TRICONTINENT™
HANDLING LIQUIDS...PRECISELY

MC6000 Precision Pump
Software Manual

8694-15 REV. 3

MC6000 Software Interface

Table of Contents

Document History

Date	Changes
03/14/2011	Updated for firmware Version 08501-03V4 • Minimum argument for cutoff Velocity [c] reduced from 50 to 1 • Added ?7 ,?19, ?45 report commands added • [H] command was modified to trigger off the falling edge of the respective input. Previously, the inputs were level sensitive. Also a de-bounce parameter was added to the EEPROM • Updated Table 3: OEM communication details • [u_13] and [u_14] EEPROM configuration parameters added • Warning added to commands [k] and [K] if the sum of the two is set greater than 200 • Any spaces in the command string are now ignored by the pump
05/24/10	Updated for firmware Version 08501-03V3 • Number of EEPROM stall steps increased from 2 to 7(u7_7) • Maximum Backlash steps (K) argument increased from 64 to 255
11/24/09	• Initial version

MC6000 Hardware Communication Interface

The MC6000 has 3 separate communication interfaces:

1) RS485

2) RS232

3) CAN

RS232 and RS485 are identical from a software standpoint. The main difference between the two is that RS485 can communicate to more than one pump. RS232 is limited to a single pump communication. RS485 is sometimes referred to as multi-drop, whereas RS232 is single drop.

The MC6000 has a built in RS232 to RS485 converter. This allows for an RS232 connection to the host computer and RS485 connection to any additional pumps.

CAN communication is completely different from RS232/RS485. Not only is the hardware layer different, but the software protocol is unique. Please refer to CAN bus section for more detail.

MC6000 Addressing Scheme

As part of the communication protocol, an address for each pump must be specified. The user has the option of addressing a single pump, two pumps (dual device), four pumps (quad device), or all 15 pumps (all devices), depending on the address byte used. Each physical address on the address switch corresponds to a hexadecimal value, as shown in Table 2.

Table 1. Hexadecimal Addressing Scheme

Address (hex)		Device
RS232/ RS485	CAN	
30	0	Master Address (master controller, personal computer, etc)
31..3F	1..F	Addresses single device
41..4F	N/A	Addresses two devices at a time (dual device)
51..5D	N/A	Addressed four devices at a time (quad device)
5F	N/A	Addresses all devices on the bus

For example, a MC6000 with address switch set to 0 is addressed as device “31h” in the RS232 or RS485 communication protocol, hardware address 1 is addressed as device “32h”, and so on.

Table 2 shows the different address switch settings for each of these configurations.

Table 2. Address Switch Settings in Hex (ASCII)

Switch Setting	Single Device		Dual Device		Quad Device		All Devices	
	Hex Address	ASCII Address	Hex Address	ASCII Address	Hex Address	ASCII Address	Address	Value to Send
0 1	31 32	1 2	41	A	51	Q	5F	-
2 3	33 34	3 4	43	C				
4 5	35 36	5 6	45	E	55	U		
6 7	37 38	7 8	47	G				
8 9	39 3A	9 :	49	I	59	Y		
A B	3B 3C	; <	4B	K				
C D	3D 3E	= >	4D	M	5D	]		
E	3F	?	4F	O				
F	Unused							

The user can communicate with all pumps in the chain by using address “5Fh”, for example, to initialize all pumps at once. Then each pump can be controlled independently by using addresses “31h” to 3Fh.”

NOTE: Multiple address commands cannot be used to determine device status, nor will they respond to Report commands. Each device must be queried separately.

Communication Protocols

Three communication protocols are available:

- OEM communications protocol
- Data Terminal (DT) protocol
- CAN protocol

NOTE: The MC6000 firmware automatically detects the OEM, DT or CAN communication protocol. The Baud rate (9600 or 38400) is not automatically detected and needs to be set correctly.

The pump can be configured to not automatically detect the protocol. In this case, the user must use the specified protocol. See the [u] command for more details.

The DT (Data Terminal) protocol is designed to be used with a Terminal emulator program such as HyperTerminal, which comes installed with Microsoft Windows. For instructions on using a Microsoft Windows Terminal Emulator, see "Using DT Protocol with Microsoft Windows" in this chapter.

TCS has its own GUI software "TCS Commander" that can be used to control the pump.

NOTE: If not using a Terminal program. OEM protocol is recommended as it provides increased error checking and recovery.

OEM Communication Protocol

OEM communication is a robust protocol that includes automatic recovery from transmission errors. Table 3 describes each setting within the OEM communication protocol.

Table 3. OEM Protocol Communication Details

Parameter	**Setting**
Character Format	
Baud Rate	9600 or 38400
Data Bits	8
Parity	None
Stop Bit	1
Command Block (see "OEM Protocol Command Block Characters" for details)	
1	SYNC (FFh)
2	STX (Crtl B or 02h)
3	Pump Address
4	Sequence Number
5+n	Data Block (length n)
6+n	ETX (Crtl C or 03h)
7+n	Checksum
Answer Block (see "OEM Protocol Answer Block Characters" for details)	
1	STX (Crtl B or 02h)
2	Master Address (0 or 30h)
3	Status Code
4+n	Data block (length n bytes)
5+n	ETX (Crtl C or 03h)
6+n	Checksum

OEM Protocol Command Block Characters

The command block characters in the OEM communication protocol are described below. All characters outside the command block are ignored.

When developing a parsing algorithm, the programmer should key on the STX as the beginning of the answer block and the checksum (character after ETX) as the end of the answer block.

SYNC (FFh)

Used for backwards compatibility to older model pumps.

STX (^B or 02h)

The STX character indicates the beginning of a command string.

Pump Address

The pump address is specific, set by the rotary switch, for each individual pump.

NOTE: The pump's address is the rotary switch setting on the back of the pump plus one. For example, if the switch is set to 0, the pump's address is 1 or "31h".

Sequence Number/Repeat Flag

The sequence number is a single byte that conveys both a sequence number (legal values: 0 to 7) and a bit-flag indicating that the command block is being repeated due to a communications breakdown. The sequence number is used as an identity stamp for each command block. Since it is only necessary that every message carry a different sequence number from the previous message (except when repeated), the sequence number may be toggled between two different values (e.g., "1" and "2") as each command block is constructed. During normal communication exchanges, the sequence number is ignored. If, however, the repeat flag is set, the pump compares the sequence number with that of the previously received command block to determine if the command should be executed or merely acknowledged without executing.

NOTE: If the operator chooses not to use this option, the sequence number can be set to a fixed value of 1 (31h).

The following two scenarios clarify this error detection mechanism.

Scenario 1

1. The computer sends a command block stamped with sequence #1 to the pump.
2. The pump receives the command, sends an acknowledgement to the PC, and executes it.
3. Transmission of the acknowledgement message is imperfect; the PC does not receive it.
4. The PC waits 100 ms for the acknowledgement then retransmits the command block with the sequence number left at 1 and the repeat bit set to indicate a retransmission.
5. The pump receives the transmission, identified as such by the repeat bit.
6. The pump checks the sequence number against that of the previously received command block. Noting a match, the pump sends an acknowledgement to the PC, but does not execute the command (since it has already been executed).
7. The PC receives the acknowledgment and continues with normal communications.
8. The next command block is stamped with sequence #2 to indicate a new command.

Scenario 2

1. The computer sends a command block stamped with sequence #1 to the pump.
2. The pump never receives the command due to a communication error and thus does not send an acknowledgement to the PC.
3. The PC waits 100 ms for the acknowledgement then retransmits the command block with the sequence number left at 1 and the repeat bit set to indicate a retransmission.
4. The pump receives the retransmission, identified as such by the repeat bit.
5. The pump checks the sequence number against that of the previously received command block. Noting a mismatch, the pump recognizes this as a new command block and sends an acknowledgement to the PC. It then executes the command.
6. The PC receives the acknowledgement and continues with normal communications.
7. The next command block is stamped with sequence #2 to indicate a new command.

The sequence number/repeat byte is constructed as follows:

Bit #	7	6	5	4	3	2	1	0
Value	0	0	1	1	REP	SQ2	SQ1	SQ0

REP: 0 for non-repeated / 1 for repeated

SQ0 – SQ2: sequence value, as follows:

Sequence Value	**SQ2**	**SQ1**	**SQ0**
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Data Block (length n bytes)

The data block consists of the data or commands sent to the pump or host (this is an ASCII string). The [Q] command has a data block length of 0 (i.e. no data string exists).

ETX (^C or 03h)

The ETX character indicates the end of the command string.

Checksum

The checksum is the last byte of the message string. All bytes (excluding line synchronization and checksums) are XORed to form an 8-bit checksum. This is appended as the last character of the block. The receiver compares the transmitted value to the computed value. If the two values match, an error free transmission is assumed; otherwise, a transmission error is assumed.

OEM Protocol Answer Block Characters

The answer block characters in the OEM communication protocol are described below.

STX (^B or 02h)

The STX character indicates the beginning of a command string.

Master Address

The master address is the address of the host system. This should always be “30h” (ASCII “0”).

Status and Error Codes

The status and error codes define pump status and signal error conditions. For a description of status and error codes, see “Error Codes and Query Status”.

Data Block (length n bytes)

This is the response from all report commands with the exception of the [Q] command.

ETX (^C or 03h)

The ETX character indicates the end of the command string.

Checksum

The checksum is the last byte of the message string. All bytes (excluding line synchronization and checksums) are XORed to form an 8-bit checksum. This is appended as the last character of the block. The receiver compares the transmitted value to the computed value. If the two values match, an error free transmission is assumed; otherwise, a transmission error is assumed.

Data Terminal (DT) Protocol

The DT Protocol can be used easily from any terminal or terminal emulator capable of generating ASCII characters at 9600 or 38400 baud, 8 bits, and no parity.

Table 4. DT Protocol Communication Details

Character Format	
Baud Rate	9600 or 38400
Data Bits	8
Parity	None
Stop Bit	1
Command Block (see "DT Protocol Command Block Characters" for details)	
1	Start Character (ASCII "/" or 2Fh)
2	Pump Address
2+n	Data Block (length n)
3+n	End Character (Carriage Return ([CR] or 0Dh)
Answer Block (see "DT Protocol Command Block Characters" for details)	
1	Start Answer (ASCII "/" or 2Fh)
2	Master Address (ASCII "0" or 30h)
3	Status Character
3+n	Data block (if applicable)
4+n	ETX (03h)
5+n	Carriage Return (0Dh)
6+n	Line Feed (0Ah)

DT Protocol Command Block Characters

The command block characters in the DT communication protocol are described below.

Start Character

The start character (ASCII "/" or "2Fh") indicates the beginning of a message.

Pump Address

The pump address is an ASCII character specific to each pump.

NOTE: The pump's address is the rotary switch setting on the back of the pump plus one. For example, if the switch is set to 0, the pump's address is 1 or "31h".

Data Block (length n)

The data block consists of the ASCII data or commands sent to the pump or host.

End Character

The end character indicates the end of the message.

DT Protocol Answer Block Characters

The answer block characters comprising the DT communication protocol are described below.

Start Answer

The start character (ASCII “/” or “2Fh”) indicates the beginning of the response.

Master Address

The master address is the address of the host system. This should always be “30h” (ASCII “0”).

Status Character

The status and error codes define pump status and signal error conditions. See the description of the [Q] command in “Error Codes and Query Status.”

Data Block

This is the response from all report commands with the exception of the [Q] command.

ETX (^C or 03h)

The ETX character indicates the end of the command string.

Carriage Return (0Dh) and/or Line Feed (0Ah)

This character terminates the reply block.

Using DT Protocol with Microsoft Windows HyperTerminal Program

The MC6000 can be controlled in DT protocol mode directly from the Microsoft Windows HyperTerminal accessory.

To communicate with the MC6000 using Windows 95/NT/XP HyperTerminal, follow these steps:

1. To connect the MC6000 to a communication ports on the PC, first select the Start->All Programs->Accessories->Communications->HyperTerminal.
2. Enter a name for the connection and select an icon, then click OK. The Phone Number dialog box appears.
3. Select the following in the field provided:

   Connect using: <Correct COM port> (Usually Com1 or Com2)
   Click OK. The COM Properties dialog box appears.

4. Select the following in the fields provided:

Bits per second:	9600 or 38400 (jumper selectable, factory default = 9600)
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None
Click:	OK

5. Select the File menu, and choose Properties. The properties dialog box appears.
6. Select the Settings tab, and enter or select these options:

   Function, arrow, and Control keys act as:
   -Select “Terminal keys”

   Emulation:
   -Select “Autodetect”
   -Enter “500” in Backscroll buffer lines
   -Click the ASCII Setup button. The ASCII Setup dialog box appears.

7. Enter or select these options:

   -Select “Send line ends with line feed:
   -Select “Echo typed characters locally”
   -Enter a Line delay of “0”
   -Enter a Character delay of “0”
   -Select “Wrap lines that exceed terminal width”

8. Click OK to close the ASCII Setup dialog box, then click OK to close the Properties dialog box.
9. Set the pump address to 1 (rotary switch on back to 0) or the appropriate address. Ensure that the Baud rate jumper is set correctly.
10. Power on the pump and initialize it by typing “/1ZR” and pressing Enter.

To run the pump, see the commands listed in “Using the MC6000 Command Set Description”.

CAN Interface Communications

Can (Controller Area Network) is a two-wire, serial communication bus. It eliminates polling sequences to verify task completion. Using CAN, the pumps asynchronously report to the master, or host, when they have finished the current task.

The CAN communication rate is set using the [U] <n> command. The factory default is 100kBits/Sec or [U51].

NOTE: All TriContinent MC6000's use CAN controller chips compatible with Philips Semiconductor CAN bus specification, version 2.0.

CAN Messages

CAN messages consist of frames. Each frame has an 11-bit Message Identifier (MID), followed by a RTR bit and EXT bit, then 4-bits to indicate the data length, and finally by up to 8 bytes of data. If more than 8 bytes of data are required to be transferred, a multi-frame Message must be used.

Can Message Frame

Message ID (MID)				**Data Block (up to 8)**			
Dir	Group	Device	Frame	RTR	EXT	Data Length code	Data byte 0...8
1 bit	3 bits	4 bits	3 bits	1 bit	1 bit	4 bits	0...64 bits

CAN Message Construction

Each message frame begins with the 11-bit MID. The data block (up to 8 bytes in length) follows the MID and length information. The MID makes up three nibbles that are transmitted first in a Message frame. The bits are grouped as shown:

MID

Dir (1 bit)
This is the direction bit. It lets the devices on the bus know whether the current message is to or from the master. "0" means that the message is from master to slave, "1" means the message is from the slave to the master.

NOTE: Peer-to-peer messaging is not supported.

Group (3 bits)
This is the group number (0 – 7). Each type device on the CAN bus has a group assignment. The MC6000 is assigned to group 2. The group number "1" is reserved for the boot request procedure.

Device (4 bits)
This is the address of the module in the particular group. Each group can have up to 16 devices. The address value is 0 – 15.

Frame Type (3 bits)
This lets the device know what type message is coming. See "CAN Frame Types."

RTR (1 bit)
This bit is not used in TriContinent's CAN implementation and should always be set to 0.

EXT (1 bit)
This bit is should always be set to 0. Extended Frames are not allowed

Data Length Code (DLC) (4 bits)
This is the length of the data block in the message. Data blocks can be from zero to eight bytes in length.

Data Block(s)
From 0 to 8 bytes can be transmitted in a single frame.

CAN Frame Types

The frame types allow each device to know what type of command is coming in and enables faster processing of commands. Pumps respond to the frame types described below.

"On-the-Fly" Commands (V and T), Type 0

Normal commands use a frame type of "1" (i.e., "Action Commands"). Since commands sent over the CAN bus with a particular frame type must complete before a subsequent command using the same frame type can be issued, a different ID must be used when issuing an "on-the-fly" command. For this reason, "on-the-fly" commands must be issued over the CAN bus with a frame type of "0" (zero).

When using "on-the-fly" commands, the frame type "0" commands will not generate completion messages and thus no pairing code is needed (these commands are simply acknowledged immediately).

Action Frames, Type 1

This frame type is used for action commands, such as Initialization commands, Movement commands, Valve commands, or to set pump operating parameters. All "task-type" commands are sent in this type message frame. When multi-frame messages are used to send an action command, this frame is the end message sent to the pump.

Common Commands, Type 2

This frame is used for commands that are common to every device on the bus. The frame type is set to 2 and the command is a single ASCII character in the data block. The single ASCII character is described below.

Command	Description
0	Reset mode. This resets the pump and begins the boot request procedure.
1	Start loaded command. Just like sending an [R] command after a string has been loaded.
2	Clear loaded command. This clears out the command buffer.
3	Repeat last command. This command does the same thing as the [X] command.
4	Stop action immediately. This acts like [T] command.

Multi-Frame Start Message, Type 3

This frame type lets the pump know that the next message will be longer than the 8-byte maximum for each frame. Subsequent frames will follow to complete the message.

Multi-Frame Data, Type 4

This frame type is used to identify a frame in the middle of a multi-frame message. The last frame of a multi-frame message for action commands must be type 1. The last frame of a multi-frame message response from the pump for report commands will be type 6.

NOTE: There is no type 5 frame

Report/Answer Commands, Type 6

This frame type is used to get information back from the pump. It is similar in operation to the query commands (i.e. [?]) used in the OEM and DT protocols. The report command is one byte long and is a single ASCII character in the data block. Report commands in ASCII format are:

Command	Description
0	Report plunger position in steps (standard/microstep), like the [?] command in OEM or DT protocols
1	Report plunger position in steps, like the [?4] command
2	Report plunger position in steps (standard/microstep), like the [?5] command
3	Report current valve position in mnemonics, like the [?6] command; i=input, b=bypass
4	Report plunger end velocity in Hertz per half-step, like the [?2] command
6	Report start velocity, like the [?1] command
7	Report cutoff velocity, like the [?3] command
10	Report buffer status, like the [?10] and [F] commands; 0=empty, 1=commands in the buffer
12	Report number of backlash steps, like the [?12] command
13	Report status of input #1. Like the [?13] command
14	Report status of input #2, like the [?14] command
15	Nonfunctional, always returns to 1
16	Nonfunctional, always returns to 1
17	Nonfunctional, always returns to 1
18	Nonfunctional, always returns to 1
20	Report checksum, like the [?20] and [#] commands
22	Report liquid sensor value Nonfunctional command, always returns 255
23	Report firmware version, like the [?23] and [&] commands
24	Report zero gap steps, like the [?24] command
29	Report current status, like the [Q] command

When the pump responds to a query, the first byte of the data block is the status byte. It is defined like the status byte in the RS232 and RS485 protocols. The next byte is a null character. The remaining six bytes are for the response in ASCII. If the pump is only reporting current status, the message is only two bytes long. If the reply consists of more than six bytes, multi-frame messages are used.

CAN Data Block

The data block tells the pump what to do. Pump commands are sent in ASCII just like in RS232 or RS485 protocols. For command strings that are more than eight bytes in length, multi-frame messages are used. This permits long program strings to be sent as with other communications interfaces.

Handling of Pump Boot Requests

When the pump is first powered up or receives a system reset command (frame type 2, command 0), the pump notifies the host of this condition by sending a boot request message at 10 to 12 second intervals until it receives a proper response. The group number is 1 for the boot request message. The frame type is 2 when the pump sends messages to the host, and the frame type must be 0 when the host replies to the boot request.

Example 1. The pump is set to address 0

Pump sends:

Dir	Group	Device	Frame	RTR	EXT	Length
1	001	0000	010	0	0	0000

MID = 100 1000 0010 = 482 (hex)

Host acknowledges:

Dir	Group	Device	Frame	RTR	EXT	Length	Node ID	Slave ID
0	001	0000	000	0	0	0010	0010 0000	0010 0000

MID = 000 1000 0000 = 080 (hex)

Host acknowledges the boot request with:

Dir = 0	Host to slave
Group = 1	Boot request response group
Device = 0	Always 0 in boot response
Frame = 0	Boot request response frame
Rtr = 0	Always 0
Ext = 0	Always 0
Length = 2	Two data bytes in return message

Node ID	Group ID (2) + Pump Address (0)	20 (hex)	Must resend with Group & Address
Slave ID	Same as Node ID	20 (hex)	

Example 2. The pump is set to address 6

Pump sends:

Dir	Group	Device	Frame	RTR	EXT	Length
1	001	0110	010	0	0	0000

MID = 100 1011 0010 = 4B2 (hex)

Host acknowledges:

Dir	Group	Device	Frame	RTR	EXT	Length	Node ID	Slave ID
1	001	0000	000	0	0	0010	0010 0110	0010 0110

MID = 100 1000 0000 = 480 (hex)

Host acknowledges the boot request with:

Dir = 0	Host to slave
Group = 1	Boot request response group
Device = 0	Always 0 in boot response
Frame = 0	Boot request response frame
Rtr = 0	Always 0
Ext = 0	Always 0
Length = 2	Two data bytes in return message

NOTE: Boot MID is the same for all nodes

Node ID	Group ID (2) + Pump Address (6)	& (ASCII)	26 (hex)
Slave ID	Same as Node ID	& (ASCII)	26 (hex)

CAN Host and Pump Exchanges

When a slave pump receives a command, finishes a command, encounters an error condition or responds to a query, its sends an answer frame to the host using the same frame type as the command it belongs to. The answer frame format is device dependent. Generally, it will have the following format:

{MID} {DLC} {Answer}

Where:

{MID}: 11-bit message identifier. The direction bit is 1. The group number and the frame type are the same as received. Device is the current device address.

{DLC}: 4-bit data length code.

{Answer}: Data bytes block. The first byte of the data block is always the status byte. It is defined as in Table 5. The second byte is a null character. The remaining bytes contain the response in ASCII format. If the reply consists of more than six bytes, the multi-frame messages are used.

NOTE: Only one command of a given frame type can be in progress at any one time; e.g., after issuing a command to a slave pump with frame type = 1, the master must wait for the answer with frame type = 1 before issuing the next command with frame type = 1. If the user insists on sending the command, a command overload status results. Several commands with different frame types can be in progress at the same time; e.g., an action command and a query command.

Following are typical exchanges between the host and slave for action commands, multi-frame commands, common commands, and query commands.

Action Command

The host commands [ZR] to a pump, and the pump is set to address 0.

Host sends:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data Bytes
0	010	0000	001	0	0	0010	ZR

Pump Acknowledges:

Dir	Group	Device	Frame Type	RTR	EXT	DLC
1	010	0000	001	0	0	0000

After executing the command, pump reports status:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data Bytes
1	010	0000	001	0	0	0010	60 00 (hex)

Multi-Frame Command

The host commands [Z2S51A3000OgHD300G10G5R] to a pump, and the pump is set to address 0.

Host sends:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0000	011	0	0	1000	Z2S51A30 (ASCII)

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0000	100	0	0	1000	00OgHD30 (ASCII)

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0000	001	0	0	0111	0G10G5R (ASCII)

Pump acknowledges:

Dir	Group	Device	Frame Type	RTR	EXT	DLC
1	010	0000	001	0	0	0000

After executing the command, pump reports status:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0000	001	0	0	0010	60 00 (hex)

NOTE: For multi-frame commands, the pump only acknowledges the last frame.

Common Command

After the host has sent command [A1000A0] to the pump, it sends command 0 of frame type 2 to a pump and makes the pump move. The pump is set to address 0.

Host sends:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0000	010	0	0	0001	0

Pump acknowledges:

Dir	Group	Device	Frame Type	RTR	EXT	DLC
1	010	0000	010	0	0	0000

After executing the command, pump reports status:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0000	010	0	0	0010	60 00 (hex)

Query Command

The host sends report command 29 of frame type 6 to a pump, and the pump is set to address 1.

Host sends:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0001	110	0	0	0010	29 (hex)

Pump reports:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0001	110	0	0	0010	60 00 (hex)

NOTE: For query commands, no acknowledge frame is needed.

The host sends report command 23 of frame type 6 to a pump, and the pump is set to address 1.

Host sends:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
0	010	0001	110	0	0	0010	23

Pump reports:

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0001	011	0	0	1000	60 00 00 (hex)C300 (ASCII)

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0001	100	0	0	0111	00 024 00 (hex) O 10(ASCII)

Dir	Group	Device	Frame Type	RTR	EXT	DLC	Data bytes
1	010	0001	110	0	0	0111	00 024 00 (hex) 0109(ASCII)

NOTE: For a multi-frame reply, the start frame is type 3, the middle frame is type 4, and the last frame is type 6.

MC6000 Command Set Description

The MC6000 features a robust command set which allows a wide range of pump actions. Many of the commands have default arguments; however, the default values may not provide the optimal settings for your application. Take a moment to familiarize yourself with each command in order to obtain the best performance for your application.

For a quick summary of all commands, see the “Command Quick Reference.”

When problems are detected, the MC6000 sends an error code. The error codes are described in MC6000 Error Codes section.

Command Execution Guidelines

To use the commands properly, keep the following in mind:

- All commands, except Report commands and most Control commands must be followed by a [R] (Run or Execute) command.

- Single or multiple command strings can be sent to the pump.

- For example:
- A single command such as [A6000R] moves the plunger to position 6000.

- A multi-command string such a [IA6000OA0R] moves the valve to the Input position, moves the Plunger to position 6000, turns the valve to the Output position, and finally returns the Plunger to position 0.

- The pump’s command buffer holds a maximum of 255 characters. If a command is sent without the [R] (Run) command, it is placed into the buffer without being executed. If a second command is sent before the first command is executed, the second command overwrites the first command (i.e., the first command string is erased).

- Once a command is executed, new commands are not accepted until the sequence is completed. If a new command is issued while the pump is busy, a error code will be returned. Exceptions to this rule include the [T] (Terminate), “on- the fly” velocity commands such as [V] (Set Top Velocity) and all Report commands.

- When a command is sent, the pump answers immediately. If an invalid command has been sent in a command string, the pump reports an error immediately. If there was a invalid parameter, in certain cases, the string will execute until the invalid parameter and then stop. In this case, the [Q] (Query) command can be used to read back the error to the host computer.

- Always run liquid through the syringe and valve when issuing a Move command. Failure to do so will eventually wear out the valve and syringe seal.

- Keep fingers out of the syringe slot while the pump is running. Failure to do so can result in injury.

Command Syntax

The syntax for each command in the command set is:

[command]	Square brackets [ ] are used to distinguish commands. [ ] should not be sent as part of the command string.
<n>	Command argument. n must be a value within the specified range.

0...6000 Valid range of numerical <argument>

Default <argument>, if no argument is given

{n} Power up default value

NOTE: All commands are case sensitive.

Initialization Commands

NOTE: On power up, the MC6000 automatically homes the Valve motor. This feature can be disabled by using the pump configuration command [u].

Initialization Sequence

Initialization sequence, in response to the [Z] or [Y] command, is as follows:

1) Valve is homed to Output Left position [Z]
2) Syringe motor speed is set by Z<n> or Y<n> arguments.
3) If Plunger is at the top of stroke, the home opto is blocked, in this case
   a) Move Valve to input position
   b) Plunger moves down until home opto is cleared
4. Valve to Output, Plunger moves up until home opto is tripped
5. Syringe motor power is set by [x<n>] arguments. Plunger is then stalled at the top of the syringe.
6. Valve to Input, the syringe then travels downward a fixed number of steps.
7. Valve to Output, syringe moves up and stops at Syringe Dead volume <k> steps away from the stall position.

The output position of the valve is assigned to the left or right side, depending upon the Initialization command [Z] or [Y].

The speed at which the plunger stalls against the top of the syringe can be controlled via the [Z<n>] or [Y<n>] arguments. The force at which the plunger stalls against the top of the syringe can be controlled by the [x<n>] command.

Z<n> Initialize Pump, Set Valve Output to the Right

The [Z] command initializes the plunger drive and sets valve output to the right (as viewed from the front of the pump).

<n> 0...40
(n) 0 Default argument

The arguments are described below:

Command	Parameter	Description
Z	<n> = 0-9	Initializes using default speed (500Hz)
	<n> = 10-40	Initializes at the defined speed (see [S] command)

Z10-Z40 are initialization speeds which correspond to S10-S40 (Speed commands [S]), found in “Set Commands” later in this chapter. These commands can be used to change the standard initialization speeds. Slower initialization speeds may be useful when working with viscous fluids or small inside diameter (ID) tubing.

Y<n> Initialize Pump, Set Valve Output to the Left

The [Y] command initializes the plunger drive and sets valve output to the left (as viewed from the front of the pump). The parameters are described below.

<n> 0...40
(n) 0 Default argument

Command	Parameter	Description
Y	<n> = 0-9	Initializes using default speed (500Hz)
	<n> = 10-40	Initializes at the defined speed (see [S] command)

Y10-Y40 are initialization speeds which correspond to S10-S40 (Speed commands [S]), found in "Set Commands" later in this chapter. These commands can be used to change the standard initialization speeds. Slower initialization speeds may be useful when working with viscous fluids or small ID tubing.

W<n> Initialize pump without valve

The [W] command initializes the plunger drive but doesn't initialize the valve drive and no valve movements are performed during the initialization. [W] accepts the same <arguments> as the [Z] command.

k<n> Syringe Dead Volume command

The [k] command allows the setting of the number of steps that the plunger travels after initialization. This is to minimize dead volume. The syntax for this command is:

<n> The offset in steps from the zero or stall position
<n> 0...255
<n> 0...2040 in microstep mode
{122} Power up default

Under default initializations, the plunger moves upward until it contacts the top of the syringe, causing a forced stall initialization. The plunger then moves downward and upward, leaving a small gap between the syringe seal and the top of the plunger. This small gap was designed so that the Teflon seal does not hit the top of the plunger each time the syringe moves to the home or zero position. This maximizes the life of the syringe seal.

Since [k] only takes effect on initialization, the [k] command is usually followed by an initialization command [Z], [Y], or [W].

Each time the unit is powered down, the [k] value will return to the default setting of {122}.

For example, to offset 10 steps away from the zero position, send the following commands:
[k10ZR]

Note: That [k] can be set or reported in half steps or microsteps. With [N] = 1 or [N] = 2 mode, the setting is in micro steps. In [N] = 0 mode, the setting is in half-steps. Switch between the different modes will automatically scale the dead volume by 8. Since the dead volume value is an integer, multiplying and dividing by 8 can cause an quantization error if the dead volume is not a multiple of 8.

Note: If the sum of the BackLash steps [K] + Syringe Dead volume [k] is greater than 200, the plunger arm could hit the bottom of the frame when commanded to go the bottom of stroke, causing a Plunger Overload error. Thus, it is recommended to keep the sum of these values less than 200.

z<n> Simulated Plunger Initialization

The [z] command simulates an initialization of the pump, however, no mechanical movement of either the plunger or valve occurs. The position counter is set to <n>.

<n> 0…6000 in standard or non-microstep, mode.
<n> 0...48000 in microstep mode.
(n) 0 Default argument

x<n> Set Initialization force

The [x] command sets the initialization current when the plunger stalls against the top of the syringe. The argument <n> is a percentage of the plunger run current. The maximum initialization current is ½ of the run current. Thus, a change in the run current using either the EEPROM [U] command or the [m] command, will affect the initialization current.

<n>	25...100	Percentage of the full plunger stall force
{n}	100	Power up default

J<n> Set Initialization Speed

<n> 10..40

J10-J40 are initialization speeds which correspond to S10-S40 (Speed commands [S]), found in “Set Commands” later in this chapter. This command duplicates the function of the arguments for the [Z], [Y] or [W] commands. This command is intended to be used in XL3000 compatibility mode only.

Valve Movement Commands

I Move all Valves to Input position

The [I] command moves all 8 valves on the MC6000 to the Input position. The Input position can be either to the Left or the Right depending on the [Y] or [Z] initialization commands.

For example:
If the [I] command is sent after the [Z] command, the valve connects the syringe to the left side port (as viewed from the front of the pump).

Figure 1 shows the positions of the valves in relation to the initialization command and valve command used.

O Move all Valves to Output position

The [O] command moves all 8 valves on the MC6000 to the Output position. The Output position can be either to the Left or the Right depending on the [Y] or [Z] initialization commands.

For example:
If the [O] command is sent after the [Z] commands, the valve connects the syringe to the right side port (as viewed from the front of the pump).

B Move all Valves to Bypass position

The [B] command connects all 8 valves to the Bypass position. The Bypass position connects the Input to the Output position, bypassing the syringe.

NOTE: When the valve is in the Bypass position, the syringe plunger cannot be moved. Sending a Plunger Movement command causes an Error 11 (Plunger Move Not Allowed).

B<n> Valve movement using binary argument

<n> 00000000 ... 11111111

The B command allows the user to set each valve pair to its own individual position.

<n>	Valve Pair 4	Valve Pair 3	Valve Pair 2	Valve Pair 1
0000	Input	Input	Input	Input
0001	Input	Input	Input	Output
0010	Input	Input	Output	Input
0011	Input	Input	Output	Output
0100	Input	Output	Input	Input
0101	Input	Output	Input	Output
0110	Input	Output	Output	Input
0111	Input	Output	Output	Output
1000	Output	Input	Input	Input
1001	Output	Input	Input	Output
1010	Output	Input	Output	Input
1011	Output	Input	Output	Output
1100	Output	Output	Input	Input
1101	Output	Output	Input	Output
1110	Output	Output	Output	Input
1111	Output	Output	Output	Output
More that 4 arguments*				
XXXX0000	Input	Input	Input	Input
XXXX1111	Output	Output	Output	Output

***NOTE:**
X = don't care
If more than 4 arguments are given for the [B] command, Only the right most arguments are used. The rest are ignored. The maximum number of arguments is 8 in all cases.

NOTE: The MC6000 has a total of 8 valves. The valves are paired such that only each of the 4 valve pairs can be individually controlled.

As viewed from the front of the MC6000, the valves are numbered from left to right. The left most valve being the highest number:
8, 7, 6, 5, 4, 3, 2, 1

Valve Number	Valve pair
1 and 2	1
3 and 4	2
5 and 6	3
7 and 8	4

E<n> Valve movement using exponential argument

<n> 0 ..255

The E command allows the user to set each valve pair to its own individual position. The argument is a decimal valve from 0 to 15. E0 sets all valves to input whereas E15 set all valves to the output position.

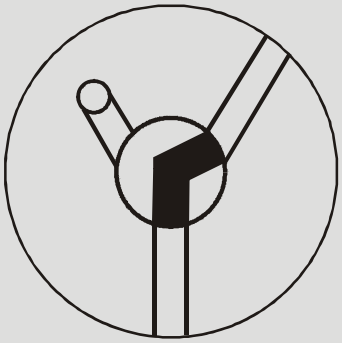
<n>	Valve Pair 4	Valve Pair 3	Valve Pair 2	Valve Pair 1
0	Input	Input	Input	Input
1	Input	Input	Input	Output
2	Input	Input	Output	Input
3	Input	Input	Output	Output
4	Input	Output	Input	Input
5	Input	Output	Input	Output
6	Input	Output	Output	Input
7	Input	Output	Output	Output
8	Output	Input	Input	Input
9	Output	Input	Input	Output
10	Output	Input	Output	Input
11	Output	Input	Output	Output
12	Output	Output	Input	Input
13	Output	Output	Input	Output
14	Output	Output	Output	Input
15	Output	Output	Output	Output
>15	Output	Output	Output	Output

Figure 1. Valve Positions for the 3-Port, 120°, Valve

Syringe Port position located
at bottom of Valve

3-Port Commands Using Z Initialization

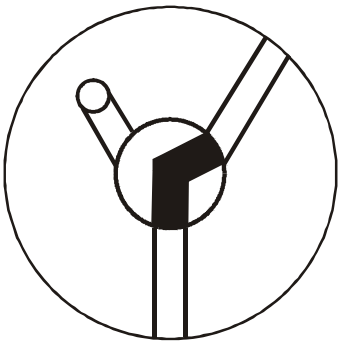
I Valve Command
sets Input to the Left



O Valve Command
sets Output to the Right

B Valve Command
Bypass Input to Output

3-Port Commands Using Y Initialization



I Valve Command
sets Input to the Right

O Valve Command
sets Output to the Left

B Valve Command
Bypass Input to Output

Plunger Movement Commands

A<n> Move Plunger to Absolute Position

The [A] command moves the plunger to the absolute position <n>.

<n>	0...6000	non-microstep mode
<n>	0...48000	microstep mode.
(0)		default argument

For example:
[A300R] moves the syringe plunger to position 300.
[A600R] moves the syringe plunger to position 600.

a<n> Move Plunger to Absolute Position (Not Busy)

This is the same as the [A] command, except the status bit within the reply string, and any subsequent Query [Q] commands, indicates that the pump is not busy.

P<n> Relative Pickup

The [P] command moves the plunger down the number of steps <n> commanded. The new absolute position is the previous position + <n>.

<n>	0...6000	non-microstep mode
<n>	0...48000	microstep mode.
(0)		default argument

For example:
The syringe plunger is at position 0. [P300] moves the plunger down 300 steps. A subsequent [P600] command moves the plunger down an additional 600 steps to an absolute position of 900.

The [P] command will return an Invalid Operand error if the final plunger position is greater than 3000 (48000 in microstep mode).

p<n> Relative Pickup (Not Busy)

This is the same as the [P] command, except that the status bit of the reply string, and any subsequent Query [Q] commands, indicates that the pump is not busy.

D<n> Relative Dispense

The [D] command moves the plunger upward the number of steps <n> commanded. The new absolute position in the previous position - <n>.

<n>	0...6000	non-microstep mode
<n>	0...48000	microstep mode.
(0)		default argument

For example:
The syringe plunger is at position 3000. [D300] will move the plunger up 300 steps to an absolute position of 2700.

The [D] command will return Invalid Operand error if the final plunger position will be less than 0.

d<n> Relative Dispense (Not Busy)

This is the same as the [D] command, except that the status bit of the reply string, and any subsequent Query [Q] commands, indicates that the pump is not busy.

Set Commands

Set Commands are used to configure the pump for specific operations. Plunger velocity and acceleration, motor current and Micro Stepping mode

Velocity and Acceleration Set Commands

Velocity and Acceleration Set commands are used to control the speed of the plunger. Plunger movement is structured into three phases:

- **Ramping Up**. Plunger movement begins with the start velocity and accelerates with the programmed slope to the constant or top speed.

- **Constant or Top Speed.** The plunger moves at the constant or top speed. Plunger speed or velocity can be programmed in Hz (half-steps/second) or in preprogrammed Set Speeds. The actual time the plunger travels it dependent on the ramping up and down. If the plunger move is short, it may never reach top speed.

- **Ramping Down.** The plunger will decelerate based on the programmed slope. To enhance fluid break-off, the Cutoff command [c] can be used to define the end velocity of the plunger just before it stops.

For each plunger move, the firmware calculates how many steps the plunger must travel during each phase in order to move the total number of steps commanded.

The top velocity can be changed on the fly (while the plunger is moving) using the [V] command. When the move completes, the speed value reverts back to its original values.

L<n> Set Acceleration Slope

During the beginning and end of a move, the plunger speed ramps up and down respectively. The ramp is programmed using the Set Slope [L] command. It is calculated as <n> x 2500 pulses/sec*sec. The syntax for this command is:

<n> 1...20 In Normal [N] = 0 or [N] = 1 mode
<n> 1...160 In velocity/acceleration microstep mode, [N] = 2
{7} Power up default

The units on the acceleration table below are in Kpulses/(sec*sec). In Half-step mode ([N] = 0 or [N] = 1) each pulse corresponds to a half-step. In Microstep velocity mode ([N] = 2), each pulse is equal to a microstep. Thus, the same [L] setting will be 8 times slower in [N] = 2 mode as opposed to [N] = 0 or [N] = 1 mode.

Slope Code	Kpulses/(sec*sec)
1	2.5
2	5.0
3	7.5
4	10.0
5	12.5
6	15.0
7	17.5
8	20.0
9	22.5
10	25.0
11	37.5
12	30.0
13	32.5
14	35.0
15	27.5
16	40.0
17	42.5
18	45.0
19	47.5
20	50.0

NOTE: In microstep velocity/acceleration mode [N] = 2, the maximum allowed argument is 160. The formula to convert the pulse code to pulses/(sec*sec) is:

Acceleration (pulses/sec*sec) = <n> * 2500

v<n> Set Start Velocity

The [v] command sets the velocity at which the plunger begins its movement. The plunger will then ramp up (slope) to the top velocity. The start velocity [v] normally should always be less than the top velocity [V]. If not, the plunger will actually decelerate to the top velocity [V] from its start velocity [v]. The syntax for this command is:

<n> 50...1000 half-steps/sec
<n> 50...8000, microsteps/sec in velocity/acceleration microstep mode, [N] = 2
{901} Power up default

V<n> Set Top Velocity

The [V] command sets the top speed in Hz (half-steps/second). The syntax for this command is:

<n> 1...6000 half-steps/sec
<n> 1...48000, microsteps/sec in velocity/acceleration microstep mode, [N] = 2
{901} Power up default

The top velocity can be changed on the fly, that is while the plunger is moving, using the [V] command. When the move completes, the speed value reverts back to its original value. Thus, on the fly velocity changes only affect the current move.

If the cutoff velocity [c] is higher than the desired top velocity [V], the cutoff velocity [c] is changed to equal the top velocity [V]. Note; the [c] cutoff velocity will not revert back to its original value if the top velocity [V] is changed back.

S<n> Set Speed

The [S] command sets the top velocity [V] to predefined speed codes shown in the table below.

<n> 0...40

These speeds do not cover the full range of speeds the plunger can travel. They are commonly used velocities provided for the convenience of the user.

The [S] command sets top velocity [V] without changing start velocity [v] , slope [L] and cutoff velocity [c] except under the following condition:

If the cutoff velocity [c] is higher than the desired top velocity [V], the cutoff velocity [c] is changed to equal the top velocity [V].

NOTE: The [c] cutoff velocity will not revert back to its original value if the top velocity [V] is changed back.

The units on the speed code are in pulses/second. In half-step mode ([N] = 0 or [N] = 1) each pulse corresponds to a half-step. In Microstep velocity mode ([N] = 2), each pulse is equal to a 1/8 half step. Thus, the same [S] setting will be 8 times slower in [N] = 2 mode as opposed to [N] = 0 or [N] = 1 mode.

Speed Code [S<n>]	Syringe Top Velocity [V] in N = 0 and N = 1 mode (pulses/second)
0	6,000
1	5,600
2	5,000
3	4,400
4	3,800
5	3,200
6	2,600
7	2,200
8	2,000
9	1,800
10	1,600
11	1,400
12	1,200
13	1,000
14	800
15	600
16	400
17	200
18	190
19	180
20	170
21	160
22	150
23	140
24	130
25	120
26	110
27	100
28	90
29	80
30	70
31	60
32	50
33	40
34	30
35	20
36	18
37	16
38	14
39	12
40	10

c<n> Set Cutoff Velocity

The [c] command sets the cutoff velocity. The cutoff velocity is the velocity at which the plunger ends its movement. The plunger will slope down [L] from the top velocity [V].

<n> 1 ...2700, half-steps/Sec
<n> 1 ...21600, microsteps/sec in velocity microstep mode, [N] = 2
{901} Power up default

The cutoff velocity [c] cannot be less than the top velocity [V]. If an attempt is made to set the cutoff velocity [c] less than the top velocity [V], the [c] cutoff velocity will be set equal to the top velocity [V].

K<n> Backlash steps

The [K] command sets the number of backlash steps <n>.

<n> 0...255
<n> 0...2040, in microstep mode, [N] = 1 or [N] = 2
{48} Power up default

When the syringe drive motor reverses direction, the plunger arm will not move until the backlash due to mechanical play within the system is compensated. To provide this compensation, during aspiration, the plunger moves down additional steps, then backs up the set number of backlash steps. This ensures that the plunger is in the correct position to begin a dispense move. Note that a small volume of fluid flows out the Input side of the valve during this operation.

Note: If the sum of the BackLash steps [K] + Syringe Dead volume [k] is greater than 200, the plunger arm could hit the bottom of the frame when commanded to go the bottom of stroke, causing a Plunger Overload error. Thus, it is recommended to keep the sum of these values less than 200.

N<n> Microstep Positioning and Velocity Mode

The [N] command enables or disables microstepping mode . In microstep position mode, plunger resolution is 8 times greater than normal or non-microstep mode. In microstep velocity mode, all velocity and acceleration parameters are set and reported in microsteps

This command also allows for velocity parameters [v], [V], [c] and [L] to be set and reported in microstep/sec units. This mode allows for finer control of the velocity settings. Please note that by changing between [N] = 0 or [N] = 1 mode and [N] = 2 mode, the existing velocity and acceleration settings are automatically changed. Thus, going to [N] = 2, with no change to the Velocity/Acceleration settings, will cause the plunger to move 8 times slower.

<n> 0...2
{0} Power up default, half-step mode

<n>	Description
0	**Normal mode**: Plunger position is set and reported in half-steps All velocity settings are in half-steps/sec Stroke is 6000 half-steps/full stroke
1	**Microstep position mode:** Plunger position is set and reported in microsteps Velocity settings are set and reported in half-steps/sec Stroke is 48000 microsteps/full stroke
2	**Microstep position and velocity/acceleration mode:** Plunger position is set and reported in microsteps Velocity and acceleration settings are set and reported in microsteps/sec Stroke is 48000 microsteps/full stroke

h<n> Set Syringe Motor Holding current

Sets the syringe's motor hold, or non-moving, current to the <value > specified in percent of maximum. On power up, this value will be overwritten to the factory set default value.

<n> 0…100 % of maximum current
{10} Power up default set at factory

m<n> Set Syringe Motor Run current

Sets the syringe motor run, or moving current, to the <value> specified <n> in percent of maximum. On power up, this value will be overwritten to the factory set default value.

<n> 0…100 % of maximum current
{100} Power up default set in EEPROM

J<n> Auxiliary Outputs

The [J] command sets the three auxiliary TTL output lines.

The syntax for this command is:

<n> 0...7
{0} Power up default

The MC6000 provides three TTL outputs on connector JP3 (pins 13, 14, 15) that correspond to outputs 1, 2, and 3. They are controlled as follows:

Command	Output 3 (pin 13)	Output 2 (pin 14)	Output 1 (pin 15)
J0	0	0	0
J1	0	0	1
J2	0	1	0
J3	0	1	1
J4	1	0	0
J5	1	0	1
J6	1	1	0
J7	1	1	1

(0 = low, Gnd; 1 = high, +5VDC)

j<ppppn> Set Auxiliary Outputs based on syringe position

The [j] command sets the three auxiliary TTL output lines to the specified state <n> when the actual syringe position is less than or equal to the specified syringe position

The syntax for this command is:

<pppp> 1…6000 Position value where Aux outputs will be set when the syringe position is less than or equal to this value. Note that step units are in either half-steps or microsteps depending on the microstep mode setting [N]

<n> 0...7 Sets TTL Output state, argument works the same as the [J] command

Normally this command is used during a dispense operation to provide an early trigger to another pump. This allows for continuous dispense flow with multiple pumps.
For example, using the following string:

[J0IA3000j5007A0GR]

J0 - Set all 3 auxiliary outputs low
I - Valve to Input position
A3000 - Aspirate, Move syringe down to bottom of stroke
j5007 - Set all 3 outputs high when syringe position is less than or equal to 500
O - Move Valve to Output
A0 - Dispense, Move syringe up to top of stroke, outputs will change state when syringe position is less than or equal to 500
G - Repeat forever
R - Execute command string

Control Commands

R Execute command sequence

The [R] command tells the pump to execute a new or previously loaded but unexecuted command string. This command will also cause the resumption of a halted [H] command string.

Commands containing [R] at the end of the string will execute immediately. If the command or program string is sent without the [R], it is placed in the command buffer.

Sending the [R] alone will execute the last unexecuted command in the buffer. Sending another [R] will not repeat the program string that has previously been executed.

X Repeat last executed command sequence

Repeats the last executed command string.

G<n> Repeat command sequence

This command repeats a command or program string the specified number of times. If a [GR] or a [G0R] is sent, the sequence is repeated indefinitely or until a [T] terminate command is issued. The [G] command can be used to nest up to 10 loops and can be repeated up to 30,000 times.

<n>	0...30000	# of times to repeat
(0)		Default argument, repeats indefinitely

Note that the argument <0> is a special case. It causes the loop to repeat indefinitely.

For example:

[A3000A0G10R] moves the syringe plunger to position 3000 then back to position 0. This sequence is repeated 10 times.

g Mark the Start of a Repeat Sequence

The [g] command is used in conjunction with the [G] command. The [g] command marks the beginning of a repeat sequence (loop) that occurs within program string (i.e., the entire string is not repeated). Both the [g] and [G] commands can be used to nest up to 10 loops.

An example of command nesting is shown below with the following string:

[A0gP50gP100D100G10G5R]

A0	-	Move plunger to position 0
g	-	Outer loop start
P50	-	Move plunger down 50 steps
g	-	Inner loop start
P100	-	Move plunger down 100 steps
D100	-	Move plunger up 100 steps
G10	-	Go to Inner loop start, repeat 10 times
G5	-	Go to Outer loop start, repeat 5 times
R	-	Execute command string

M<n> Delay Command Execution

The [M] command delays execution of a command in milliseconds. This command is typically used to allow time for liquid in the syringe and tubing to stop oscillating thereby enhancing precision. The syntax for this command is:

<n> 0...30000 milliseconds

H<n> Halt Command Execution

The [H] command is used within a program string to halt execution of the string. To resume execution, a [R] command or TTL signal must be sent.

<n> 0...2
(0) Default argument

Two TTL inputs are available, Input 1 (JP3 pin 7), and Input 2 (JP3 pin 8). They control execution as follows:

<n> = 0 Waits for [R] or either Input 1 or 2's falling edge
<n> = 1 Waits for [R] or Input 1's falling edge
<n> = 2 Waits for [R] or Input 2's falling edge

The status of the TTL input lines can also be read using [?13] and [?14]. These commands are described in Report Commands section.

Note: Both Input 1 and 2 are falling edge sensitive. That is, execution is halted until a High to Low transition occurs on the respective input. Firmware version prior to V4 were level-sensitive. Also, note that both inputs are internally pulled up to +5V

Note When using [H]<0> and using only 1 input, with the other input left floating. Since the unused input is pulled high, this eliminates the requirement for the active input to go high, the firmware will then look for the low-level on the active input. Thus, in this case, the inputs will appear to be level sensitive.

Auxiliary input de-bounce parameter added to EEPROM u13_X, where X<=50
Units are in mS. After the [H] command, the respective Input must be High continuously for at least X milliseconds before a Low transition will be recognized. Upon transition from High to Low, the falling edge, the input must remain Low for X milliseconds continuously before the string will resume operation. A special case is when the de-bounce delay parameter is set to zero. In this case the inputs will be level sensitive, not falling edge

For example, the following command string is sent:

[ZgH1IA1000H2OA0G0R]

The pump initializes, waits for the trigger, or Input 1 to go Low, then aspirates 1000 steps. It then waits for Input 2 to go Low before dispensing.

Z	-	Initialize pump
G	-	Start loop
H1	-	Wait for Input1 to go Low
I	-	Valve to Input position (Aspirate)
A1000	-	Move syringe to position 1000
H2	-	Wait for Input2 to go Low
O	-	Valve to Output position (Dispense)
A0	-	Move syringe to position 0
G0	-	Loop back to [g], run forever
R	-	Run

T Terminate Command Execution

The [T] command immediately terminates any executing command string. All plunger or valve movements are terminated immediately.

y<n> Execute next command in sequence based on Aux inputs

Execute the next command in the string based on the Auxiliary inputs.

<n> 0...3

<arg>	Input2	Input1	Action
<0>	Low	Low	Will execute next command in the string buffer if Input2 and Input1 are held Low
<1>	Low	High	Will execute next command in the string buffer if Input2 and Input1 are set as indicated
<2>	High	Low	Will execute next command in the string buffer if Input2 and Input1 are set as indicated
<3>	High	High	Will execute next command in the string buffer if Input2 and Input1 are both High

Input 1 is located on connector JP3 pin 7, and Input 2 on JP3 pin 8.

NOTE: Auxiliary inputs are internally pulled High. Thus, if left floating, they will be High.

The [y] command in conjunction with the [e] command can be used as an IF..ELSE, IF..ELSE statement based on the auxiliary TTL inputs. This is especially useful in stand-alone operation.

For example:
The example below allows the MC6000 in stand alone mode to switch between three modes of operation depending on the states of Input 1 and Input 2. The MC6000 is configured to Auto Run String 0 on power up.

If Input 2 is Low and Input 1 is High, [e1] will be executed. Thus, the pump will Aspirate/Dispense cycle 50 steps.

If Input 2 is High and Input 1 is Low, [e2] will be executed. Thus, the pump will Aspirate/Dispense cycle 1000 steps.

If none of the above states of Input 2 and Input 1 are realized, then [e3] will be executed. Thus, the pump will Aspirate/Dispense the full stroke of 3000 steps.

The following four strings are programmed into the EEPROM locations 0, 1, 2 and 3.

[s0Zy1e16y2e2e3R]
[s1gIA50OA0G0R]
[s2gIA1000OA0G0R]
[s3gIA3000OA0G0R]

String 0 [e0]
s0 Store following in EEPROM location 0 (executed on power up)
Z Initialize Pump
y1 Execute next instruction if Input 2 is tied Low and Input 1 is High (or left open)
e1 Jump to String 1
y2 Execute next instruction if Input 2 is High (or left open) and Input 1 is tied Low
e2 Jump to String 2
e3 Else, jump to String 3
R Run

String 1 [e1]
s1 Store following in EEPROM location 1
g Start loop
I Valve to Input position (Aspirate)
A50 Move syringe to position 50
O Valve to Output position (Dispense)
A0 Move syringe to position 0
G0 Loop back to [g], run forever
R Run

String 2 [e2]
s2 Store following in EEPROM location 2
g Start loop
I Valve to Input position (Aspirate)
A1000 Move syringe to position 1000
O Valve to Output position (Dispense)
A0 Move syringe to position 0
G0 Loop back to [g], run forever
R Run

String 3 [e3]
s3 Store following in EEPROM location 3
g Start loop
I Valve to Input position (Aspirate)
A3000 Move syringe full stroke to position 3000
O Valve to Output position (Dispense)
A0 Move syringe to position 0
G0 Loop back to [g], run forever
R Run

NOTE: Using the [e] command in a string will force a jump to the specified string, it will not return to the departure point once the "jumped to" string has completed.

Non-Volatile Memory (EEPROM) Commands

The non-volatile memory in the MC6000 can store up to 15 separate program strings. Thus providing the user with the option of computer-free stand-alone operation. The pump can be programmed to run stored programs using the [s] command.
On power up, if the Auto Run jumper is installed on the back of the pump, the string corresponding to the rotary switch position will be automatically executed.

For example:

The following 2 strings are programmed into location 0 and 1:
[s0ZIP1000H0OD1000R]
[s1ZP500H0OD500R]

The Auto Run jumper is installed on the back of the pump.
If the rotary switch on the back of the pump is set to 0, [s0] will run automatically. If the rotary switch is set to 1, [s1] will run automatically.

s<n> Load Program String into Non-Volatile Memory

The non-volatile memory in the MC6000 can store a program string thus providing the user with the option of computer-free operation. The [s] command is placed at the beginning of a program string to load the string into the non-volatile memory.

<n> 0...14

Up to 15 program strings (numbered 0 through 14) can be loaded into the non-volatile memory. Each string can use up to 128 characters. For example, [IA3000OA0R] requires 10 characters.

To run a stored string automatically on power up, the auto run jumper must be installed and the rotary address switch set to the proper address. The table below shows the relationship between the stored strings [s<n>] and the rotary switch.

Rotary Switch setting	String referenced
0	s0
1	s1
2	s2
3	s3
4	s4
5	s5
6	s6
7	s7
8	s8
9	s9
A	s10
B	s11
C	s12
D	s13
E	s14

The [e<n>] command can also be used to execute the string.

NOTE: An Initialization command should always be included in the non-volatile memory command string if the pump will be used in stand alone or auto run mode.

Example program string: [s8ZS1gIA30000H0OA0GR]

Command Segment	Description
s8	Loads string into the program 8 of non-volatile memory (Address switch position 8)
Z	Initializes pump
S1	Sets plunger speed
g	Marks start of loop
I	Turns valve to Input position
A3000	Moves plunger to position 3000
H0	Halt operation until either TTL inputs go Low
O	Turns valve to Output position
A0	Moves plunger to position 0
G	Go to [g], repeat indefinitely
R	Executes command string

e<n> Execute non-volatile memory program string

Execute EEPROM string stored at location <n>.

<n> 0...14

For example:

[e8R] will run the string stored in EEPROM location 8

Linking Program Strings in the Non-Volatile memory

Non-volatile memory program strings can be linked by ending one program string with an [e] command that refers to a second program string.

Example program strings: [sIZgIA3000OA0G5e2R]
[s2gIA3000OgH0D300G10GR]

The first string loads an initialization and prime sequence into program 1 of the non-volatile memory (address switch position 1). It then links to string 2 in the non-volatile memory.

The second string loads an aspirate and dispense sequence into program 2 of the non-volatile memory. The second non-volatile memory program string fills the syringe, then performs 10 dispenses of 300 steps each. Due to the [H0], the dispenses are triggered by the proper auxiliary input, or an [R] command. This sequence is repeated endlessly until the pump is powered down.

If the Auto Run jumper is installed and the address switch is set correctly, on power-up, the pump will automatically initialize, prime and perform the multiple dispenses until it is again powered down.

NOTE: When linking program strings, a jumped to string will not return to the calling string.
In the example above: [s1] jumps to [e2]. Once [e2] has executed, control will **not** return to the calling sting [s1]. Thus, [e] commands are normally placed at the end of a string just before the [R].

u<n_xxx> **Set Pump Configuration EEPROM Parameters** (For factory use only)

Loads pump configuration and calibration info into the internal EEPROM.
NOTE: These parameters are only read on power up. Thus they will only take effect when the power is cycled or the [!] command is given. Note this command, unlike the other Set commands, does not require and [R] to execute.

Command	Description	Factory Default
u1_XXX	Motor hold current, 0 – 100%	10
u2_XXX	Motor run current, 0 – 100%	100
u3_XXX	Plunger stroke in half-steps * 100	60
u4_X	X = 1, Automatically initialize valve on power up. X = 0, don't auto initialize valve	1
u5_X	CAN baud rate X = 0, CAN buss disabled X = 1, 100K X = 2, 125K X = 3, 500K X = 4, 1M	1
u6_XX	Power up default Backlash half-steps. [K] command	48
u7_XXX	Number of stall half-steps. On homing after plunger stops on the opto, number of steps to stall the plunger in the top of syringe. Units are half-steps * 100.	7
u8_XXX	Power up default Dead Volume half-steps. [k] command	122
u9_X	X = 1, AutoRun on power-up X = 0, Normal operation Normally set with [U30] and [U31] commands	0
u10_X	X = 0, Automatically detect OEM/DT protocol X = 1, OEM protocol X = 2, DT protocol	0
u11_X	X = 8, 8 valve pump X = 6, 6 valve pump X = 4, 4 valve pump	8
u12_X	X = 0, XMP6000 mode X = 1, XMP3000 X = 2, XL3000 compatible mode	0
u13_X	De-bounce delay for trigger inputs in mS. Valid arguments are 0-50. After the [H] command, the respective Input must be High for at least X milliseconds before a Low transition will be recognized. Upon transition from High to Low, the falling edge, the Input must remain Low for X milliseconds before the string will resume operation. An argument of zero is a special case. In this case the inputs will be level sensitive, not falling edge.	1
u14_XXXXXXX	7-character alpha-numeric	0000000

U<n> Set Pump Configuration EEPROM Parameters

The <U> command writes pump configuration parameters to EEPROM. Works similar to the [u] command, it is provided for firmware compatibility.

Argument <n>	Description
30	Sets auto-run into EEPROM. On power up, pump will run the stored string specified by the address switch. For example if the address switch is set to 1, the string in location 1 (set using the <s1> command) will be run. **NOTE**: Auto-run can also be set by installing a jumper on the back of the pump. If the jumper is installed, auto-run is enabled regardless of the EEPROM setting.
31	Clears auto-run from EEPROM
35	Auto detect DT or OEM serial protocol (Factory default)
36	OEM serial protocol only
37	DT serial protocol only
51	Set CAN baud rate to 100K (Factory default)
52	Set CAN baud rate to 250K
53	Set CAN baud rate to 500K
54	Set CAN baud rate to 1Meg
57	Set CAN baud rate to 125K

> <n1,n2> Set User Data Command

Writes specified user data in bytes to EEPROM
<n1> 0..15 Address where byte is stored
<n2> 0..255 Byte to be stored.

For example, the following command:
>10,120

Would set location 10 with a value of 120

User set data can be read out using the [>] command

! Reload Pump Configuration parameters from EEPROM

Set the MC6000 to use the updated EEPROM information. Avoids having to cycle the power when a pump configuration parameter, [u] command, is given.

Report Commands

Report commands report various pump parameters. The response is returned immediately and can be used when the pump is busy executing another command string.

Report commands do not require a [R] command

? or ?0 Report Absolute Plunger Position

Reports the absolute position of the plunger in half-steps, N = 0 mode (0…6000). In N = 1 or N = 2 mode, microsteps [0…48000]

?1 Report Start Velocity

Reports the start velocity [v] in pulses/sec [50…1000], [400...8000] N = 2 mode

?2 Report Top Velocity

Reports the top velocity [V] in pulses/sec [5...6000], [40...48000] N = 2 mode

?3 Report Cutoff Velocity

Reports the cutoff velocity [c] in pulses/sec [50...2700], [400...21600] N = 2 mode

?4, ?5 Report Absolute Plunger Position

Same as ?

?6 Report Valve positions

Reports the valve position in mnemonics (i = input, o = output, b = bypass)

?7 Report acceleration slope

Reports the acceleration slope set with the [L] command

?10 or F Report Command Buffer Status

Reports the command buffer status. If the buffer is empty, the pump returns status code 0. If the buffer is not empty, the pump returns a 1. If a program string is sent to the pump without an [R] command, the string is loaded into the buffer and the buffer status becomes 1. An [R] command will then execute the command stored in the buffer.

0 = empty
1 = commands in buffer

?12 Report Number of Backlash Steps

Reports the backlash steps [K]

?13 Report Status of Auxiliary Input #1 (JP3, Pin 7)

0 = Low (0 Volts)
1 = High (+5 Volts)

?14 Report Status of Auxiliary Input #2 (JP3, Pin 8)

0 = Low (0 Volts)
1 = High (+5 Volts)

?15 Non-functional

Always reports 1

?16 Non-functional

Always reports 1

?17 Non-functional

Always reports 1

?18 or % Report Number of Valve Movements (since last report)

Reports the number of valve movements since the last [?18] or [%] command.

?19 Reports if pump is initialized

If pump is not initialized with a Z,Y or similar command, will report 0. Reports 1 if initialized

?20 or # Report Firmware Checksum

Reports the firmware checksum.

?21 Motor Hold Current

Reports the syringe motor holding current, set with [h] ,in %.

?22 Motor Run Current

Reports the syringe motor running current, set with [m], in %.

?23 or & Report Firmware Version

Reports the firmware version. Format is "MC6000: MMDDYY".
MM = month
DD = day
YY = year

?24 Report the Dead Volume

Reports Dead Volume [k].

?25 Report the acceleration slope code

Reports the acceleration slope code set with [L].

?26 or * Report +24V

Reports the measured +24V input. The value is multiplied by 10. For example, if V = 24V, the response will be 240.

?27 or ?76 Report configuration EEPROM

Reports the confirmation EEPROM data set by the [u] command.

?28 Report Syringe step motor position mode

Reports the Plunger step mode as set by the [N] command.

?29 or Q Report pump status

Reports device status

?30 - ?44 Reports user programmed Auto-Run strings

[?30] reports [s0] string
[?31] reports [s1] string
[?32] reports [s2] string
[?33] reports [s3] string
[?34] reports [s4] string
[?35] reports [s5] string
[?36] reports [s6] string
[?37] reports [s7] string
[?38] reports [s8] string
[?39] reports [s9] string
[?40] reports [s10] string
[?41] reports [s11] string
[?42] reports [s12] string
[?43] reports [s13] string
[?44] reports [s14] string

?45 Reports EEPROM Alpha-numeric value

< Reports User set EEPROM data

Reports the valve of the user data stored at the specified location <n>.

Potential Software Incompatibilities with Competitor's Pumps

Command	Description	Discrepancy
Y<n>	Initialize pump, Set output to left	New feature
W<n>	Initialize pump without valves	New feature
B<n>	Set individual valve positions using Binary argument	Number of arguments, MC6000 controls valve by pairs. Thus only first 4 arguments are used.
E<n>	Set individual valve positions using Exponential argument	Argument range. MC6000 controls valve by pairs
h<n>	Set syringe motor hold current	New feature
m<n>	Set syringe motor run current	New feature
V<n>	Set Top Velocity	MC6000 allows a minimum speed of 1
y<n>	Execute next command in buffer based on Auxiliary inputs	New feature
U41 and U47	Set serial baud rate	Serial baud rate set by jumpers, thus this command has no effect on pump operation
u	Set pump configuration EEPROM parameters	New feature
j<ppppn>	Set Auxiliary outputs based on syringe position	New feature
#	Report Firmware checksum	New response
&	Report firmware version	New response
?6	Report Valve positions	New feature
?7	Report Acceleration slope	New feature
?15,?16 and ?17	Report various pump parameters	Not implemented on MC6000, however command will always report a "1"
?19	Reports if pump is initialized	New feature
?21	Report Motor Hold current	New feature
?22	Report Motor Run current	New feature
?30-?44	Reports user programmed Auto-Run strings.	New feature
?27	Report configuration EEPROM	New feature
?76	Report configuration EEPROM	New response

XMP3000 and XL3000 Compatibility Modes

Setting the MC6000 to XMP3000 compatibility mode [u12_1] causes the following differences:

- Stroke is limited to 3000 steps. 24000 in microstep mode.

Setting the MC6000 to XL3000 compatibility mode [u12_2] causes the following differences:

- Stroke is limited to 3000 steps. 24000 in microstep mode.
- Default Start Velocity is 186 pulses/sec
- Default Top Velocity is 454 pulses/sec
- Default Cutoff Velocity is 454 pulses/sec
- For the [Z]<n> and [Y]<n> commands, the argument <n> sets the number of valves. <N> = 4,6,8 valves. the [J]<n> command, where n = 10..40 sets the initialization speed to [S]<10> to [S]<40>.

MC6000 Status and Error Codes

The [Q] command reports error codes and pump status (Idle or Busy). The user should send a [Q] command before sending a program string or individual command to ensure that the pump has completed the previous command successfully.

The response to the [Q] command (the status byte) provides two items of information: Pump status (bit 5) and error code (bits 0-3).

Status Bit

Bit 5 is the status bit. It indicates when the pump is busy or not busy. The designations for bit 5 are listed below.

Status Bit 5	Description
X = 1	Pump is Idle. It is ready to accept new commands.
X = 0	Pump is Busy and will only accept Report, Terminate (T) or (V) commands.

In response to uppercase Move commands ([A], [P] and [D]), the [Q] command reports that the pump is Busy. In response to lowercase Move commands ([a], [p], and [d]), the [Q] command reports that the pump is Idle. Additionally, commands addressed to multiple pumps at once cannot be used to obtain pump status; each pump must be queried separately.

NOTE: Although the answer block for other commands contains a status bit, it should not be used for determining pump status. A [Q] command is the only valid method to determine if the pump is busy. The error information in the status byte of the answer block is always valid.

Error Codes

Error codes describe problem conditions that may be detected in the MC6000. Error codes are returned in the least significant four bits of status byte. If an error occurs, the pump stops executing commands, clears the command buffer, and inserts the error code into the status byte. Some errors continue to appear, such as Plunger Overloads, until they are cleared by the Initialization command. On a Plunger Overload, the device will not execute another plunger Move command until it is reinitialized.
The last error has precedence in the status byte. For example, if a Command Overflow occurs, an error 15 results. If the next command causes an error #3, the status byte reflects the error #3 (Invalid Operand).

MC6000 Error Codes

Error Code	Status Byte if Idle (ACSII)	Status Byte if Busy (ASCII)	Description
0 (00h)	`	@	**Error Free** Condition
1 (01h)	a	A	**Initialization error.** This error occurs when the pump fails to initialize. Check for blockages and loose connections before attempting to reinitialize. The pump will not accept commands until it has been successfully initialized. This error can only be cleared by successfully initializing the pump.
2 (02h)	b	B	**Invalid Command.** This error occurs when an unrecognized command is issued. Correct the command and operation will continue normally.
3 (03h)	c	C	**Invalid Operand.** This error occurs when an invalid parameter (<n>) is given with a command. Correct the parameter and pump operation will continue normally.
4 (04h)	d	D	**Invalid CheckSum.** In OEM mode, the checksum did not match the received string
6 (06h)	f	F	**EEPROM Failure.** This error occurs when the EEPROM is faulty. If you receive this error, please call TriContinent Customer Service.
7 (07h)	g	G	**Device Not Initialized**. This error occurs when the pump is not initialized. To clear the error, initialize the pump.
8 (08h)	h	H	**CAN bus failure**
9 (09h)	i	I	**Plunger Overload.** This error occurs when movement of the syringe plunger is blocked by excessive back pressure. The pump must be reinitialized before normal operation can resume. This error can only be cleared by reinitializing the pump
10 (0Ah)	j	J	**Valve Overload.** This error occurs when the valve drive loses steps by blockage or excess back pressure. The pump must be reinitialized before normal operation can resume. Sending another Valve command reinitializes the valve and sets it to the correct location. Continual valve overload errors could be an indication the valve should be replaced.
11 (0Bh)	k	K	**Plunger Move Not Allowed.** When the valve is in the bypass or throughput position, Plunger Movement commands are not allowed
15 (0Fh)	o	O	**Command Overflow.** This error occurs when the command buffer contains more than 255 characters. Commands in the buffer must be executed before more commands can be sent. A command overflow will also occur if the pump is busy executing a command and another command string is requested to run.

The pump handles errors differently depending on the error type. There are four error types, which are described below.

Immediate Errors

These include “Invalid Command” (error 2), “Invalid Operand” (error 3), “Invalid Checksum” (error 4), and “Plunger Move Not Allowed” (error 11). After the command is sent, the answer block immediately returns an error. Once a valid command is sent, the pump will continue to function normally. Since the [Q] command is a valid command, the pump will not return an error. In this case, the [Q] command is not required.
There is no need to reinitialize the pump following this error type.

Initialization Errors

These include “Initialization errors” (error 1) and “Device not Initialized” (error 7).

To ensure that the pump initializes successfully, send a [Q] command after the Initialization [Z], [Y] or [W] commands.

- If an error occurs during the initialization, an “Initialization error” (error 1) will be returned with the [Q]’s response. The pump must be reinitialized until the [Q} command indicates successful initialization.

- If the [Q] command indicates both a successful initialization and that the pump is ready, subsequent Move commands can be sent.

- If Initialization is not successful, or if any Move command is sent prior to an Initialization command, a “Device Not Initialized” error is returned with any Plunger Move command.

Overload Errors

These include the “Plunger Overload” and “Valve Overload’ errors (error 9 and 10). If the pump returns either a Plunger or Valve Overload, the pump must be reinitialized before continuing. If another command is sent without reinitializing the pump, another Overload error will be returned.

Command Overflow Error

This error occurs when certain commands are sent to the pump while it is busy executing a previous command string. Any Move, Set (except [V]), or Valve commands that are sent while the pump is busy, will cause this error to be issued. The pump ignores the command and issues a Command Overflow (error 15).
The [Q] command allows the controller to determine when the command is complete and the pump ready to accept new commands.
There is no need to re-initialize the pump following this error type.

Report commands, top velocity [V] and the terminate [T] command, will not return a Command Overflow error. These commands are considered valid even when the pump is Busy.

CAUTION: All errors reported by the pump should be captured by the user’s software and the physical cause corrected before continuing operation. Failure to do so may result in damage to the pump or adversely affect pump performance, and void the warranty.

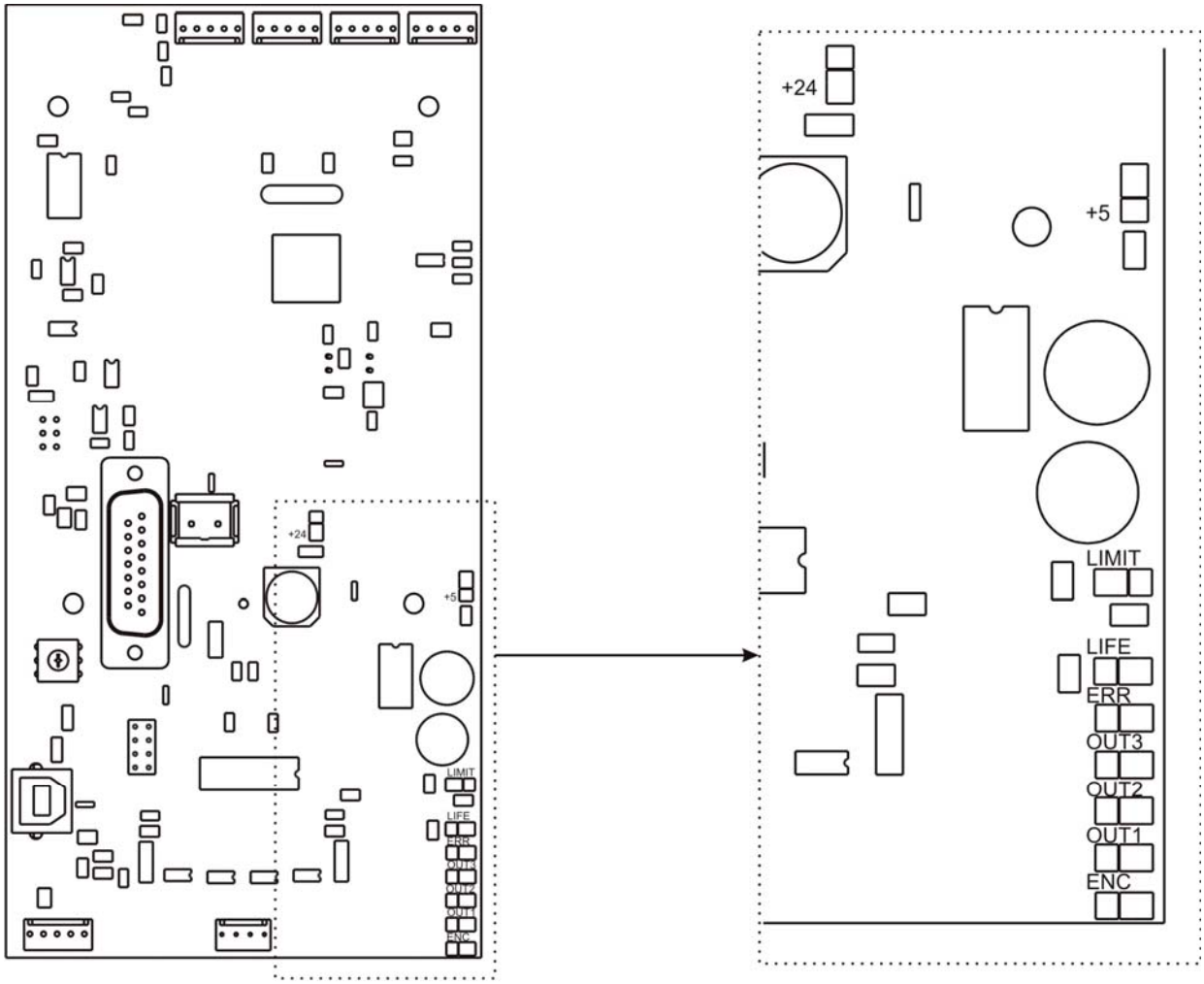
Table 5. Error Codes, Status Byte with ASCII and Hexadecimal Values

Status Byte	Hex if Status Bit 5 =		ASCII if Status Bit 5 =		Error Code	Description
7 6 5 4 3 2 1 0	X= 0 (Busy)	X=1 (Idle)	X= 0 (Busy)	X=1 (Idle)	Number	Error
0 1 X 0 0 0 0 0	40h	60h	@	`	0	No Error
0 1 X 0 0 0 0 1	41h	61h	A	a	1	Initialization failure
0 1 X 0 0 0 1 0	42h	62h	B	b	2	Invalid Command
0 1 X 0 0 0 1 1	43h	63h	C	c	3	Invalid Operand
0 1 X 0 0 1 0 0	44h	64h	D	d	4	Invalid CheckSum
0 1 X 0 0 1 0 1	45h	65h	E	e	5	Unused
0 1 X 0 0 1 1 0	46h	66h	F	f	6	EEPROM Failure
0 1 X 0 0 1 1 1	47h	67h	G	g	7	Device not Initialized
0 1 X 0 1 0 0 0	48h	68h	H	h	8	CAN bus failure
0 1 X 0 1 0 0 1	49h	69h	I	i	9	Plunger Overload
0 1 X 0 1 0 1 0	4Ah	6Ah	J	j	10	Valve Overload
0 1 X 0 1 0 1 1	4Bh	6Bh	K	k	11	Plunger Move Not Allowed
0 1 X 0 1 1 1 1	4Fh	6Fh	O	o	15	Command Overflow

Error Reporting Examples

[A7000R]	Since <4000> is greater than the stroke of the MC6000, this returns an error immediately in the command response. When queried ([Q] command), does not return error.
[A3000P3500R]	Moves to position 3000 then stops. The [P3500] is past the stroke of the MC6000, a [Q] command returns an error.
[e200R]	<14> is the maximum argument for [e]. Returns an invalid command error immediately in the command response. The pump status is Idle.
[A3000e2000R]	Returns an invalid command error immediately in the command response. The pump is Idle.
[BA1000R]	Since valve is in the bypass position, plunger movements are not allowed. This will returns an error immediately. When queried with the [Q] command does not return an error.

Figure 2. MC6000 Diagnostic LEDs



LED	Function	Normal/Typical State
+24	24VDC Supply	ON
+5	5VDC Regulator	ON
LIMIT	Top of stroke sensor	ON at top of stroke
LIFE	Processor Heartbeat	Blink slowly
ERR	Error	OFF (See Table 2) RED WHEN ON
OUT3	Auxiliary output 3	ON (OFF if output set)
OUT2	Auxiliary output 2	ON (OFF if output set)
OUT1	Auxiliary output 1	ON (OFF if output set)
ENC	Encoder sensor	Blinks as plunger moves

Table 1

ERR LED	Error
OFF	No error
1 Blink	Initialization error
2 Blinks	EEPROM failure
3 Blinks	Plunger overload
4 Blinks	Valve overload

Table 2

MC6000 RS232 and RS485 Command Summary

Control Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
R	N/A	N/A			Execute command string
y<n>	0…3	N/A		<0> - Execute if Input2 is Low and Input1 is Low <1> - Execute if Input2 is Low and Input1 is High <2> - Execute if Input2 is High and Input1 is Low <3> - Execute if Input2 is High and Input1 is High	Execute next command in buffer based on Aux inputs
X	N/A	N/A			Re-execute last executed command string
G<n>	0…30000	0		0 = loop forever	Repeat command sequence
g	N/A	N/A			Mark the start of a repeat sequence
M<n>	0…30000	N/A		milliseconds	Delay command execution
H<n>	0-2	0		<0> - Wait for [R] or either input 1 or 2 to go low <1> - Wait for [R] or input 1 to go low <2> - Wait for [R] or input 2 to go low	Halt command execution
T	N/A	N/A			Terminate command

Initialization Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
Z<n>	0,10…40	0		<0> or no arg, default initialization speed <10-40> initialize at defined plunger speed	Initialize plunger, valve to the right
Y<n>	0,10…40	0		<0> or no arg, default initialization speed <10-40> initialize at defined plunger speed	Initialize plunger, valve to the left
W<n>	0,10…40	0		<0> or no arg, default initialization speed <10-40> initialize at defined plunger speed	Initialize plunger without valve
z<n>	0…6000	0			Set pump's internal position counter to value specified
k<n>	0…255		122		Syringe dead volume command
x<n>	25 ..100		100		Set initialization force as a percentage of the maximum force
J<n>	10…40			XL mode only	Set initialization speed

Plunger Movement Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
A<n>	0...6000	0			Move plunger to absolute position
a<n>	0...6000	0			Move plunger to absolute position; Not busy
P<n>	0…6000	0			Relative pickup
p<n>	0...6000	0			Relative pickup; Not busy
D<n>	0...6000	0			Relative dispense
d<n>	0...6000	0			Relative dispense; Not busy

Valve Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
I	N/A	N/A			Move all valves to Input position
O	N/A	N/A			Move all valves to Output position
B	N/A	N/A			Move all valves to Bypass position
B<n>	0000...1111	N/A			Move each valve pair to the input or output position defined by the binary argument
E<n>	0..255	N/A			Move each valve pair to the input or output position defined by the exponential argument

Set Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
K<n>	0…255	0	48		Backlash steps
L<n>	1...20		7		Set acceleration/deceleration slope
v<n>	0...1000		901		Set start velocity in Hz
V<n>	1 ...6000		901		Set top velocity in Hz
S<n>	0...40				Set speed
c<n>	1 ...2700		901		Set cutoff velocity in Hz
h<n>	0...100		10		Set motor hold current in %
m<n>	0...100		100		Set motor run current in %
N<n>	0...2		0	<0> = normal mode <1> = microstep position mode <2> = microstep position and velocity mode	
J<n>	0...7			0 = all outputs Low 7 = all outputs High	Sets the 3 TTL auxiliary outputs
J<ppppn>	<pppp> 1…6000			<pppp> = syringe position <n> = auxiliary outputs	Sets the 3 auxiliary outputs based on syringe position

EEPROM Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
s<n>	0...14				Load program string into EEPROM
e<n>	0...14				Execute EEPROM string
><n1><n2>	<n1> 0..15 <n2> 0..255				Store user data <n2> at specified location <n1>
U30	N/A				Sets auto run
U31	N/A				Clears auto-run
u_	0...14			For factory use only	Set system configuration parameter into EEPROM
!	N/A				Load EEPROM settings

Report Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
Q	N/A	N/A			Report system status
?	N/A	N/A			Report absolute plunger position
?0	N/A	N/A			Same as ?
?1	N/A	N/A			Report start velocity in Hz
?2	N/A	N/A			Report peak velocity in Hz
?3	N/A	N/A			Report cutoff velocity
?4	N/A	N/A			Report plunger position, same as ?
?5	N/A	N/A			Same as ?
?6	N/A	N/A			Reports Valve Position
?7	N/A	N/A			Reports Acceleration slope set with [L]
?10	N/A	N/A			Report Command Buffer Status, same as F
?12	N/A	N/A			Report number of backlash steps
?13	N/A	N/A			Report Status of Aux 1 input
?14	N/A	N/A			Report Status of Aux 2 input
?15	N/A	N/A			Non-Functional, Will always report 1
?16	N/A	N/A			Non-Functional, Will always report 1
?17	N/A	N/A			Non-Functional, Will always report 1
?18	N/A	N/A			Number of valve movements since last ?18, same as %
?19	N/A	N/A			Reports 1 if pump is initialized, 0 otherwise
?20	N/A	N/A			Report firmware checksum, same as #
?21	N/A	N/A			Reports plunger hold current as a percentage
?22	N/A	N/A			Reports plunger run current as a percentage
?23	N/A	N/A			Report firmware version, same as &
?24	N/A	N/A			Report the syringe's dead volume as set by the [k] command

Report Commands

Command	Operand Range <n>	Default Operand	Power Up Default	Operand Description	Command Description
?26 or *	N/A	N/A			Reports +24V
?27	N/A	N/A			Reports configuration EEPROM data as set using the [u] command
?28	N/A	N/A			Reports syringe motor step mode
?30 - ?44	N/A	N/A			Reports user EEPROM execution strings, ?30 = s0, ?31 = s1 and so on
?45	N/A	N/A			Reports alpha-numeric value
?76	N/A	N/A			Same as ?27
F	N/A	N/A			Report Command buffer status, same as ?10
&	N/A	N/A			Report firmware version, same as ?23
#	N/A	N/A			Report firmware checksum, same as ?20
< <n>	0 ..15	N/A			Reports user stored data.
%	N/A	N/A			Number of valve movements since last report, same as ?18

MC6000 CAN Bus Command Summary

On-the-fly Commands
Frame Type = 0

Command	Operands	Command Description
V	same as RS-232/485	Top velocity
T	N/A	Terminate

Action Commands
Frame Type = 1

Command	Operands	Command Description
		All RS-232/485 commands, with the exception of Report commands, are valid Action commands in CAN bus mode.

Common Commands
Frame Type = 2

Command	Operands	Command Description
0	N/A	Reset mode
1	N/A	Start Loaded command
2	N/A	Clear Loaded command
3	N/A	Repeat last command, like X
4	N/A	Stop action immediately, same as T command

Report Commands
Frame Type = 6

Command	Operands	Command Description
0	N/A	Plunger position
1	N/A	Report start speed
2	N/A	Report top speed
3	N/A	Report cutoff speed
6	N/A	Report Valve position
10	N/A	Buffer status, like F
13	N/A	Input 1 status, like ?13
14	N/A	Input 2 status, like ?14
15	N/A	Number of pump initializations, like ?15. Note, currently not implemented, will always return a 1.
16	N/A	Number of plunger movements, like ?16. Note, currently not implemented, will always return a 1.
17	N/A	Number of valve movements, like ?17. Note, currently not implemented, will always return a 1.
18	N/A	Number of valve movements since last report, like ?18
20	N/A	Firmware checksum, like ?20
23	N/A	Firmware version, like &
24	N/A	Syringe dead volume, like ?24
29	N/A	Current status, like Q