
    ENDIF
  ENDFOR
ENDIF

SET     REPL = 0      ; reset command for next time

IF      CLEN > 0      ; if order isn't empty
  SET    IDXWR = OVLN[CLEN] + OSUBP[CLEN] ; first unused work line number
ELSE
  SET    IDXWR = 1     ; if order empty get first work line number
ENDIF

SET     CLEN1 = CLEN + 1 ; first unused order line number

```

```

LABEL      1

READ      PART          ;part number (<0 in end order)
IF        PART < 0 ;end of order
  GOTO    8
ENDIF

IF        CLEN1 > 100   ;no place for more lines
  LABEL   2
  READ    PART
  IF      PART >= 0 ;waiting for end of order
    GOTO  2
  ENDIF

  PRIN TLN "gea" ;indicate no place for order
  PRIN TLN ">"
  GOTO    9
ENDIF

READ      QTYA          ;quantity of part to do
READ      QRAW          ;number of raw materials (0 if not assembly)
FOR       J = 0 TO QRAW
  READ    RAW           ;raw material
  IF      QRAW > 0      ;if assembly
    READ  QTYB          ;quantity of this raw material
  ELSE
    SET    QTYB = 1     ;always 1 if not assembly
  ENDIF
  READ    LEN           ;number of subprocess

  SET     CPART[CLEN1] = PART ;put in order table
  SET     CRAW[CLEN1] = RAW
  SET     CRAWN[CLEN1] = J ;raw material number (for assembly)
  SET     CWIN[CLEN1] = IDXWR ;first subprocess work table line number
  SET     CSUBP[CLEN1] = LEN;number of subprocess
  SET     CQTY[CLEN1] = QTYA * QTYB ;quantity of this raw material
  IF      CRAW > 0
    ANDIF J = 0
      SET  CPTS[CLEN1] = QRAW ;first line in assembly is the part itself
      SET  CRDL[CLEN1] = 0
    ELSE
      SET  CPTS[CLEN1] = 0
      SET  CRDL[CLEN1] = CQTY[CLEN1]
    ENDIF
  ENDIF
  SET     OASM[CLEN1] = 0
  SET     OAQ[CLEN1] = QTYB ;quantity of this raw material
  SET     OWRK[CLEN1] = 0 ;in process
  SET     OFIN[CLEN1] = 0 ;ready
  SET     ORLS[CLEN1] = 2 ;can release

  FOR     I = 1 TO LEN ;loop on the entire process
    READ  VST ;station
    READ  VMA ;machine
    READ  VTASK ;task
    SET   VMA = VMA * 16 ;compress in one var
    SET   VTASK = VTASK * 256
    SET   WORKT[IDXWR] = VST + VMA ;put in work table
    SET   WORKT[IDXWR] = WORKT[IDXWR] + VTASK
    SET   IDXWR = IDXWR + 1 ;next line in work table
  ENDFOR
  SET    CLEN1 = CLEN1 + 1 ;next line in order table
ENDFOR
GOTO    1

```



```

LABEL      8
SET        CLEN = CLEN1 - 1      ;update order length
PRI NTLN   "fdz"                  ;getting all order
PRI NTLN   ">"
LABEL      9
SET        SUREP = 0
END

#      <<<<<  STSTAT  >>>>>

#stations report
PROGRAM    RSTAT
DEFINE    I          ;for loop on all stations
DEFINE    TM          ;used for timeout
DEFINE    FLAG        ;flag for stop waiting after all stations have answered

FOR      I = 1 TO 6
  IF      SREF[I] > 0      ;only on defined station
    SET    RSTAT[I] = 1    ;send check command
  ELSE
    SET    RSTAT[I] = 0
  ENDIF
ENDFOR

SET      TM = TIME + 200    ;start timer for timeout

LABEL    1
SET      FLAG = 0
FOR      I = 1 TO 6
  IF      RSTAT[I] > 0    ;still no answer
    SET    FLAG = 1      ;set flag
  ENDIF
ENDFOR
IF      FLAG = 1          ;if any stations did not answer
  ANDIF   TM > TIME      ;wait until timeout
  GOTO    1
ENDIF

PRI NTLN  "      +-----+-----+"
PRI NT    "-----+"
PRI NTLN  "      ! STATION # ! LOCATION !"
PRI NT    " REMARKS          !"
PRI NTLN  "      +-----+-----+"
PRI NT    "-----+"

FOR      I = 1 TO 6
  IF      SREF[I] > 0      ;only on defined station
    PRI NTLN "      ! " I "      ! " SREF[I]
    PRI NT  "      ! "
    IF      IN[I] = 1      ;station input on
      IF RSTAT[I] = 0      ; station answered
        PRI NT "COMMUNICATION O.K."
      ELSE
        PRI NT "NOT RESPONDING " ; station did not answer
      ENDIF
    ELSE
      ;station input off
      IF RSTAT[I] = 0      ; station answered
        PRI NT "I / O PROBLEM "
      ELSE
        PRI NT "NOT ACTIVE " ; station did not answer
      ENDIF
    ENDIF
  ENDIF
  PRI NT "!"

```

```

    ENDIF
ENDFOR

PRI NTLN "      +-----+-----+"
PRI NT  "-----+"
PRI NTLN ">"
END

#st at i on changes
PROGRAM CSTAT
DEFIN E C ;command
DEFIN E ST ;st at i on number
DEFIN E P ;loc at i on of st at i on

LABEL 1
DELAY 70

PRI NTLN "      +-----+-----+"
PRI NTLN "      ! A -  ADD NEW STATI ON      !"
PRI NTLN "      ! C -  CHANGE STATI ON LOCATI ON !"
PRI NTLN "      ! D -  DELETE STATI ON      !"
PRI NTLN "      ! S -  PRI NT STATI ON STATUS  !"
PRI NTLN "      ! X -  EXI T TO ACL          !"
PRI NTLN "      +-----+-----+"
PRI NTLN " COMMAND ( A/ C/ D/ S/ X): "
LABEL 2
GET C

IF C = 'A' ;add st at i on
  ORIF C = 'a'
  PRI NT "A"
  PRI NTLN "ADD STATI ON NUMBER : "
  READ ST

  IF ST > 6
    ORIF ST < 1
    PRI NTLN
    PRI NTLN "ONLY STATI ONS 1 TO 6 ARE ALLOWED"
    PRI NTLN
    GOTO 1
  ENDIF

  IF S[ST] > 0
    PRI NTLN
    PRI NTLN "STATI ON " ST " ALREADY EXI STS, "
    PRI NT " USE 'C' COMMAND TO CHANGE STATI ON LOCATI ON"
    PRI NTLN
    GOTO 1
  ENDIF

  PRI NT " I N LOCATI ON : "
  READ P

  IF P > 32
    ORIF P < 1
    PRI NTLN
    PRI NTLN "ONLY LOCATI ONS 1 TO 32 ARE ALLOWED"
    PRI NTLN
    GOTO 1
  ENDIF

  SET S[ST] = P

```

```

    PRINTLN " Done. "
    GOTO 1
ENDIF

IF C = 'D' ; delete station
    ORIF C = 'd'
    PRINT "D"
    PRINTLN "DELETE STATION NUMBER : "
    READ ST

    IF ST > 6
        ORIF ST < 1
        PRINTLN
        PRINTLN "ONLY STATIONS 1 TO 6 ARE ALLOWED"
        PRINTLN
        GOTO 1
    ENDIF

    IF S[ST] = 0
        PRINTLN
        PRINTLN "STATION " ST " DOES NOT EXIST"
        PRINTLN
        GOTO 1
    ENDIF

    SET S[ST] = 0
    PRINTLN " Done. "
    GOTO 1
ENDIF

IF C = 'C' ; change station location
    ORIF C = 'c'
    PRINT "C"
    PRINTLN "CHANGE STATION NUMBER : "
    READ ST

    IF ST > 6
        ORIF ST < 1
        PRINTLN
        PRINTLN "ONLY STATIONS 1 TO 6 ARE ALLOWED"
        PRINTLN
        GOTO 1
    ENDIF

    IF S[ST] = 0
        PRINTLN
        PRINTLN "STATION " ST " DOES NOT EXIST, "
        PRINT " USE 'A' COMMAND TO ADD A NEW STATION"
        PRINTLN
        GOTO 1
    ENDIF

    PRINT " NEW LOCATION : "
    READ P

    IF P > 32
        ORIF P < 1
        PRINTLN
        PRINTLN "ONLY LOCATIONS 1 TO 32 ARE ALLOWED"
        PRINTLN
        GOTO 1
    ENDIF

    SET S[ST] = P

```

```

      PRI NTLN  " Done. "
      GOTO     1
    ENDF

    IF      C = 'S'          ;st at us
      ORIF   C = 's'
      PRINT  "S"
      GOSUB  RSTAT
      GOTO   1
    ENDF

    IF      C = 'X'          ;exi t
      ORIF   C = 'x'
      PRINT  "X"
      PRI NTLN
      PRI NTLN ">"
      GOTO   9
    ENDF

    GOTO    2

    LABEL   9
    END

#      <<<<<  PLTSTAT  >>>>>

#pallet report
PROGRAM  RPLT
DEFIN E I      ;pallet number
DEFIN E P      ;locati on

DELAY  30

PRI NTLN
PRI NTLN " +-----+-----+-----+-----+---"
PRI NT  "-----+-----+-----+-----+-----+"
PRI NTLN " ! PALLET! PLACE ! TEMPL. ! PART ! "
PRI NT  " RAW ! SUBPR ! NEXT ST! I N ST. ! STATUS ! "
PRI NTLN " +-----+-----+-----+-----+---"
PRI NT  "-----+-----+-----+-----+-----+"

FOR I = 1 TO 12
  SET    P = PPLAQ[I]
  PRI NTLN " !" I      " !" PPLAQ[I]
  PRI NT  "          !" PTMPL[I] "          !" PPART[I]
  PRI NT  "          !" PRAW[I] "          !" PSUBP[I]
  PRI NT  "          !" PNEXT[I] "          !"
  IF     CST[P] > 0 ;station in locati on
    PRI NT  CST[P]
  ELSE
    PRI NT  " "
  ENDF
  PRI NT  "          !"
  IF     PSTOP[I] = 0
    PRI NT  " MOVE"
  ELSE
    PRI NT  " STOP"
  ENDF
  PRI NT  "          !"
ENDFOR

PRI NTLN " +-----+-----+-----+-----+---"
PRI NT  "-----+-----+-----+-----+-----+"

```

```

PRI NTLN ">"

END

#pallet changes
PROGRAM CPLT
DEFIN E C ;command
DEFIN E P ;pallet number
DEFIN E VAL ;new value of an item

LABEL 1
DELAY 40

PRI NTLN " +-----+-----"
PRI NT "-----+"
PRI NTLN " ! S - STATUS ! C - CHANGE PALLET"
PRI NT " CONTENTS ! X - EXIT TO ACL !"
PRI NTLN " +-----+-----"
PRI NT "-----+"

PRI NTLN " COMMAND (S/C/X) : "
LABEL 11
GET C

IF C = 'X' ;exit
  CRIF C = 'x'
  PRI NT "X"
  GOTO 9
ENDIF

IF C = 'S' ;status
  CRIF C = 's'
  PRI NT "S"
  GOSUB RPLT
  GOTO 1
ENDIF

IF C = 'C' ;change
  CRIF C = 'c'
  PRI NT "C"

PRI NTLN "ENTER PALLET NUMBER:"
READ P

IF P > 12
  CRIF P < 1
  PRI NTLN
  PRI NTLN " ONLY PALLET NUMBERS 1 - 12 ARE ALLOWED"
  PRI NTLN
  GOTO 1
ENDIF

# ----- TEMPLATE -----
LABEL 3
PRI NTLN "TEMPLATE ON PALLET IS " PTMPL[P]
PRI NT " CHANGE (Y/N)? (N)"
GET C
PRI NTLN
IF C = 'Y'
  CRIF C = 'y'

```

```

    PRI NTLN "NEW TEMPLATE NUMBER : "
    READ VAL
    SET     PTMPL[ P] = VAL
ENDIF

# ----- PART -----
LABEL 4
PRI NTLN "PART ON PALLET I S " PPART[ P]
PRI NT  " CHANGE (Y/N)? (N)"
GET     C
PRI NTLN
IF      C = 'Y'
    CRIF C = 'y'
    PRI NTLN "NEW PART NUMBER "
    READ VAL
    SET     PPART[ P] = VAL
ENDIF

# ----- RAW-----
LABEL 5
PRI NTLN "RAW MATERIAL NUMBER I S " PRAW[ P]
PRI NT  " CHANGE (Y/N)? (N)"
GET     C
PRI NTLN
IF      C = 'Y'
    CRIF C = 'y'
    PRI NTLN "NEW RAW MATERIAL NUMBER "
    READ VAL
    SET     PRAW[ P] = VAL
ENDIF

# ----- SUBPROCESS -----
LABEL 6
PRI NTLN "NEXT SUBPROCESS FOR PART I S " PSUBP[ P]
PRI NT  " CHANGE (Y/N)? (N)"
GET     C
PRI NTLN
IF      C = 'Y'
    CRIF C = 'y'
    PRI NTLN "NEW SUBPROCESS NUMBER "
    READ VAL
    SET     PSUBP[ P] = VAL
ENDIF

# ----- NEXT STATI ON -----
LABEL 7
PRI NTLN "NEXT STATI ON I S " PNEXT[ P]
PRI NT  " CHANGE (Y/N)? (N)"
GET     C
PRI NTLN
IF      C = 'Y'
    CRIF C = 'y'
    PRI NTLN "NEW STATI ON NUMBER "
    READ VAL
    IF VAL > 8
        CRIF VAL < 1
        PRI NTLN "ONLY STATI ON NUMBERS 1 - 8 ARE ALLOWED"
        GOTO 7
    ENDIF
SET     PNEXT[ P] = VAL
ENDIF

# ----- STATUS -----
LABEL 8

```

```

;PRI NTLN "PALLET STATUS IS:"
;IF PSTOP[P] = 0
; PRINT "MOVING"
;ELSE
; PRINT "STOPPED"
;ENDIF
;PRINT ", CHANGE STATUS TO "
;IF PSTOP[P] > 0
; PRINT "MOVING"
;ELSE
; PRINT "STOPPED"
;ENDIF
;PRINT " (Y/N)? (N)"
;GET C
;PRI NTLN
;IF C = 'Y'
; ORIF C = 'y'
; SET PSTOP[P] = 1 - PSTOP[P]
;ENDIF

PRI NTLN
GOTO 1
ENDIF

GOTO 11

LABEL 9
PRI NTLN
PRI NTLN ">"
END

# <<<<< CRDSTAT >>>>>

#order report
PROGRAM RORD
DEFINE I ;line in "order table"
DEFINE CK;counter to stop at end of each page
DEFINE FP;flag to print title on the beginning of page

DELAY 30

IF CLEN = 0
PRI NTLN "ORDER IS EMPTY!"
PRI NTLN
GOTO 9
ENDIF

SET FP = 1 ;print title first

FOR I = 1 TO CLEN
IF FP = 1
PRI NTLN " +-----+-----+-----+-----+"
PRINT " +-----+-----+-----+-----+"
PRI NTLN " ! LINE ! PART ! RAW ! QTY ! "
PRINT " PARTS ! IN PROC ! READY ! REMARKS ! "
PRI NTLN " +-----+-----+-----+-----+"
PRINT " +-----+-----+-----+-----+"
SET FP = 0
ENDIF

PRI NTLN " !" I " " !" CPART[I]
PRINT " !" CRAW[I] " " !" CQTY[I]
PRINT " !" CPTS[I] " " !" CWORK[I]

```

```

PRINT "!" OF N[I] " !"
IF CRLS[I] > 0
  PRINT "READY (" CRLS[I] ") "
ENDIF
IF CRLS[I] = 0
  PRINT "WORKING "
ENDIF
IF CRLS[I] = -1
  PRINT "RELEASED "
ENDIF
IF CRLS[I] = -2
  PRINT "OUT OF MATERIAL"
ENDIF
IF CRLS[I] = -3
  PRINT "DONE "
ENDIF
IF CRLS[I] = -4
  PRINT "DONE (MISSING) "
ENDIF
PRINT "!"
SET CK = I / 16
SET CK = CK * 16
IF CK = I
  PRINTLN "PRESS ANY KEY TO CONTINUE ...."
  GET CK
  SET FP = 1
ENDIF
ENDFOR

PRINTLN " +-----+-----+-----+-----+"
PRINT "-----+-----+-----+-----+"

LABEL 9
PRINTLN ">"
END

```

```

#order subprocess report
PROGRAM RPROC
DEFINE I ;line in "order table"
DEFINE FROM ;first line in work table
DEFINE TO ;last line in work table
DEFINE J ;line in work table
DEFINE WORK ;value from work table
DEFINE ST ;station
DEFINE MACH ;machine
DEFINE TASK ;task
DEFINE F ;flag =1 if remark on subprocess was printed

DELAY 30

IF QLEN = 0
  PRINTLN "ORDER IS EMPTY !"
  PRINTLN
  GOTO 9
ENDIF

FOR I = 1 TO QLEN

  PRINTLN " +-----+-----+-----+-----+"
  PRINTLN " ! LINE ! PART ! RAW ! QTY !"
  PRINTLN " +-----+-----+-----+-----+"

  PRINTLN " !" I " !" CPART[I]

```



```

PRINT " !" CRAMP[1] "      !" CQTY[1]
PRINT " !"

PRINTLN " +-----+-----+-----+-----+"

SET     FROM = OVLN[1]
SET     TO = FROM + OSUBP[1]
SET     TO = TO - 1

PRINTLN "+-----+-----+-----+"
PRINT  "-----+"
PRINTLN "! STATION ! MACHINE ! TASK  !"
PRINT  "  REMARKS      !"
PRINTLN "+-----+-----+-----+"
PRINT  "-----+"

FOR     J = FROM TO TO
  SET   WORK = WORKT[J]
  SET   ST  = WORK MOD 16
  SET   WORK = WORK / 16
  SET   MACH = WORK MOD 16
  SET   TASK = WORK / 16

  PRINTLN "          !" ST "          !" MACH
  PRINT  "          !" TASK "          !"
  SET   F = 0
  IF    J = FROM
    PRINT "  RELEASE          "
    SET   F = 1
  ENDIF
  IF    TASK = 999
    PRINT "  UNLOADING          "
    SET   F = 1
  ENDIF
  IF    TASK = 998
    PRINT "  UNLOAD FOR ASSEMBLY  "
    SET   F = 1
  ENDIF
  IF    F = 0
    PRINT "          "
  ENDIF
  PRINT "!"
ENDFOR
PRINTLN "+-----+-----+-----+"
PRINT  "-----+"

IF     I < CLEN
  PRINTLN " PRESS ANY KEY TO CONTINUE !"
  GET   J
ENDIF
ENDFOR

LABEL  9
PRINTLN ">"
END

```

Station Controller System Programs

```
#          <<<<  VARS  >>>>

#for com

#variable for getting char from rs232 ports
DIMG  GCOM[8]

#used when receiving R command
GLOBAL  RPART      ;part number
GLOBAL  RRAW       ;raw material number
GLOBAL  RRAWN      ;raw material number (assembly)
GLOBAL  RTMPL      ;template number

#used when receiving Y reply
GLOBAL  YMA        ;machine number
GLOBAL  YTASK      ;task number
GLOBAL  YTMPL      ;template number
GLOBAL  YPART      ;part number
GLOBAL  YRAW       ;raw material number
GLOBAL  YRAWN      ;raw material number (assembly)
GLOBAL  YSUBP      ;required subprocess
GLOBAL  YINFO      ;information on part (after q.c.)

#used when receiving C reply
GLOBAL  CMA        ;machine number
GLOBAL  CTASK      ;task number
GLOBAL  CBF       ;buffer number

#used when received M or F replies
GLOBAL  MBFR       ;buffer number

#used for sending V or E replies
GLOBAL  FREL       ;releasing status
                ; 0: don't need to release
                ; 1: after release command is received
                ;      and material is found
                ; 2: send 'E' (missing)
                ; 3: send 'V' (o.k.)
GLOBAL  TMPLT      ;template used
GLOBAL  PART       ;part number
GLOBAL  RAW        ;raw material number
GLOBAL  RAWN       ;raw material number (assembly)
GLOBAL  TASK
GLOBAL  SUBP
GLOBAL  INFO
GLOBAL  FV         ;=0 when not found,/ buffer no. (1 for ASRS)
SET     FREL = 0
SET     TMPLT = 1
SET     PART = 0
SET     RAW = 0
SET     RAWN = 0
SET     FV = 0

GLOBAL  TASK1      ;task number for each machine
GLOBAL  TASK2
GLOBAL  TASK3
GLOBAL  TASK4
```

```

#used for sending F replies
DIMG   CFIN[8]           ;queue for finished tasks (according to buffer numbers)
GLOBAL FFIN             ;=1 when any task has finished
GLOBAL FBFR             ;buffer number where task finished
SET    FFIN = 0

```

```

#used for sending P or Q
GLOBAL PALET           ;current pallet number
GLOBAL FPLT           ;=1 if info required from center
SET    PALET = 0
SET    FPLT = 0

```

```

#used for sending O
GLOBAL FONP           ;need to send 'O'
GLOBAL OBFR           ;buffer number
GLOBAL OSUBP          ;subprocess number
GLOBAL OINFO          ;information on part (after q.c.)
SET    FONP = 0

```

```

#for send 'a' 't' 'f' 'n' (in ASRS moving)
GLOBAL ATFLG
SET    ATFLG = 0

```

```

#for sending W
GLOBAL FWRK           ;=1: need to send 'W'
GLOBAL VBFR
SET    FWRK = 0

```

```

#for sending I
GLOBAL FIGN           ;=1: need to send 'I'
SET    FIGN = 0

```

```

#used for send N
GLOBAL FNEW           ;>0 when new part on buffer
SET    FNEW = 0

```

```

#used for send L
GLOBAL LFLAG
GLOBAL LBFR
GLOBAL LTASK
SET    LFLAG = 0

```

```

#used for send U
GLOBAL UFLAG
GLOBAL UBFR
SET    UFLAG = 0

```

```

#used for send C
GLOBAL CFLAG           ;>0 source buffer number
GLOBAL CBFR1
GLOBAL CBFR2           ;target buffer number
SET    CFLAG = 0

```

```

#used for send M (machine work)
GLOBAL MFL
GLOBAL MMACH
SET    MFL = 0
DIMG   QM[5]

```

```

#used for send X (flipping)
GLOBAL XFLAG
GLOBAL XMACH
SET    XFLAG = 0

```

```

#used for send J (loading)
GLOBAL JFLAG
GLOBAL JMACH
SET JFLAG = 0

#used for send H (homming)
GLOBAL HFLAG
SET HFLAG = 0

#used for send K (assembly)
GLOBAL KFLAG
GLOBAL KPRT
SET KFLAG = 0

#EMPTY pallets needed
GLOBAL NEMP
SET NEMP = 0 ;number of empty pallets needed

#number of EMPTY pallets waiting
GLOBAL FEMP
SET FEMP = 0 ;=1 when empty pallet is in station

GLOBAL FTMP ;template flag
; 0 : nothing
; 1 : send T
; 2 : after sending
; 3 : received V (o.k.)
; 4 : received K (missing)

GLOBAL TPART ;part number need template
GLOBAL BTEMP ;buffer number of empty template for part

GLOBAL APART ;part number for assembly
DIM QASM[40] ;parts need to assembly

GLOBAL AFLAG ;assembly flag
; 1 : send A (o.k.)
; 2 : send D (can't assembly)
GLOBAL APRT ;part number

GLOBAL ASMBF ;>0 after start assembly until getting
SET ASMBF = 0

#buffer contents: [ index is (MACHINE-1) * 2 + (machine BUFFER) ]
DIM MTASK[8] ;task required
DIM MFLC[8] ;0 : empty, 1: with part, 2: in mach., 3: ready
DIM MTMPL[8] ;template number
DIM MPART[8] ;part number
DIM MRAM[8] ;raw material number
DIM MRAMV[8] ;raw material number (assembly)
DIM MSUBP[8] ;required subprocess
DIM MNFC[8] ;information on part (after q.c.)
DIM MBUSY[4] ;0-machine free, 1/2 - number of buffer in use
DIM MFIN[4] ;>0 when machine has finished
DIM LASTT[4] ;last task completed at machine
DIM FTASK[4] ;=1 when program loading to machine
GLOBAL TFLAG
SET TFLAG = 0 ;=1 when changing programs
GLOBAL UTASK ;=0 when task loading in use
SET UTASK = 1

#using ROBOT, only ONE program can use a robot at any time

```

```

GLOBAL ROBOT          ;flag set to 0 when robot is in use
SET    ROBOT = 1

#station buffer number for robot to go to
#      [(MACHINE-1) * 2 + (machine BUFFER number)]
GLOBAL BFR
GLOBAL MACH           ;machine number load/unload

#maximum buffers , normal 2, asrs 1, cnc 2 for each machine
# normally (2 buffers * 1 machine)
GLOBAL MAXBF
SET    MAXBF = 2 ;=4 in a station with 2 machines

#number of free buffers at this station
GLOBAL NWORK
SET    NWORK = MAXBF

#for debugging
GLOBAL PRNT
SET    PRNT = 0 ;1 involves printing of communication

#station id
GLOBAL STNUM         ;station number

# flag = 1 when at ASRS station
GLOBAL ASRS
SET    ASRS = 0

#use for robot moves
DI MPA  C M[500] ;robot points
DI MPB  C M[500] ;slidebase points

#flags, used when robot program runs
GLOBAL MMF          ;by MACH
GLOBAL PMF          ;by PTOV
GLOBAL TVF          ;by TOW
GLOBAL VPF          ;by WTCP

#if using slidebase, axis number for homing
GLOBAL SLIDE
SET    SLIDE = 7 ;default

#when system is busy don't stop pallets
GLOBAL SBUSY
SET    SBUSY = 0 ;=1 when busy

#reset flag = 1 -> send 'S' to central controller
GLOBAL FRST
SET    FRST = 0

#for using customize number of buffer each machine
DI MG   MBMAX[4]
DI MG   BUFMA[8]

SET     MBMAX[1] = 2 ;default 2 buffer each machine
SET     MBMAX[2] = 2
SET     MBMAX[3] = 2
SET     MBMAX[4] = 2

#      <<<<<< CNTID1 >>>>>>

SET     STNUM = 1 ;station number

# <<<< VARASR (ASRS station) >>>>

```

```

SET    MAXBF = 2;1 buffer + 1 the ASRS
SET    NWORK = MAXBF
SET    ASRS = 1 ;i ndi cate "at ASRS st at i on"
#    <<<<<  VAR2  (2 MACH NES in station)  >>>>>

SET    MAXBF = 4;2 buffers * 2 machi nes
SET    NWORK = MAXBF
#    <<<<<<  START  >>>>>>>>

PROGRAM HOMES          ;1 homi ng
END

PROGRAM CIM            ;2 r un al l CIM ACL syst em progr ams
END

PROGRAM RESET         ;3 r eset CIM syst em
END

PROGRAM $CLR          ;4 for user appl i cat i ons
END

PROGRAM CIMP          ;5 att ach to CIM poi nt
END

PROGRAM $U1           ;6 att ach to 1ST user progr am
END

PROGRAM $L2           ;7 att ach to 2ND user progr am
END

PROGRAM $L3           ;8 att ach to 3RD user progr am
END

PROGRAM $L4           ;9 att ach to 4TH user progr am
END

PROGRAM $I N I T      ;10 i n i t i a l i z e ASRS carousel posi ti on vect or
END

PROGRAM DTTP          ;11 t empl at e to pal let
END

PROGRAM DTFP          ;12 t empl at e fr om pal let
END

PROGRAM DTTB          ;13 t empl at e to buf fer
END

PROGRAM DTFB          ;14 t empl at e fr om buf fer
END

PROGRAM DPTB          ;15 part to buf fer
END

PROGRAM DPFB          ;16 part fr om buf fer
END

PROGRAM DPTM          ;17 part to machi ne
END

PROGRAM DPFM          ;18 part fr om machi ne
END

```

```

PROGRAM DVTM      ;19 start machine and wait for it to finish
END

PROGRAM MM        ;20 for future use
END

PROGRAM SELB1     ;21 select buffer 1
SET BFR = 1
END

PROGRAM SELB2     ;22 select buffer 2
SET BFR = 2
END

PROGRAM SELB3     ;23 select buffer 3
SET BFR = 3
END

PROGRAM SELB4     ;24 select buffer 4
SET BFR = 4
END

PROGRAM SELM1     ;25 select machine 1
SET MACH = 1
END

PROGRAM SELM2     ;26 select machine 2
SET MACH = 2
END

PROGRAM OGRI P    ;27 open gripper
OPEN
END

PROGRAM CGRI P    ;28 close gripper
CLOSE
END

PROGRAM $$29      ;29
END

PROGRAM $$30      ;30
END

PROGRAM $$31      ;31
END

PROGRAM $$32      ;32
END

PROGRAM $$33      ;33
END

PROGRAM $$34      ;34
END

PROGRAM $$35      ;35
END

PROGRAM $$36      ;36
END

PROGRAM $$37      ;37

```

```

END

PROGRAM $$38      ; 38
END

PROGRAM $$39      ; 39
END

PROGRAM $$40      ; 40
END

PROGRAM $ST1      ; 41 attach to 1ST station program
END

PROGRAM $ST2      ; 42 attach to 2ND station program
END

PROGRAM $ST3      ; 43 attach to 3RD station program
END

PROGRAM $ST4      ; 44 attach to 4TH station program
END

PROGRAM $ST5      ; 45 attach to 5TH station program
END

PROGRAM $ST6      ; 46 attach to 6TH station program
END

PROGRAM $ST7      ; 47 attach to 7TH station program
END

PROGRAM $ST8      ; 48 attach to 8TH station program
END

PROGRAM $ST9      ; 49 attach to 9TH station program
END

PROGRAM $ST10     ; 50 attach to 10TH station program
END

PROGRAM SFLP1
END

PROGRAM SFLP2
END

PROGRAM SFLP3
END

PROGRAM SFLP4
END

PROGRAM WTL1
WAIT FTASK[ 1] = 0
END

PROGRAM WTL2
WAIT FTASK[ 2] = 0
END

PROGRAM WTL3

```



```
WAIT FTASK[ 3] = 0  
END
```

```
PROGRAM WTL4  
WAIT FTASK[ 4] = 0  
END
```

```
# <<<<< STPOINT >>>>
```

```
#default template manipulation positions  
# buffer n must be point n  
# pallet must be point 9  
# for n (1 .. 9)  
# n+10 is relative to n 100 mm up
```

```
TEACHR C M [ 11] C M [ 1]  
0  
0  
1000  
0  
0
```

```
TEACHR C M [ 12] C M [ 2]  
0  
0  
1000  
0  
0
```

```
TEACHR C M [ 13] C M [ 3]  
0  
0  
1000  
0  
0
```

```
TEACHR C M [ 14] C M [ 4]  
0  
0  
1000  
0  
0
```

```
TEACHR C M [ 15] C M [ 5]  
0  
0  
1000  
0  
0
```

```
TEACHR C M [ 16] C M [ 6]  
0  
0  
1000  
0  
0
```

```
TEACHR C M 17] C M 7]
0
0
1000
0
0
```

```
TEACHR C M 18] C M 8]
0
0
1000
0
0
```

```
TEACHR C M 19] C M 9]
0
0
1000
0
0
```

```
# <<<<< TMLTA >>>>
```

```
# get template from pallet
PROGRAM FROMP /N
SPEEDA 40
MOVED C M 19]
GOSUB CGR P
SPEEDA 5
MOVELD C M 9]
DELAY 50
GOSUB CGR P
SPEEDA 10
MOVELD C M 19]
END
```

```
# put template on pallet
PROGRAM TOPLT /N
SPEEDA 40
MOVED C M 19]
SPEEDA 5
MOVELD C M 9]
DELAY 50
GOSUB CGR P
SPEEDA 10
MOVELD C M 19]
END
```

```
# get template from buffer (BFR)
PROGRAM FROMB /N
DEFINE B1

SET B1 = BFR + 10
SPEEDA 40
MOVED C M B1]
GOSUB CGR P
SPEEDA 5
MOVELD C M BFR]
GOSUB CGR P
SPEEDA 10
```

```
MOVELD C M B1]
END
```

```
# put template on buffer (BFR)
PROGRAM TCBFR / N
DEFINE B1
```

```
SET B1 = BFR + 10
SPEEDA 40
MOVED C M B1]
SPEEDA 5
MOVELD C M BFR]
GOSUB CGR P
SPEEDA 10
MOVELD C M B1]
END
```

```
# <<<<< TMLTB >>>>
```

```
# get template from pallet
PROGRAM FROMP / N
SPEED 40
MOVED C M B[ 9]
MOVED C M 19]
GOSUB CGR P
SPEEDA 5
MOVELD C M 9]
DELAY 50
GOSUB CGR P
SPEEDA 10
MOVELD C M 19]
END
```

```
# put template on pallet
PROGRAM TOPLT / N
SPEED 40
MOVED C M B[ 9]
MOVED C M 19]
SPEEDA 5
MOVELD C M 9]
DELAY 50
GOSUB CGR P
SPEEDA 10
MOVELD C M 19]
END
```

```
# get template from buffer (BFR)
PROGRAM FROMB / N
DEFINE B1
```

```
SET B1 = BFR + 10
SPEED 40
MOVED C M B[ BFR]
MOVED C M B1]
GOSUB CGR P
SPEEDA 5
MOVELD C M BFR]
GOSUB CGR P
SPEEDA 10
MOVELD C M B1]
END
```

```
# put template on buffer (BFR)
PROGRAM TCBFR /N
DEFINE B1
```

```
SET B1 = BFR + 10
SPEED 40
MOVED C M[ BFR]
MOVED C M[ B1]
SPEEDA 5
MOVELD C M[ BFR]
GOSUB CGRIP
SPEEDA 10
MOVELD C M[ B1]
END
```

```
# <<<<< RCBA >>>>
```

```
#home group B
PROGRAM STB /N
END
```

```
#cim point
PROGRAM CIMP /Y
DELAY 30
@ATTACH CIM
END
```

```
# <<<<< ROBB >>>>
```

```
#home group B
PROGRAM STB /N
H-HOME SLIDE
SPEEDB 40
MOVED C M[ 9]
END
```

```
#cim point
PROGRAM CIMP /Y
DELAY 30
@ATTACH CIM
DELAY 30
@ATTACH CMB
END
```

```
# <<<<< CONTIN >>>>>
```

```
# assign 'static number' programs to execute user programs
```

```
#home robot
PROGRAM STA /N
HOME
GOSUB STB
SPEEDA 40
MOVED C M[ 19]
END
```

```
#home group
PROGRAM HOMES /Y
;SET HFLAG = 1
```

```
GOSUB STA
;SET HFLAG = 3
STOP CIM
DELAY 10
RUN CIM
END
```

```
#get part from buffer (BFR)
PROGRAM FB / N
CLOSE
END
```

```
#put part on buffer (BFR)
PROGRAM TCB / N
OPEN
END
```

```
#load part to machine 1
PROGRAM TOM1 / N
OPEN
END
```

```
#load part to machine 2
PROGRAM TOM2 / N
OPEN
END
```

```
#load part to machine 3
PROGRAM TOM3 / N
OPEN
END
```

```
#load part to machine 4
PROGRAM TOM4 / N
OPEN
END
```

```
#load part on machine (MACH)
PROGRAM TOM / N
```

```
IF MACH = 1
  GOSUB TOM1
ELSE
  IF MACH = 2
    GOSUB TOM2
  ELSE
    IF MACH = 3
      GOSUB TOM3
    ELSE
      IF MACH = 4
        GOSUB TOM4
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
```

END

```
#unload part from machine 1
PROGRAM FM1 / N
OPEN
END
```

```
#unload part from machine 2
PROGRAM FM2 / N
OPEN
END
```

```
#unload part from machine 3
PROGRAM FM3 / N
OPEN
END
```

```
#unload part from machine 4
PROGRAM FM4 / N
OPEN
END
```

```
#unload part from machine (MACH)
PROGRAM FM / N
```

```
IF MACH = 1
  GOSUB FM1
ELSE
  IF MACH = 2
    GOSUB FM2
  ELSE
    IF MACH = 3
      GOSUB FM3
    ELSE
      IF MACH = 4
        GOSUB FM4
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

END

```
PROGRAM LMOD1 / N; load mode machine 1 program (deny if user not need)
END
```

```
PROGRAM LMOD2 / N; load mode machine 2 program (deny if user not need)
END
```

```
PROGRAM LMOD3 / N; load mode machine 3 program (deny if user not need)
END
```

```
PROGRAM LMOD4 / N; load mode machine 4 program (deny if user not need)
END
```

```
PROGRAM AMOD1 / N; auto mode machine 1 program (deny if user not need)
END
```

```
PROGRAM AM02 /N; auto mode machine 2 program (dummy if user not need)
END
```

```
PROGRAM AM03 /N; auto mode machine 3 program (dummy if user not need)
END
```

```
PROGRAM AM04 /N; auto mode machine 4 program (dummy if user not need)
END
```

```
#clear
PROGRAM CLEAR /N ; 'clear' program (dummy if user not need)
END
```

```
PROGRAM $CLR /Y
GOSUB CLEAR
END
```

```
#user
PROGRAM USER1 /N ; user 1 program (dummy if user not need)
END
```

```
PROGRAM USER2 /N ; user 2 program (dummy if user not need)
END
```

```
PROGRAM USER3 /N ; user 3 program (dummy if user not need)
END
```

```
PROGRAM USER4 /N ; user 4 program (dummy if user not need)
END
```

```
PROGRAM $U1 /Y
GOSUB USER1
END
```

```
PROGRAM $U2 /Y
GOSUB USER2
END
```

```
PROGRAM $U3 /Y
GOSUB USER3
END
```

```
PROGRAM $U4 /Y
GOSUB USER4
END
```

```
#init
PROGRAM INT /N
END
```

```
PROGRAM $INT /Y
GOSUB INT
END
```

```
PROGRAM DTTP /Y ; 11 template to pallet
GOSUB TOPLT
END
```

```
PROGRAM DTFP /Y ; 12 template from pallet
GOSUB FROMP
```

```

END

PROGRAM DTTB / Y ; 13 template to buffer
GOSUB TCBFR
END

PROGRAM DTFB / Y ; 14 template from buffer
GOSUB FROMB
END

PROGRAM DPTB / Y ; 15 part to buffer
GOSUB TCB
END

PROGRAM DPFB / Y ; 16 part from buffer
GOSUB FB
END

PROGRAM DPTM / Y ; 17 part to machine
GOSUB TOM
END

PROGRAM DPFM / Y ; 18 part from machine
GOSUB FM
END

PROGRAM $WTM / N
END

PROGRAM DWTM / Y ; 19 start machine and wait for it to finish
GOSUB $WTM
END

PROGRAM FLI P1 / N ; flipping part on machine 1 (dummy if user not need)
END

PROGRAM FLI P2 / N ; flipping part on machine 2 (dummy if user not need)
END

PROGRAM FLI P3 / N ; flipping part on machine 3 (dummy if user not need)
END

PROGRAM FLI P4 / N ; flipping part on machine 4 (dummy if user not need)
END

PROGRAM SFLP1 / Y ; system flipping part using robot
DEFINE R
SET SBUSY = 1
PEND R FROM ROBOT
SET XMACH = 1
SET XFLAG = 1
GOSUB FLI P1
SET XFLAG = 3
WAIT XFLAG = 0
SET SBUSY = 0
POST 1 TO ROBOT
END

PROGRAM SFLP2 / Y ; system flipping part using robot
DEFINE R
SET SBUSY = 1
PEND R FROM ROBOT
SET XMACH = 2

```



```

SET      XFLAG = 1
GOSUB   FLI P2
SET      XFLAG = 3
WAIT    XFLAG = 0
SET      SBUSY = 0
POST    1 TO ROBOT
END

```

```

PROGRAM  SFLP3 / Y      ; system flipping part using robot
DEFINE   R
SET      SBUSY = 1
PEND     R FROM ROBOT
SET      XMACH = 3
SET      XFLAG = 1
GOSUB   FLI P3
SET      XFLAG = 3
WAIT    XFLAG = 0
SET      SBUSY = 0
POST    1 TO ROBOT
END

```

```

PROGRAM  SFLP4 / Y      ; system flipping part using robot
DEFINE   R
SET      SBUSY = 1
PEND     R FROM ROBOT
SET      XMACH = 4
SET      XFLAG = 1
GOSUB   FLI P4
SET      XFLAG = 3
WAIT    XFLAG = 0
SET      SBUSY = 0
POST    1 TO ROBOT
END

```

```

PROGRAM  START / N
END

```

```

PROGRAM  URST / N
END

```

```

#      <<<<<  CVER  >>>>>

```

```

# printing: version number, date, and station number

```

```

PROGRAM  CVER

```

```

PRI NTLN " ER - CIM  VERSION 1.83"
PRI NTLN " DATE 26/ 10/ 92"
PRI NTLN " STATION " STNUM
PRI NTLN ">"

```

```

END

```

```

#      <<<<<  FINISH  >>>>>

```

```

# get number of machine that has finished from queue

```

```

PROGRAM  FINIS
LABEL    1
QPEND   FBFR FROM QFIN      ; queue of finished machines
SET      FFIN = 1          ; flag : a machine has finished
WAIT    FFIN = 0
GOTO    1
END

```

```

#      <<<<<  PALLETS  >>>>>

```

```

GLOBAL   TMAX
SET      TMAX = 100
GLOBAL   TD
SET      TD = 800

# stop the pallet
PROGRAM STOPP          ; does not perform any function at this time
                    ; since all pallets stop at station
END

# release the pallet
PROGRAM MOVEP /Y
DEFINE   TNEXT
DEFINE   T1
SET      T1 = TNEXT - TIME
IF       T1 > TMAX
    SET   TNEXT = TIME + TMAX
ENDIF
WAIT    TIME > TNEXT
SET     TNEXT = TIME + TD

SET     PALET = 0
SET     OUT[6] = 1
DELAY   100
SET     OUT[6] = 0
END

# get pallet number
PROGRAM PLTS
DEFINE   PLT

IF       IN[6] = 1          ; pallet already here
    SET   OUT[6] = 1      ; release it
    WAIT  IN[6] = 0      ; be sure the pallet is released
    SET   OUT[6] = 0
ENDIF

LABEL    1
SET      OUT[5] = 1      ; command: stop pallet
WAIT    IN[6] = 1      ; wait until pallet comes

SET      PLT = IN[4] * 2  ; get pallet id
SET      PLT = PLT + IN[3]
SET      PLT = PLT * 2
SET      PLT = PLT + IN[2]
SET      PLT = PLT * 2
SET      PLT = PLT + IN[1]

SET      PALET = PLT
SET      FPLT = 1

WAIT    PALET = 0      ; wait until pallet is released
WAIT    IN[6] = 0

GOTO    1
END

# <<<< I SVALID >>>>

# dummy program for stations that can't release parts
PROGRAM GETP /N

```

```
#example : for assembly get part for assembly place  
END
```

```
PROGRAM ISVLD / N  
DEFINE R  
SET BTEMP = 0 ; buffer number contain empty template for part  
SET TPART = RPART ; part number needed template  
SET FTMP = 1 ; send command "need template"  
WAIT BTEMP > 0 ; wait for empty template  
PEND R FROM ROBOT  
SET BFR = BTEMP  
SET PART = RPART  
SET RAW = 0  
SET RAWN = 0  
SET SBUSY = 1 ; don't stop pallets  
SET ATFLG = 7  
WAIT ATFLG = 0  
GOSUB GETP ; get part  
SET ATFLG = 9  
WAIT ATFLG = 0  
GOSUB TCB ; put the part on buffer  
SET SBUSY = 0  
POST 1 TO ROBOT  
SET ASMBF = 0 ; can assemble another part  
SET FV = BTEMP ; buffer number contain the ready part  
END
```

```
# <<<<<< ASMB >>>>>>
```

```
PROGRAM ASMBL / N ; user assembly program  
#user program to assembly (PART)  
END
```

```
PROGRAM ASMB1  
DEFINE LPART  
DEFINE R  
  
LABEL 1  
QPEND LPART FROM QASM  
WAIT ASMBF = 0 ; don't assemble the second part  
SET ASMBF = 1 ; before getting the first  
PEND R FROM ROBOT  
SET SBUSY = 1  
SET PART = LPART  
SET KPRT = LPART  
SET KFLAG = 1 ; send 'H' starting assembly  
GOSUB ASMBL  
WAIT KFLAG = 0  
SET KFLAG = 3 ; send 'h' end assembly  
WAIT KFLAG = 0  
SET SBUSY = 0  
POST 1 TO ROBOT  
SET APRT = LPART  
SET AFLAG = 1  
WAIT AFLAG = 0  
GOTO 1  
END
```

```
PROGRAM ASMB  
QPOST APART TO QASM  
SET APART = 0  
END
```

```

# <<<<<  RELEASE  >>>>>

#execute part release
PROGRAM RELES

GOSUB  ISVLD          ;check if part is valid

IF     FV = 0
  SET   FREL = 2      ; not found
ELSE
  SET   FREL = 1      ;'found material'
  IF    MFLQ[FV] < 4
    SET   MFLQ[FV] = 3 ;ready pary
    SET   MITMPL[FV] = RTMPL ;template number
  ENDF
  SET   MFLQ[FV] = 3 ;ready pary
  SET   MPART[FV] = RPART;part number
  SET   MRAW[FV] = RRAW;raw material number
  SET   MRAWN[FV] = RRAWN;raw material number (assembly)
  SET   MSUBP[FV] = 0
  SET   MTASK[FV] = 0
  SET   MNFQ[FV] = 0
  SET   FNEW = FV
  WAIT  FNEW = 0
  QPOST FV TO CFIN    ;finish release ready to continue
ENDF

END

# <<<<<  TASK  >>>>>

GLOBAL  $MCH1
GLOBAL  $MCH2
GLOBAL  $MCH3
GLOBAL  $MCH4
SET     $MCH1 = 1
SET     $MCH2 = 2
SET     $MCH3 = 3
SET     $MCH4 = 4

GLOBAL  $ENDT

#waiting for characters to be received on the rs232 #2
PROGRAM WAIT2
LABEL  1

SET     GCOM[2] = 0
                ; command to finish being executed
GETCOM  2, GCOM[2] ;get the command

IF     PRINT = 1 ;in DEBUG mode
  PRINTLN "$ in2 : " GCOM[2]
ENDF

IF     GCOM[2] = 'S' ;start
  SET   TFLAG = 1
ENDF

IF     GCOM[2] = 'E' ;end
  SET   TFLAG = 0
ENDF

```

```

IF      GCOM[ 2] = ' R           ; rest art
  SET   LASTT[ 1] = 0
  SET   LASTT[ 2] = 0
  SET   LASTT[ 3] = 0
  SET   LASTT[ 4] = 0
  SET   TFLAG = 1
ENDIF

IF      GCOM[ 2] = ' G           ; start loadi ng
ENDIF

IF      GCOM[ 2] = ' V
  DELAY 20
  SET   $ENDT = 1
ENDIF

IF      GCOM[ 2] = ' K           ; cannot loadi n (missi ng file or definiti on)
  ; NEXT VERSI ON ...
ENDIF

GOTO   1                       ;for the next
END

PROGRAM TASK1 / N
DEFIN E  RCODE
DEFIN E  U
IF      TFLAG = 1
  IF      TASK1 <> LASTT[ 1]
    SET   FTASK[ 1] = 1
    GOSUB LMDD1
    PEND  U FROM UTASK
    SET   JMACH = 1
    SET   JFLAG = 1
    SET   $ENDT = 0
    PROCM 2 " %L"
    PROCM 2 " " $MCH1
    PRLNCOM 2 " " TASK1

    WAIT  $ENDT = 1

    SET   JMACH = 1
    SET   JFLAG = 3
    POST  1 TO UTASK
    GOSUB AMDD1
    SET   LASTT[ 1] = TASK1
  ENDIF
ENDIF
END

PROGRAM TASK2 / N
DEFIN E  RCODE
DEFIN E  U
IF      TFLAG = 1
  IF      TASK2 <> LASTT[ 2]
    SET   FTASK[ 2] = 1
    GOSUB LMDD2
    PEND  U FROM UTASK
    SET   JMACH = 2
    SET   JFLAG = 1
    SET   $ENDT = 0
    PROCM 2 " %L"
    PROCM 2 " " $MCH2
    PRLNCOM 2 " " TASK2
  ENDIF
ENDIF

```

```

    WAIT    $ENDT = 1

    SET     JMACH = 2
    SET     JFLAG = 3
    POST    1 TO UTASK
    GOSUB   AMDD2
    SET     LASTT[ 2] = TASK2
  ENDF
ENDF
END

```

```

PROGRAM TASK3 / N
DEFIN E ROODE
DEFIN E U
IF      TFLAG = 1
  IF    TASK3 <> LASTT[ 3]
    SET  FTASK[ 3] = 1
    GOSUB LMDD3
    PEND U FROM UTASK
    SET  JMACH = 3
    SET  JFLAG = 1
    SET  $ENDT = 0
    PROOM 2 "%L"
    PROOM 2 " " $MCH3
    PRLNCOM 2 " " TASK3

    WAIT    $ENDT = 1

    SET     JMACH = 3
    SET     JFLAG = 3
    POST    1 TO UTASK
    GOSUB   AMDD3
    SET     LASTT[ 3] = TASK3
  ENDF
ENDF
END

```

```

PROGRAM TASK4 / N
DEFIN E ROODE
DEFIN E U
IF      TFLAG = 1
  IF    TASK4 <> LASTT[ 4]
    SET  FTASK[ 4] = 1
    GOSUB LMDD4
    PEND U FROM UTASK
    SET  JMACH = 4
    SET  JFLAG = 1
    SET  $ENDT = 0
    PROOM 2 "%L"
    PROOM 2 " " $MCH4
    PRLNCOM 2 " " TASK4

    WAIT    $ENDT = 1

    SET     JMACH = 4
    SET     JFLAG = 3
    POST    1 TO UTASK
    GOSUB   AMDD4
    SET     LASTT[ 4] = TASK4
  ENDF
ENDF

```

END

```
PROGRAM $TSK1
GOSUB  TASK1
SET    FTASK[ 1] = 0
END
```

```
PROGRAM $TSK2
GOSUB  TASK2
SET    FTASK[ 2] = 0
END
```

```
PROGRAM $TSK3
GOSUB  TASK3
SET    FTASK[ 3] = 0
END
```

```
PROGRAM $TSK4
GOSUB  TASK4
SET    FTASK[ 4] = 0
END
```

```
PROGRAM $TASK
IF     MACH = 1
  SET  TASK1 = TASK
  RUN  $TSK1
ELSE
  IF   MACH = 2
    SET TASK2 = TASK
    RUN $TSK2
  ELSE
    IF   MACH = 3
      SET TASK3 = TASK
      RUN $TSK3
    ELSE
      IF   MACH = 4
        SET TASK4 = TASK
        RUN $TSK4
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
END
```

<<<<<< MACHINE >>>>>>

```
PROGRAM $MFB
IF     PART > 0
  IF   TASK < 900
    SET  FTASK[ MACH] = 2
    GOSUB $TASK
  ; ELSE
  ; SET  FTASK[ MACH] = 0
  ENDIF
  GOSUB FB           ;get part from buffer
ENDIF
SET   MF = 0
END
```

```
PROGRAM $MITM
IF     PART > 0
```

```

    GOSUB   TCM           ; put part in machine
  ENDF
  SET     MF = 0
  END

```

```

PROGRAM $MFM
  IF     PART > 0
    GOSUB   FM           ; get part from machine
  ENDF
  SET     MF = 0
  END

```

```

PROGRAM $MTB
  IF     PART > 0
    GOSUB   TCB           ; put part on buffer
  ENDF
  SET     MF = 0
  END

```

```

PROGRAM VTM1 / N
  DELAY  100
  END

```

```

PROGRAM VTM2 / N
  DELAY  100
  END

```

```

PROGRAM VTM3 / N
  DELAY  100
  END

```

```

PROGRAM VTM4 / N
  DELAY  100
  END

```

```

PROGRAM $TM1
  IF     PART > 0
    GOSUB   VTL1
    IF     TASK1 < 900
      QPOST 1 TO QM
    ENDF
    GOSUB   VTM1
    IF     TASK1 < 900
      QPOST 101 TO QM
    ENDF
  ENDF
  SET     MF[N 1] = 1
  END

```

```

PROGRAM $TM2
  IF     PART > 0
    GOSUB   VTL2
    IF     TASK2 < 900
      QPOST 2 TO QM
    ENDF
    GOSUB   VTM2
    IF     TASK2 < 900
      QPOST 102 TO QM
    ENDF
  ENDF
  SET     MF[N 2] = 1
  END

```



```

PROGRAM $TM8
IF PART > 0
  GOSUB VTL3
  IF TASK3 < 900
    QPOST 3 TO QM
  ENDF
  GOSUB VTM8
  IF TASK3 < 900
    QPOST 103 TO QM
  ENDF
ENDF
SET MFI N[3] = 1
END

```

```

PROGRAM $TM4
IF PART > 0
  GOSUB VTL4
  IF TASK4 < 900
    QPOST 4 TO QM
  ENDF
  GOSUB VTM4
  IF TASK4 < 900
    QPOST 104 TO QM
  ENDF
ENDF
SET MFI N[4] = 1
END

```

```

PROGRAM $REPM
DEFINE M

LABEL 1
QPEND M FROM QM
SET MMACH = M MOD 100
IF M > 100
  SET MFL = 3
ELSE
  SET MFL = 1
ENDIF
WAIT MFL = 0
GOTO 1
END

```

```

PROGRAM $VTM / Y
IF MACH = 1
  SET TASK1 = TASK
  RUN $TM1
ENDIF
IF MACH = 2
  SET TASK2 = TASK
  RUN $TM2
ENDIF
IF MACH = 3
  SET TASK3 = TASK
  RUN $TM8
ENDIF
IF MACH = 4
  SET TASK4 = TASK
  RUN $TM4
ENDIF
END

```

```

#check if can move part to or from machine
PROGRAM MACH
DEFINE I ; machine (1 - 4) up to 4 machines
DEFINE J ; machine buffer (1 - 2)
DEFINE B ; absolute buffer number (1 - 8)
DEFINE R

LABEL 1
FOR I = 1 TO 4 ; up to 4 machines
  IF MBUSY[I] = 0 ; machine empty! check part ready for machine
    FOR J = 1 TO MAXBF
      SET B = J
      IF BUFMA[B] = 1
        ANDIF MFLG[B] = 1 ; part waiting for machine
          SET SBUSY = 1 ; don't stop pallets (robot is busy)

          PEND R FROM ROBOT ; waiting for robot to be available
          SET BFR = B ; buffer
          SET MACH = I ; machine
          SET MMF = 1 ; get part from buffer
          SET PART = MPART[B]
          SET RAW = MRAW[B]
          SET RAVW = MRAVW[B]
          SET TASK = MTASK[B]
          SET SUBP = MSUBP[B]
          SET INFO = MNFC[B]
          SET LTASK = TASK
          SET LBFR = B
          SET LFLAG = 1 ; report start load machine
          RUN $MFB
          WAIT MMF = 0 ; wait to finish getting part
          WAIT LFLAG = 0
          SET LFLAG = 3
          SET MMF = 2 ; load machine with part
          RUN $MTM
          WAIT MMF = 0 ; wait until loading is finished
          SET MNFC[B] = INFO
          SET SBUSY = 0 ; continue to stop pallets
          SET MFLG[B] = 2 ; part in the machine
          WAIT LFLAG = 0
          SET LFLAG = 5
          WAIT LFLAG = 0
          SET MBUSY[I] = J ; specify buffer number
          POST 1 TO ROBOT ; free robot
          SET TASK = MTASK[B]
          GOSUB $VTM ; wait for machine to finish task
          SET J = 9 ; last machine buffer + 1
          ENDIF ; (don't continue check this machine)
        ENDFOR
      ELSE
        IF MFIN[I] > 0 ; if machine is finished
          SET SBUSY = 1 ; don't stop pallets
          PEND R FROM ROBOT ; waiting for robot to be available
          SET B = MBUSY[I]
          SET PART = MPART[B]
          SET RAW = MRAW[B]
          SET RAVW = MRAVW[B]
          SET TASK = MTASK[B]
          SET SUBP = MSUBP[B]
          SET INFO = MNFC[B]
          SET BFR = B ; buffer number
          SET MACH = I ; machine number
          SET MMF = 3 ; unload part from machine
          IF TASK < 900

```

```

        SET      UBFR = B
        SET      UFLAG = 1 ;report start unload machine
    ENDF
    RUN      $MFM
    WAIT     MMF = 0 ;wait until unload is finished
    IF      TASK < 900
        WAIT     UFLAG = 0
        SET      UFLAG = 3
    ENDF
    SET      MMF = 4 ;put part on buffer
    RUN      $MTB
    SET      SBUSY = 0 ;continue to stop pallets
    WAIT     MMF = 0 ;wait to finish putting part on buffer
    IF      TASK < 900
        SET      MFLQBJ = 3 ;tag that task is finished
    ELSE
        SET      MFLQBJ = 4 ;empty template
    ENDF
; and part ready for next subprocess
    SET      MNFQBJ = INFO
    IF      TASK < 900
        WAIT     UFLAG = 0
        SET      UFLAG = 5 ;for send U to central controller
        WAIT     UFLAG = 0
    ENDF
    SET      SBUSY = 0 ;continue to stop pallets
    SET      MFIN[I] = 0 ;reset 'finish' and 'busy' flags
    SET      MBUSY[I] = 0
    CPOST    B TO QFIN ;queue of buffer numbers that contain
; parts ready for next process
    POST     1 TO ROBOT ;release the robot to other program
    ENDF
    ENDF
    ENDFOR
    DELAY    10

    GOTO    1
    END

# <<<<<< PTOW >>>>>

PROGRAM PP
GOSUB FROMP ;get template from pallet
SET PVF = 0
END

PROGRAM PB
GOSUB TOBFR ;put template on buffer
SET PVF = 0
END

# move template from pallet to machine buffer for work
PROGRAM PTOW/N
DEFINE B ;machine buffer number
DEFINE I ;absolute buffer number

SET B = MAXBF + 1 ;search free buffer of machine
LABEL 1
SET B = B - 1
IF B = 0 ;no free buffer for that machine
    SET YMA = 0 ;reset the command parameters
    SET YTMPL = 0
    SET YTASK = 0

```

```

SET      YPART = 0
SET      YRAW = 0
SET      YRAVN = 0
SET      YSUBP = 0
SET      YINFO = 0
SET      FIGN = 1      ; and release the pallet
GOTO     9
ENDIF
IF      YMA = 0 ; machine number must be > 0
SET      YMA = 1
ENDIF

SET      I = B
IF      BUFMA[I] <> YMA
  CRIF   MFLQ[I] > 0 ; =0 when buffer is free
  GOTO   1
ENDIF
SET      NWORK = NWORK - 1 ; decrement 'free buffers in station'
PEND     B FROM ROBOT      ; wait for free robot to be available

SET      BFR = I          ; buffer number
                          ; update buffer contents information
SET      MTASK[BFR] = YTASK ; task number
SET      MITPL[BFR] = YTMPL ; template number
SET      MPART[BFR] = YPART ; part number
SET      MRAW[BFR] = YRAW   ; raw material number
SET      MRAVN[BFR] = YRAVN ; raw material number
SET      MSUBP[BFR] = YSUBP ; subprocess
SET      MNFC[BFR] = YINFO  ; part information
SET      YMA = 0           ; reset the command
SET      YTMPL = 0
SET      YTASK = 0
SET      YPART = 0
SET      YRAW = 0
SET      YRAVN = 0
SET      YSUBP = 0
SET      YINFO = 0

IF      MRAW[BFR] < 0
  SET    MFLQ[BFR] = 4
ELSE
  SET    MFLQ[BFR] = 1      ; buffer contain part
ENDIF

SET      VBFR = BFR
SET      FVRK = 1
SET      PVF = 1          ; get template from pallet
RUN      PP
WAIT     PVF = 0          ; wait to finish getting template

WAIT     FVRK = 0
SET      FVRK = 3        ; send 'W' to center

SET      PVF = 2 ; put template on buffer
RUN      PB
WAIT     PVF = 0 ; wait to finish putting template on buffer
WAIT     FVRK = 0
SET      FVRK = 5
WAIT     FVRK = 0

IF      MRAW[BFR] < 0
  SET    BTEMP = BFR
ENDIF

```

```

POST    1 TO ROBOT    ; release the robot

LABEL 9
END

#      <<<<< TOW >>>>>

PROGRAM TF
GOSUB  FROMB          ; get template from buffer
SET    TVF = 0
END

PROGRAM TT
GOSUB  TOBFR          ; put template on buffer
SET    TVF = 0
END

#continue work in station
PROGRAM TOW
DEFINE B              ; machine buffer number
DEFINE I              ; absolute buffer number

IF      CBFR = 0 ; buffer number must be > 0
  SET   CBFR = 1
ENDIF
SET    B = CBFR ; calculate machine number
IF     CMA = BUFMA[B] ; if continue with the same machine only
      ; update information without moving the template
  SET   MTASK[CBFR] = CTASK    ; new task number
  SET   MSUBP[CBFR] = MSUBP[CBFR] + 1 ; new subprocess
  SET   MFLQ[CBFR] = 1
  SET   CMA = 0    ; reset the continue command
  SET   CTASK = 0
  SET   CBFR = 0
  SET   FFIN = 0  ; reset finish flag (can work with next one)
ELSE   ; continue with other machine (move the template)
  SET   B = MAXBF + 1 ; search free buffer on new machine
  LABEL 1
  SET   B = B - 1
  IF    B = 0
    *save for later    ; can't find free buffer
  ENDIF
  SET   I = B ; calculate absolute buffer number
  IF    BUFMA[I] <> CMA
    CRIF MFLQ[I] > 0    ; =0 when buffer is empty
    GOTO 1
  ENDIF
PEND   B FROM ROBOT    ; wait for robot to be available
      ; update information in new buffer
SET    MTASK[I] = CTASK    ; task number
SET    MIMPL[I] = MIMPL[CBFR] ; template number
SET    MPART[I] = MPART[CBFR] ; part number
SET    MRAV[I] = MRAV[CBFR] ; raw material number
SET    MRAW[I] = MRAW[CBFR] ; raw material number (assembly)
SET    MSUBP[I] = MSUBP[CBFR] + 1 ; subprocess
SET    MNFC[I] = MNFC[CBFR] ; part information
SET    MFLQ[I] = 1
SET    MTASK[CBFR] = 0    ; reset old buffer information
SET    MIMPL[CBFR] = 0
SET    MPART[CBFR] = 0
SET    MRAV[CBFR] = 0
SET    MRAW[CBFR] = 0

```

```

SET     MSUBP[CBFR] = 0
SET     M N F Q [CBFR] = 0
SET     BFR = CBFR           ; old buffer number
SET     CBFR1 = CBFR
SET     CBFR2 = 1
SET     CFLAG = 1
SET     TVF = 1             ; get template from old buffer
RUN     TF
WAIT    TVF = 0             ; wait to finish getting template
WAIT    CFLAG = 0
SET     CFLAG = 3
SET     MFLQ[CBFR] = 0
SET     BFR = 1             ; new buffer number
SET     TVF = 2             ; put template on new buffer number
RUN     TT
WAIT    TVF = 0             ; wait to finish putting template on buffer
SET     CBFR = 0
SET     CTASK = 0
SET     CVA = 0             ; reset the continue command

WAIT    CFLAG = 0
SET     CFLAG = 5
WAIT    CFLAG = 0

POST    1 TO ROBOT          ; release robot for other program
SET     FFIN = 0            ; reset finish flag (can work with next one)
ENDIF

```

END

```
# <<<<<  WTCP  >>>>>
```

```

PROGRAM VB
GOSUB  FROMB           ; get template from buffer
SET    WPF = 0
END

```

```

PROGRAM VP
GOSUB  TOPLT           ; put template on pallet
SET    WPF = 0
END

```

```
# move template from machine buffer to pallet
```

```

PROGRAM WTCP
DEFINE B

```

```

IF     FFIN = 3         ; unloading
  IF     ASRS = 0       ; in non ASRS station send the empty template
                        ; to ASRS station
    SET   MPART[MBFR] = 0 ; reset template contents information
    SET   MRAV[MBFR] = 0 ; to empty
    SET   MRAVW[MBFR] = 0
    SET   MSUBP[MBFR] = 0
    SET   M N F Q [MBFR] = 0
  ELSE
    SET   FFIN = 0     ; finish part in place
    SET   MFLQ[MBFR] = 0
    GOTO  9
  ENDIF
ENDIF
ENDIF

```

```

SET     NEMP = NEMP + 1 ; need one more empty pallet
WAIT    FEMP > 0       ; FEMP > 0 when empty pallet in station

```

```

PEND      B FROM ROBOT          ; wait for robot to be available

SET       OSUBP = MSUBP[ MBFR]
SET       CINFO = M NFC[ MBFR] ; part information
SET       CBFR = MBFR
SET       FONP = 1              ; send 'O' to center (putting on pallet)

SET       BFR = MBFR           ; buffer number
SET       WPF = 1              ; get template from buffer
RUN       VB
WAIT      WPF = 0              ; wait to finish getting template
WAIT      FONP = 0
SET       FONP = 3
SET       WPF = 2              ; put template on pallet
RUN       VP
WAIT      WPF = 0              ; wait to finish putting template on buffer
WAIT      FONP = 0
SET       FONP = 5
WAIT      FONP = 0

SET       MTASK[ MBFR] = 0      ; reset buffer contents information
SET       MPART[ MBFR] = 0      ; to empty buffer
SET       MRAW[ MBFR] = 0
SET       MRAV[ MBFR] = 0
SET       MSUBP[ MBFR] = 0
SET       MITPL[ MBFR] = 0
SET       M NFC[ MBFR] = 0
SET       MFLC[ MBFR] = 0
SET       FFIN = 0             ; finish last 'finish task'

POST      1 TO ROBOT           ; release robot for other program

LABEL     9
END

# <<<<  RESET  >>>>

#procedure called by RESET and AUTO (at stations)
PROGRAM $STRT /N
POST      1 TO ROBOT           ; enable robot work
GOSUB     START
END

#clear all buffers and flags
PROGRAM RESET /Y
DEFINE    I                    ; for loop on buffers and machines

CLRROOM  0                    ; clear communication ports
STOP      MACH                 ; stop only when reset then run again
STOP      RELES                 ; releasing raw material
STOP      PTOV                 ; from pallet to work
STOP      TOV                   ; continue work at the same station
STOP      VTOP                 ; move template from station to pallet
STOP      TF                    ; template from buffer (to other buffer)
STOP      TT                    ; template to buffer (from other buffer)
STOP      PP                    ; template from pallet (to buffer)
STOP      PB                    ; template to buffer (from pallet)
STOP      VB                    ; template from buffer (to pallet)
STOP      VP                    ; template to pallet (from buffer)
STOP      $MFB                 ; part from buffer (to machine)
STOP      $MTM                 ; part to machine (from buffer)
STOP      $MFM                 ; part from machine (to buffer)

```

```

STOP $MTB           ;part to buffer (from machine)
STOP $VTM          ;exec VTM1 or VTM2
STOP $TM1          ;start machine 1 and wait for it to stop
STOP $TM2          ;start machine 2 and wait for it to stop
STOP $TM3          ;start machine 3 and wait for it to stop
STOP $TM4          ;start machine 4 and wait for it to stop
STOP $TSK1
STOP $TSK2
STOP $TSK3
STOP $TSK4
DELAY 10
CLRBUF             ;stop any robot movement
DELAY 10

FOR I = 1 TO 8     ;clear buffer information
  SET MTASK[I] = 0 ;task number
  SET MIMPL[I] = 0 ;template number
  SET MPART[I] = 0 ;part number
  SET MRAW[I] = 0 ;raw material number
  SET MRAWN[I] = 0 ;raw material number (assembly)
  SET MSUBP[I] = 0 ;subprocess
  SET MNFC[I] = 0 ;information about part
  SET MFLQ[I] = 0 ;empty buffer
ENDFOR

FOR I = 1 TO 4     ;clear machine information
  SET MBUSY[I] = 0 ;buffer number of part that machine is working on
  SET MFIN[I] = 0 ;flag =1 when machine is finished
  SET MLASTT[I] = 0 ;last task performed by machine
  SET MFTASK[I] = 0 ;=1 when waiting for machine loading
ENDFOR

SET FEMP = 0       ;empty pallet arrived
SET NEMP = 0       ;number of empty pallets needed
SET AFLAG = 0      ;send 'A' / 'D'
SET ATFLG = 0      ;send 'a' 't' 'f' 'e' 'n'
SET FIGN = 0       ;send 'I' (ignore)
SET FWRK = 0       ;send 'W' (getting part from pallet to work)
SET FONP = 0       ;send 'O' (putting part on pallet)
SET FNEW = 0       ;send 'N' (new part on buffer)
SET LFLAG = 0      ;send 'L' (load part into machine)
SET UFLAG = 0      ;send 'U' (unloading part from machine)
SET CFLAG = 0      ;send 'C' (move template from buffer to buffer)
SET MFL = 0        ;send 'M' 'm' (machine work)
SET JFLAG = 0      ;send 'J' 'j'
SET KFLAG = 0      ;send 'K' 'k'
SET HFLAG = 0      ;send 'H' (assembly)
SET FTMP = 0       ;send 'T' (need template)
SET XFLAG = 0      ;send 'X' 'x'
SET FPLT = 0       ;pallet arrived
SET FFIN = 0       ;finish flag
                   ; flag > 0 if:
SET PVF = 0        ;robot moved template from pallet to buffer
SET VPF = 0        ;robot moved template from buffer to pallet
SET MF = 0         ;robot moved part between machine and buffer
SET TVF = 0        ;robot moved template from buffer to buffer

SET FREL = 0       ;release flag
SET RTMPL = 1001   ;default template number
SET RPART = 0
SET RRAW = 0
SET RRAWN = 0
SET PART = 0
SET RAW = 0

```



```

SET    RAWN = 0
SET    FV = 0
SET    NVCBK = MAXBF; number of free buffers
#SET   CUT[7] = 1 ; signal to central controller
SET    YMA = 0 ; number of machine to process part taken from pallet
SET    YTMPL = 0 ; template to get from pallet
SET    CMA = 0 ; number of machine to continue work
SET    SBUSY = 0 ; =1 when robot is busy (don't stop pallets)
SET    MBFR = 0
SET    ASMBF = 0 ; can assemble next part
SET    UTASK = 1 ; can loading CNC program

```

```

@CON
DELAY 40

```

```

RUN    MOVEP ; release pallet
GOSUB $STRT
RUN    MACH ; continue running after reset is finished
GOSUB URST

```

```

SET    FRST = 1 ; send 'S' reset o.k.

```

```

END

```

```

# <<<<<< COM >>>>>>

```

```

#waiting for characters to be received on the rs232
PROGRAM WAIT1
LABEL 1
WAIT   GCOM[1] = 0 ; after getting one command wait for
; command to finish being executed
GETCOM 1, GCOM[1] ; get the command
IF     PRNT = 1 ; in DEBUG mode
  PRINTLN "$ in : " GCOM[1]
ENDIF
GOTO 1 ; for the next
END

```

```

# communicate with rs232

```

```

PROGRAM COM1

```

```

STOP   WAIT1 ; if already running, don't run twice
DELAY 20

```

```

SET    GCOM[1] = 0 ; signal to WAIT1 to get next command
RUN    WAIT1 ; program waiting for command on rs232
; from the central controller

```

```

LABEL 1
IF     GCOM[1] > 0 ; command has arrived

```

```

IF     GCOM[1] = 'A' ; check
  SENDCOM 1, 'B' ; if central controller receives the 'B'
; then communication is o.k.
ENDIF

```

```

IF     GCOM[1] = 'C' ; part continues work at same station
; get information on part
  READCOM 1, CMA ; machine number
  READCOM 1, CTASK ; task number
  READCOM 1, CBFR ; buffer number
  IF     MSUBP[CBFR] = 0 ; if continue is received immediately

```

```

                                ; after release
ANDIF FREL = 1
SET FREL = 3 ; send 'V' "release finished o.k."
ENDIF
RUN TOW ; continue to next subprocess
ENDIF

IF GCOM1] = 'E' ; pallet empty
IF NEMP > 0 ; only if empty pallet is needed
RUN STOPP ; stop the pallet (if not already stopped)
SET FEMP = FEMP + 1 ; flag 'empty pallet here'
SET NEMP = NEMP - 1 ; one of empty pallets needed came
ELSE
SET FIGN = 1 ; ignore if empty pallet is not needed
ENDIF
ENDIF

IF GCOM1] = 'F' ; part completely finished
READCOM 1, MBFR
SET FFIN = 3 ; indicate "part will not return to conveyor"
RUN WTCP 8 ; if not ASRS station, return the template
; to conveyor
ENDIF

IF GCOM1] = 'K'
SET FTMP = 4
ENDIF

IF GCOM1] = 'M' ; part to conveyor
READCOM 1, MBFR
RUN WTCP 8
ENDIF

IF GCOM1] = 'N' ; part on pallet is for another station
SET FIGN = 1 ; ignore (only release the pallet)
ENDIF

IF GCOM1] = 'R' ; release raw material for requested part
READCOM 1, RPART ; part to release
READCOM 1, RRAW ; raw material for this part
READCOM 1, RRAWN ; raw material number for this part
RUN RELES ; release the raw material for the part
ENDIF

IF GCOM1] = 'S' ; reset
GOSUB RESET
ENDIF

IF GCOM1] = 'V'
SET FTMP = 3
ENDIF

IF GCOM1] = 'W'
READCOM 1, APART

```

```

GOSUB ASMB
ENDIF

IF GCOM1] = 'Y' ; part on pallet for this station
  RUN STOPP ; stop the pallet to get the part
                ; get information on part
  READCOM 1, YMA ; machine to work with
  READCOM 1, YTASK ; task number to do
  READCOM 1, YTMPL ; template number
  READCOM 1, YPART ; part number
  READCOM 1, YRAW ; raw material
  READCOM 1, YRAWN ; raw material number
  READCOM 1, YSUBP ; current subprocess
  READCOM 1, YINFO ; part information
  RUN PTOW ; get template from pallet and put on buffer
ENDIF

SET GCOM1] = 0 ; ready for next
ENDIF

IF AFLAG > 0
  IF AFLAG = 1
    SENCOM 1, 'A'
  ELSE
    SENCOM 1, 'D'
  ENDIF
  SET AFLAG = 0
ENDIF

IF ATFLAG > 0
  IF ATFLAG = 1
    SENCOM 1, 'a'
  ENDIF
  IF ATFLAG = 3
    SENCOM 1, 't'
  ENDIF
  IF ATFLAG = 5
    SENCOM 1, 'f'
  ENDIF
  IF ATFLAG = 7
    SENCOM 1, 'n'
  ENDIF
  SET ATFLAG = 0
ENDIF

IF CFLAG > 0 ; move part between buffers
  IF CFLAG = 1
    SENCOM 1, 'C'
    PRLNCOM 1, CBFR1 ; buffer
    PRLNCOM 1, CBFR2 ; target buffer
    PRLNCOM 1, MNFQ[CBFR2]
  ELSE
    SENCOM 1, 'c'
  ENDIF
  SET CFLAG = 0
ENDIF

IF FF1N = 1 ; finish work
  SENCOM 1, 'F'

```

```

PRLNCOM 1, FBFR ; bu f er
SET     FF I N = 2
END I F

I F     HFLAG > 0
  I F     HFLAG = 1
    SENCOM 1, 'H'
  ELSE
    SENCOM 1, 'h'
  END I F
SET     HFLAG = 0
END I F

I F     FI GN > 0 ; i gn ore
  SENCOM 1, 'I'
  PRLNCOM 1, PALET
  RUN     MOVEP ; rel ease the pal let
  SET     FI GN = 0
END I F

I F     JFLAG > 0
  I F     JFLAG = 1
    SENCOM 1, 'J'
  ELSE
    SENCOM 1, 'j'
  END I F
  PRLNCOM 1, JMACH
  SET     JFLAG = 0
END I F

I F     KFLAG > 0
  I F     KFLAG = 1
    SENCOM 1, 'K'
    PRLNCOM 1, KPRT
  ELSE
    SENCOM 1, 'k'
  END I F
SET     KFLAG = 0
END I F

I F     LFLAG > 0 ; l oadi ng par t
  I F     LTASK > 900
    I F     LFLAG = 1
      SENCOM 1, 'Z'
      PRLNCOM 1, LBFR ; bu f er
    ELSE
      SENCOM 1, 'z'
    END I F
  ELSE
    I F     LFLAG = 1
      SENCOM 1, 'L'
      PRLNCOM 1, LBFR ; bu f er
    ELSE
      SENCOM 1, 'l'
    END I F
  END I F
SET     LFLAG = 0
END I F

```

```

IF      MFL > 0
  IF      MFL = 1
    SENCOM 1, 'M
  ELSE
    SENCOM 1, 'm'
  ENDIF
  PRLNCOM 1, MMACH
  SET     MFL = 0
ENDIF

```

```

IF      FNEW > 0      ; new part
  SENCOM 1, 'N
  PRLNCOM 1, FNEW      ; buffer
  PRLNCOM 1, MIMPL[FNEW] ; template
  PRLNCOM 1, MPART[FNEW] ; part
  PRLNCOM 1, MRAW[FNEW] ; raw
  PRLNCOM 1, MRAWN[FNEW] ; raw number
  SET     FNEW = 0
ENDIF

```

```

IF      FONP > 0      ; put part on empty pallet
  IF      FONP = 1
    SENCOM 1, 'O      ; information
    PRLNCOM 1, PALET      ; pallet
    PRLNCOM 1, CBFR      ; subprocess
    PRLNCOM 1, CINFO      ; part information
  ELSE
    SENCOM 1, 'o'
  ENDIF
  IF      FONP = 5
    RUN     MOVEP      ; release the pallet
    IF      OSUBP = 0      ; subprocess 0 is the released part
      ANDIF  FREL = 1
      SET     FREL = 3      ; send 'V' to center "end of release"
    ENDIF
    IF      FEMP > 0      ; one empty pallet loaded and released
      SET     FEMP = FEMP - 1
    ENDIF
    SET     NWORK = NWORK + 1      ; one more buffer free
  ENDIF
  SET     FONP = 0
ENDIF

```

```

IF      FPLT = 1      ; pallet arrived
  IF      IN[7] = 0      ; if station is not working
    RUN     MOVEP      ; free the pallet
  ELSE
    IF      NEMP = 0      ; if station does not need an empty pallet
      ANDIF  NWORK = 0      ; and no places are available to work
      CRIF  SBUSY = 1      ; or the robot is busy
      SENCOM 1, 'P      ; then only report to central controller
      PRLNCOM 1, PALET
      RUN     MOVEP
    ELSE
      SENCOM 1, 'Q      ; ask central controller: what is on pallet ?
      PRLNCOM 1, PALET
    ENDIF
  ENDIF
  SET     FPLT = 0      ; pallet checked

```

```

ENDIF

IF      FRST = 1          ; after reset
  SENCOM 1, 'S'
  SET    FRST = 0
ENDIF

IF      FTMP = 1
  SENCOM 1, 'T'
  PRLNCOM 1, TPART
  SET    FTMP = 2
ENDIF

IF      UFLAG > 0       ; unloading part
  IF      UFLAG = 1
    SENCOM 1, 'U'
    PRLNCOM 1, UBFR      ; buffer
  ELSE
    SENCOM 1, 'u'
  ENDIF
  SET    UFLAG = 0
ENDIF

IF      FREL > 1        ; after receiving release command
  IF      FREL = 3
    SENCOM 1, 'V'        ; o.k.
  ELSE
    SENCOM 1, 'E'        ; missing
  ENDIF
  SET    FREL = 0
ENDIF

IF      FVRK > 0        ; after getting part from pallet
  IF      FVRK < 10
    IF      FVRK = 1
      SENCOM 1, 'W'
      PRLNCOM 1, PALET
      PRLNCOM 1, VBFR
    ELSE
      SENCOM 1, 'w'
    ENDIF
  ELSE
    IF      FVRK = 11
      SENCOM 1, 'Y'
      PRLNCOM 1, PALET
    ELSE
      SENCOM 1, 'y'
    ENDIF
  ENDIF
  IF      FVRK = 3
    CRIF   FVRK = 13
    RUN    MOVEP
  ENDIF
  SET    FVRK = 0
ENDIF

IF      XFLAG > 0
  IF      XFLAG = 1

```

```

        SENCOM 1, 'X'
    ELSE
        SENCOM 1, 'x'
    ENDF
    PRLNCOM 1, XMACH
    SET     XFLAG = 0
ENDF

```

```

DELAY 20
SET    COUT[7] = IN[7]

GOTO 1
END

```

```

# <<<< MAIN >>>>

```

```

# put machine number for each buffer
PROGRAM INTB
DEFINE MNI

```

```

SET     N = 1
FOR     M = 1 TO 4
    IF     MBMAX[M] > 0
        FOR     I = 1 TO MBMAX[M]
            SET     BUFMA[N] = M
            SET     N = N + 1
            IF     N > 8
                GOTO 99
            ENDF
        ENDFOR
    ENDF
ENDFOR

```

```

LABEL 99

```

```

END

```

```

#program runs automatically when ever the controller is turned on
PROGRAM CIM /Y

```

```

IF     BUFMA[1] = 0
    GOSUB INTB
ENDF

```

```

GOSUB $STRT ;enable robot to work

```

```

CLRCOM 0 ;clear communication ports
@CON ;control on

```

```

DELAY 40
GOSUB CIMP ;attach CIM positions
DELAY 30

```

```

STOP PLTS ;pallet tracking
STOP FINIS ;machine finish tracking
STOP COMI ;communication with central controller
STOP MACH ;load and unload part to/from machines
STOP ASMB1 ;assembly program
STOP $REPM ;report machine
STOP VAI T2 ;com2 using to autonc

```

```

RUN    PLTS           ; pallet tracking
RUN    FINIS         ; machine finish tracking
RUN    COMI          ; communication with central controller
RUN    MACH          ; load and unload part to/from machines
RUN    ASMB1         ; assembly program
RUN    $REPM         ; report machine
IF     ASRS = 0
  RUN  WAIT2         ; com 2 using to autocnc
ENDIF

WAIT   1 = 2         ; HALT, prevents program from being run twice

END

```

```

PROGRAM AUTO
DELAY  40
GOSUB  CIMP          ; attach CIM positions
DELAY  30
GOSUB  RESET
DELAY  20
STOP   MACH
END
#      <<<<< BFSTAT >>>>>

```

```

PROGRAM RBFR
DEFINE MAXM
DEFINE I
DEFINE J
DEFINE B
DEFINE B2
DEFINE FB TB

```

```

DELAY  30
SET    MAXM = MAXBF / 2

```

```

FOR    I = 1 TO MAXM
  SET   B = I - 1
  SET   B = B * 2
  SET   B2 = MBUSY[I]
  SET   B = B + B2
  PRINTLN "MACHINE " I
  IF    B2 = 0
    PRINT " EMPTY"
  ELSE
    PRINT " CONTAINS PART " MPART[B]
    PRINT " , FROM BUFFER " B
    PRINT " , EXECUTE TASK " MTASK[B]
    IF   MFIN[I] > 0
      PRINT " >>> FIN SHED"
    ENDIF
  ENDIF
ENDIF

```

```

SET    FB = I - 1      ; buffers of machine
SET    FB = FB * 2
SET    FB = FB + 1
SET    TB = FB + 1

```

```

PRINTLN " +-----+-----+-----+-----"
PRINT   "+-----+-----+-----+-----+"
PRINTLN "! BUFFER! TEMPL! PART ! RAW !"
PRINT   "! SUBPR! INFO ! TASK ! REM !"

```



```

PRI NTLN " +-----+-----+-----+-----"
PRI NT " +-----+-----+-----+-----+"

FOR J = FB TO TB
  PRI NTLN " !" J " !" MIMPL[ J]
  PRI NT " !" MPART[ J] " !" MRAW[ J]
  PRI NT " !" MSUBP[ J] " !" M NFO[ J]
  PRI NT " !" MTASK[ J] " !"
  IF MFLQ[ J] = 0
    PRI NT " EMPTY "
  ENDIF
  IF MFLQ[ J] = 1
    PRI NT " WAITING "
  ENDIF
  IF MFLQ[ J] = 2
    PRI NT " WORKING "
  ENDIF
  IF MFLQ[ J] = 3
    PRI NT " FINISH "
  ENDIF
  PRI NT " !"
ENDFOR
PRI NTLN " +-----+-----+-----+-----"
PRI NT " +-----+-----+-----+-----+"

ENDIF
IF PALET > 0
  PRI NTLN " PALLET #" PALET " IN STATION"
  IF YMA > 0
    PRI NTLN " CONTAINS TEMPLATE " YTMPL
    PRI NT " PART " YPART
    PRI NT " RAW MATERIAL " YRAW
    PRI NT " SUBPROCESS " YSUBP
    PRI NTLN " FOR MACHINE " YMA
    PRI NT " TASK " YTASK
    PRI NT " INFO = " YINFO
  ENDIF
ENDIF

IF MF = 1
  PRI NTLN " ROBOT GETS PART FROM BUFFER" BFR
ENDIF

IF MF = 2
  PRI NTLN " ROBOT LOADS PART TO MACHINE" MACH
ENDIF

IF MF = 3
  PRI NTLN " ROBOT GETS PART FROM MACHINE" MACH
ENDIF

IF MF = 4
  PRI NTLN " ROBOT PUTS PART ON BUFFER" BFR
ENDIF

IF PWF = 1
  PRI NTLN " ROBOT GETS TEMPLATE FROM PALLET"
ENDIF

IF PWF = 2
  PRI NTLN " ROBOT PUTS TEMPLATE ON BUFFER" BFR
ENDIF

IF TVF = 1
  PRI NTLN " ROBOT GETS TEMPLATE FROM BUFFER" BFR

```

```

ENDIF

IF TWF = 2
  PRINTLN "ROBOT PUTS TEMPLATE ON BUFFER" BFR
ENDIF

IF VPF = 1
  PRINTLN "ROBOT GETS TEMPLATE FROM BUFFER" BFR
ENDIF

IF VPF = 2
  PRINTLN "ROBOT PUTS TEMPLATE ON PALLET"
ENDIF

PRINTLN ">"

```

```
END
```

```

PROGRAM CBFR
DEFINE B BF BT
DEFINE C
DEFINE M
DEFINE MAXM

```

```

DELAY 30
SET MAXM = MAXBF / 2

```

```

LABEL 1
PRINTLN
PRINTLN "      +-----+"
PRINTLN "      ! M - MACH NE CHANGES !"
PRINTLN "      ! B - BUFFER CHANGES !"
PRINTLN "      ! S - STATUS          !"
PRINTLN "      ! X - EXIT TO ACL    !"
PRINTLN "      +-----+"
PRINTLN " COMMAND ( M B/ S/ X ) : "
GET C

```

```

IF C = 'S'
  ORIF C = 's'
  PRINT "S"
  GOSUB RBFR
  PRINTLN " PRESS ANY KEY .... "
  GET C
  GOTO 1
ENDIF

```

```

IF C = 'M'
  ORIF C = 'm'
  PRINT "M"
  IF MAXM > 1
    PRINTLN " MACH NE NUMBER ( 1 - " MAXM " ) : "
    READ M
    IF M > MAXM
      ORIF M < 0
        PRINTLN " ONLY MACH NE NUMBERS 1 - " MAXM
        PRINT " ARE ALLOWED "
        GOTO 1
      ENDIF
    IF M = 0
      GOTO 1
    ENDIF
  ELSE

```

```

      SET      M = 1
    ENDIF
    SET      B = MBUSY[M]
    IF      B = 0
      PRINTLN "MACHINE IS NOW EMPTY "
      PRINTLN "ENTER BUFFER NUMBER ("
      SET      BF = M - 1
      SET      BF = BF * 2
      SET      BF = BF + 1
      SET      BT = BF + 1
      PRINT   BF " - " BT " ) : "
      READ    B
      IF      B < BF
        ORIF  B > BT
          PRINTLN "ONLY BUFFER NUMBERS " BF " - "
          PRINT   BT " ARE ALLOWED"
          GOTO  1
        ENDIF
      SET      MBUSY [ M ] = B
      PRINTLN "Done. "
      PRINTLN
    ELSE
      IF      MFIN[M] > 0
        PRINTLN "MACHINE HAS FINISHED, "
        PRINT   "CAN NOT CHANGE MANUALLY"
        GOTO  1
      ENDIF
      PRINTLN "MACHINE CONTAINS PART FROM BUFFER "
      PRINT   B " AND IS WORKING ON IT"
      PRINTLN "YOU CAN TAG IT AS FINISHED OR EMPTY"
      PRINTLN "TAG IT AS FINISHED (Y/N) ? (N)"
      GET     C
      IF      C = 'Y'
        ORIF  C = 'y'
          PRINT   "Y"
          SET      MFIN[M] = 1
          PRINTLN "Done. "
          PRINTLN
          GOTO  1
        ENDIF
      PRINTLN "TAG IT AS EMPTY (Y/N) ? (N)"
      GET     C
      IF      C = 'Y'
        ORIF  C = 'y'
          PRINT   "Y"
          SET      MBUSY[M] = 0
          PRINTLN "Done. "
          PRINTLN
        ENDIF
      ENDIF
    ENDIF
    SET      C = 'M'
  ENDIF

  IF      C = 'B'
    ORIF  C = 'b'
    PRINT   "B"
    PRINTLN "BUFFER NUMBER (1 - " MAXBF " ) : "
    READ    B
    IF      B > MAXBF
      ORIF  B < 0
        PRINTLN "ONLY BUFFER NUMBERS 1 - " MAXBF
        PRINT   " ARE ALLOWED"
        GOTO  1
      ENDIF
    ENDIF
  
```

```

IF      B = 0
  GOTO  1
ENDIF

PRI NTLN "TEMPLATE VALUE IS " MTMPL[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'
  PRINT "Y"
  PRI NTLN "NEW TEMPLATE NUMBER : "
  READ  C
  SET   MTMPL[B] = C
ENDIF

PRI NTLN "PART VALUE IS " MPART[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'
  PRINT "Y"
  PRI NTLN "NEW PART NUMBER : "
  READ  C
  SET   MPART[B] = C
ENDIF

PRI NTLN "RAW MATERIAL VALUE IS " MRAV[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'
  PRINT "Y"
  PRI NTLN "NEW RAW MATERIAL NUMBER : "
  READ  C
  SET   MRAV[B] = C
ENDIF

PRI NTLN "SUBPROCESS NUMBER IS " MSUBP[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'
  PRINT "Y"
  PRI NTLN "NEW SUBPROCESS NUMBER : "
  READ  C
  SET   MSUBP[B] = C
ENDIF

PRI NTLN "TASK NUMBER IS " MTASK[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'
  PRINT "Y"
  PRI NTLN "NEW TASK NUMBER : "
  READ  C
  SET   MTASK[B] = C
ENDIF

PRI NTLN "INFO VALUE IS " MNFC[B]
PRINT  " CHANGE (Y/N) ? (N)"
GET    C
IF     C = 'Y'
  ORIF C = 'y'

```

```

    PRINT "Y"
    PRINTLN "NEW INFO VALUE : "
    READ C
    SET MFLQBJ = C
ENDIF

PRINTLN "STATUS IS "
IF MFLQBJ = 0
    PRINT "EMPTY"
ENDIF
IF MFLQBJ = 1
    PRINT "WAITING"
ENDIF
IF MFLQBJ = 2
    PRINT "WORKING"
ENDIF
IF MFLQBJ = 3
    PRINT "FINISH"
ENDIF
PRINT "CHANGE (Y/N) ? (N)"
GET C
IF C = 'Y'
    ORIF C = 'y'
    PRINT "Y"
    PRINTLN "NEW STATUS (0: EMPTY, 1: "
    PRINT "WAITING 2: WORKING 3: FINISH) : "
    READ C
    SET MFLQBJ = C
ENDIF

#MFLAG

    SET C = 'B'

ENDIF

IF C = 'X'
    ORIF C = 'x'
    PRINT "X"
    GOTO 9
ENDIF

GOTO 1

LABEL 9
PRINTLN
PRINTLN ">"
END
# <<<<< SCORDEF >>>>>

# scorbse definition
DI MPA SCORA[464]
DI MPB SCORB[464]
# <<<<< SCORDEFA >>>>>

# scorbse definition (not using group B)
DI MPA SCORA[464]

```

ASRS Station Program Listing

```
;*** ASRS-ACL version 3.2
;*** Last changed: 24/05/92.
;*** Update by Shmulik :7/8/92
;*** Last update :7/9/92

;*** Note: these ACL routines for the ASRS should be run with ASRS.EXE
;*** version 2.30.

#include asrs.dmc

DIMB SRSP[18] ;The circular points of the ASRS,
; SRSP[1] for position of cell 1,
; offset for each cell is 1706.66 encoder pulses.
; Program MARS creates all positions (relative to SRSP[1])

;--- DIMPA CIM[500]

; CIM] (Group A (Robot) points)
; -----
; Pallet conveyor:
; 9=on pallet, 19=above 9
;
; Barcode scanner:
; 10=above buffer1 in front of scanner
; 8=a few centimeters below point 10
;
; Buffers:
; For a template:
; 1=on buffer 1, 11=above 1
; For a part (on a template which is on the buffer):
; Parts are grouped into 100 families of 10 parts each
; Parts 1-10, 11-20, ... , 991-1000. Each family has
; 2 points: one above and one on a template. For part
; number n (1<=n<=1000) the relevant points are: CIM]p1]
; (above the template) and CIM]p2] (on the template),
; where p1=201+int((n-1)/10), p2=p1+100 where int(x) is
; the integer part of x (e.g int(3/2)=1, int(4/5)=0).
; for example:
; Parts 1-10:
; p1=201=Above template,
; p2=301=On template;
; Parts 11-20:
; p1=202=Above template,
; p2=302=On template;
; etc.
;
;
; ASRS-Robot :
; In the following points the right digit is the level
; number (e.g. point 82 refers to level 2).
; 31,32,33=aside ASRS;
; 41,42,43=in front of ASRS;
; 51,52,53=inside ASRS ABOVE template;
; 61,62,63=inside ASRS ON template;
; 71,72,73=in front of ASRS for feeder;
; 81,82,83=inside of ASRS for feeder IN the feeder;
; 91,92,93=inside of ASRS for feeder ABOVE the feeder;
```

```

GLOBAL SPDB ;The speed of the carousel (Group B speed)
SET SPDB = 17;Default value is 17

GLOBAL A ;Used to abort manual operations, 0=Continue, 1=Abort
GLOBAL OUTPL ;Used in manual operations, 1=buff1, 2=buff2,
; ,3=conv <- not used.

GLOBAL TMP ;Used in manual operations, the # of part or template
GLOBAL GTERR ;Used in manual operations, Result of retrieval:
; 0=Ck, 1=Nb part, -1=Nb template for the part.

DIMG SRS[54] ;The contents of each cell in the ASRS,
; The contents of cell C in level L is SRS[18*(L-1)+C]
; Each entry is in this format LL+1024*HH
; LL should satisfy 0<LL<1000
; The different objects are coded as follows:
; Empty cell : LL=HH=0
; Empty template : HH=template #, LL=0
; Unknown : HH=100, LL=0
; Part on a template: LL=Part ID, HH=0 (Or HH<=1000)
; Feeder : LL=Part ID, HH=2000+feeder_quantity

DIMG SRSI[54] ; Index for searching in SRS[] array

DIMG TMTT[270] ;Template to part matching. Each line is made of
; 9 numbers: Temp_Num P1, P2, ..., P8 which means that
; part Pi can fit on template # Temp_Num
DIMG PRTTM[1000] ;Part to template matching
; PRTTM[i] is the template # matching part # i

GLOBAL NCHG ;# of changes since last update (read by program 'U)
DIMG CHG1[8] ;CHG1[] & CHG2[] record the changes, CHG1[i] is the cell
DIMG CHG2[8] ; number that was changed, CHG2[i] is the new content.
GLOBAL LOCK ; For PC check of locking (i.e. ASRS is in use)
GLOBAL PLOCK ; For ACL sync of locking

;---GLOBAL RRAW ; # of part to be fetched (used in ISVLD) ***
GLOBAL CELL ; # of cell where a part 'RRAW was found on a template
GLOBAL FDOEL ; # of cell where a part 'RRAW was found in a feeder
GLOBAL TCELL ; # of cell where a template for 'RRAW was found
;---GLOBAL FV ; In use in program ISVLD (FV=0 not found, FV<>0 found) ***
;---GLOBAL RTMPL ; # of template type ***

GLOBAL MCELL ; # of cell to be moved from or to ASRS
GLOBAL COONT ; Cell contents in coded (SRS[]) format
GLOBAL BCODE ; Value of last read barcode # on a template
GLOBAL CKOOD ; =1 If barcode scanned ok, =0 otherwise.

GLOBAL MSS ; Status of ASRS for PC tracking:
; 10=Retrieval of part, 20=Retrieval of empty template,
; 30=Part can't be retrieve,
; 40=Template can't be retrieve,
; 50=Storage of part,
; 60=Storage of empty template,
; 80=Taking part from buffer to conveyor
; 90=Taking an empty template from buffer to conveyer

GLOBAL MSP ; # of a part for PC tracking
GLOBAL MST ; # of template for PC tracking
GLOBAL MVP ; # of position to move to (used in program MV)
GLOBAL BARC ; Is barcode scanner active? (0=Nb, 1=Yes)
GLOBAL BCK ; Does the barcode match (0=Nb, 1=Yes)
GLOBAL BTMP ; Template type # for barcode match

```

```

SET    LOCK = 0 ; Init'ze for 1st loading
SET    PLOCK = 1 ; " " " "
SET    ROBOT = 1 ; " " " "
SET    BARC = 0 ; " " " "

```

```

;***** Various output routines *****

```

```

PROGRAM LON ; Turns on the yellow light on the ASRS

```

```

SET    OUT[ 3] = 1
SET    OUT[ 4] = 1

```

```

END

```

```

PROGRAM LOFF ; Turns off the yellow light on the ASRS

```

```

SET    OUT[ 3] = 0
SET    OUT[ 4] = 0

```

```

END

```

```

PROGRAM MASP9 ; Move to SRSP[ 9] with lights on

```

```

GOSUB  LON
MOVED  SRSP[ 9]
GOSUB  LOFF

```

```

END

```

```

;***** Hbrng *****

```

```

PROGRAM STA / Y ; Hbrng routine

```

```

SET    LOCK = 0
SET    PLOCK = 1
SET    ROBOT = 1
HOME
GOSUB  LON
HOME  7
GOSUB  LOFF
SPEEDB SPDB
GOSUB  MASP9

```

```

END

```

```

PROGRAM MV ; Move to SRSP[ MVP] with lights on

```

```

@CON
GOSUB  LON
MOVED  SRSP[ MVP]
GOSUB  LOFF

```

```

END

```



```

PROGRAM NCRET ; When the request for a retrieval of a part is not fulfilled
SET    CUI[ 16] = 1
DELAY  400
SET    CUI[ 16] = 0
END

```

```

PROGRAM NOCOD ; When the barcode # on a template is not correct

```

```

DEFINE I
FOR    I = 1 TO 20
  SET  CUI[ 15] = 1
  DELAY 10
  SET  CUI[ 15] = 0
  DELAY 10
ENDFOR

END

```

```

PROGRAM SRSI ; Creates array SRSI[] (Index for SRSI[])
              ; This will control the order at which the cells are
              ; searched.
; This specific arrangement first searches the 1st and 2nd floors
; and then the 3rd floor

```

```

SET  SRSI [ 1] = 9
SET  SRSI [ 2] = 27

SET  SRSI [ 3] = 10
SET  SRSI [ 4] = 28

SET  SRSI [ 5] = 8
SET  SRSI [ 6] = 26

SET  SRSI [ 7] = 11
SET  SRSI [ 8] = 29

SET  SRSI [ 9] = 7
SET  SRSI [10] = 25

SET  SRSI [11] = 12
SET  SRSI [12] = 30

SET  SRSI [13] = 6
SET  SRSI [14] = 24

SET  SRSI [15] = 13
SET  SRSI [16] = 31

SET  SRSI [17] = 5
SET  SRSI [18] = 23

SET  SRSI [19] = 14
SET  SRSI [20] = 32

SET  SRSI [21] = 4
SET  SRSI [22] = 22

SET  SRSI [23] = 15

```

```

SET   SRSI [ 24] = 33
SET   SRSI [ 25] = 3
SET   SRSI [ 26] = 21

SET   SRSI [ 27] = 16
SET   SRSI [ 28] = 34

SET   SRSI [ 29] = 2
SET   SRSI [ 30] = 20

SET   SRSI [ 31] = 17
SET   SRSI [ 32] = 35

SET   SRSI [ 33] = 1
SET   SRSI [ 34] = 19

SET   SRSI [ 35] = 18
SET   SRSI [ 36] = 36

SET   SRSI [ 37] = 45
SET   SRSI [ 38] = 46
SET   SRSI [ 39] = 44
SET   SRSI [ 40] = 47
SET   SRSI [ 41] = 43
SET   SRSI [ 42] = 48
SET   SRSI [ 43] = 42
SET   SRSI [ 44] = 49
SET   SRSI [ 45] = 41
SET   SRSI [ 46] = 50
SET   SRSI [ 47] = 40
SET   SRSI [ 48] = 51
SET   SRSI [ 49] = 39
SET   SRSI [ 50] = 52
SET   SRSI [ 51] = 38
SET   SRSI [ 52] = 53
SET   SRSI [ 53] = 37
SET   SRSI [ 54] = 54

```

```
END
```

```
;***** 'Low level' Barcode routines *****
```

```
PROGRAM LSON ; Send the scanner command to start scanning,
```

```
DEFINE I ; Should be run at a high priority for timing reasons.
```

```
FOR I = 1 TO 2 ; The sequence 2, 83, ..., 8 is sent twice
```

```
  DELAY 15 ; to be on the safe side
```

```
  SENCOM 2, 2
```

```
  DELAY 5
```

```
  SENCOM 2, 83
```

```
  DELAY 5
```

```
  SENCOM 2, 84
```

```
  DELAY 5
```

```
  SENCOM 2, 73
```

```
  DELAY 5
```

```
  SENCOM 2, 69
```

```
  DELAY 5
```

```
  SENCOM 2, 3
```

```
  DELAY 5
```

```
  SENCOM 2, 8
```

```
  DELAY 30
```

```
ENDFOR
```

```
END
```

```

PROGRAM LSCFF ; Send the scanner command to stop scanning

DEFIN E I
CLROOM 2
FOR I = 1 TO 5 ; The ASCII-ESC char is sent 5 times
  SENCOM 2,27 ; to be on the safe side
  DELAY 20
ENDFOR

END

```

```

PROGRAM RBCOD ; Read the barcode from scanner

```

```

DEFIN E I M X
CLROOM 2
GOSUB LSON ; Start the scanner
SET M = 100000
SET I = 1
SET BCODE = 0
SET CKCOD = 0

LABEL 1
GETCOM 2,X
IF X >= 48
  ANDIF X <= 57 ; Ignore non-digit characters
  SET X = X - 48
  SET X = X * M
  SET BCODE = BCODE + X
  SET M = M / 10
  IF I < 6
    SET I = I + 1
    GOTO 1
  ELSE
    SET CKCOD = 1
    GOTO 2
  ENDF
ENDIF
GOTO 1

LABEL 2
RUN LSCFF

END

```

```

;***** PC <-> Controller programs *****

```

```

PROGRAM PUTSR ; Send SRS[] array to the PC

```

```

DEFIN E I
GET I
PRI NTLN "t d1"
PRI NTLN

FOR I = 1 TO 54
  PRI NTLN SRS[I]
ENDFOR

PRI NTLN "et 1"
PRI NTLN

```

```
SET      NCHNG = 0
```

```
END
```

```
PROGRAM GETSR      ; Get SRS[] array from PC
```

```
DEFINE I  
GET      I  
PRI NTLN "rs3"  
PRI NTLN
```

```
FOR      I = 1 TO 54  
  READ   SRS[I]  
ENDFOR
```

```
PRI NTLN  "ek3"  
PRI NTLN
```

```
END
```

```
PROGRAM GETTP      ; Get TMPTT[] array from PC (and build PRITM[] array)
```

```
DEFINE I J K P  
DEFINE T
```

```
GET      I  
PRI NTLN "rd2"  
PRI NTLN
```

```
FOR      I = 1 TO 270  
  READ   TMPTT[I]  
ENDFOR
```

```
FOR      I = 1 TO 1000  
  SET PRITM[I] = 0  
ENDFOR
```

```
PRI NTLN "er2"  
PRI NTLN
```

```
SET      K = 1      ; Build PRITM table
```

```
FOR      I = 1 TO 30  
  SET     T = TMPTT[K]  
  SET     K = K + 1  
  FOR     J = 1 TO 8  
    SET   P = TMPTT[K]  
    SET   K = K + 1  
    IF    P > 0  
      AND I F P <= 1000  
        SET PRITM[J] = T  
    END I F  
  ENDFOR  
ENDFOR
```

```
END
```

```
PROGRAM PUTTP ; Send TMPTT[] array to the PC
```

```
DEFINE I J K  
GET      I  
PRI NTLN "td2"
```

```

PRI NTLN

SET      K = 1
FOR      I = 1 TO 30
  FOR    J = 1 TO 9
    PRI NT  TMPTT[ K]
    SET    K = K + 1
  ENDFOR
PRI NTLN
ENDFOR
PRI NT    " et 2"
PRI NTLN

END

```

PROGRAM U ; PC calls this program to get the changes in the ASRS

```

DEFINE I
GET I
PRI NTLN "t"

IF      LOCK = 1
  SET   I = NCHNG + MSS
  SET   I = 0 - I
  PRI NTLN I
  PRI NTLN MSP
  PRI NTLN MST
ELSE
  PRI NTLN NCHNG
ENDIF

IF      NCHNG >=1
  ANDIF NCHNG <=8
  FOR   I = 1 TO NCHNG
    PRI NTLN CHG1[I] CHG2[I]
  ENDFOR
  PRI NTLN "e"
ENDIF

PRI NTLN

END

```

;***** Move Robot & ASRS routines *****

PROGRAM TTCBF ; Take a template from ASRS to buffer 1 (physical)

```

DEFINE LCXLV ; If barcode is active then BTMP=Template type # to check
DEFINE TT

SET   X = MCELL - 1
SET   L = X / 18
SET   L = L + 1
SET   C = X MOD 18
SET   C = C + 1

STOP  MSP9
CLRBUF 7
GOSUB LCN

MOVE  SRSPT Q

```

```

SET      LV = L + 30
MOVED   CI M LV]
SET      LV = L + 40
MOVED   CI M LV]
OPEN
MOVED   SRSPI C]
GOSUB   LOFF

SET      LV = L + 50
MOVELD  CI M LV]
SPEEDA  10
SET      LV = L + 60
MOVED   CI M LV]
CLOSE
SET      NCHNG = NCHNG + 1
IF      NCHNG <= 8
    SET   CHG1[ NCHNG] = MCELL
    SET   CHG2[ NCHNG] = 0
ENDIF

SET      SRS[ MCELL] = 0
SET      LV = L + 50
MOVED   CI M LV]
SPEEDA  50
SET      LV = L + 40
MOVELD  CI M LV]
SET      LV = L + 30
MOVED   CI M LV]
.sendat f 9
IF      FDCEL > 0
    SET   C = FDCEL - 1
    SET   C = C MOD 18
    SET   C = C + 1
    GOSUB LON
    MOVE  SRSPI C]
ENDIF

IF      BARC = 1 ; Barcode is active
    RUN   RBCOD
    MOVED CI M 8]
    MOVED CI M 10]
    DELAY 100
    IF    CKCOD = 1
        GOTO 1
    ENDIF
    DELAY 200
    LABEL 1
    STOP  RBCOD
    RUN   LSCFF
    SET   BCK = 0
    IF    CKCOD = 1
        SET   TT = BCODE / 10000
        IF    TT = BTMP
            SET   BCK = 1
        ENDIF
    ENDIF
    IF    BCK = 0
        GOTO 99
    ENDIF
ENDIF

MOVED   CI M 11]
SPEEDA  20
MOVELD  CI M 1]

```

```

OPEN
MOVELD  Q M 11]
SPEEDA  50

```

```

LABEL 99

```

```

END

```

```

PROGRAM FT0BF ; Take a part from a feeder to buffer 1 (physical)

```

```

DEFINE  L C X LV
DEFINE  A1 A2 P

```

```

SET     X = MCELL - 1
SET     L = X / 18
SET     L = L + 1
SET     C = X MOD 18
SET     C = C + 1
SET     LV = 70 + L
.sendat f 5

```

```

MOVED   Q M LV]
OPEN
GOSUB   LON
MOVED   SRS[ MCELL]
GOSUB   LOFF
SPEEDA  10
SET     LV = 80 + L
MOVELD  Q M LV]
CLOSE
SET     NCHNG = NCHNG + 1
SET     X = SRS[ MCELL] - 1024 ; Adjusting the quantity
IF      NCHNG <= 8
    SET   CHG1[ NCHNG] = MCELL
    SET   CHG2[ NCHNG] = X
ENDIF

```

```

SET     SRS[ MCELL] = X
SET     LV = 90 + L
MOVELD  Q M LV]
SET     LV = 70 + L
MOVELD  Q M LV]
SPEEDA  50
.sendat f 9
SET     P = SRS[ MCELL]
SET     P = P MOD 1024
SET     A1 = P - 1
SET     A1 = A1 / 10
SET     A1 = A1 + 201
SET     A2 = A1 + 100
MOVED   Q M A1]
SPEEDA  20
MOVELD  Q M A2]
OPEN
SPEEDA  50
MOVELD  Q M A1]
.sendnf

```

```

END

```

PROGRAM CVTOT ; Take a template into ASRS (physical),

```
DEFINE LC ; If BARC=1 then check the barcode.
DEFINE XLVTTN
SET X = MCELL - 1
SET L = X / 18
SET L = L + 1
SET C = X MOD 18
SET C = C + 1
STOP MSP9
CLRBUF 7

GOSUB LCN
MOVE SRSF[ C]

IF BARC = 1 ; If barcode is active
  RUN RBCOD
  MOVED C[ M] 8]
  MOVED C[ M] 10]
  DELAY 100
  IF CKCOD = 1
    GOTO 1
  ENDF
  DELAY 200
  LABEL 1
  STOP RBCOD
  RUN LSCFF
  IF CKCOD = 1
    SET TTN = BCODE / 10000
    IF TTN <> YTMPL ; Barcode <> Information
      RUN NCCOD
      SET CCONT = 100 * 1024 ; Unknown template
    ENDF
  ELSE ; No barcode decode
    RUN NCCOD
    SET CCONT = 100 * 1024 ; Unknown template
  ENDF
ENDIF

SET LV = 30 + L
MOVED C[ M] LV]
SET LV = 40 + L
MOVED C[ M] LV]
MOVED SRSF[ C]
GOSUB LOFF
SET LV = 50 + L
MOVED C[ M] LV]
SPEEDA 10
SET LV = 60 + L
MOVED C[ M] LV]
OPEN

SET NCHNG = NCHNG + 1
IF NCHNG <= 8
  SET CHG1[ NCHNG] = MCELL
  SET CHG2[ NCHNG] = CCONT
ENDIF
SET SRS[ MCELL] = CCONT
SET LV = 50 + L
MOVED C[ M] LV]
SPEEDA 50
SET LV = 40 + L
MOVED C[ M] LV]
SET LV = 30 + L
```


MOVED C M LV]

END

PROGRAM FROMP / Y ; Takes a template from the pallet

OPEN
MOVED C M 19]
SPEEDA 20
MOVELD C M 9]
CLOSE
SPEEDA 30
MOVELD C M 19]
SPEEDA 50

END

PROGRAM FROMB / Y ; Takes a template from buffer 1

DEFINE X
PEND X FROM PLOCK
SET LOCK = 1
IF MPART[1] > 0
SET MSS = 80
SET MSP = MRAW[1] ; The raw part # is needed
ELSE
IF MPART[1] = 0
SET MSS = 90
SET MSP = -MRAW[1]
IF MSP <= 1000
ANDIF MSP >= 1
SET MST = PRITM[MSP]
ELSE
SET MST = 0
ENDIF
ENDIF
ENDIF

SPEEDA 50
MOVED C M 11]
OPEN
SPEEDA 25
MOVELD C M 1]
CLOSE
MOVELD C M 11]
SPEEDA 50

END

PROGRAM TOPLT / Y ; Takes a template to the pallet conveyor

MOVED C M 19]
SPEEDA 20
MOVELD C M 9]
OPEN
MOVELD C M 19]
SPEEDA 50
SET LOCK = 0

POST 1 TO PLOCK

END

PROGRAM MOVEP / N
; Release the pallet (Dummy definition)
END

;***** Store/Retrieve routines *****

PROGRAM ISVLD; Fetch a part (part ID=RRAW) with a template to buffer 1

DEFINE I L T I RAW
DEFINE J P1 T1 FD
DEFINE MS MP MT

PEND I FROM PLOCK
PEND I FROM ROBOT
SET LOCK = 1
SET SBUSY = 1

IF RRAW > 0
SET MSS = 10 ; Part retrieval operation
SET MSP = RRAW
ELSE
IF RRAW < 0
SET MSS = 20 ; Empty template retrieval operation
SET MSP = -RRAW
IF MSP >= 1
ANDIF MSP <= 1000
SET MST = PRITM[MSP]
ELSE
SET MST = 0
ENDIF
ENDIF
ENDIF

IF SRSI [1] = 0
GOSUB SRSI
ENDIF

SET FV = 0

IF RRAW < 0 ; Request to release an empty template
SET IRAW = -RRAW
IF IRAW <= 1000
SET RTMPL = PRITM[IRAW]
FOR J = 1 TO 54
SET I = SRSI [J]
SET T = SRSI [I]
IF T >= 1024
SET P1 = T MOD 1024
IF P1 = 0
SET T1 = T / 1024
IF T1 = RTMPL
SET MCELL = I
SET BTMP = RTMPL ; Template type #
.sendat f 3
GOSUB TTCBF ; Take the empty template to buffer 1
; after barcode

```

.sendnt 1
IF      BARC = 1 ; Barcode active
  IF      BCK = 1
    SET    FV = 1
    GOTO   99
  ELSE    ; Barcode<>Information
    RUN    NOCOD
    SET    COCNT = 100*1024 ; Mark as unknown
    SET    BARC = 0          ; Don't re-read the barcode
    GOSUB  CVTOT           ; Restore into ASRS
    SET    BARC = 1
  ENDIF
ELSE ; Barcode not active
  SET    FV = 1
  GOTO   99
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
GOTO   99
ENDIF

```

```

IF      RRAW >= 1 ; Request to release a part (on a template)
ANDIF  RRAW <= 1000
SET    RTMPL = PRITM RRAW
IF      RTMPL = 0
  SET    RTMPL = 1001
ENDIF
SET    FDCEL = 0
; First try to find a part on template (not from a feeder)
FOR    J = 1 TO 54
  SET    I = SRSI [J]
  SET    T = SRS [I]
  IF      T > 0
    SET    P1 = T MOD 1024
    SET    T1 = T / 1024
    IF      P1 = RRAW
      IF      T1 <= 1000 ; A part on template
        SET    MCELL = I
        SET    BTMP = RTMPL
        .sendatf 1
        GOSUB  TTCBF
        .sendna
        IF      BARC = 1 ; Barcode active
          IF      BCK = 1
            SET    FV = 1
            GOTO   99
          ELSE ; Barcode<>Information
            RUN    NOCOD
            SET    COCNT = 100 * 1024 ; Mark as unknown
            SET    BARC = 0          ; Don't re-read the barcode
            GOSUB  CVTOT           ; Restore into ASRS
            SET    BARC = 1
          ENDIF
        ELSE ; Barcode not active
          SET    FV = 1
          GOTO   99
        ENDIF
      ELSE ; A feeder
        IF      FDCEL = 0 ; A feeder wasn't found yet
          IF      T1 > 2000 ; Not an empty feeder
            SET    FDCEL = 1
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDFOR

```

```

        ENDF
      ENDF
    ENDF
  ENDF
ENDFOR

```

```

; If we are here then the part requested can't be found on a template.
; If a feeder was found (FDOEL>0) then find an empty template,
; retrieve it and then retrieve the part from the feeder.

```

```

IF      FDOEL > 0 ; A feeder was found
; Find an empty template
FOR      J = 1 TO 54
  SET     I = SRSI [J]
  SET     T = SRSI [I]
  IF      T >= 1024
    SET     P1 = T MOD 1024
    IF      P1 = 0
      SET     T1 = T / 1024
      IF      T1 = RTMPL
        SET     MCELL = I
        SET     BTMP = RTMPL ; Template type #
        .sendatf 3
        GOSUB  TTOBF ; Take the empty template to buffer 1 after barcode
        .sendnt 2
        IF      BARC = 1 ; Barcode active
          IF      BCK = 1
            GOTO 11
          ELSE    ; Barcode<->Information
            RUN     NOCOD
            SET     COCNT = 100 * 1024 ; Mark as unknown
            SET     BARC = 0           ; Don't re-read the barcode
            GOSUB  CVTOT           ; Restore into ASRS
            SET     BARC = 1
          ENDF
        ELSE    ; Barcode not active
          GOTO 11
        ENDF
      ENDF
    ENDF
  ENDF
ENDFOR
ENDF
GOTO 99

```

```

LABEL 11
SET     SBUSY = 0
; Retrieve the part from the feeder
SET     MS = MSS ; Save MS? variables
SET     MP = MSP
SET     MT = MST
POST    1 TO ROBOT ; Robot can be released for PTOW program
DELAY 50
PEND    1 FROM ROBOT
SET     SBUSY = 1
SET     MSS = MS ; Restore MS? variables
SET     MSP = MP
SET     MST = MT
SET     MCELL = FDOEL
GOSUB  FTOTBF ; Take the part from the feeder to a buffer 1
SET     FV = 1

```

```

LABEL 99
IF FV = 0
  RUN NCRET
  IF MSS = 10
    SET MSS = 30
    DELAY 700
  ELSE
    IF MSS = 20
      SET MSS = 40
      DELAY 700
    ENDIF
  ENDIF
ENDIF

```

```

SET LOCK = 0
SET SBUSY = 0
POST 1 TO ROBOT
POST 1 TO PLOCK
RUN M/SP9

```

```
END
```

PROGRAM PTOW; Take a template from above a pallet into ASRS (logical)

```

DEFINE J F K
PEND K FROM ROBOT
SET SBUSY = 1

```

```

SET LOCK = 1
IF YPART > 0
  SET MSS = 50
  SET MSP = YPART
ELSE
  IF YPART = 0
    SET MSS = 60
    SET MST = YTMPL
  ENDIF
ENDIF
IF SRSI [ 1 ] = 0
  GOSUB SRSI
ENDIF
SET F = 0
FOR J = 1 TO 54
  SET K = SRSI [ J ]
  IF SRSI [ K ] = 0
    SET F = 1 ; An empty place was found
    GOTO 1
  ENDIF
ENDFOR

```

```

LABEL 1
IF F = 0 ; ASRS is full
  SET FIN = 1
  IF PLOCK <> 0
    SET LOCK = 0
  ENDIF
  SET SBUSY = 0
  POST 1 TO ROBOT
ELSE
  .sendy 11
  GOSUB FROMP ; Take a template from above the pallet
  .sendy 13
  IF YPART = 0

```

```

    SET      COCNT = YTMPL * 1024
ELSE
    SET      COCNT = YPART
ENDIF
SET      MPART[ 2] = YPART
SET      MRAW[ 2] = YRAW
SET      MSUBP[ 2] = YSUBP
SET      YMA = 0
SET      YTASK = 0
SET      YPART = 0
SET      YRAW = 0
SET      YSUBP = 0
SET      YINFO = 0
SET      VBFR = 2
; SET      FWRK = 1
SET      MCELL = K
GOSUB    CVTOT ; Take the template from above the conveyor to the ASRS
SET      YTMPL = 0
IF      PLOCK <> 0
    SET      LOCK = 0
    RUN      M/SP9
ENDIF
.sendy 15
SET      SBUSY = 0
POST     1 TO ROBOT
; QPOST   2 TO CFIN; removed by Shmulik at 20/10/92
ENDIF

END

```

;***** PC <-> Controller sync routines *****

PROGRAM WCA ; PC wait to change ASRS contents

```

DEFINE   V1
SET      LOCK = 1 ; Assume other processes are running (i.e. ISVLD or PTOV)
IF      PLOCK <> 0
    ANDIF  ROBOT <> 0
    PEND   V1 FROM PLOCK
    PEND   V1 FROM ROBOT
    SET    LOCK = 0
ENDIF
END

```

```

PROGRAM  FCA ; When PC has finished updating
POST     1 TO ROBOT
POST     1 TO PLOCK
END

```

;***** Manual Store/Retrieve routines

```

PROGRAM  STRP ; Forward definition
END

```

```

PROGRAM  RETP ; Forward definition
END

```

```

PROGRAM  TESTB ; Forward definition
END

```

PROGRAM XSTRP ; Used to abort program STRP

```
DEFINE SBARC
SET SBARC = BARC

LABEL 1
IF A = 0
GOTO 1
ELSE
STOP STRP
CLRBUF
GOSUB LOFF
STOP RBCOD
GOSUB LSCFF
SET LOCK = 0
SET BARC = SBARC
POST 1 TO ROBOT
POST 1 TO PLOCK
STOP XSTRP
ENDIF
```

END

PROGRAM XRETP ; Used to abort program RETP

```
DEFINE SBARC
SET SBARC = BARC

LABEL 1
IF A = 0
GOTO 1
ELSE
STOP RETP
CLRBUF
GOSUB LOFF
SET LOCK = 0
STOP RBCOD
GOSUB LSCFF
SET BARC = SBARC
POST 1 TO ROBOT
POST 1 TO PLOCK
STOP XRETP
ENDIF
```

END

PROGRAM XTSTB ; Used to abort program TESTB

```
LABEL 1
IF A = 0
GOTO 1
ELSE
STOP TESTB
STOP RBCOD
CLRBUF
GOSUB LSCFF
SET LOCK = 0
SPEEDA 50
POST 1 TO ROBOT
POST 1 TO PLOCK
STOP XTSTB
ENDIF
```

END

```
PROGRAM STRP /Y
      ; Manual storage of part/template # TMP
DEFINE K L C LV
DEFINE I J TTN BTTN

GET I
IF SRSI [ 1 ] = 0
  GOSUB SRSI I
ENDIF

SET L = 0
FOR J = 1 TO 54
  SET K = SRSI [ J ]
  IF SRSI [ K ] = 0
    SET L = 99
    GOTO 1
  ENDF
ENDFOR

LABEL 1
IF L = 0
  PRIN TLN "np"
PRIN TLN
ELSE
  PRIN TLN "pt"
  PRIN TLN
  PEND I FROM PLOCK
  PEND I FROM ROBOT
  SET LOCK = 1
  SET A = 0
  RUN XSTRP
  OPEN
  MOVED CI M [ 11 ]
  SPEEDA 20
  MOVE LD CI M [ 1 ]
  CLOSE
  MOVE LD CI M [ 11 ]
  IF BARC = 1 ; If bar code is active
    RUN RBCOD
    IF TMP >= 1 ; Request to release a part (on a template)
      ANDIF TMP <= 1000
      SET TTN = PRITM [ TMP ]
      IF TTN = 0
        SET TTN = 1001
      ENDF
    ELSE
      SET TTN = TMP / 1024
    ENDF
    MOVED CI M [ 8 ]
    MOVED CI M [ 10 ]
    DELAY 100
    STOP RBCOD
    RUN LSOFF
    IF CKCOD=1
      SET BTTN = BCODE / 10000
      IF BTTN <> TTN ; Bar code <= I n f or mat i on
        RUN NCCOD
        SET TMP = 100 * 1024 ; Unknown template
      ENDF
    ELSE ; No bar code decode
```



```

    RUN    NCCOD
    SET    TMP = 100 * 1024 ; Unknown template
  ENDF
ENDIF
SPEEDA   50
SET      K = K - 1
SET      L = K / 18
SET      L = L + 1
SET      C = K MOD 18
SET      C = C + 1
SET      K = K + 1
GOSUB    LON
MOVE     SRSF[C]
SET      LV = 30 + L
MOVED    C[M]LV]
MOVED    SRSF[C]
GOSUB    LOFF
SET      LV = 40 + L
MOVED    C[M]LV]
SPEEDA   10
SET      LV = 50 + L
MOVED    C[M]LV]
SET      LV = 60 + L
MOVED    C[M]LV]
OPEN
SET      SRS[K] = TMP
SET      NCHNG = NCHNG + 1
IF       NCHNG <= 8
  SET    CHG1[NCHNG] = K
  SET    CHG2[NCHNG] = TMP
ENDIF
SET      LV = 50 + L
MOVED    C[M]LV]
SET      LV=40+L
MOVED    C[M]LV]
SPEEDA   50
SET      LV = 30 + L
MOVED    C[M]LV]
STOP     XSTRP
SET      LOCK = 0
POST     1 TO ROBOT
POST     1 TO PLOCK
RUN      M/SP9
PRINT    "fs"
PRINTLN
ENDIF

END

PROGRAM  RETP / Y
        ; Manual retrieval of part/template # TMP
DEFINE  I T
DEFINE  J K
DEFINE  P1
DEFINE  T1
DEFINE  FD TTN BTTN
DEFINE  TMPP FVV

SET     GTERR = 0
GET     I
PRINTLN "gt"
PRINTLN

PEND    I FROM PLOCK

```

```

PEND      I FROM ROBOT
SET       LOOK = 1
SET       A = 0
RUN       XRETP

IF        SRSI [1] = 0
  GOSUB   SRSI I
ENDIF

SET       FDCEL = 0
SET       TCELL = 0
SET       FW = 0

IF        TMP > 1000 ; Request to release an empty template
FOR       J = 1 TO 54
  SET     I = SRSI [J]
  SET     T = SRSI [I]
  IF     T = TMP
    SET   MCELL = I
    SET   BTMP = T / 1024 ; Template type #
    GOSUB TTCBF ; Take the empty template to buffer 1 after barcode
    IF   BARC = 1 ; Barcode active
      IF   BCK = 1
        SET   FW = 1
        GOTO 99
      ELSE ; Barcode<>Information
        RUN   NCCOD
        SET   CCONT = 100 * 1024 ; Mark as unknown
        SET   BARC = 0 ; Don't re-read the barcode
        GOSUB CVTOT ; Restore into ASRS
        SET   BARC = 1
      ENDIF
    ELSE ; Barcode not active
      SET   FW = 1
      GOTO 99
    ENDIF
  ENDIF
ENDFOR
SET       GTERR = 1
GOTO     99
ENDIF

IF        TMP >= 1 ; Request to release a part (on a template)
ANDIF    TMP <= 1000
SET      TTN = PRITM [TMP]
IF      TTN = 0
  SET    TTN = 1001
ENDIF
; First try to find a part on template (not from a feeder)
FOR      J = 1 TO 54
  SET    I = SRSI [J]
  SET    T = SRSI [I]
  IF    T > 0
    SET   P1 = T MOD 1024
    SET   T1 = T / 1024
    IF   P1 = TMP
      IF   T1 <= 1000 ; A part on template
        SET   MCELL = I
        SET   BTMP = TTN
        GOSUB TTCBF
        IF   BARC = 1 ; Barcode active
          IF   BCK = 1
            SET   FW = 1
            GOTO 99
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDFOR

```

```

ELSE ; Barcode<>Information
  RUN      NCOOD
  SET      CCONT = 100 * 1024 ; Mark as unknown
  SET      BARC = 0          ; Don't re-read the barcode
  GOSUB   CVTOT           ; Restore into ASRS
  SET      BARC=1
ENDIF
ELSE ; Barcode not active
  SET      FW=1
  GOTO    99
ENDIF
ELSE ; A feeder
  IF      FDCEL = 0 ; A feeder wasn't found yet
  IF      T1 > 2000 ; Not an empty feeder
    SET      FDCEL = 1
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```

; If we are here then the part requested can't be found on a template.
; If a feeder was found (FDCEL>0) then find an empty template,
; retrieve it and then retrieve the part from the feeder.

```

```

IF      FDCEL > 0 ; A feeder was found
; Find an empty template
FOR      J = 1 TO 54
  SET      I = SRS[J]
  SET      T = SRS[I]
  IF      T >= 1024
    SET      P1 = T MOD 1024
    IF      P1 = 0
      SET      T1 = T / 1024
      IF      T1 = TTN
        SET      MCELL = 1
        SET      TCELL = 1
        SET      BTMP = TTN ; Template type #
        GOSUB   TTOBF ; Take the empty template to buffer 1
                  ; after barcode
        IF      BARC = 1 ; Barcode active
          IF      BOK = 1
            GOTO    11
          ELSE ; Barcode<>Information
            RUN      NCOOD
            SET      CCONT = 100 * 1024 ; Mark as unknown
            SET      BARC = 0          ; Don't re-read the barcode
            GOSUB   CVTOT           ; Restore into ASRS
            SET      BARC = 1
          ENDIF
        ELSE ; Barcode not active
          GOTO 11
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
SET      GTERR = -1 ; No empty template for the part was found
GOTO    99
ENDIF
ENDIF
SET      GTERR = 1
GOTO    99

```

```

LABEL 11
; Retrieve the part from the feeder
SET MCELL = FDCEL
GOSUB FTGBF ; Take the part from the feeder to a buffer 1

```

```

LABEL 99
STOP XRETP
SET LOCK = 0
PRINT "fr"
PRINTLN
POST 1 TO ROBOT
POST 1 TO PLOCK
RUN M/SP9

```

```

END

```

```

PROGRAM TESTB / Y ; Manual test of barcode scanner

```

```

DEFINE I ; Takes an empty template from buffer 1
GET I
PEND I FROM PLOCK
PEND I FROM ROBOT
SET LOCK = 1
SET A = 0
RUN XTSTB

```

```

OPEN
MOVED C[M]11]
SPEEDA 20
MOVELD C[M]1]
CLOSE
RUN RBCOD
MOVELD C[M]11]
SPEEDA 40
MOVED C[M]8]
MOVED C[M]10]
DELAY 100
MOVED C[M]11]
STOP RBCOD
RUN LSCFF
SPEEDA 20
MOVELD C[M]1]
OPEN
SPEEDA 50
MOVELD C[M]11]
STOP XTSTB
SET LOCK = 0
POST 1 TO ROBOT
POST 1 TO PLOCK
PRINTLN "vg"
PRINTLN

```

```

END

```

```

PROGRAM MKSRS ; Used to make SRSP[] points when SRSP[1] is defined.

```

```

DEFINE I J OFFS ; Counter clockwise order assumed.
FOR I = 2 TO 18
  HERER SRSP[I] SRSP[1]
  SET J = I - 1
  SET OFFS = J * 170666
  SET OFFS = OFFS / 100
;; SET OFFS = 0 - OFFS

```

```
    PRI NTLN " position " | " offset " OFFS
    SETPV   SRSP[1] 7 OFFS
ENDFOR
```

END

;***** Other routines *****

```
PROGRAM CIM / Y
DELAY 30
@ATTACH CIM
DELAY 40
@ATTACH SRSP
END
```

```
PROGRAM $INIT / Y
GOSUB MRSRS
GOSUB INT
END
```

```
PROGRAM $STRT ; Init. ASRS
STOP WCA
DELAY 80
SET LOCK = 0
SET PLOCK = 1
SET ROBOT = 1
SET NOHNG = 0
GOSUB LOFF
GOSUB LSOFF
GOSUB SRSI
SPEEDA 50
SPEEDB SPDB
@PRI ORTY LSON 10
DELAY 40
@PRI ORTY LSOFF 10
DELAY 40
@PRI ORTY RBOD 10
DELAY 40
@CON
GOSUB START
END
```

```
PROGRAM AVER
PRI NTLN
PRI NTLN " ASRS-ACL VERSION 3.2"
PRI NTLN "- Sep 1992."
PRI NTLN
END
```