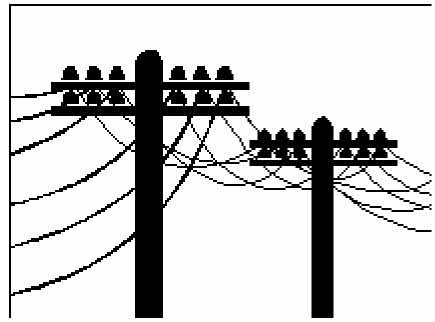
MERRICK

Merrick Industries, Inc.



Serial Communications Concepts Manual

VERSION 3.0b

MERRICK INDUSTRIES, INC.
10 ARTHUR DRIVE
LYNN HAVEN, FL 32444
(850) 265-3611

February 21, 1999

Date	Description
2004-05-27	**Formatting**

TABLE OF CONTENTS

INTRODUCTION

The purpose of this manual is to provide information on Serial Communication Options and Functions available with the Merrick MC² Controller. This manual covers the following MC² Controller models:

Controller Application Charts

The following chart contains the list of Merrick controller applications with the respective model identification numbers. See Get Model Identification 'c' (99) on page 10 for more information.

Model	DEC	HEX	Description	Supported?
20.00	1	1	Basic Feeder	Version K Only
10.00	2	2	Basic Belt Scale	NO
24.00	3	3	Feeder	NO
36.00	4	4	Batching Loss in Weight	NO
16.00	5	5	Batching Belt Scale	NO
30.00	6	6	Basic Loss in Weight	Version D Only
21.00	7	7	Feeder W/ Lag Control	NO
90.00	9	9	Digital Master Rate Setter	NO
91.00	10	A	Analog Master Rate Setter	NO
22.00	11	B	Generic Feedrate Controller	Version B Only
94.00	12	C	Remote Bezel	NO
24.80	13	D	Belt Scale w/Bypass mode	NO
35.00	14	E	Batching Loss in Weight	NO
99.00	15	F		NO
S10.00	16	10	Spanish Belt Scale	NO
31.00	17	11	Gain in Weight Batcher	NO

Model	DEC	HEX	Description	Supported?
20.00.HP	33	21	Feeder	ALL
10.00.HP	34	22	Belt Scale	ALL
S10.00.HP	35	23	Spanish Belt Scale	ALL
11.00.HP	36	24	Static Scale	ALL
35.00.HP	37	25	Loss in Weight Batcher	ALL
30.00.HP	38	26	Loss in Weight	ALL
24.81.HP	39	27	Feeder with Watchdog	ALL
S20.00.HP	40	28	Spanish Feeder	ALL

Model	DEC	HEX	Description	Supported?
30.10.EX	50	32	Advanced Loss in Weight	ALL
24.96.EX	51	33	Batching Feeder	ALL
24.10.EX	52	34	Complete Feeder	ALL
30.20.EX	54	35	Batching Loss-In-Weight	ALL
40.10.EX	55	36	Impact Flow-meter	ALL

MC^2	MC^2 HP Series	MC^3
20.00.K	10.00.HP	24.10.EX
22.00.B	S10.00.HP	24.96.EX
30.00.D	11.00.HP	30.10.EX
	20.00.HP	30.20.EX
	S20.00.HP	40.10.EX
	24.81.HP	
	30.00.HP	
	35.00.HP	

This manual contains information intended to enable qualified personnel to perform the following tasks:

- Connect a printer or other monitoring device to an MC^2 or MC^3 Controller and test for correct operation.
- Connect a computer or other intelligent device to an MC^2 or MC^3 Controller and test for correct operation.
- Connect two or more MC^2 or MC^3 Controllers together and test for correct operation of communications.
- Develop Computer-Based programs to communicate with the MC^2 or MC^3 Controller.

This manual is intended to be used in conjunction with the Operator's Manual and schematic diagrams supplied to you by Merrick.

OVERVIEW

Serial Communication in Merrick Controllers ~~the MC^2/MC^3~~ can be separated into two independent types of communication. These are Computer Communication and Printer Communication.

Computer Communication

Computer Communication in ~~the MC^2~~ Merrick Controllers provides the means to exchange information between the controller and other intelligent serial devices. For example, values in the MC^3 Controller can be read from and written to by another device such as a Personal Computer or a PLC.

Serial Port #1 on the LTI board of MC^3 controllers are used for normal Computer communications. Communication using the DF-1 protocol of the MC^3 is established via Serial port #2 normally the the rs-232 serial printer port. Certain MC^3 controllers allow basic DF-1 communications using the RS232 port normally used by the printer.

Serial Port #2 on the Serial Interface Board in MC^2 controllers is used for Computer Communications (see HARDWARE on page 18). The type and amount of data that can be exchanged between a computer and the MC^2 Controller, is determined by the Merrick Communications Protocol.

Printer Communication

Printer Communication provides the means to send data to a printing or monitoring device. For example, this type of communication could be used to send a string of data representing the Net Weight to a panel ticket printer. Printer Communication is initiated at the controller. Normally a button is pushed or a digital input is closed to signal that a printout is desired. Printer communications is accomplished using Serial Port #1 on the Serial Interface Board (see HARDWARE on page 18). Refer to the Operator's Manual of your specific controller and software version to determine what print data options are available.

Hardware Options

There are two different Serial Interface Boards used in MERRICK MC^2 Controllers. The original Serial Interface Board has been available since the release of the MC^2. The newer Advanced Serial Interface Board has only recently been developed. Serial Interface Boards section on page 18 explains the original Serial Interface Board and the newer Advanced Serial Interface Board. Interconnection section on page 20 of this manual provides generic examples of Interconnection Diagrams for both Printer and Computer Communication.

COMPUTER COMMUNICATIONS

Communications with a computer or other intelligent serial device can be established in most MC² Applications. The data exchanged between a controller and a computer is determined by the Merrick Communications Protocol and the communications program written for the computer. It is normal however to be able to read and write to almost all variables in the controller. To be able to communicate with a computer, the following conditions must be met:

1. A Serial Interface Board must be correctly installed in the Controller. Refer to HARDWARE on page 18 to verify correct installation and set-up of the Serial Interface Board.
2. Interconnection Wiring between the controller and the other serial device must be installed and correct. Always refer to the wiring diagrams supplied to you by Merrick for your particular system for verification of correct interconnection. Interconnection on page 20 contains examples of standard wiring diagrams for different types of systems. These may be used as generic guides for connecting a communications device to the controller if a wiring diagram supplied by Merrick is not available.
3. Serial Communications parameters must be programmed into the controller to match the remote serial device. These include Baud Rate, Number of Data Bits, Parity Type, ASCII Start Code, ASCII End Code and Controller Address. Refer to the MC² Operator's Manual for a description of these items and instructions on how to enter them into your controller. In addition, a Specification Sheet is sent with each controller and it will contain suggested values for all of the Computer Communication Parameters.
4. A communications program must be utilized which was written for the Merrick Communications Protocol and which is running on the computer or device which will be communicating with the controller.

Merrick Protocol

The Merrick Protocol is based on character messages transferred from a Master Unit to a Slave Unit. The master will send a character message giving a command to the slave. This command could be a request for information, an instruction to perform, or data to be stored. The slave will then perform the command and respond with a data message reply, or if no data is to be returned a Positive Acknowledgment **(ACK)** message. If an error occurred or if a message is not valid the slave may reply with a Negative Acknowledgment Message **(NACK)** or not at all. The slave will not send a message to the master without having first received a command from the master.

Depending on the system configuration, a controller could be either a master or slave unit. Certain Merrick MC² applications, such as 9000 and 9100 MasterSet Systems, are programmed to control other Merrick controllers. This would mean that the special application would be the master and the other applications would be slaves. Conversely, if the controller was connected to a computer, then normally the controller would be a slave and the computer would be the master.

Building a Command Message in a Master

Many different Command Messages are available in the Merrick Protocol. Telegrams, starting on page 10, details the function, format and contents of each message. A generic format for any Command Message is as follows:

[START] [DEST] [CMD] [DATA] [CHKSUM] [END]

The **[START]** and **[END]** characters in any message are always the same. The **[DEST]** is determined by the Controller Number of the Slave Unit the message is addressed to. The **[CMD]** character is determined by the Master and corresponds to the command it needs to transmit to the Slave Unit. The **[DATA]** component is optional and may not be included for every Command Message. The **[CHKSUM]** is included with every Command Message. If error checking is not required "**??**" can be sent as the **[CHKSUM]** and the slave MC² will not perform any error checking and assume the Command Message is valid.

Decoding a Reply Message from a Slave

Many different Reply Messages are possible in the Merrick Protocol. Telegrams, starting on page 10, details the function, format and contents of each Reply Message. A generic format for any Reply Message is as follows:

[START] [SRC] [DATA] [CHKSUM] [END]

The **[SRC]** is determined by the Controller Number of the Slave Unit the Reply is from. The **[DATA]** component is optional and may not be included for every Reply Message. The **[CHKSUM]** is included with every Reply Message.

[START] - Start of Message

This is a single character which is sent at the beginning of each message to inform the receiving unit that a new message is being started. The default value of the START Character is an ASCII Line Feed (10). but the value may be changed by the operator. Refer to the Operator's Manual for instructions on entering the Start Character Value. The **[START]** and **[END]** characters in any message are always the same for all messages.

Controller Address

By using a Multi-Drop configuration up to 32 different slave units could be connected to one master. This would allow the master to access information in all of the slave units. In this type of configuration each message is sent to and received by every slave on the Multi-Drop Loop. It is important that only the controller the master intended the message for, take action and respond. To determine which slave the master is sending the message to, a unique Controller Number is programmed into each MC² Controller. If two of the slave units on the same multi-drop circuit have the same Controller Number both will attempt to respond to the message and the results are normally result in communications errors. Refer to the Operator's Manual for the correct procedure to set the Controller Number.

[DEST] - Destination

This is a single character representing the destination of the message sent from a master unit. This corresponds to the Controller Number stored in the slave MC² to which the message is addressed. An example would be if the Controller Number of the slave was 1 then the [DEST] would be the character '1'.

[SRC] - Source

This is a single character representing the source of a reply message sent from a slave. This corresponds to the Controller Number stored in the MC² generating the reply message. An example would be if the Controller Number of the slave was 1 then the [SRC] would be the character '1'.

Messages

There are two types of messages in the Merrick Protocol. The master will transmit a Command Message and a slave will respond with an Acknowledgment Message. Each message can be broken down into individual components all of which have a specific definition. Some components are common to both types of message and other components are specific to one message or the other. See Telegrams (page 10) for a detailed explanation of the possible characters included for each message in the Merrick Protocol. Listed below are the components used for both types of messages:

[CMD] - Command Letter

CMD is one character which determines which command the master requires to be performed by the slave. Reply messages normally do not contain the CMD component.

[DATA] - Command Message or Reply Message Data

The number of DATA characters and the values they represent are determined by the particular message in which they are contained. Some messages do not contain the DATA component.

[CHKSUM] - Checksum Value

The CHKSUM consists of two characters representing a hexadecimal value between 0 and 255. This value is used to verify that the message received has arrived intact and has not been modified through noise or some other external factor. If this value is equal to two question marks, ("??") the CHKSUM check is disabled and is the message is assumed to be valid. To calculate the checksum perform the following steps:

1. Add the ASCII values of the characters in the message together. For example the ASCII value of an 'A' would be 65. Use only the **[DEST]** or **[SRC]**, **[CMD]** and **[DATA]** in the calculations. Do not include the **[START]**, **[END]** and **[CHKSUM]** characters into the calculations.
2. Keep only the eight <u>least</u> significant bits of the sum.
3. Complement the bits of the value by changing each 1 to a 0 and each 0 to a 1.
4. Add 1 to the value

Example: **[START]** ***"1A001"*** **[CHKSUM] [END]**

Add '1' + 'A' + '0'-+ '0' + '1' =	Bin #			Hex Value
49 + 65 + 48 + 48 + 49 = 259 Dec	0001	0000	0011	103
Keep only the eight least significant bits of the sum.		0000	0011	003
Complement the bits of the value by changing each 1 to a zero and each zero to a '1'.		1111	1100	0FC
Add 1 to the value.		0000	0001	001
Result:		1111	1101	0FD

The checksum for this example are the two characters "fd". To verify the checksum, add it to the total of all the characters and the least significant byte (8 bits) of the result should be zero.

fd Hex		1111	1101
+ 03		0000	0011
	0001	0000	0000

Verification (Lower 8 bits are equal to zero)

ACK - Positive Acknowledgment Character

ACK is the character used to indicate a reply message which is a Positive Acknowledgment of a Command Message. The ACK Character itself is an Exclamation Mark, **'!'** (ASCII (33)). An ACK message will contain only the ACK Character as data and informs the Master that the message just sent was received by the slave with the correct checksum, was a legal message, and was executed successfully. The ACK message is only used when no return data is expected from the Slave.

NACK - Negative Acknowledgment Character

NACK is the character used to indicate a reply message which is a Negative Acknowledgment of a Command Message. A NACK message informs the Master that the message just sent was received by the slave with the correct checksum, but could no be handled by the Slave for some reason. The NACK Character (**'?'** ASCII (63)) is the first character after the Controller Address in the NACK Message. The NACK message will also contain and error code to help the Master determine what was wrong with the command message. See the definition of ERRCODE below for a list of all possible errors. Please note that if a slave receives a message from a Master for which the checksum does not match, no NACK Message will be generated by the Slave.

ERRCODE - NACK Message Error Code

ERRCODE is a single character which represents a numeric error code in a Negative Acknowledgment (NACK) Message. The ERRCODE will be the first character after the NACK Character in a NACK Message. Listed below are the possible ERRCODE characters in a NACK Message:

'1' Format Error — This error normally occurs when a Command Message was received by the Slave which did not have the correct number of characters in it for the data. In other words the length of the Command Message was not correct for the desired command.

'2'	Busy Error	This error normally indicates that a valid Command Message was received, but due to operating conditions at the time of receiving the Command Message the Slave was unable to complete the desired command.
'3'	Access Error	This error normally occurs when a Command is sent to a Slave for which the Master is refused access to in the Slave. Access can be refused in different MC² Controller for different reasons. See the description of each individual message for a more specific cause of access NACK.
'4'	Bad Data	This error normally indicates that the data contained in the Command Message was not in the allowable range for that command. For example if you commanded the Slave to turn on Digital Output #17 and Digital Output #17 did not exist, a NACK Message with the ERRCODE of '4' would be generated.
'5'	Power Up Error	This error normally indicates that a Command Message was received by a Slave which has its Power-up Flag set. When a Power-up Flag is set in an MC² Controller it indicates that the Slave Controller has lost power since the last Command Message had been received by the Slave. Until the Master instructs the Slave to clear the Power-up Flag all other attempted communications with the Slave will result in a NACK Message with an error code of '5'. See a description of message 'i' which can clear the Power Up Flag in the Slave for more information.
'6'	Bad Command	This error normally indicates that a Command Message was received by the Slave which was valid except that the command contained in the message was not a valid command for the Slave Controller to execute. This can occur if different types of controllers are connected to one master and what could be a valid command for one controller may not be recognized by another.

END - End of Message

This is a single character which is sent to signal the end of the current message. The ASCII value of the END Character can be entered by the operator, but the default value of the END Character is 13 which represents an ASCII Carriage Return. Refer to the Operator's Manual for instructions on entering the End Character Value.

TELEGRAMS

All telegrams use the format presented in COMPUTER COMMUNICATIONS (page 6) for sending and receiving data.

Sent:	**[START] [DEST] [CMD] [DATA] [CHKSUM] [END]**
Sent:	LF 1 a 017 d6 CR
	Read register 023 CKSUM = d6 hex
Received:	**[START] [SRC] [DATA] [CHKSUM] [END]**
Received:	LF 1 0000000 ff CR
	The value of register 23 is 15 decimal.

NOTE: For the examples of telegrams following the start codes, source or destination, checksum and end codes will be left off for clarity.

Read Register Value 'a' (97)'

Request the value of a register in an MC² Controller. The register to be read must be sent as three digit hexadecimal characters. The MC² will return the value of the register in eight hexadecimal integer characters.

Message to Controller: ahhh - (hex)

a0f3 - read register 243.

Reply From Controller: hhhhhhhh - (hex)

00000403 - value in register 243 dec (0f3 h) is 1027 dec (403 h).

Get Model Identification 'c' (99)

The value read from this telegram contains Model of controller, which version this model is, the type of CPU used and the highest register number this application contains. The following chart shows the format used by the returning form of this telegram (where X represents hexadecimal characters).

Model	Vers	CPU	Hi Reg
XX	XX	X	XXXX

The Model number returns as two characters representing the model number is the following chart. The version number returns as two characters representing a hexadecimal integer. The CPU type returns either ‘1’ for a normal or ‘2’ for a fast CPU. Four characters are returned representing a hexadecimal integer for the highest register number of the application.

Error! Reference source not found. (page **Error! Bookmark not defined.**) contains a chart of Merrick Applications with the respective model numbers for each application.

Message to Controller: c - no parameters are sent with this message.

c

Reply From Controller: hhhhhhhhh - (hex)

264320139

26 - Model is a 30.00.HP

__43 - Version is ‘C’

____2 - CPU board is the fast type.

_____0139 - This model has 313 registers

Get Digital Status 'd' (100)

Read Status of Digital Inputs, Digital Outputs and General Alarm Values The following chart shows the format used by the returning form of this telegram (where X represents hexadecimal characters).

DIN	DOUT	ALARMS
XX	XXXX	XXXX

The next two charts show how to read the Digital I/O portion of the telegram.

Digital Input the first character is not used.

N/A	N/A	N/A	N/A	DIN4	DIN3	DIN2	DIN1

Digital Output - characters 1 and 2 are not used.
The following is the format for characters 3 and 4.

N/A	DO7	DO6	DO5	DO4	DO3	DO2	DO1

See Digital I/O in Appendix (page 34) for digital I/O tables for various models of controllers.

Message to Controller: d - no parameters are sent with this message.

d

Reply From Controller: hhhhhhhhhh - (hex)

0300300000

03 - Digital Inputs 1 and 2 are closed

__0030 - Digital Outputs 5 and 6 are closed

______0000 - There are no General Alarms

Read Front Panel 'e' (101)

Read Upper Display, Lower Display and LEDs from front panel.

Format of returned message.

Upper Display	Lower Display	LEDs
CCCCCCCCC	CCCCCCCCCCCCCCC C	XXXX

LED format Upper Byte. The upper four bits are used for the Alarm LED.

Alarm LED	Alarm LED	Alarm LED	Alarm LED	Green 4	Green 3	Green2	Green1

LED format Lower Byte.

Yel 8	Yel 7	Yel 6	Yel 5	Yel 4	Yel 3	Yel 2	Yel 1

Message to Controller: e - no parameters are sent with this message.

e

Reply From Controller: cccccccccccccccchhhh - (hex)

_____5.00Feedrate lb/min 0d40

_____5.00 - Upper display

_________Feedrate lb/min - Lower display

______________________0d40 - Grn Led’s 1, 3 &4 on, Yel Led 7 on

Read Calibration Parameters 'f' (102)

Read # of Decimal places for Rate, Speed, Load, Belt, and Totals. Also read Weigh Span Calibration Value and EMT Divide Value.

The first 5 characters represent the decimal points used by the controller. The following chart shows the format for the decimal points used.

Spee d	Feedrate	Belt Length	Load/Weight	Total
X	X	X	X	X

For Controller models 20.00.K, 22.00.B and 30.00.D the following format is returned when a 'f' is sent:

Decimal Points	Weigh Span	EMT Divide Value
XXXXX	XXXX	XXXX

The next 4 characters represent the Weigh Span value (in Hex).

The last 4 characters represent the EMT Divide value (in Hex).

For models 10.00.HP.O through B and 20.00.HP.O through B the following format is returned when this telegram is issued to the controller:

Decimal Points	N/A	Scale Counts
XXXXX	0000	XXXXXXXX

Note: This telegram is not available for the following models:

10.00.HP.C 11.00.HP.O - A 20.00.HP.C 30.00.HP.O - C 35.00.HP.O

Message to Controller: f - no parameters are sent with this message.

f

Reply From Controller: hhhhhhhhhhhh - (hex)

2222203fc0064

2 - Speed DP's

_2 - Feedrate DP's

__2 - Belt length DP's

___2 - Load/Weight DP's

____2 - Total DP's

_____03fc - Weigh Span of 1020 dec

_________0064 - EMT divide value of 100 dec

Read Masterset Values 'g' (103)

Read Reset Flag, Feedrate, Total and Pacing Flag.

The format of the data returned is:

Reset Flag	Feedrate	Total	Pacing Flag
X	XXXXXXXX	XXXXXXX X	X

The First character represents whether the controller has had a possible change to a calibration or rerate parameter. If the value is '1' then the possibility exists that a parameter has been changed. The master device should read the calibration and rerate parameters if they are used in any calculations made by the master. The flag then can be reset by issuing a Clear Reset Flag 'K' (75) (page 16). If the value is '0' then no changes have occurred.

The next 8 characters represent the feedrate of the controller.

Note: On the 11.00.HP.A The gross weight is returned rather than the feedrate. The 11.00.HP.O does not use this telegram.

The next 8 characters represent the Total of the controller at the time the 'g' telegram was sent.

The last character determines if the controller status is in a condition for pacing to occur. If the character is a '1' then pacing can occur. If the character is a '0' then no pacing will occur. See the Operator Manual for your specific application to see if pacing is available. "Pacing" allows a system to react to a controller that cannot achieve its setpoint. When Pacing starts the setpoints of the other controllers in the system are reduced to maintain an accurate blend of ingredients at a lower system rate based on the controller which caused Pacing to occur.

Message to Controller: g - no parameters are sent with this message.

g

Reply From Controller: hhhhhhhhhhhhhhhhh - (hex)

1000003e80000e01c0

1 - Calibration or Rerate of controller has occurred

_000003e8 - Feedrate is equal to 10.00 (assuming 2 DP's)

_________0000e01c - Total = 573.72 (assuming 2 DP's)

_________________0 - No Pacing

Read Miscellaneous Values 'h' (104)

Read Reset Flag, Speed, Weight or Load, and Batch Total.

The controller will return data to the issuing device in the following format:

Speed	Weight or Load	Batch Total
XXXXXXXX	XXXXXXXX	XXXXXXXX

The numbers will be hexadecimal integers.

This command is only available on the following models:

10.00.HP.O - B   20.00.HP.O - B   20.00.K   22.00.B   30.00.D

Message to Controller: h - no parameters are sent with this message.

h

Reply From Controller: hhhhhhhhhhhhhhhhhhhhhhhh - (hex.)

2222203fc0064

000002c5 - 7.09 Belt Speed (assuming 2 DP's)

________000004d3 - 12.35 Load (assuming 2 DP's)

________________000001cd - 4.61 Batch Weight (assuming 2 DP's)

Clear Power-up Flag 'i' (105)

When AC Power is removed from and then reapplied to an MC² Controller, a Power-Up Flag is set. Until the Power-Up Flag is reset to zero all communications messages received by the controller (other than this one) will receive a Negative Acknowledgment Message containing the NACK Character '5'. This flag must be reset by this communications message before normal communications can resume. The purpose of the Power-Up Flag is to inform a Master that power had been lost by the controller and to make the Master acknowledge that this has happened by resetting the Power-Up Flag. In addition to resetting the Power-Up Flag, an initial Communications Timer Value must be sent to the MC² Controller with this message. The timer value is eight hexadecimal characters which represent a 32-bit integer value in tenths of a second. This value is used as a Master Communications Timer. Sending a value of 100 to the MC² using this or the 'k' message would indicate that a Master would be expected to send a valid communications message to the controller at least once every 10.0 seconds. If a valid communications message addressed to that particular MC² Controller is not received within the allotted time (initially set by this message) a General Alarm is generated which is called "Master Comm Lost!". Messages addressed to other controllers will be ignored as usual and will not reset the timer. This alarm is used to inform the operator that no communications has taken place in the allotted time frame. This could indicate a breakdown in the communications between a Master and the controller and should not be ignored. If you wish to disable the communications timer and its associated alarm, send a zero value to the MC² using this message or the "k" message.

Message to Controller: ihhhhhhhh - (hhhhhhhh is the hexadecimal number for Communication Timeout.)

i00000032 - Clear Power up flag and set Comm Timeout at 5.00 seconds

Reply From Controller: ! - no parameters are returned

Read Single Digital Output 'j' (106)

Read the Status (Open (0) or Closed (1)) of one Digital Output. See Digital I/O in Appendix (page 34) for a list of valid digital outputs.

Message to Controller: jhh - (hh is the hexadecimal number for the digital output to check.)

j02

Reply From Controller: 1 - Digital output 2 is ON.

Set Comm Timer Value 'k' (107)

The Communications Timer Value must be sent to the MC² Controller with this message. The timer value is eight hexadecimal characters which represent a 32-bit integer value in tenths of a second. This value is used as a Master Communications Timer. Sending a value of 100 to the MC² using this message would indicate that a Master would be expected to send a valid communications message to the controller at least once every 10.0 seconds. If a communications message is not received by the MC² Controller within the allotted time (initially set by message 'i'), a General Alarm is generated which is called "Master Comm Lost!". This alarm is used to inform the operator that no communications has taken place in the allotted time frame. This could indicate a breakdown in the communications between a Master and the controller and should not be ignored. If you wish to disable the communications timer send a zero value to the MC² using this message. In addition the time value may be set after a controller has lost power, using the "i" message. See the description of message 'i' in this chapter for more information on the Communications Timer.

Message to Controller: khhhhhhhh -

k00000032 -

Reply From Controller: ! - no parameters are returned

Transmit Last Message 'l' (108)

Requests that the MC² Controller re-transmit the last message.

Message to Controller: l - no parameters are sent with this message.

l

Assuming that _____5.00Feedrate lb/min 0d40 was the previous message.

Reply From Controller: _____5.00Feedrate lb/min 0d40

Set Register Value 'A' (65)

Change the value of a register in an MC² Controller.

format in hexadecimal:

Reg #	Value
XXX	XXXXXXXX

Message to Controller: Ahhhhhhhhhhh - (hex)

A00200002710 - Place 10000(dec) in register 002.

Reply From Controller: ! - no parameters are returned

Reset Controller 'C' (67)

Reset or Restart Controller. A single character is accepted as a command flag. If this flag is a '**1**' a "warm start" is accomplished. The controller would be in state just as if it had lost power and power was returned. No register values or other memory values would be modified. If the input flag is a '**2**', the controller will do a "cold start". This will reset all of the internal variables to default values preset at the factory. The unit would have to be rerated and calibrated after this was done.

Message to Controller: C# - (1 or 2)

C1 - Controller to perform “warm start”.

Reply From Controller: No Reply occurs

Lock Keyboard 'F' (70)

Locks keyboard so that no keys can be input from the front panel. Usually used with a Remote Keyboard Application. Most MC² Applications have a timer which will unlock the keyboard automatically if no keystroke commands are received within 90 seconds. Keystroke Commands are sent using Message 'G'. The keyboard of a remote controller may be unlocked by using Message 'H'. See a description of both Message 'G' and Message 'H' for further information.

Message to Controller: F - no parameters are sent with this message.

F

Reply From Controller: ! - no parameters are returned

Send Remote Keyboard Input 'G' (71)

Sends the value of a keystroke to the controller, causing the controller to perform any function just as if the key had been entered at the physical keyboard of the controller. Usually used with a Remote Keyboard Application. The keyboard of the remote controller must have been locked using the Message 'F' before it will accept a remote key input by using Message 'G'. See the description of Message 'F' and Message 'H' for more information.

Bit	MC² keyboard key	Hex Code	Binary Code
0	"1"	0001	0000 0000 0000 0001
1	"2"	0002	0000 0000 0000 0010
2	"3"	0004	0000 0000 0000 0100
3	"4"	0008	0000 0000 0000 1000
4	"5"	0010	0000 0000 0001 0000
5	"6"	0020	0000 0000 0010 0000
6	"6"	0040	0000 0000 0100 0000
7	"8	0080	0000 0000 1000 0000
8	"9"	0100	0000 0001 0000 0000
9	"0"	0200	0000 0010 0000 0000
10	"↑"	0400	0000 0100 0000 0000
11	"←"	0800	0000 1000 0000 0000
12	"→"	1000	0001 0000 0000 0000
13	"↓"	2000	0010 0000 0000 0000
14	"X"	4000	0100 0000 0000 0000
15	"ENT"	8000	1000 0000 0000 0000

Message to Controller: Ghhhh - no parameters are sent with this message.

G8000 - Send an [ENT] keypress to the controller.

Reply From Controller: ! - no parameters are returned

Unlock Keyboard 'H' (72)

Unlocks keyboard so that keys can be input from the front panel. Usually used with a Remote Keyboard Application to release a previously locked keyboard. See a description of Message 'F' and Message 'G' for more information.

Message to Controller: H - no parameters are sent with this message.

H

Reply From Controller: ! - no parameters are returned

Send Computer Setpoint 'I' (73)

Sends new Setpoint to MC² Controller which must be in Computer Setpoint Mode. The Setpoint is not changed until the controller receives the Use Setpoint 'J' (74) (page 16). This allows the Setpoint to be verified before making it permanent.

Note: For model 10.00.HP.C this telegram is used to change the High Flowrate alarm value. For model 11.00.HP.A this telegram changes the weight setting of Limit Switch 3. The new setting will not occur until the 'J' is sent.

Message to Controller: Ihhhhhhhh - (hex)

I00000444 - Send a new setpoint of 10.92 (assumes 2 DP's)

Reply From Controller: ! - no parameters are returned

Use Setpoint 'J' (74)

Commands controller to move the new Computer Setpoint from storage and make it the current Setpoint. This is normally used after a new computer Setpoint has been sent to the controller and has been verified..

Message to Controller: J - no parameters are sent with this message.

J

Reply From Controller: ! - no parameters are returned

Clear Reset Flag 'K' (75)

The Reset Flag in an MC² Controller is used to inform a user through communication that a change has been made to one of the Rerate or Calibration Values through the Menu system by the operator. This includes all values on the Rerate or Calibrate Menus as well as any values affected by the calibration routines. This flag is read using message 'g'. If this flag is set to a one, any Rerate or Calibration Parameters which were previously read through communications should be read again to insure that the information is current. For example if you are using the Weigh Span Value to scale registers read from the controller, if the reset flag is true (equal to 1) you should read the Weight Span Value again to insure that the current value is correct. After reading any Rerate or Calibration Parameters which are being used, this message should be used to clear the reset flag back to zero.

Message to Controller: K - no parameters are sent with this message.

K

Reply From Controller: ! - no parameters are returned

Emit Register Type Mask 'O' (79)

This telegram returns four hexadecimal characters representing the format of a particular register. See Property Word on page 26 for reading the property word returned by the controller.

Message to Controller: Ohhh - (hex)

O043

Reply From Controller: hhhh - (hex)

8112 - Integer which is included in checksum read always/write protected by needle switch with 2 DP's.

Emit Register Formatted 'W' (87)

Request the value of a register in an MC² Controller. The MC² will return the value as a string of characters representing a decimal value including a minus and a decimal point if applicable. This telegram is used to read a formatted register value without the need to scale the value or insert a decimal point. The response to this telegram is variable length rather than the fixed length hexadecimal value read using the 'a' telegram.

Message to Controller: Whhh -(hex)

W043 -

Reply From Controller: String - (variable length)

10.01

HARDWARE

This section describes MC² serial ports. For MC³ serial communication, refer to the MC³ hardware reference manual, at http://www.merrick-inc.com/mct.

Serial Interface Boards

A portion of the circuitry needed for Serial Communications is located on the CPU Board in the MC². However, for Serial Communications to take place in an MC² Controller, a Serial Interface Board is necessary.

Original Serial Interface Board

The Serial Interface Board contains the circuitry for both RS-232 (V28) and 20 mA Current Loop on the same board. It has a standard DB-25 connector attached to it. The Serial Interface Board can handle two serial channels with communication rates up to 9600 baud. The two channels may be independently configured for RS-232(C) or 20 mA current loop. Channel #1 is utilized for printer communications. Channel #2 is utilized for Computer Communication. The Serial Interface Board is mounted at the rear-upper right corner of the controller. The 25-pin connector protrudes through a rectangular opening in the rear cover. The Serial Interface Board is connected to the CPU Board by a ribbon cable. The ribbon cable plugs into the CPU Board at the top edge.

The operator must insure that the switches on the Serial Interface Board matches the Specification Sheet sent with the MC² Controller Documentation. There are two knife switches located on the Serial Interface Board. The switches are set for each port independently. They switch is set to the 'V' position for RS-232 and the 'C' position for 20 mA Current Loop. The Specification sheet sent with each MC² Controller will indicate the position of these switches.

Advanced Serial Interface Board

The Advanced Serial Interface Board contains the circuitry for RS-232 (V28), 20 mA Current Loop and RS-485 (duplex V11) on the same board. It has a standard DB-25 connector attached to it. The Advanced Serial Interface Board can handle two serial channels with communication rates up to 19200 baud. Channel #1 which is utilized for printer communications may be independently configured for RS-232(C) or 20 mA current loop. Channel #2 is utilized for Computer Communication and can be used for either RS-232 or RS-485 communications. The Advanced Serial Interface Board is mounted at the rear-upper right corner of the controller. The 25-pin connector protrudes through a rectangular opening in the rear cover. The Advanced Serial Interface Board is connected to the CPU Board by a ribbon cable. The ribbon cable plugs into the CPU Board at the top edge.

There are no switches which need to be set on this board. The Advanced Serial Interface Board detects which interface circuits are active and automatically switches to them. When using the RS-485 on Channel #2, up to 32 Advanced Serial Interface Boards may be connected together in a multi-drop system.

LEDs (Light Emitting Diodes)

Two different colors of LEDs are provided on the Advanced Serial Interface Board in order to provide some visual indication of the status of the two Channels. The LEDs are active regardless of which type of hardware protocol is being utilized. This includes RS-232, 20 mA Current Loop and RS-485.

D1 - Yellow LED which indicates Channel 1 Receive Line is active when illuminated.

D2 - Yellow LED which indicates Channel 2 Receive Line is active when illuminated.

D3 - Green LED which indicates Channel 1 Transmit Line is active when illuminated.

D4 - Green LED which indicates Channel 2 Transmit Line is active when illuminated.

Terminating Resistors

When connecting more than two serial devices with RS-485, it is necessary to remove the Terminating Resisters R6 and R7 from the Advanced Serial Interface Boards on the MC² Controllers which are not on the ends of the connection wiring. See Figure 5-8 in Chapter 5 for an example of this type of interconnection. The Figure shows three Slave MC² Controllers connected to a Master

MC² Controller. The Master MC² Controller begins the RS-485 wiring and the third MC² Slave Controller is at the end of the wiring. Only the Advanced Serial Interface Boards in those two controllers should have the Terminating Resisters installed. Terminating Resisters R6 and R7 are mounted in sockets for easy removal and installation.

R6 - 1/4 W, 249 Ohm Terminating Resister for RS-485 Channel #2 Transmit Lines.

R7 - 1/4 W, 249 Ohm Terminating Resister for RS-485 Channel #2 Receive Lines.

INTERCONNECTION

Merrick Interconnection Drawings

A Wiring Diagram supplied by Merrick is normally provided as a reference for connecting all components together in a system. Wiring diagrams for the Serial Communications are included in these. In addition to the Merrick Wiring Diagrams, this chapter provides generic examples of diagrams for interconnection of serial communications devices with the MC² Controller. These may not apply to your specific system and should be used as a reference and not as a replacement for the wiring diagrams supplied by Merrick.

Distances

The maximum cable length in a Merrick Communications system is based on the serial communications hardware specification being utilized. When utilizing RS-232 the cable must be less than 50 feet in length. . RS-232 is a point to point connection only. If a distance of greater than 50 feet is necessary, either 20 mA Current Loop or RS-485 is recommended. Both of these allow many serial devices to be connected together. In the case of 20 mA Current Loop the connection is in a loop. With RS-485 the connection is a bus connection. The overall cable length for these specifications can be up to 2000 feet without sacrificing signal integrity

Multiple MC² Controllers for Computer Communications on Channel #2

It is possible using 20 mA Current Loop or RS-485 for a Master Serial Device to communicate with more than one Slave MC² Controller. Each slave is addressed by the Master individually. A slave will ignore all commands from the master unless the destination field in the message matches the Controller # programmed into the Slave MC².

Examples: The following pages contain wiring examples for generic systems. Figures 5-1 and 5-2 present examples of connecting an IBM PC or compatible to one MC². Figures 5-3 through 5-5 present examples of connection of a Master MC² to one or more Slave MC². Figures 5-6 through 5-7 present examples of connection of a printer to the MC².

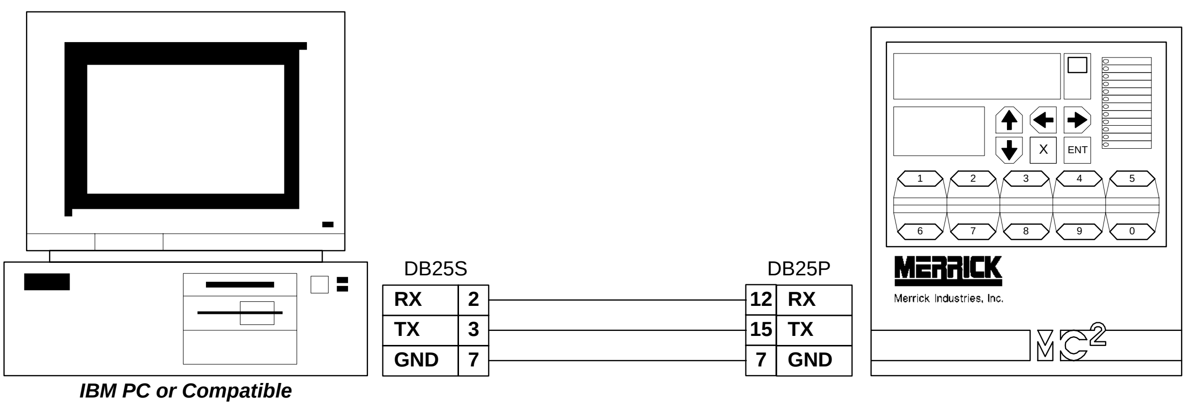


Fig. 5-1 Connecting a 25 Pin RS-232 PC with MC² Controller

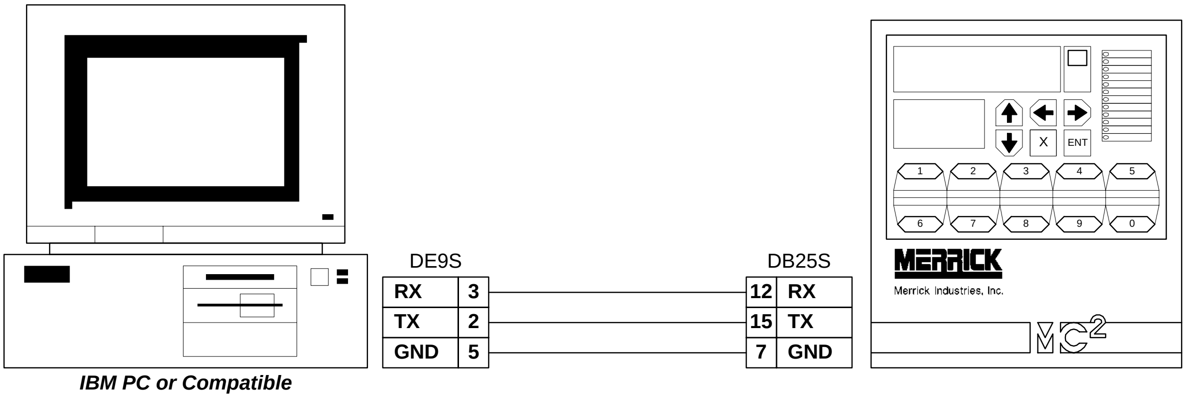


Fig. 5-2 Connecting a 9 Pin RS-232 PC with MC² Controller

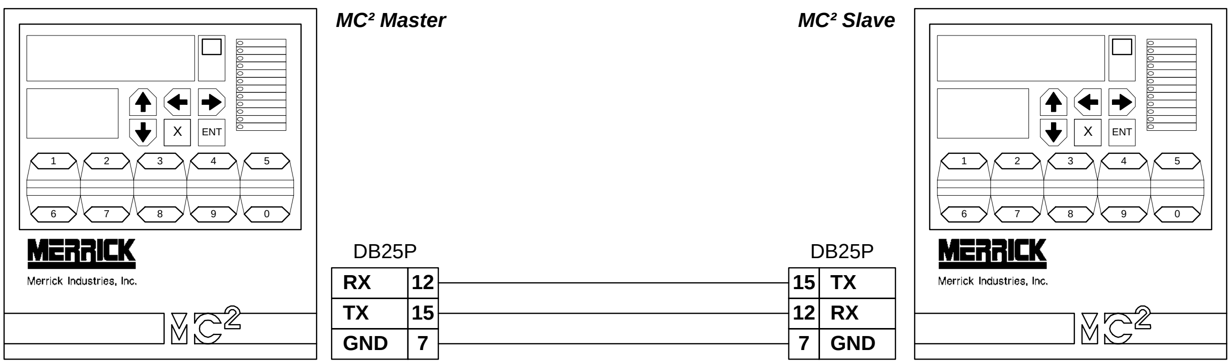


Fig. 5-3 Connecting 2 MC² Controllers via RS-232

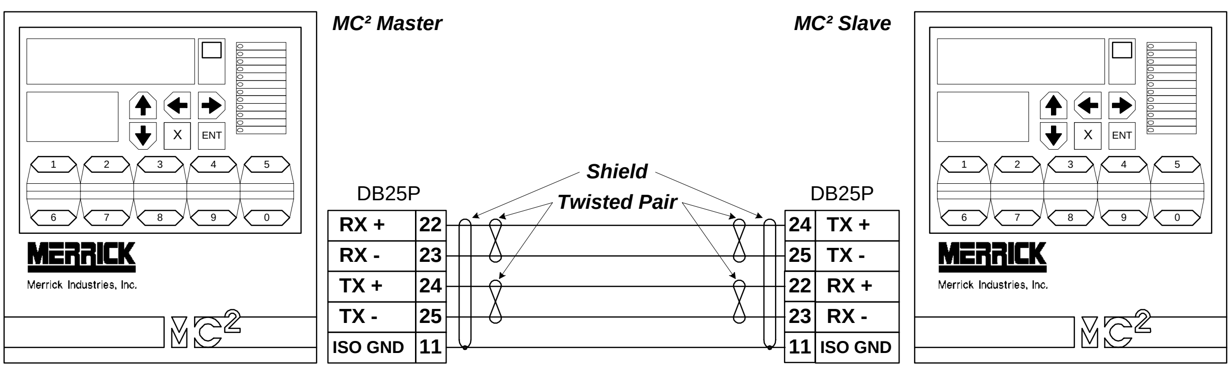


Fig. 5-4 Connecting 2 MC² Controllers via RS-485

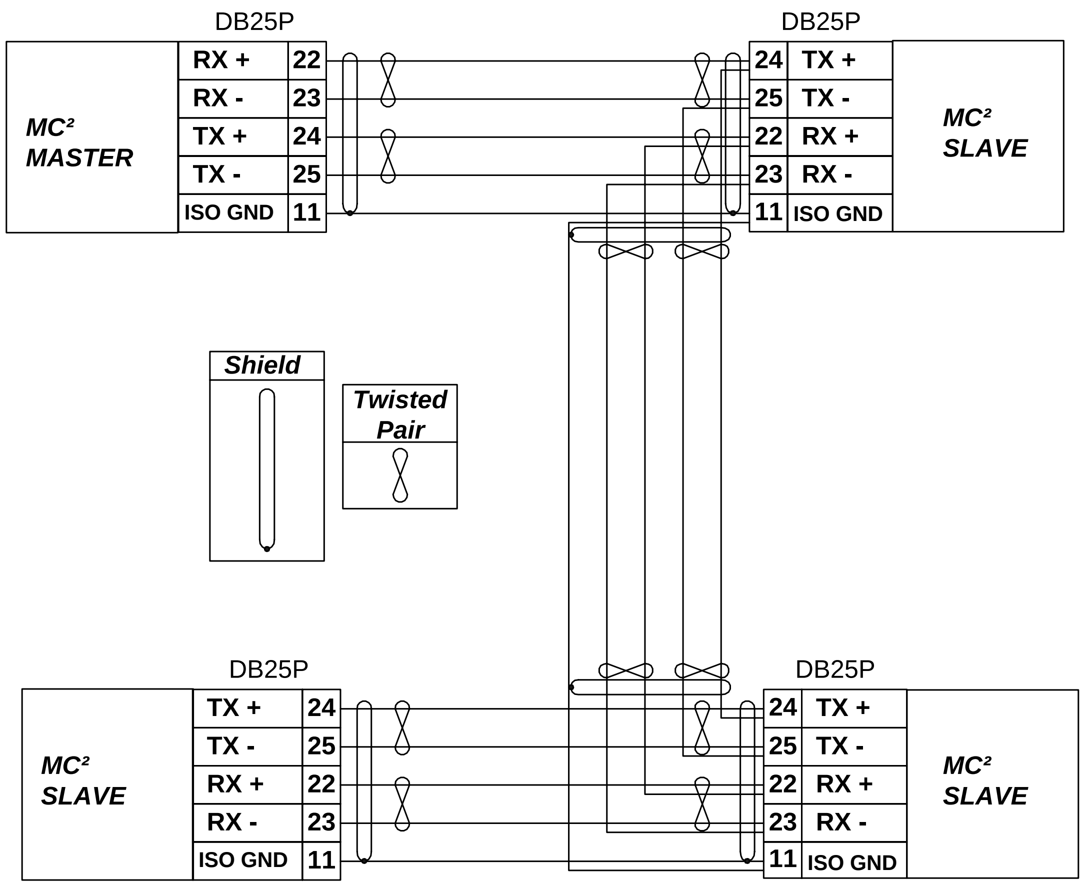


Fig. 5-5 Connecting Multiple MC² Controllers via RS 485 Interface

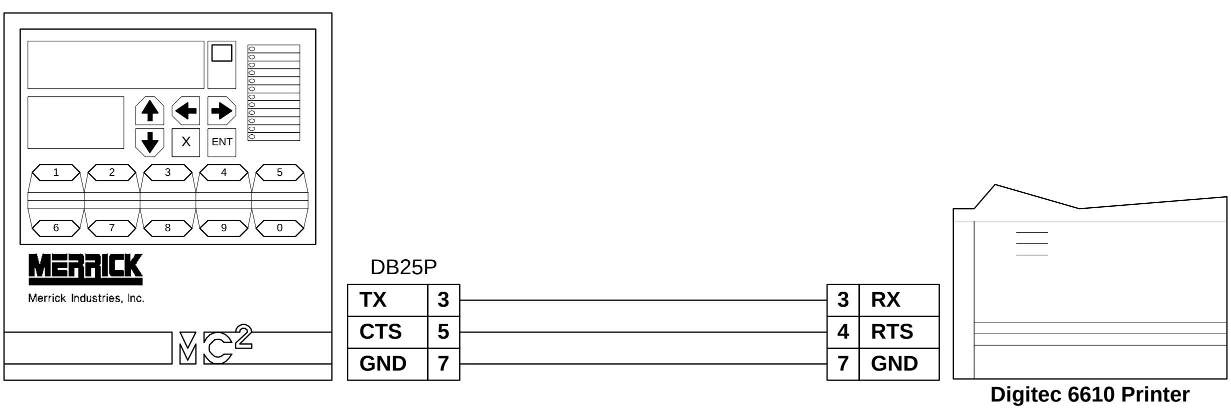


Fig. 5-6 Connecting a Digitec 6610 Printer to an MC² Controller

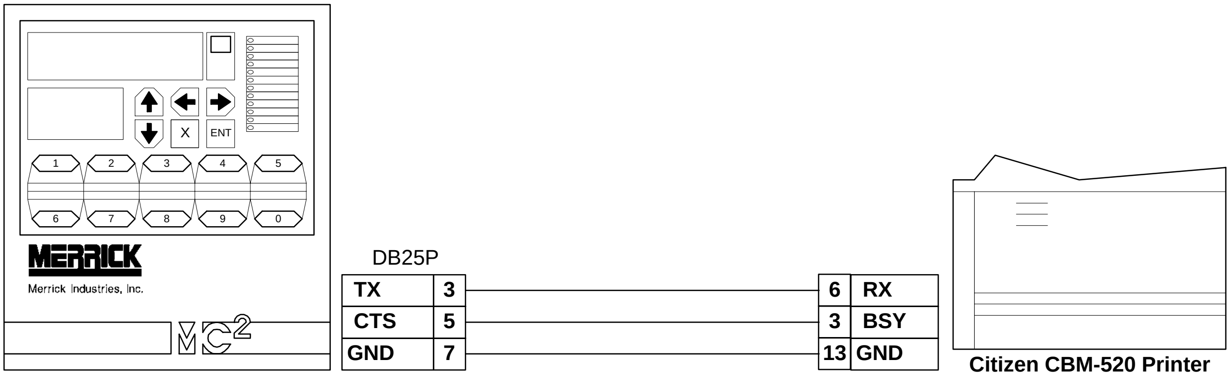


Fig. 5-7 Connecting a Citizen CBM 520 Printer to an MC² Controller

Serial Interface Boards - Connector Pinouts

Advanced Serial Interface Board Pin Assignments

Pin #	Code	Description
2	RxD1	Receive Data RS-232 Channel #1
3	TxD1	Transmit Data RS-232 Channel #1
4	RTS1	Request to Send RS-232 Channel #1
5	CTS1	Clear to Send RS-232 Channel #1
7	GND	Ground
9	+ 5VDC	
11	GND	Local RS-485 Ground
12	RxD2	Receive Data RS-232 Channel #2
13	Rx1 +	Receive Data 20 mA Current Loop Channel #1 (Positive)
14	Rx1 -	Receive Data 20 mA Current Loop Channel #1 (Negative)
15	TxD2	Transmit Data RS-232 Channel #2
17	Tx1 +	Transmit Data 20 mA Current Loop Channel #1 (Positive)
18	Tx1 -	Transmit Data 20 mA Current Loop Channel #1 (Negative)
19	RTS2	Request to Send RS-232 Channel #2
20	DTR	Data Terminal Ready (Both channels)
21	CTS2	Clear to Send RS-232 Channel #2
22	Rx2 +	Receive Data RS-485 Channel #2 (Positive)
23	Rx2 -	Receive Data RS-485 Channel #2 (Negative)
24	Tx2 +	Transmit Data RS-485 Channel #2 (Positive)
25	Tx2 -	Transmit Data RS-485 Channel #2 (Negative)

Original Serial Interface Board Pin Assignments

Pin #	Code	Description
2	RxD1	Receive Data RS-232 Channel #1
3	TxD1	Transmit Data RS-232 Channel #1
4	RTS1	Request to Send RS-232 Channel #1
5	CTS1	Clear to Send RS-232 Channel #1
7	GND	Ground
9	+ 5 VDC	

Pin #	Code	Description
12	RxD2	Receive Data RS-232 Channel #2
13	Rx1	Receive Data 20 mA Current Loop Channel #1 (Positive)
14	Rx1	Receive Data 20 mA Current Loop Channel #1 (Negative)
15	TxD2	Transmit Data RS-232 Channel #2
17	Tx1	Transmit Data 20 mA Current Loop Channel #1 (Positive)
18	Tx1	Transmit Data 20 mA Current Loop Channel #1 (Negative)
19	RTS2	Request to Send RS-232 Channel #2
20	DTR	Data Terminal Ready (Both channels)
21	CTS2	Clear to Send RS-232 Channel #2
22	Tx2 +	Transmit Data 20 mA Current Loop Channel #2 (Positive)
23	Tx2 -	Transmit Data 20 mA Current Loop Channel #2 (Negative)
24	Rx2 +	Receive Data 20 mA Current Loop Channel #2 (Positive)
25	Rx2 -	Receive Data 20 mA Current Loop Channel #2 (Negative)

Channel Reference Charts - Advanced

RS-232 Channel #1

Pin #	Code	Description
2	Rx	Receive Data
3	Tx	Transmit Data
4	RTS	Request to Send
5	CTS	Clear to Send

RS-232 Channel #2

Pin #	Code	Description
12	RxD2	Receive Data
15	TxD2	Transmit Data
19	RTS2	Request to Send
21	CTS2	Clear to Send

20 mA Current Loop Channel #1

Pin #	Code	Description
13	Rx1	Receive Data +
14	Rx1	Receive Data -
17	Tx1	Transmit Data +
18	Tx1	Transmit Data -

RS-485 Channel #2

Pin #	Code	Description
22	Rx2	Receive Data +
23	Rx2	Receive Data -
24	Tx2	Transmit Data +
25	Tx2	Transmit Data -

Miscellaneous

Pin #	Code	Description
7	GND	RS-232 Ground (Both Channels)
11	GND	Local RS-485 Ground
20	DTR	Data Terminal Ready (Both Channels)

Channel Reference Charts - Original

RS-232 Channel #1

Pin #	Code	Description
2	Rx	Receive Data
3	Tx	Transmit Data
4	RTS	Request to Send
5	CTS	Clear to Send

RS-232 Channel #2

Pin #	Code	Description
12	RxD2	Receive Data
15	TxD2	Transmit Data
19	RTS2	Request to Send
21	CTS2	Clear to Send

20 mA CURRENT LOOP Channel #1

Pin #	Code	Description
13	Rx1	Receive Data +
14	Rx1	Receive Data -
17	Tx1	Transmit Data +
18	Tx1	Transmit Data -

20 mA CURRENT LOOP Channel #2

Pin #	Code	Description
22	Rx2	Receive Date +
23	Rx2	Receive Data -
24	Tx2	Transmit Data +
25	Tx2	Transmit Data -

Miscellaneous

Pin #	Code	Description
7	GND	RS-232 Ground (Both Channels)
20	DTR	Data Terminal Ready (Both Channels)

REGISTER LISTS

All MC² applications contain a numbered table of parameters, called MC registers. They are useful in a variety of environments (exp. SuperBridge) for monitoring and control purposes. For example, message **'a'** will read an integer register value and message **'A'** will send a new integer value to a register. Message **'W'** will read a register with the value returned in floating point format including the sign and decimal point.

Property Word

The property word describes access rules, decimal places and scaling of the MC registers. The property word has the following layout:

Bit 15..14 — Internal storage format, according to the following table:

Bit 15	Bit 14	Storage format
0	0	long (32 bit integer)
0	1	char (8 bit integer)
1	0	int (16 bit integer)
1	1	float (32 bit IEEE floating point)

Bit 13..10 — Not used

Bit 9 — Set if the MC register is initialized to zero at controller cold start.

Bit 8 — Set if the MC register is included in the MC register checksum, that is, is safely retained when the controller is powered down.

Bit 7..6 — At least one of the bits are set if the MC register is scaled. SuperBridge will unscale the MC register and convert it into a float.

Bit 5..4 — Access mask. 00: Read and write permitted. 01: Read permitted. Write permitted if needle switch 1 is open on the display board. 10: Read access only. 11: No access.

Bit 3..0 — Decimal place codes.
0..4: 0..4 decimal places, respectively.
5..9 according to the values in MC registers 005..009, respectively, with the exception of models 20.00.K and 22.00, where the number of decimal places for code 009 is found in MC register 170, and 30.00.D, where the number of decimal places for code 009 is found in MC register 134

20.00.K

Reg	Prop	Register Content
002	8110	Scale Factor for load indication
005	8110	# Dec places for speed
006	8110	# Dec places for feedrate
007	8110	# Dec places for belt length
008	8110	# Dec places for load
020	0107	Belt length
021	0148	Design load
026	0146	Design Feedrate
029	0105	Design belt speed
030	0148	Tare Load
031	0248	Net load

Reg	Prop	Register Content
032	0248	Gross load
033	0109	Sub total
034	0109	Total
037	0205	Belt speed
039	0100	Tacho pulse counter
040	0246	Feedrate
041	0246	High resolution feedrate
083	8010	Power down counter
089	8210	General alarm status bits
118	0105	Internal speed setpoint
119	0101	Manual setpoint, %
120	0146	Internal feedrate setpoint

Reg	Prop	Register Content
121	4000	Setpoint selector
122	0101	External rate ratio %
123	0101	Controller Prop Band, %
124	0100	Controller Integral, s/reset
125	0102	Controller derivative, s
136	8202	Controller output, %
137	0202	Rate component of change in controller output, %
138	0202	Proportional component of change in controller output, %
139	0202	Integral component of change in controller output, %
163	8101	Low alarm delay, s
164	8101	Low setpoint deviation alarm limit, %
165	8101	High setpoint deviation alarm limit, %
166	8101	High alarm delay, s

Reg	Prop	Register Content
170	8100	# Dec places for belt total
182	0106	High feedrate alarm limit
183	0106	Low feedrate alarm limit
184	0105	High speed alarm limit
185	0105	Low speed alarm limit
186	8100	Alarm mode selector
187	8100	Totalizer cutoff mode selector
188	8100	Cutoff value, %
189	8101	Cutoff delay, s
210	8102	Allowed change in autotare, %
212	8101	Min load for autotare, %
214	8100	Autotare enable selector
216	8101	Autotare delay, s

22.00.B

Reg	Prop	Register Content
006	8110	# Dec places for feedrate
014	8101	Totalization cut-off, %
015	8101	Totalization cut-off flag
026	0146	Design feedrate
027	0146	Blend feedrate
033	0109	Subtotal
034	0109	Total
035	0100	Remainder for subtotal
036	0100	Remainder for total
040	0246	Feedrate
041	0246	High Resolution feedrate
089	8210	General alarm status bits
110	0101	Manual speed setpoint, %
111	0146	Internal setpoint
112	4000	Setpoint selector
113	0101	Setpoint multiplier, %
115	0101	Controller Prop Band, %
116	0101	Controller Integral, rep/min
117	0102	Controller derivative, s

Reg	Prop	Register Content
125	8202	Controller output, %
126	0202	Rate component of change in controller output, %
127	0204	Proportional component of change in controller output, %
128	0202	Integral component of change in controller output, %
163	8101	Low alarm delay, s
164	8101	Low setpoint deviation alarm limit, %
165	8101	High setpoint deviation alarm limit, %
166	8101	High alarm delay, s
187	8100	Totalization cut-off flag
188	8101	Totalization cutoff value, %
189	8101	Cutoff delay, %
202	0106	Actual setpoint

30.00.D

Reg	Prop	Register Content
002	8110	Scale Factor for weight indication
006	8110	# Dec places for feedrate
008	8110	# Dec places for weight
014	8101	Total cut-off value, %
015	8101	Total cut-off flag
019	8100	Auto-fill flag
021	0108	Fill weight
023	0108	Heel point
024	0101	Stabilization time, s
025	0148	Design weight
026	0146	Design feedrate
027	0101	Fill time, s
028	8101	Empty weight, %
029	8101	Clean-out time, s
030	0148	Tare weight
031	0248	Net weight
032	0248	Gross weight
033	0109	Sub total
034	0109	Total
040	0246	Feedrate
041	0246	High resolution feedrate
083	8010	Power down counter
089	8210	General alarm status bits
100	0101	Manual setpoint, %
101	0146	Internal feedrate setpoint
102	4000	Setpoint selector

Reg	Prop	Register Content
103	0101	External rate ratio %
104	0101	Controller Prop Band, %
105	0102	Controller Integral, s/reset
106	0102	Controller derivative, s
110	8202	Controller output, %
111	0202	Rate component of change in controller output, %
112	0202	Proportional component of change in controller output, %
113	0202	Integral component of change in controller output, %
127	8101	Low alarm delay, s
128	8101	Low setpoint deviation alarm limit, %
129	8101	High setpoint deviation alarm limit, %
130	8101	High alarm delay, s
134	8100	# Dec places for belt total
152	0106	High feedrate alarm limit
153	0106	Low feedrate alarm limit
154	8100	Feedrate alarm mode selector

10.00.HP

Reg	Prop	Register Content
002	0110	Scale Factor
005	8110	# Dec places for speed
006	8110	# Dec places for feedrate
007	8110	# Dec places for belt length
008	8110	# Dec places for load
009	8110	# Dec places for total
021	4100	HPAD cal setting
022	4100	HPAD gain setting
023	4100	HPAD tare setting
024	8110	HPAD ticks per sample
031	0200	Raw HPAD counts
038	0148	Design Load
040	0106	Design feedrate

Reg	Prop	Register Content
042	0106	Blend feedrate
044	0105	Design belt speed
045	0107	Belt length
046	0100	Pulses per belt rev
050	0148	Tare counts
051	0248	Net belt load
052	0248	Gross belt load
053	0205	Belt Speed
055	0200	Tacho pulse counter
057	0206	Feedrate
058	0206	High resolution feedrate
065	0119	Subtotal
066	0119	Total

Reg	Prop	Register Content
067	0118	Remainder for subtotal
068	0118	Remainder for total
132	8010	Power down counter
139	8210	General alarm status bits
217	0106	High feedrate alarm limit
218	0106	Low feedrate alarm limit
219	0105	High speed alarm limit
220	0105	Low speed alarm limit
221	8100	Alarm mode selector
222	8101	Low alarm delay, s

Reg	Prop	Register Content
225	8101	High alarm delay, s
230	8100	Totalizer cutoff mode selector
231	8101	Cutoff value, %
232	8101	Cutoff delay, %
235	8102	Allowed change in autotare, %
236	8101	Min load for autotare, %
237	8100	Autotare enable selector
238	8101	Autotare delay, s

11.00.HP

Reg	Prop	Register Content
003	0110	Scale Factor
005	8110	# Dec places for weight
006	8110	# Dec places for total
007	8110	# Dec places for enhanced resolution
021	4110	HPAD cal setting
022	4110	HPAD gain setting
023	4110	HPAD tare setting
025	8110	HPAD fixed ticks per sample
028	8210	External sync pulse counter
029	8110	External sync pulse divider
030	8110	Min ticks per samples external sync
031	8110	Max ticks per samples external sync
032	8110	External sync mode selector
038	0117	Design weight
039	0117	Overweight limit
040	0117	Underweight limit
041	0227	Absolute weight
042	0227	Gross weight
043	0227	Net weight
045	0227	Last stable weight

Reg	Prop	Register Content
046	0220	Raw HPAD counts
051	8220	Scale stable flag
055	0116	Total
056	0116	Subtotal
057	0117	Remainder for total
058	0117	Remainder for subtotal
076	4200	Printer transmitter status
082	0100	Item number for printout
083	0100	Item number increment
118	8010	Power down counter
124	8210	Actual alarm status
190	8110	Number of samples for stability
191	0107	Permitted span for stability
192	0107	Weight interval for center zero
230	0197	Zero tracking weight
231	0107	Max change in zero tracking weight
232	0107	Max zero tracking weight

20.00.HP, 24.81.HP

Reg	Prop	Register Content
002	0110	Scale Factor
005	8110	# Dec places for speed
006	8110	# Dec places for feedrate
007	8110	# Dec places for belt length

Reg	Prop	Register Content
008	8110	# Dec places for load
009	8110	# Dec places for total
021	4100	HPAD cal setting
022	4100	HPAD gain setting

Reg	Prop	Register Content
023	4100	HPAD tare setting
024	8110	HPAD ticks per sample
031	0200	Raw HPAD counts
038	0148	Design Load
040	0106	Design feedrate
042	0106	Blend feedrate
044	0105	Design belt speed
045	0107	Belt length
046	0100	Pulses per belt rev
050	0148	Tare counts
051	0248	Net belt load
052	0248	Gross belt load
053	0205	Belt Speed
055	0200	Tacho pulse counter
057	0206	Feedrate
058	0206	High resolution feedrate
065	0119	Subtotal
066	0119	Total
067	0118	Remainder for subtotal
068	0118	Remainder for total
139	8210	General alarm status bits
186	0105	Internal speed setpoint
187	0101	Manual setpoint, %
188	0106	Internal feedrate setpoint
189	4110	Setpoint selector
190	0101	External rate ratio %
191	0206	Analog input feedrate setpoint
192	0205	Analog input speed setpoint
193	0101	Controller Prop Band, %
194	0101	Controller Integral, rep/min
195	0102	Controller derivative, s
200	0202	Last controller raw output, %
203	8101	Max controller acceleration, %/s

Reg	Prop	Register Content
204	8101	Max controller deceleration, %/s
210	0202	Controller output, %
211	0202	Rate component of change in controller output, %
212	0204	Proportional component of change in controller output, %
213	0202	Integral component of change in controller output, %
214	0101	Belt speed used for tare and calibration procedures
217	0106	High feedrate alarm limit
218	0106	Low feedrate alarm limit
219	0105	High speed alarm limit
220	0105	Low speed alarm limit
221	8100	Alarm mode selector
222	8101	Low alarm delay, s
223	8101	Low setpoint deviation alarm limit, %
224	8101	High setpoint deviation alarm limit, %
225	8101	High alarm delay, s
230	8100	Totalizer cutoff mode selector
231	8101	Cutoff value, %
232	8101	Cutoff delay, %
235	8102	Allowed change in autotare, %
236	8101	Min load for autotare, %
237	8100	Autotare enable selector
238	8101	Autotare delay, s

30.00.HP

Reg	Prop	Register Content
003	0110	Scale Factor
005	8110	# Dec places for weight
006	8110	# Dec places for total
007	8110	# Dec places for enhanced resolution
008	8110	# Dec places for feedrate
021	4110	HPAD cal setting

Reg	Prop	Register Content
022	4110	HPAD gain setting
023	4110	HPAD tare setting
025	8110	HPAD fixed ticks per sample
028	8210	External sync pulse counter
029	8110	External sync pulse divider
030	8110	Min ticks per samples external sync

Reg	Prop	Register Content
031	8110	Max ticks per samples external sync
032	8110	External sync mode selector
040	0117	Design Weight
041	0117	Overweight limit
042	0117	Underweight limit
043	0118	Design feedrate
045	0227	Gross weight
054	8220	Scale stable flag
055	0116	Total
056	0116	Subtotal
057	0117	Remainder for total
058	0117	Remainder for subtotal
112	8200	Calibration menu flag
113	8210	General alarm status bits
121	8110	Number of samples for stability
122	0107	Permitted span for stability
140	4110	Setpoint Selector
141	0112	External rate ratio %
142	0118	Internal setpoint
143	0111	Manual setpoint in %
144	0218	Analog input setpoint
145	0118	Communications setpoint
146	0118	Preliminary comm setpoint
147	0218	Actual (used) setpoint
150	0100	State variable for LIW machine
151	8200	Manual fill request flag
165	8100	Alarm mode
166	0101	Low alarm delay
167	0101	High alarm delay

Reg	Prop	Register Content
168	0118	Low rate alarm
169	0118	High rate alarm
170	0101	Low setpoint deviation %
171	0101	High setpoint deviation %
175	0117	Empty weight
176	0117	Heel Point
177	0117	Fill Weight
200	0218	Actual Feedrate
205	0101	Controller Prop Band in percent
206	0101	Controller repeats/minute
207	0102	Controller rate time in secs
212	0202	Last Controller raw output in percent
215	8101	Max Acceleration percent/sec
216	8101	Max deceleration percent/sec
217	0202	Controller output signal
218	0202	Rate component of change in controller output
219	0204	Prop component of change in controller output
220	0202	Integer component of change in controller output
221	0100	Zero counts
226	0107	Calibration weight

35.00.HP

Reg	Prop	Register Content
003	0110	Scale Factor
005	8110	# Dec places for weight
006	8110	# Dec places for total
007	8110	# Dec places for enhanced resolution
021	4110	HPAD cal setting
022	4110	HPAD gain setting
023	4110	HPAD tare setting
025	8110	HPAD fixed ticks per sample
028	8210	External sync pulse counter

Reg	Prop	Register Content
029	8110	External sync pulse divider
030	8110	Min ticks per samples external sync
031	8110	Max ticks per samples external sync
032	4110	External sync mode selector
038	0117	Design weight
039	0117	Overweight limit
040	0117	Underweight limit
042	0227	Gross weight

Reg	Prop	Register Content
043	0227	Net weight
045	0227	Last stable weight
046	0220	Raw HPAD counts
051	8220	Scale stable flag
055	0116	Total
056	0116	Subtotal
057	0117	Remainder for total
058	0117	Remainder for subtotal
075	4200	Printer transmitter status
082	0100	Item number for printout
083	0100	Item number increment
124	8210	Actual alarm status
190	8110	Number of samples for stability
191	0107	Permitted span for stability
215	8100	Fill request flag
218	0197	Current batch total
223	0112	External rate ratio, %
224	0115	External batch setpoint
225	0115	Internal batch setpoint
226	0115	External batch setpoint
227	0115	Comm batch setpoint
228	8100	Setpoint selector

Reg	Prop	Register Content
229	0117	Batch tolerance
230	8100	Batch state variable
238	0117	Heel point
241	0117	Fill weight
242	0117	Current preact
243	0117	Max preact change
244	0117	Max preact absolute
245	0110	Preact adaptation, %
246	8101	Preact block selector
247	0101	Wait time before checking stability, s
248	0101	Max wait time for stability, s
249	0101	Max batch time, s
250	0101	Max fill time, s
251	8100	Ticks per sample when feeding
252	8100	Ticks per sample default
254	0101	Fine feed desired time
256	0117	Min batch weight
257	0117	Max batch weight
259	0115	Previous batch total

GENERAL ALARMS

Detailed information about the meaning of the alarms is available in the operations manual for the controllers.

The General Alarm bits are mapped out according to the following table. Detailed information about the meaning of the alarms is available in the operations manual for the controllers.

Controller	Bit	Meaning
20.00.K	1	Overflow
	2	A/D Overrange
	3	Auto-Tare Reject
	4	Master Comm Lost
	6	Display Failure
	7	Display Failure
30.00.D	0	A/D Overrange
	1	Hopper Empty
	2	Slow Fill
	3	Over Fill
	4	Master Comm Lost
	6	Display Failure
	7	Display Failure
22.00	4	Master Comm Lost
	6	Display Failure
	7	Display Failure
10.00.HP	1	A/D Overrange
	2	Auto-Tare Reject
	3	Master Comm Lost
	4	A/D Underrange
	5	Display Failure
	6	Display Failure
	7	HPAD Not Set-Up
	8	Test OverFlow
11.00.HP	1	Scale Overload
	2	Scale Underload
	3	A/D Underrange
	4	A/D Overrange
	5	Bad Tare
	7	HPAD Not Set
	8	Bad Low Display
	9	Comm Lost
20.00.HP	1	A/D Overrange
	2	Auto-Tare Reject

Controller	Bit	Meaning
	3	Master Comm Lost
	4	A/D Underrange
	5	Display Failure
	6	Display Failure
	7	HPAD Not Set-Up
	8	Test OverFlow
30.00.HP	1	Scale Overload
	2	Scale Underload
	3	A/D Underrange
	4	A/D Overrange
	5	Slow Fill
	6	Hopper Empty
	7	HPAD Not Set
	8	Bad Low Display
	9	Comm Lost
	11	Overfill
	14	No HPAD Data
35.00.HP	1	Scale Overload
	2	Scale Underload
	3	A/D Underrange
	4	A/D Overrange
	5	Stable Timeout
	6	Batch Timeout
	7	HPAD Not Set
	8	Bad Low Display
	9	Comm Lost
	10	Slow Fill
	11	Fill When Batch
	14	No HPAD Data

APPENDIX A

Digital I/O

The Digital I/O bits are mapped out according to the following table. Detailed information is available in the operations manual for the controllers.

Controller	I/O	Bit	Meaning
20.00.K	O	0	High Alarm
	O	1	Low Alarm
	O	2	Low Speed Cut off
	O	4	In Control
	O	6	General Alarm
	I	8	Soft Start
	I	9	Control Master Reset
30.00.D	O	0	High Alarm
	O	1	Low Alarm
	O	2	Filling
	O	3	Slow Fill
	O	4	In Control
	O	5	Feeder Running
	O	6	General Alarm
	I	8	Soft Start
	I	9	Control Master Reset
	I	10	Remote Fill
22.00	O	0	High Alarm
	O	1	Low Alarm
	O	4	In Control
	O	6	General Alarm
	I	8	Soft Start
	I	9	Control Master Reset
10.00.HP	O	0	High Alarm
	O	1	Low Alarm
	O	2	Low Speed Cut off
	O	4	Calibration
	O	6	General Alarm
11.00.HP	O	0	In Center Zero
	O	1	Scale Stable
	O	2	Print Complete
	O	3	Limit Switch 1
	O	4	Limit Switch 2
	O	5	Limit Switch 3
	O	6	General Alarm
	I	8	Print String A

Controller	I/O	Bit	Meaning
	I	9	Print String B
	I	10	Clear Sub-Total
	I	11	Tare
20.00.HP	O	0	High Alarm
	O	1	Low Alarm
	O	2	Low Speed Cut off
	O	4	In Control
	O	6	General Alarm
	I	8	Soft Start
	I	9	Control Master Reset
30.00.HP	O	0	High Alarm
	O	1	Low Alarm
	O	2	Filling
	O	3	Slow Fill
	O	4	In Control
	O	5	Feeder Running
	O	6	General Alarm
	I	8	Soft Start
	I	9	Control Master Reset
	I	10	Remote Fill
35.00.HP	O	0	Fast Feed
	O	1	Fine Feed
	O	2	Fill valve open
	O	3	Batch Out Of Tolerance
	O	4	Ready For Start
	O	5	Batch Complete
	O	6	General Alarm
	I	8	Remote Print
	I	9	Start Batch
	I	10	Stop / Reset Batch

Note that the status of unused digital inputs in MCs are reported to the Master device via telegram Get Digital Status 'd' (100) (page 10), even if they are not used by the MC application. They can be used as remote inputs for the ladder logic.

APPENDIX B

Internal State

Some cyclic controller applications have an internal state variable, useful for indication of what's going on in. The state variable is numerical, and can not be used for bit monitoring.

30.00.HP

State	Meaning
0	Check for fill requirement at startup
1	Prepare for normal feed
2	Wait for filter values to stabilize
3	Normal LIW feed
4	Prepare for a fill cycle
5	Filling
6	Check for auto-fill condition
7	Preparations after fill cycle
8	Stabilization time after filling

State	Meaning
9	Prepare for normal feed after filling
10	Prepare for Cleanout cycle
11	Run Cleanout cycle to low weight
12	Run Cleanout cycle (time) after low weight
12	Waiting for fill after Cleanout complete

35.00.HP

State	Meaning
0	Test for autofill
1	Stopped by button 7, "STOP BATCH"
2	Ready for start of new batch
3	Preparing for a batch
4	Prepare for mandatory wait before stable
5	Wait before stable, before batching
6	Stable check before batching
7	Check if fast feed needed
8	Start fast feed
9	Fast feeding
10	Check if skip fine feed
11	Start fine feed
12	Fine feeding

State	Meaning
13	Init wait after feed
14	Wait before stable after batching
15	Wait for stable after batching
16	Calc weight batched out so far
17	Calculate new preact
18	Prepare for filling
19	Arm timer before filling
20	Wait before fill
21	Wait for stability for filling
22	Start filling
23	Filling, check for overfill, done
24	Stop filling

APPENDIX C

Telegrams allowed by model.

Model	a	c	d	e	f	g	h	i	j	k	l	A	B	C	F	G	H	I	J	K	M	N	O	P	W
MC^2 20.00.K	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●					
MC^2 30.00.D	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	●	●						
MC^2 10.00.HP.O	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	2		●					
MC^2 10.00.HP.A	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	2		●					
MC^2 10.00.HP.B	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	2		●					
MC^2 10.00.HP.C	●	●	●	●		●		●	●	●	●	●		●	●	●	●	3	3	●			●		●
MC^2 10.00.HP.D	●	●	●	●		●		●	●	●	●	●		●	●	●	●	3	3	●			●		●
MC^2 10.00.HP.E	●	●	●	●		●		●	●	●	●	●		●	●	●	●	3	3	●			●		●
MC^2 S10.00.HP.C	●	●	●	●		●		●	●	●	●	●		●	●	●	●	3	3	●			●		●
MC^2 11.00.HP.O	●	●	●	●				●	●	●	●	●		●	●	●	●			●					●
MC^2 11.00.HP.A	●	●	●	●		●		●	●	●	●	●		●	●	●	●	3	3	●			●		●
MC^2 20.00.HP.O	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●					
MC^2 20.00.HP.A	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●					
MC^2 20.00.HP.B	●	●	●	●	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●					
MC^2 20.00.HP.C	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 20.00.HP.D	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 S20.00.HP.C	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 24.81.HP.O	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 22.00.B	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.O	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.A	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.B	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.C	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.D	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.E	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 30.00.HP.F	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^2 35.00.HP.O	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●			●		●
MC^3 24.10.EX.O	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.10.EX.A	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.96.EX.O	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●	●	●	●		●
MC^3 24.96.EX.A	●	●	●	●		●		●	●	●	●	●		●	●	●	●	●	●	●	●	●	●		●
MC^3 24.96.EX.B	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.96.EX.C	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.96.EX.D	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.96.EX.E	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 24.96.EX.F	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 30.10.EX.O	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 30.10.EX.A	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 30.20.EX.ß	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MC^3 40.10.EX.a	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

1. Refer to Read Calibration Parameters 'f' (102) on page 11 for proper format of message return.

2. This message is acknowledged, but has no effect on the controller.

3. Refer to Send Computer Setpoint 'I' (73) on page 16 for an explanation of this specific model's value to be changed.

APPENDIX D

Maximum Baud Rates for each model.

Model	***Max Baud***
MC² 20.00.K	9600
MC² 22.00.B	19200
MC² 30.00.D	9600
MC² 10.00.HP.O,A,B	9600
MC² 10.00.HP.C	19200
MC² S10.00.HP.C	19200
MC² 11.00.HP.A	19200
MC² 20.00.HP.O,A,B	9600
MC² 20.00.HP.C	19200
MC² S20.00.HP.C	19200
MC² 24.81.HP.O	19200
MC² 30.00.HP.O,A-D	19200
MC² 35.00.HP.O,A	19200
All MC³ Controllers	38400

APPENDIX E

Ascii Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00 Null	01 SOH ☺	02 STX ☻	03 ETX ♥	04 EOT ♦	05 ENQ ♣	06 ACK ♠	07 BEL •	08 BS ◘	09 HT ○	10 LF ◙	11 VT ♂	12 FF ♀	13 CR ♪	14 SO ♫	15 SI ☼
1	16 DLE ►	17 DC1 ◄	18 DC2 ↕	19 DC3 ‼	20 DC4 ¶	21 NAK §	22 SYN ▬	23 ETB ↨	24 CAN ↑	25 EM ↓	26 SUB →	27 ESC ←	28 FS ∟	29 GS ↔	30 RS ▲	31 US ▼
2	32	33 !	34 "	35 #	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 -	46	47 /
3	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
4	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
5	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
6	96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
7	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL