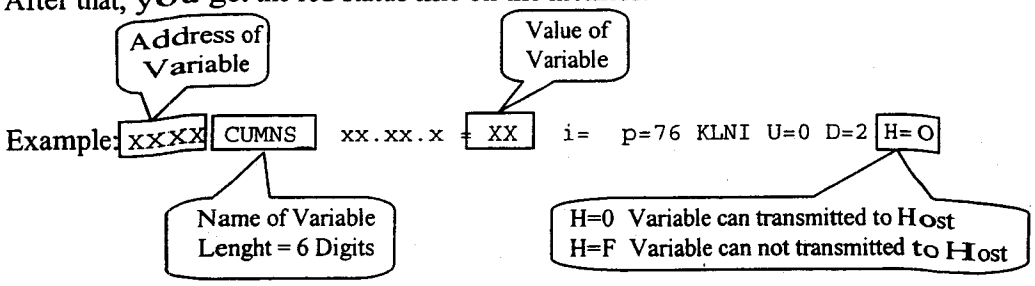
KLOCKNER PRESS

Status Line Description for Host

When you call for a special variable of the control with the host command,
TV (transmit variable) for example you need to know the exact variable name
of the MPC80.= Screen Host = Press
With the status line, which you can turn ON, on page 80, you can:
a. find out the variable name
b. if that variable can be transmitted to the host or not.

For to get the Status line on the screen, you have to enter on page 80:
In line 1 the number " 1 "
In line 3 the number " 2 "

After that, you get the red status line on the monitor.



The control will never send anything by itself to the host!!!!
If you (host) needs any information from the control, you have to call for!!!
The control will only respond to the commands from host.

MOULD # - PAGE 0 LINE 5

E09E CUMNS_8.00.00= 23 i= p=00 KLDV U=5 D=1 H=0

THIS DOCUMENT DISCRIBES THE COMMUNICATION BETWEEN A KLÖCKNER WINDSOR MPC 80 PRESS CONTROL AND A HOST COMPUTER

CONTENTS

1.0 PRESS CONTROL COMMUNICATION INTERFACE DOCUMENT

1.1 SCOPE

The purpose of this document is to define the communication interface between the MPC-80 and the a Host Computer. This definition will be the basis for Klöckner Windsor to develop the required hardware and software for the communications link between the MPC-80 and the Host Computer.

This document will describe in detail the asynchronous communication protocol used in this communication link. Asynchronous communications is a form of data transfer and is a character oriented link control which will be used as the medium of communications between the Host Computer and the MPC-80 control for the Klöckner Windsor injection molding machine.

The Communication System Interface should be capable of supporting:

1. Herein described data transfer protocol and XON/XOFF control.
2. RS 232 C Voltage interface.
3. Modem Connection.
4. 20 mA current loop connection.

1.2 COMMUNICATION CRITERIA

The Klöckner Windsor MPC-80 will communicate with any computer which uses either an RS 232 C interface with either voltage or XON/XOFF transmission control, or 20 mA current loops using XON/XOFF transmission control.

The communication parameters shall conform to the following:

1. The MPC-80 will interface with any computer which communicates via Asynchronous interface hardware, and is not a "Batch Processing" type computer, that is it must accept and issue variable length data and command strings.

2. Standard for communications shall be the EIA RS 232C voltage Interface.

3. Shall support data and command transfer with parity bit and checksum error checking and XON/XOFF control signals for preventing buffer overflow. This method requires that the communication software being used on both ends of the communication link recognize and respond to the same st of ASCII control characters to facilitate transfer of variable length strings of data or commands.

4. Shall support communications between the MPC-80 and the Host Computer by use of either a MODEM or Direct Cable connection. Strapping options must be made in the cable connecting the two computers.

5. Baud Rage - 300 to 9600 Bits Per Second.

6. Mode of communication - Full Duplex.

7. Mode of initiating communication - Polling.

Klöckner Windsor will supply to the customer a sample communications program demonstrating the use of all control characters necessary to communicate with the MPC-80. Instructions will be provided showing how to access or upload data from the MPC-80 to the Host computer, and how to download data from the Host computer to the MPC-80. all parameters, alarms, messages, and any data that exists in

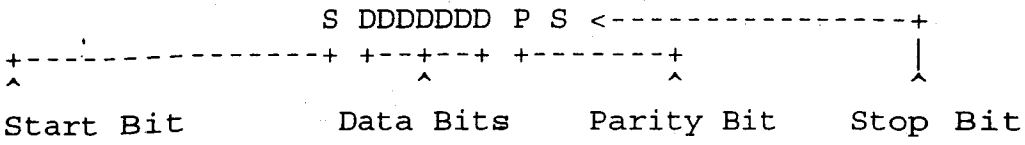
the MPC-80 can be accessed by the Host computer, and some parameters changed by the Host computer.

1.3 DATA BITS

After a start bit is transmitted, the actual information to be transferred is transmitted as "data bits". the figure below shows the "seven data bit configuration" which shall be used in the Asynchronous data link control. The maximum number of characters which can be supported by this configuration is also shown.

Number of Data Bits	Serial Data Configuration	Maximum No. Characters
7	SDDDDDDDPS	128

Data Configuration



1.4 ASYNCHRONOUS COMMUNICATION INTERFACE CONNECTION

1. RS 232C or 20 mA current loops, selectable by hardware jumpers or switches.

2. The following signals are used:

Shield (Connected to MPC-80 Ground)	Pin 1
TxD	Pin 2
RxD	Pin 3
RTS	Pin 4
CTS	Pin 5
DSR	Pin 6
Signal Ground	Pin 7
TXD (20 Ma. +)	Pin 9
TxD (20 MA. -)	Pin 10
DTR	Pin 20
RxD (20 Ma. +)	Pin 24
RxD (20 MA. -)	Pin 25

3. Baud rate hardware jumper or switch selectable:
   300, 600, 1200, 2400, 4800, 9600 Baud.

4. 1 Start Bit, 7 Data Bits, 1 Parity Bit, 1 Stop Bit

5. Data Error Checking:
   Odd Parity - 1 Parity Bit
   Four (4) ASCII Checksum characters at the end of each string.

1.5 ERROR CHECKING CHECKSUM

After the Acknowledgement (ACK) or Negative Acknowledgement (NAK) is received, a Start of Header (SOH) is sent, followed by the data, followed by the checksum, and concluded with the End of Text (ETX) character.
The checksum is calculated by adding the ASCII values of each character in the data string. The sum is divided by 255 and the remainder is retained as the checksum. This is transmitted as two ASCII characters. The receiving computer computes it's own checksum and compares it's result with that received from the transmitting computer.
If the two values are the same, the receiving computer sends an Acknowledge (ACK) to the transmitting computer to tell it that the data has been received in good order.
If the two values are not the same, the receiving computer sends an Negative Acknowledgement to the transmitting computer to tell it to send the data string again.
This retransmission process is repeated until the data is properly received, or until nine attempts have been made to transmit the block. If the data cannot be properly transmitted after nine attempts, the communication is aborted, and must be restarted.

1.6 XON/XOFF CONTROL

The XON/XOFF "data flow control protocol" is a mechanism for controlling the flow of data when communication is accomplished by 20 mA current loops. This is necessary because there is no other means available for the receiving computer to control transmission by the sending computer.

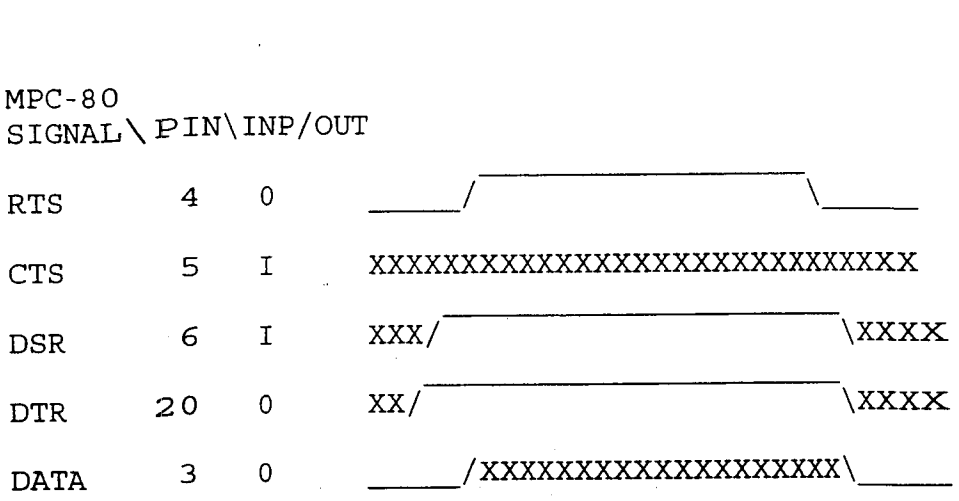
The capacity of a communications buffer can be exceeded if the mismatch in flow of data is significant and the data file being transmitted is large, but the XON/XOFF flow control toggles the flow of data on and off to prevent buffer overflow.

If the XON/XOFF protocol is properly implemented at both the MPC-80 and the IBM Personal Computer ends of a communications link, proper data flow speed-matching is handled by the receiving software at each end of the communications link. When the volume of data capacity of the buffer, the software sends an XOFF to the transmitting computer. (The XOFF character is an ASCII Device Control 3, and is computer temporarily halts data transmission, allowing the receiving computer to process the data contained in it's communications receiver buffer. When the buffer has been emptied to a pre-determined low level, receiving computer software send an XON character, (This is an ASCII Device Control 1 equivalent to a Control Q), and signals the transmitting computer to resume data transmission. This cycle may be repeated many times during a data transmission without user knowledge or involvement.

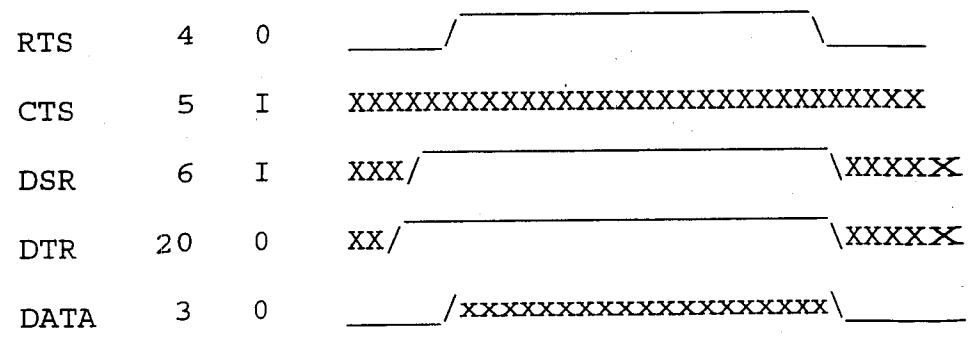
To handle the upload and download communications properly, both he MPC-80 and the IBM Personal computer must have the same data flow control protocol, so that data may be transmitted to the MPC-80 or the IBM Personal Computer without error.

1.7 HANDSHAKING BETWEEN THE MPC-80 AND THE HOST COMPUTER

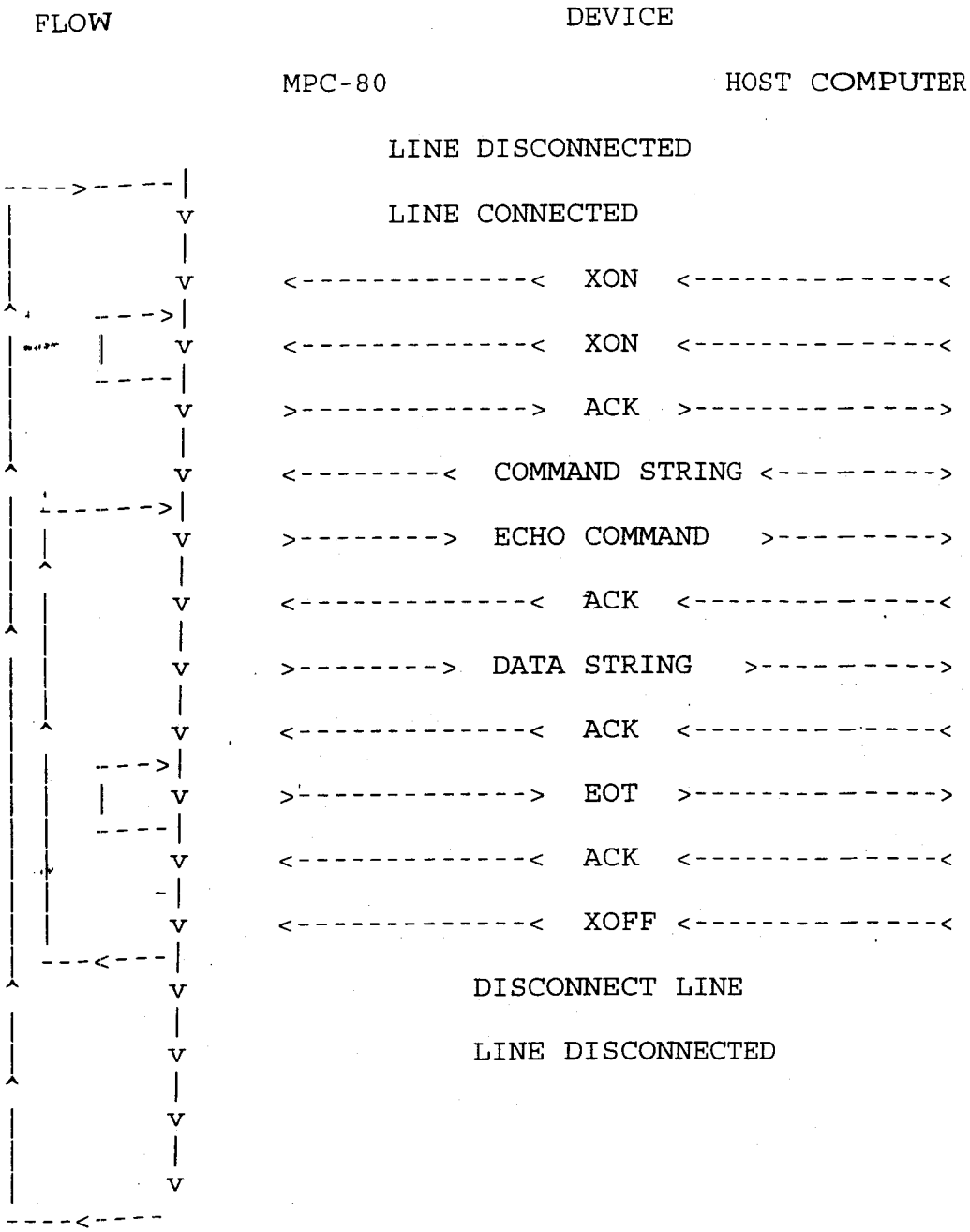
MPC-80 RECEIVING FROM THE HOST COMPUTER



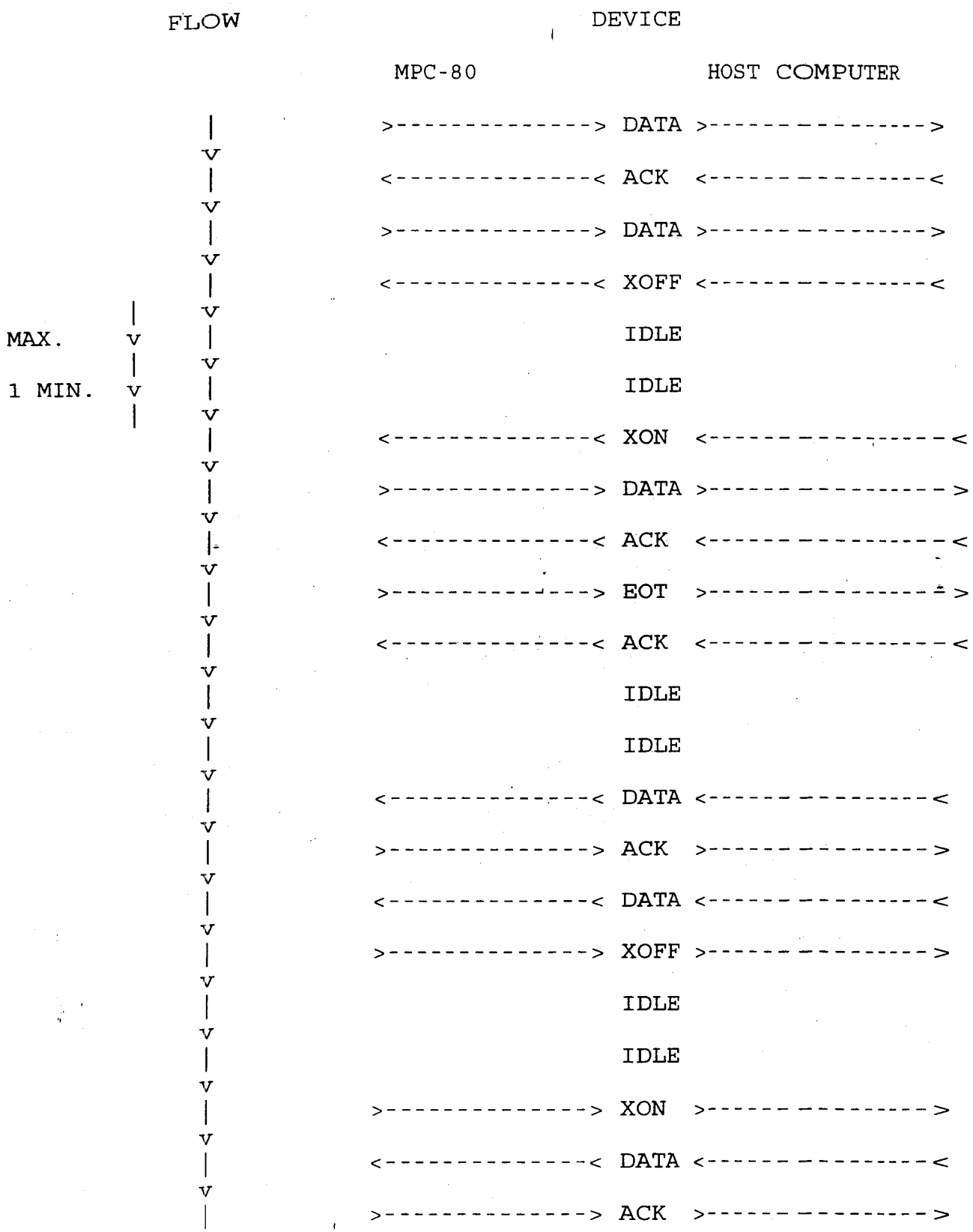
HOST COMPUTER RECEIVING FROM THE MPC-80

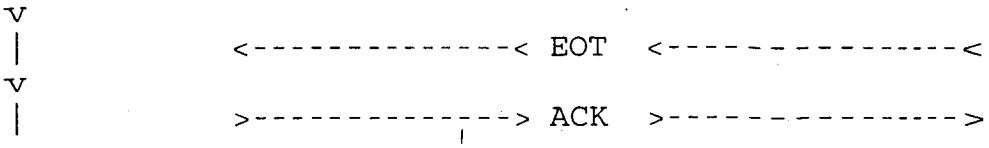


1.8 FLOW OF COMMUNICATION BETWEEN THE MPC-80 AND THE HOST COMPUTER

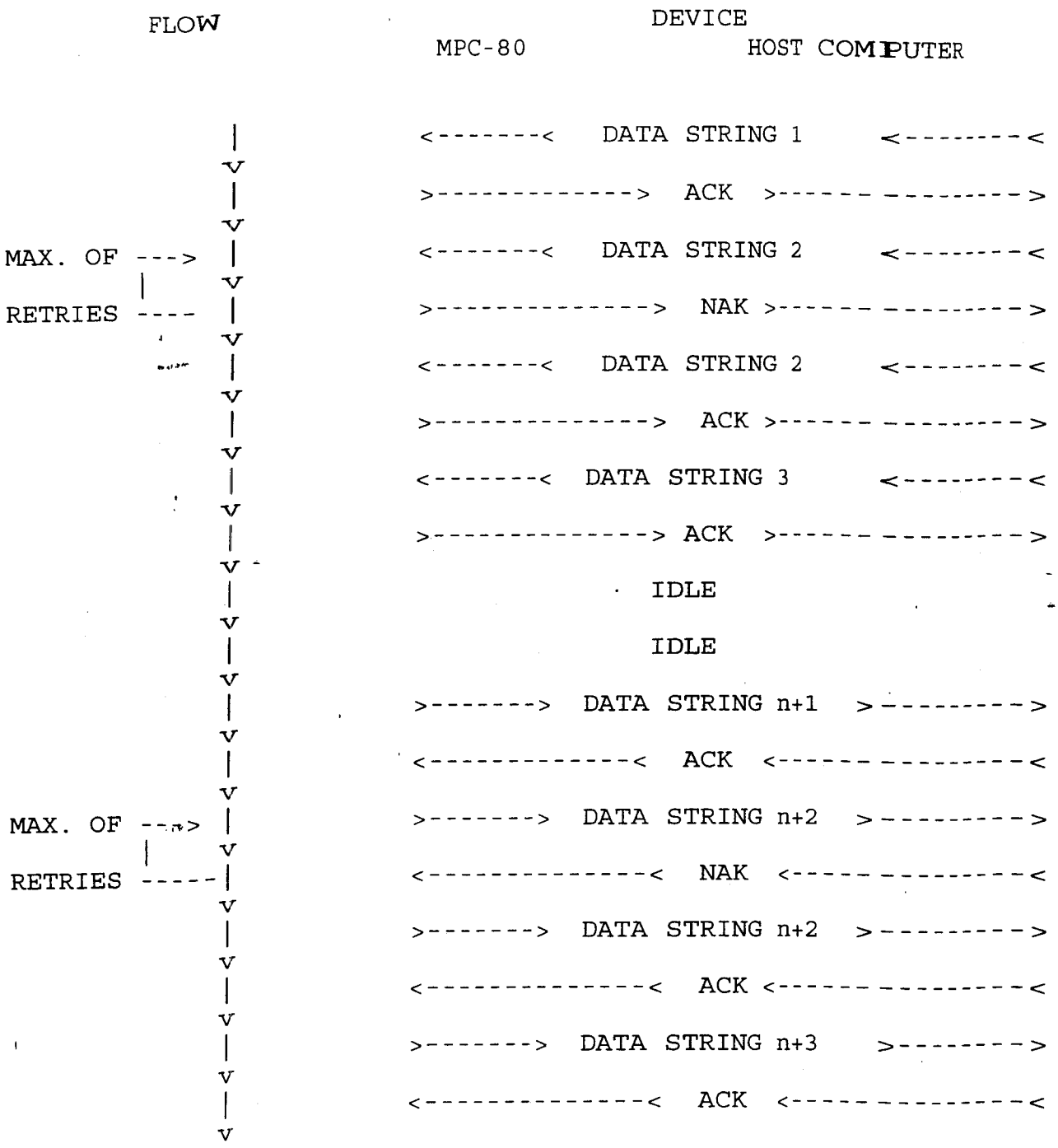


1.9 FUNCTIONING OF XON AND XOFF BETWEEN THE MPC-80 AND THE HOST COMPUTER




EOT
ACK

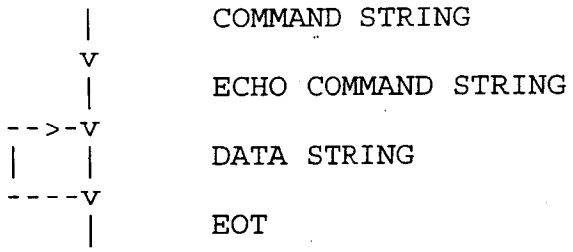
1.18 FUNCTIONING OF ACK AND NAK BETWEEN THE MPC-80 AND THE HOST COMPUTER



1.11 FORMAT OF ALL COMMANDS OR DATA STRINGS

<- - - - - STX / DATA OR MESSAGE / CHECKSUM / ETX - - - - -<

1.12 FORMAT OR ALL COMMUNICATION TELEGRAMS



2.0 PRESS CONTROL COMMUNICATIONS SYNTAX

This document describes the syntax protocol for data communications between a Windsor MPC-80 Injection Moulding Machine Control Unit and a Host Computer.

The herein described syntax generally can work with any type of Host Computer by using a Serial Data Communications Line. The host is master of the link, MPC-80 is slave, eg. communication is always controlled by the host.

The Hardware necessary for the data transfer is described in the document Communications between the MPC-80 and a Host Computer.

2.1 HOW ARE THE PRESS SET UP PARAMETERS TRANSFERRED TO THE HOST COMPUTER?

The machine setup parameters (program) will be transferred to and received from the external world (host equipment) by using a special communications protocol.
That protocol uses a sequence of statements to handle all communications.

2.2 HOW DOES THE MPC-80 COMMUNICATE WITH THE HOST COMPUTER

The MPC-80 communicates over a serial interface link. MPC-80 is the Slave of the link. The Host is Master. The following chapters describe the syntax for that communications.

Asynchronous Communication

1. RS 232 C
2. Following Signals are used:

   RTS
   DTR
   DSR
   CTS
   RXD
   TXD
   SIGNAL GND
   SHIELD connected to MPC-80 GND

3. Baud rate Hardware jumper selectable

   300, 600, 1200, 2400, 4800, 9600 Baud

4. 7bit data, 1 start bit, 1 stop bit, 1 parity bit

5. Data error checking:

   Odd parity - 1 parity bit
   16 bit 1's comliment checksum at end of each data or command string.

2.2 -cont. FORMAT

All transmission begins with STX and consist of 1 or more data or command strings. Each data - or command string is ended with a checksum and an ETX. The transmission telegram is ended by a EOT.

A complete transmission Telegram consists of 1 command string, an echo command string, and 1 or more data strings.

A complete data or command string consists of a STX character text, formed by ASCII - characters. The characters can also include control characters below 20H.
Last 4 characters of a data or command string is a 16 bit checksum. (refer chapter communication between MPC-80 and Host.)

```
S (I I) P P P P P P C C C C E
T                           T
X                           X
```

STX : Start of Text ASCII 02
(II): Identifier, fixed length of 2 ASCII characters (optional)
PPP : Parameterset, up to 249 ASCII characters
ETX : End of Text ASCII 03

The flow of data between Host and MPC-80 is controlled by the command string. The command string is the first message of each telegram, and therefore issued by the Host.
Due to the fact that the MPC-80 is slave of the link the Host must initiate all communication telegrams. There is no way for the MPC-80 to generate interrupts to the Host. This enables the Host to control all actions of the communications link while still having the possibility to multiplex the communications line between multiple MPC-80 units and to organize data backups and to do protected calculations for data concentration.

The fast response to the real time process in maintained by the capability of the MPC-80 to collect data and storing them into buffers.
A polling sequence can be used by the Host, to get knowledge of data availability.
This polling system reduces the amount of data to be transferred over the communications link.

2.3 SUMMARY : LIST OF IDENTIFIERS

TP	TRANSMIT PAGE	(h)
TL	TRANSMIT LINE OF PAGE	(h)
MS	TRANSMIT CURRENT MACHINESTATUS	(h)
TV	TRANSMIT A VARIABLE	(h)
DD	DOWNLOAD DATA	(h)
UD	UPLOAD DATA	(m)
EB	TRANSMIT ERROR BUFFER	(h)
LC	TRANSMIT LAST CYCLES BUFFER	(h)
CB	TRANSMIT CHANGE OF STATE BUFFER	(h)
DM	DOWNLOAD MESSAGE	(h)
ST	SET TRACE INSTRUCTION REGISTER	(h)
TB	TRANSMIT TRACE BUFFER	(h)
IT	INTERRUPT from MPC to Host	(Int m)

2.4 EXAMPLES OF COMMUNICATION

The examples of communication give a desCRLFiption of syntax being used for the communication between Host and MPC-80. The direction of the flow of information is shown in this document by a label at the end of each line or each group of line.

(h) : Host issues the string
(m) : MPC-80 issues the string

2.4.1 TP TRANSMIT PAGE

The Transmit Page command can be used to get a desired page of the MPC-80 screen pages along with all on the page displayed actual and set values. The data of the actual and set values are read out from the memory of the MPC-80 at time of access. Any page can be accessed by the host, regardlessly to the actual page displayed on the sCRLFeen of the MPC-80.

TRANSMIT PAGE (h)

```
TP 10 CRLF      (h)
TP 10 CRLF      (m)
      SEITE 10 ZYLNDERTEMPERATUREN                    LF CRLF
                                                      LF CRLF
1     ZYLINDERHEIZUNG 0 = AUS  1 = EIN                LF CRLF
EOT    (m)
```

2.4.2 TL TRANSMIT LINE OF PAGE

The Transmit Line of Page command may be used to read out an individual line of a page. The line number refers to the physical line number of the screen layout at the MPC-80 screen.

TRANSMIT LINE OF PAGE (h)

```
TL 10 3 CRLF                    (h)
TL 10 3 CRLF                    (m)
1  ZYLINDERHEIZUNG 0 = AUS  1 = EIN
                   2 = ABSENKUNG SOLL 0              LF CRLF
```

EOT (m)

2.4.3 MS TRANSMIT CURRENT MACHINE STATUS

The Machine Status Bytes contains a set of representative status information concerning the actual machine function and host computer interface status. The data are gathered in a 16 word data set as follows:

1. word	Error Buffer Status	0=empty 1=data pending
2. word	Change of State Buffer Status	"
3. word	Last Cycle Buffer Status	"
4. word	Up/Download Request	0=no 1=upload 2=download
5. word	End of Batch	0=not reached 1=reached
6. word	Trace Buffer	0=empty 1=data pending
7. word	Safety Gate	0=closed and tested 1=closed 2=closing 3=opening 4=open 5=Tapeswich or S31 6=undefined position
8. word	Mold	0=closed and clamp pressure 1=closed 2=opening 3=open 4=closing 5=cooling time 6=undef.position
9. word	Carriage	0=back 1=moving forward 2=is forward 3=moving backward 4=undef.position
10. word	Injection	0=plasticised 1=injection 2=hold pressure 3=plasticising 4=suckback 5=no funktion
11. word	Ejector	0=back 1=mov.forward 2=is forward 3=mov. backward 4=undef.position
12. word	Core 1	0=not present 1=setting 2=is set 3=pulling 4=is pulled 5=undef.pos.
13. word	Core 2	
14. word	Core 3	
15. word	Core 4	
16. word	Mode	0=undefined 1=Setup 2=manualy 3=semi-automatic 4=automatic 6=cycle stop 7=mold change

This 16 words of Information are transferred to the host as a 32 bytes ASCII character string containing hexadecimal digits 0....F .

TRANSMIT CURRENT MACHINE STATUS

```
MS CRLF          (h)
MS 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 LF CRLF(m)
EOT              (m)
```

MPC-80 reports in word 1...4 Bufferstatus (Errorbuffer, Change of State Buffer, Last Cycles Buffer, Request Parameter Up/Download) and words 5 ... 16 Machinestatus.

2.4.4 TV TRANSMIT A VARIABLE

Set values, actual values and process data are contained in a set of variables.
This variables reside in memory of the Z15 - Processor of the MPC-80. Most of them can be seen on the sCRLFeen, some are not available to the operator.
All variables can be accessed by the use of symbolic names.

NOMENCLATURE OF VARIABLE NAMES

All names are 6 Characters long. The first character is an uppercase ASCII character.
The names are formed of memnonic substrings.
The substrings have coordination to the meaning of the individual variable.
4 groups of substrings form the 6 character names:

```
NAMEXY
----!!____physical type (stroke=s time=t
!  ! !                  pressure=p velocity=s ...
!  ! !_____effect       (set value, actual value,
!  !                    machine constant, number ...
!  !_______additional label
!__________function
```

FUNCTION:

MC	Mould Closing
MO	Mould Opening
IN	Injection
HP	Hold Pressure
CA	Carriage
EJ	Ejector
BH	Barrel Heater
MH	Mould Heater
AV	Air valve
CO	Core
CY	Cycle
PO	Production

EFFECT

Effect discribes the effect of the variable with respect of the process.

There are:
S Set Value
A Actual Value
K Constant
N Number

PHYSICAL SIZE
Identifies the physical size as there are:
s Stroke
p Pressure
T Temperature
t Time
v velocity
q Quantity

All values of variables can be read out of the MPC-80, most of them can be written into the MPC-80. Whether a variable can be written depends of the host user status defined in the MPC-80.
The host user status is a machine specific Parameter and is defined by Klöckner Windsor. It allows the host to write into a variable or not.
All Variables must be transferred as data and dimension in ASCII-character. This is to ensure that the Variable has been sent in the correct physical size.

```
TRANSMIT A VARIABLE          (h)
    TV  MOPOSs CRLF          (h)   Host reads a Variable
    TV  MOPOSs 1420.3 mm CRLF (m)
    EOT                      (m)
```

MPC-80 reports mould open position set value of opening stroke

```
    TV MOPOSs 1400.0 mm  CRLF    (h) Host writes a Variable
    TV MOPOSs CRLF               (m)
    EOT
```

The next example shows, how the MPC-80 answers, if a variable has been accessed which is not available.

```
    TV MOPOSs 1400.0 mm  (m)     (h)  Host writes a Variable
    ER V?        CRLF    (m)
```