HAMILTON

Microlab® 600 RS-232

Communication Manual

TABLE OF CONTENTS

Table of Contents

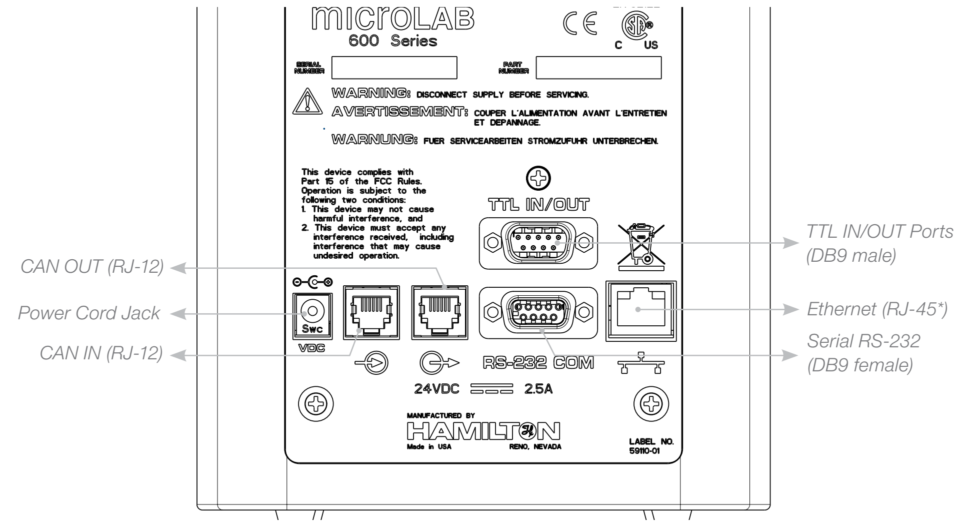
CHAPTER 1:
General Overview

The Microlab® 600 receives commands via three different communication methods. These methods are Ethernet, Serial RS-232 and Digital TTL. This manual provides detailed information on connecting and controlling a Microlab 600 or a daisy chain of Microlab 600 instruments using Hamilton's Protocol 1/RNO+. Protocol 1 is a serial communication language that sends ASCII commands to the Microlab 600 via the electrical standard RS-232C.

This chapter discusses the following topics:

1.1 Microlab 600 Communication Ports

The diagram below shows the communication ports on the back of the Microlab 600 syringe pump. The pump receives inputs via RS-232C, Ethernet, CAN, and Digital TTL. With the exception of the Digital TTL port, once the instrument is connected to a communication port it ignores commands sent over the other ports. The instrument must be power cycled or reset to connect via a new port. Below is a description as to when each input port could be used.



Before connecting the Ethernet (RJ-45) port to any device the Power over Ethernet (PoE) must be turned off to avoid damage to the connected device. This is achieved by starting with the instrument off. Press and hold the power button to turn the instrument on. After 3 seconds the Ready light will illuminate solid indicating the PoE is turned off.

1.1.1 Ethernet Port

The Ethernet protocol provides maximum control of the Microlab 600 functions. The command set exposes greater functionality than the serial protocol and allows for remote connection of the instrument. Commands sent over the internet from a PC anywhere in the world can be received and executed by the Microlab 600. However, there is an important caveat to using the Ethernet control. Ethernet control commands must be sent from a programming language that is Microsoft® .NET 2.0 compatible. This means the commands must be sent from a PC running a Windows® operating system. Compatible programming languages include Visual C#®, Visual Basic®, and LabVIEW™. An electronic programmer's guide, example files and the .Net 2.0 Application Programming Interface (API) are shipped on a CD with each Advanced or standalone Microlab 600.

1.1.2 Serial RS-232 Port

The serial protocol controls the most commonly used commands. For the majority of applications there is no need for enhanced Ethernet commands. The main benefit of RS-232 communication is platform independence. Commands can be sent from any device with a serial port. This means PC, Linux, Mac and embedded controllers can all send commands to RS-232 port.

1.1.3 CAN IN/OUT

The CAN IN port is only used when daisy chaining multiple pumps. The CAN IN port must be connected to the CAN OUT port of a Microlab 600 earlier in the chain. The first instrument in the chain must be connected to the control device using either Ethernet or RS-232. There is no way to control the instrument directly using the CAN port.

1.1.4 Digital TTL IN/OUT Port

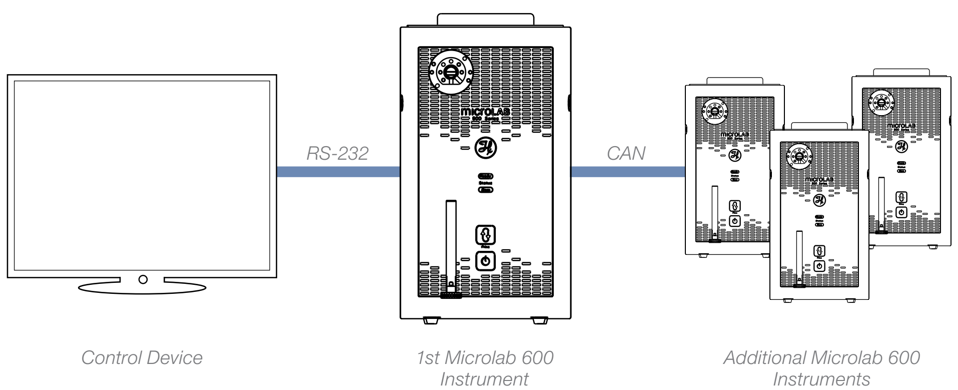
The Digital TTL port can be used to send and receive simple triggers from other devices. Using the Microlab 600 controller, it is possible to create a custom method that is programmed to wait for triggers sent via the TTL input pins and send signals out through the output pins when a specific task is completed. This method can then be downloaded to the pump and run independently from any control device.

This port can also be utilized by both the Ethernet and the RS-232 protocols to send and receive signals from other instruments that are not connected to the control device.

1.2 Daisy Chain Compatibility

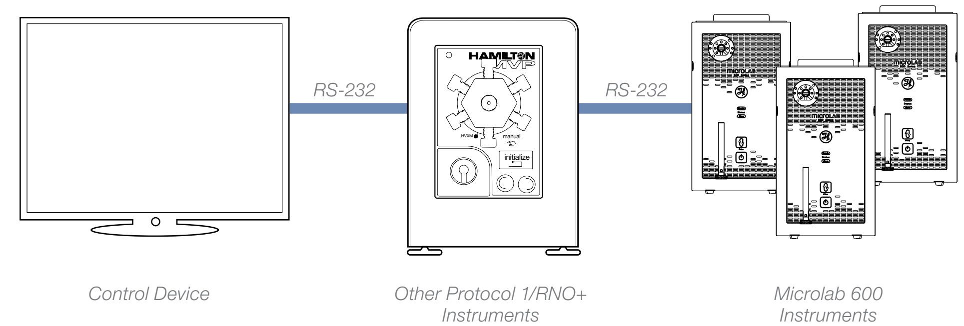
1.2.1 Microlab 600 Units Only

The Microlab 600 supports the daisy chaining and control of up to 16 instruments via RS-232. The control device communicates with the first instrument in the daisy chain. Then, the first device communicates with the other 15 instruments via a proprietary CAN interface.



1.2.2 Microlab 600 Units With Other RNO Protocol Devices

The Microlab 600 is compatible with other devices that communicate via Hamilton's Protocol 1/RNO+. These devices include the older Microlab 500 series and the Modular Valve Positioner. Only the first Microlab 600 in a daisy chain communicates with RS-232. This means that all other RNO protocol devices must come before the Microlab 600 in a daisy chain.

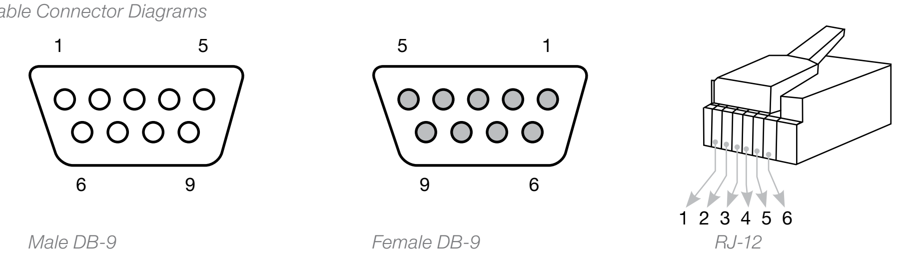


1.2.3 Required Cables For Daisy Chaining

A variety of cables are required to connect the Microlab 600 with a control device or other Protocol 1 devices. The major cables required for this purpose are outlined in the table below. If other connections are required, please contact Hamilton Company for a cable or pin diagram.

Part Number	Description	Upstream Cable Connector	Downstream Cable Connector	Pinout
3553-01	PC to Microlab 600	DB-9 Female	DB-9 Male	Upstream pins 1–9 are directly connected to downstream pins 1–9
66650-01	Microlab 600 to Microlab 600	RJ-12	RJ-12	Upstream pins 1–6 are directly connected to downstream pins 1–6
66626-01	Modular Valve Positioner to Microlab 600	DB-9 Male	DB-9 Male	Upstream pins 1, 2 & 3 are connected to downstream pins 2, 3 & 5 respectively
66627-01	Microlab 500 to Microlab 600	RJ-12	DB-9 Male	Upstream pins 2, 3 & 4 are connected to downstream pins 3, 2 & 5 respectively

Cable Connector Diagrams



CHAPTER 2:

Communication via RS-232

Hamilton Company provides a protocol for communication with this instrument: Protocol 1/RNO+. This protocol uses the industry standard RS-232C interface (using RXD, TXD and GND) and allows up to 16 instruments to be linked in a daisy chain configuration. The instruments may be individually accessed via their own address, while a broadcast addressing scheme allows all instruments to be accessed simultaneously using broadcast addressing.

> **Note:** All information transferred between the controlling device and the instrument is case sensitive and must be sent exactly as shown.

This chapter discusses the following topics:

2.1 Hamilton Protocol 1/RNO+ Overview

The Hamilton Protocol 1/RNO+ syntax is used to communicate with instruments (Microlab diluters/dispensers, PSD pumps and Modular Valve Positioners) which are designed and manufactured by Hamilton Company.

To successfully communicate using this protocol the serial port must be configured as follows:

Data Format

Baud Rate: 9,600 (default)
Parity: Odd
Data bits: 7
Stop bits: 1

All commands sent to the instrument must be terminated using a carriage return. When a command is received by an instrument it is processed, and the instrument responds back to the control device with an acknowledgement. There are two types of acknowledgements. The first is a positive acknowledgement that communicates the command is understood and can be executed. The second is a negative acknowledgement that communicates the command is not understood or cannot be executed. The communication is performed using three basic control characters (Table 2-1).

Notation	Name	ASCII Code (decimal)	Description
<CR>	Carriage return	13	A control character that terminates a data string
<ACK>	Acknowledge	06	A control character transmitted by the instrument indicating an affirmative response to the controlling device
<NAK>	Negative acknowledge	21	A control character transmitted by the instrument indicating a non-affirmative response to the controlling device

Table 2-1 Protocol 1/RNO+ Control Characters

2.2 Data Transfer Format

Commands sent from a controlling device to an instrument must begin with the instrument's address and end with a carriage return. Instruments configured in a daisy chain will only respond to commands that start with their address. If it is desirable to send a single command to all instruments, a ":" can be used for the instrument address. This is the broadcast address which will be acted upon by all instruments in the chain. The control device will not receive a response from broadcasted commands.

<adr><data string><CR>

- Instrument address
- Commands and/or status request
- Carriage return

The instrument will respond with one of the following:

<ACK><CR>

- Acknowledge (no parity/syntax errors)
- Carriage return*

or

<ACK><response string><CR>

- Acknowledge (no parity/syntax errors)
- Status data
- Carriage return

or

<NAK><CR>

- Negative acknowledge (no parity/syntax errors)
- Carriage return

**A minimum 1 millisecond delay must occur between the controlling device receiving the <CR> of the response and the transmission of any data on a daisy chain.*

2.3 Establishing Communication (Auto-addressing and Initializing)

When an instrument is first turned on (prior to auto-addressing) it does not know if it is a single instrument or one instrument in a daisy chain of other instruments. Without knowing its position in the chain, it will not know which commands to respond to and which to ignore. For this reason, the auto-address command should be the first sequence of characters transmitted to the instrument(s). Until the auto-address command is sent and addresses are assigned to the instrument(s), the instrument(s) will ignore all commands and requests.

Note: Instruments are assigned their addresses by auto-addressing. Hardware addressing is not supported with this protocol.

Protocol 1/RNO+ instruments are auto-addressed using the following sequence:

1a<CR>

The first instrument in the daisy chain will be assigned the address "a" and will transmit the sequence 1b<CR> to the next instrument. The process of assigning the address received and transmitting the next address continues for all instruments in the daisy chain. The last instrument in the daisy chain responds to the controlling device with

1<last address + 1><CR>

Example: Four instruments are in a daisy chain, and the controlling device transmits 1a<CR>. The controlling device will receive 1e<CR>, indicating that the instruments have been assigned addresses a, b, c and d.

If the daisy chain has already been auto-addressed sending the auto-address command again will not readdress the instruments. Instead the expected response will be 1a.

Broadcast Addressing

In addition to addressing a single unit in a daisy chain, the Protocol 1/RNO+ allows all units in a daisy chain to be addressed at once using the broadcast address. Instead of the instrument-specific lettered address, the broadcast instrument address is ":". Note, however, that the instrument(s) will NOT transmit protocol or status information when addressed with the broadcast address (i.e., to ensure no corruption of data on the serial line when multiple instruments attempt to transmit data at the same time).

Instrument Initialization

Commands can be sent to the instruments after auto-addressing. Before sending commands that move the valves and syringes, it is necessary to initialize the drives. Commands sent to the syringe drive are ignored until initialization is complete. Commands sent to the valve drives result in the automatic initialization of the valve drive prior to rotation to the desired position. Depending on the user case, there are several initialize commands that can be found in Table 3.1.2.

Recovering from Resets and Power Failures

When controlling a single Microlab 600 an instrument reset or a power failure will result in the need to auto-address the instrument using the 1a<CR> command

When controlling a daisy chain of Microlab 600s and other Protocol 1/RNO+ devices the recovery will require a few steps. The first step is to broadcast a reset command :!<CR> to all the pumps. Then issue the 1a<CR> command to auto-address the instruments. Next issue the reset command again followed by an additional auto address. Repeat the reset and auto-address commands until the same auto-address response is received twice. These extra steps are required because the reset command will only be seen by instruments that have been auto-addressed. If multiple instruments have power failures throughout the chain multiple resets may be required.

2.4 Data String Components

A data string may contain one status request and/or one or more commands. Multiple status requests in a single data string should be avoided as they are not explicitly supported. The instrument buffers the commands received until the execute command is received. Once the execute command is received, the commands are executed in the order received. The instrument command buffer can contain up to two valves, one syringe, one timer delay and one digital output command per syringe pump. If the command buffer is already full, new commands replace existing commands.

While the instrument is executing commands, it will ignore any new commands received for that given side of the instrument with the exception of execution commands such as Halt.

Command Example 1: Auto-addressing the instrument

The following example assumes there is only one instrument connected to the control device.

Control device transmits:

1a<CR>

- Auto-address sequence
- Starting address
- End of data string

Controlling device receives:

1b<CR>

- Auto-address sequence
- Last address + 1
- End of data string

Command Example 2: Initializing the instrument

The following example assumes the instrument has already been auto-addressed as per the previous example.

Control device transmits:

aXR<CR>

- Instrument address
- Initialize instrument
- Execute
- End of data string

Controlling device receives:

<ACK><CR>

- Data string acknowledged
- End of response string

Command Example 3: A complex command

The following example assumes the instrument has already been auto-addressed and initialized as per the previous examples.

Control device transmits:

aIP100S3N5O>T100R<CR>

- Instrument address
- Move valve to input
- Move syringe down 100 steps
- Speed 3
- 5 return steps
- Move valve to output
- Time delay, 100 ms
- Execute
- End of data string

Controlling device receives:

<ACK><CR>

- Data string acknowledged
- End of response string

Command Example 4: A status or parameter request

The following example assumes the instrument has already been auto-addressed as per the previous example. The response shown will vary depending upon the instrument model used.

Control device transmits:

aU<CR>

- Instrument address
- Firmware version request
- End of data string

Controlling device receives:

<ACK>NV01.72.A<CR>

- Data string acknowledged
- Firmware version
- End of data string

CHAPTER 2: COMMUNICATION VIA RS-232

CHAPTER 3:
Protocol Command Summary

This section provides detailed information on the Protocol 1/RNO+ supported by Hamilton Company:

3.1 General Commands

These are the individual command strings that can be used together to create a synchronized dispense or fill action. The commands are used to move the valves and syringes, and set the state of the Digital I/O port. Some commands have optional parameters listed in the tables that control things like speed and return steps. If these parameters are not specified, the instrument will use the default settings. These general commands will be stored in the command buffer until the Execute command "R" is sent by the control device.

3.1.1 Syringe Pump Selection Commands

The Dual Syringe Microlab 600 has two complete syringe drives one left and one right. For this instrument is it critical to indicate which syringe drive should execute the command. If a channel selection command is not sent before a given command, the left (default) side is assumed as the target for commands and parameter change requests. For status requests, the response will be for the whole instrument. Exceptions to this rule are noted after the given commands.

For instruments without two syringe drives, all devices are considered to be on the left (default) side. Commands attempted on the right side will result in an error.

Command	Description
B	Select left syringe drive
C	Select right syringe drive

Example of Syringe Select commands:

This example assumes the syringes have just been initialized. Sending this command will fill the left syringe full and the right syringe to half stroke.

aBP48000CP24000R

- Instrument address
- Select the left syringe drive
- Move the left syringe down 48,000 steps
- Select the right syringe drive
- Move the right syringe down 24,000 steps
- Execute command

3.1.2 Initialization Commands

Before commands can be sent to move an object like a valve or a syringe, the object must first be initialized. During initialization the valve and syringe find their home position which acts as the starting position for all subsequent movements. If the left or right syringe pump is not explicitly specified via one of the syringe selection commands, the initialization commands are performed on all available sides of the instrument.

The instrument will not execute any initialization commands until an Execute command is received.

Optional parameters can be used to temporarily set the syringe speed during initialization. If the optional parameter is not used the syringe will initialize at the default speed stored in the memory. When sending a speed parameter it is not necessary to include leading zeros. For example speed 2 can be sent as S2 or S0002.

Command	Optional Parameters	Description
X	Syringe speed Sxxxx	Command Name: Initialize Instrument Command Description: 1) Valve(s) to initialize and rotate to the output position 2) Syringe(s) drive up until overload (bottomed out) 3) Valve(s) to rotate to input position 4) Syringe(s) to back off the default back-off steps Parameter Values: xxxx = This is an optional parameter that sets the syringe speed used during initialization. If not specified, the syringes initialize at the speed currently stored in memory. Valid values are 2-3692 seconds per stroke.
X1	Syringe speed Sxxxx	Command Name: Initialize Syringe(s) Only Command Description: 1) Syringe(s) drive up until overloaded (bottomed out) 2) Syringe(s) to back off the default back-off steps Parameter Values: xxxx = This is an optional parameter that sets the syringe speed used during initialization. If not specified, the syringes initialize at the speed currently stored in memory. Valid values are 2-3692 seconds per stroke.
X2	Syringe speed Sxxxx	Command Name: Initialize Syringe(s) Only with Error Bit Command Description: This initialization command can only be performed after a previous successful syringe initialization. After this command is completed, the pump compares the new initialization position to the original and sets the “syringe move” error bit if the syringe overloads before it reaches the top of the stroke. Parameter Values: xxxx = This is an optional parameter that sets the syringe speed used during initialization. If not specified, the syringes initialize at the speed currently stored in memory. Valid values are 2-3692 seconds per stroke.
LX		Command Name: Initialize Valve(s) Only Command Description: 1) Rotate the valve(s) at least 395° 2) Stop valve(s) at the input position

Example of Initialization commands:

Sending the command below initializes the left and right valve drives. Then it initializes the right syringe at speed 10 and the left syringe at speed 5.

aBXS10CXS5R

- Instrument address
- Select the left syringe
- Initialize the left valve then syringe at speed 10
- Select the right syringe drive
- Initialize the right valve then syringe at speed 5
- Execute command

3.1.3 Syringe Positioning Commands

Syringe positioning commands are used to move the syringe drive up, down, or to an absolute position. One syringe positioning command per syringe drive can be held in the command buffer at a time. New syringe commands will overwrite commands in the command buffer that have not been executed.

To determine the volume dispensed per step the total syringe volume is divided by 48,000 steps. All Hamilton instrument syringes are designed with a 60 mm stroke length and the Microlab 600 is designed to move 60 mm in 48,000 steps. For example to dispense 9 mL from a 10 mL syringe you would determine the number of steps by multiplying 48000 steps (9 mL/10 mL) to get 43,200 steps.

Optional parameters can be used to temporarily set the syringe speed and return steps during a move. If the optional parameter is not used the syringe will move at the default speed and return steps stored in the memory. When sending a speed and return step parameters it is not necessary to include leading zeros. For example speed 2 can be sent as S2 or S0002.

Command	Optional Parameters	Description
PXXXXX	Syringe speed Snnnn Return steps Nrrrr	Command Name: Syringe Pickup Command Description: This command moves the syringe a set number of steps down from the current syringe position. Parameter Values: xxxxx = This is the number of steps to move the syringe. Valid values are 1–52,800 steps. nnnn = This sets the syringe speed used during the positioning command. Valid values are 2–3692 seconds per stroke. rrrr = This is the number of return steps to apply to the end of a downward syringe movement. This removes the mechanical slack from the system. Valid values are 0–1000 steps.
DXXXXX	Syringe speed Snnnn	Command Name: Syringe Dispense Command Description: This command will move syringe a set number of steps up from the current syringe position Parameter Values: xxxxx = This is the number of steps to move the syringe. Valid values are 1–52,800 steps. nnnn = This sets the syringe speed used during the positioning command. Valid values are 2–3692 seconds per stroke.
MXXXXX	Syringe speed Snnnn Return steps Nrrrr	Command Name: Syringe Absolute Move Command Description: This command moves the syringe from its current position to any step position along the stroke. It automatically calculates how far it needs to move in either the up or down direction. Parameter Values: xxxxx = This is the step position where syringe should move. Valid values are 1–52,800 steps. nnnn = This sets the syringe speed used during the positioning command. Valid values are 2–3692 seconds per stroke. rrrr = If the syringe move is in the downward direction, this is the number of return steps to apply to the end of the movement. This removes the mechanical slack from the system. Valid values are 0–1000 steps.

Example of syringe positioning commands:

This example assumes that both syringe drives have been initialized and the left syringe is empty. Sending this command fills the left syringe full and moves the right syringe to half stroke.

aBP48000CM24000S25N4R

- Instrument address
- Select the left syringe
- Move the left syringe down 48,000 steps
- Select the right syringe drive
- Move the right syringe from its current position to step 24,000 at speed 25 (25 seconds per stroke). If the movement is in the downward direction, return 4 steps at the end of the stroke
- Execute command

3.1.4 Valve Positioning Commands

Valve positioning commands are used to move the valve drives to a desired position name or a defined angle. Up to two valve positioning commands per syringe drive can be held in the command buffer at a time. Once the command buffer is full new valve commands will overwrite the last valve command in the buffer. The reason for accepting two valve commands per syringe drive is to allow for positioning the valve, moving the syringe, and then positioning the valve to a new location all in one command string.

When sending valve position names and valve angles leading zeros will be ignored. For example to rotate the valve to 15 degrees LA115 and LA1015 will perform the same action.

Command	Description
I	**Command Name:** Valve to Input Position (Position 9) **Command Description:** This command rotates the valve to the default input position for the valve type that is currently stored in the pump.
O	**Command Name:** Valve to Output Position (Position 10) **Command Description:** This command rotates the valve to the default output position for the valve type that is currently stored in the pump.
W	**Command Name:** Valve to Output Position (Position 11) **Command Description:** This command rotates the valve to the default wash position for the valve type that is currently stored in the pump.
LPdpp	**Command Name:** Valve Positioning by Name **Command Description:** This command rotates the valve to the valid valve position names for the valve type that is currently stored in the pump. Position 9 is always the default input position. Position 10 is always the default output position. Position 11 is always the wash position, but not all valves have a wash position. **Parameter Values:** d = This parameter determines the direction the valve turns when moving to the desired valve position. Valid values are 0 = clockwise (CW), 1 = counter-clockwise (CCW). pp = These are the named valve ports. The port names are mapped to different ports depending on the current instrument valve configuration. Valid values are positions 1–11 but not all valves have 11 valid positions.
LAdaaa	**Command Name:** Valve Positioning by Degree **Command Description:** This command rotates the valve to a defined position in degrees. The valve drive's home position is specified as zero degrees. All movements are absolute so the pump automatically calculates how far it needs to move the valve to reach the desired angle. **Parameter Values:** d = This parameter determines the direction the valve turns when moving to the desired valve position. Valid values are 0 = CW, 1 = CCW. aaa = This is the desired valve position specified by the angles in degrees. Valid values are from 0° to 359° in whole degree increments.

CHAPTER 3: PROTOCOL COMMAND SUMMARY

Example of Valve Positioning commands:

This example assumes that both syringe drives have been initialized and the left syringe is empty. Sending this command rotates the left valve to the output and the right valve to 195°. When the left valve finishes its move, the left syringe drive fills completely. Once the left syringe drive finishes its move, the left valve rotates to Position 1.

aBOP48000LP11CLA0195R

- a — Instrument address
- B — Select the left syringe
- O — Rotate the left valve to the Output position
- P48000 — Move the left syringe down 48,000 steps
- LP11 — After the left syringe finishes its move rotate the left valve CCW to Position 1
- C — Select the right syringe drive
- LA0195 — Rotate the right valve drive CW to 195°
- R — Execute command

3.1.5 Timer Commands

Command	Description
>Txxxxxxxx	**Command Name:** Timer Delay **Command Description:** This command adds a delay before a command is executed or between the execution of two commands. The order of the commands determines when the delay will occur. Leading zeros will be ignored so sending >T100 is the same as >T00000100. **Parameter Values:** xxxxxxxx = This is the duration of the delay in milliseconds. Valid values are between 0–99999999 milliseconds.

Example of the Timer command:

This example assumes both syringe drives have been initialized and the left syringe is empty. Sending this command fills the left and right syringe. The left syringe starts 1 second after the right syringe. After the right syringe completes its move, there will be a 1 second delay before the right valve rotates to 195 degrees.

aB>T1000P48000CP48000>T1000LA1195R

- Instrument address
- Select the left syringe
- Timer delay left syringe pump 1,000 ms
- After the delay move the left syringe down 48,000 steps
- Select the right syringe drive
- Move the right syringe down 48,000 steps
- After syringe move timer delay the right syringe pump 1,000 ms
- After the delay rotate the right valve drive CW to 195°
- Execute command

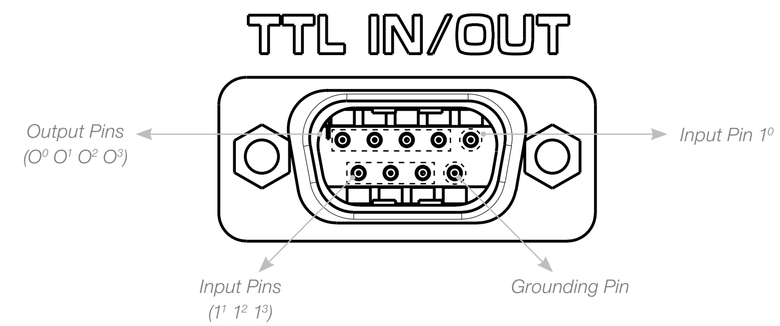
3.1.6 Digital I/O Commands

Command	Description
>Dxx	Command Name: TTL Data Output Command Description: This command turns on and off the four output pins of the TTL IN/OUT port. This can be used to send a 5V signal to trigger an action in another device. There is only one Digitial I/O port so it works independent of the syringe pump selection. Leading zeros will be ignored so sending >D1 is the same as <D01. Parameter Values: xx = This is the decimal value corresponding to the pins that will be turned on and off. Valid values are 0–15. The decimal value maps to a binary value which can be seen in the table below. Since there are only four output pins, only the last four binary digits are used. For example, a decimal value of 0 is equal to the binary value 0000 and results in all pins being turned off. A decimal value of 15 is equal to the binary value 1111 and results in all pins being turned on.

Decimal values mapped to output pins

Pin	Function
1	Output 1 (O^0)
2	Output 2 (O^1)
3	Output 3 (O^2)
4	Output 4 (O^3)
5	Input 1 (O^0)
6	Input 2 (O^1)
7	Input 3 (O^2)
8	Input 4 (O^3)
9	Ground

Table 2-4 TTL Pin Configuration



Decimal	Output 0	Output 1	Output 2	Output 3
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

A value of 0 indicates the pin will be turned off. A value of 1 indicates the pin will be turned on.

Example of the Digital I/O command:

a>D13R

- Instrument address
- Turns on Output Pins 0, 2, and 3 while Output Pin 1 is turned off
- Execute command

3.1.7 Execution Commands

Command	Description
R	Command Name: Execute command Command Description: This command executes commands stored in the instrument's command buffer. The command can be included at the end of a command string to execute the string or can be sent as separate command to execute the commands in the instruments command buffer.
K	Command Name: Halt execution of commands Command Description: This command immediately stops a command the instrument is already executing.
$	Command Name: Resume execution of commands Command Description: This command restarts a command that has been stopped by the Halt command.
V	Command Name: Clear all buffered commands Command Description: This command clears the command buffer of all commands that have not been executed.

CHAPTER 3: PROTOCOL COMMAND SUMMARY

3.1.8 Instrument Control Commands

Command	Description
!	Command Name: Total system reset Command Description: This command powers the instruments off and back on. It can take more than 2 seconds for a single Microlab 600 to power cycle and be ready to receive commands. For a daisy chain of Microlab 600s, it can take as much as 12 seconds to power cycle all instruments and be ready for the auto-address command.

3.2 Parameter Change Commands

The parameter change commands changes the default syringe settings and valve configuration.

3.2.1 Syringe Parameter Change

Command	Description
YSSxxxx	Command Name: Set syringe default speed Command Description: This command changes the speed of the syringe drive. All changes are temporary unless they are saved to non-volatile memory using the parameter save commands. Suggested speeds are defined per syringe in the table below. Leading zeros will be ignored so sending YSS2 is the same as YSS0002. Parameter Values: xxxx = This is the syringe speed parameter that is temporarily saved in memory as the default syringe speed. Valid values are 2–3692 seconds per stroke.
YSNxxxx	Command Name: Set default return steps Command Description: This command is used to change the default return steps. Syringe return steps are used to compensate for the mechanical drive system backlash, which improves accuracy and precision. Return steps are added to all downward movements, which ultimately causes them to go past the desired position. The return steps are then subtracted by an upward stroke so the syringe ends at the exact desired location. All changes will be temporary unless they are saved to non-volatile memory using the parameter save commands. Return steps are independent of syringe size and the default for the Microlab 600 is 24 steps. Leading zeros will be ignored so sending YSN25 is the same as YSN0025. Parameter Values: xxxx = This is the return step parameter that is temporarily saved in memory as the new default. Valid values are 0–1000 steps.
YSBxxxx	Command Name: Set default back-off steps Command Description: This command changes the default back-off steps. Syringe back-off steps are used during initialization. The syringe is driven to the top of stroke until the drive overloads. Then the syringe is adjusted a small distance away from the overload point, and this final position is used as the zero point for future syringe movements. The small distance moved from the overload point is defined by the back-off steps. Larger syringes generally require more back-off steps because the plunger tips are larger and tend to compress, requiring more back-off distance before being in the right position to draw liquid. All changes are temporary unless they are saved to non-volatile memory using the parameter save commands. Suggested back-off steps are defined per syringe in the table below. Leading zeros will be ignored so sending YSB25 is the same as YSB0025. Parameter Values: xxxx = This is the back-off step parameter that is temporarily saved in memory as the new default. Valid values are 0–1000 steps.

CHAPTER 3: PROTOCOL COMMAND SUMMARY

Microlab 600 Syringe Recommended Settings

Syringe Size	Default Speed (sec/stroke)	Default Back-off Steps
10 µL	2	80
25 µL	2	80
50 µL	2	80
100 µL	2	80
250 µL	2	80
500 µL	2	80
1 mL	2	80
2.5 mL	4	96
5 mL	4	96
10 mL	4	96
25 mL	8	96
50 mL	16	96

3.2.2 Valve Parameter Change

Command	Description
LSTxx	**Command Name:** Set Valve Configuration **Command Description:** This command sets the type of valve attached to the instrument. Values 11–17 are used to control PSD/4 valves. Values 18-19 are the standard valve configurations for the Microlab 600. When used with a dual syringe system, values 19–20 automatically configure the left and right valves. **Parameter Values:** xx = This parameter corresponds to the valve types found in the table below. Valid values are 11–20. LST Parameter — Valve Name 11 — 8-5 12 — 6-5 13 — 4-5 14 — 3-2 15 — 3-5 16 — 3-3 17 — Y 18 — Single/Dual Dispense 19 — Continuous Dispense 20 — Dual Diluter
LSFxxx	**Command Name:** Set Valve Speed **Command Description:** This command adjusts the default valve rotation speed. The default speed is 240 degrees per second. Faster speeds less torque and could result in more frequent valve drive stalls. This is application-dependent and affected by valve type, duty cycle, solvent, etc. Leading zeros will be ignored so sending LSF15 is the same as LSF015. **Parameter Values:** xxxx = This parameter sets the valve drive speed. Valid values are 15–720 degrees per second.

Microlab 600 Valid Valve Positions by Name and Degree

LST Parameter	Valve Name	Valid Position Names		Positions in Degrees	
11	8-5 (8-Port Distribution)	1-11		0, 45, 90, 135, 180, 225, 270, 315, 0, 270, 90	
12	6-5 (6-Port Distribution)	1-6 & 9-11		45, 90, 135, 180, 225, 270, 45, 270, 135	
13	4-5 (4-Port Distribution)	1-4 & 9-11		0, 90, 180, 270, 0, 270, 90	
14	3-2 (2-Port Distribution)	1-4 & 9-11		0, 90, 180, 270, 0, 270, 90	
15	3-5 (3-Port Distribution)	1-3 & 9-11		0, 90, 180, 0, 180, 90	
16	3-3 (2-Port T valve)	1-4 & 9-11		0, 90, 180, 270, 0, 180, 270	
17	Y (2-Port Y valve)	1-3 & 9-11		0, 120, 240, 0, 240, 120	
		Left Valve	**Right Valve**	**Left Valve**	**Right Valve**
18	Single/Dual Dispense	1, 3, 9 & 10	1, 2, 9 & 10	0, 135, 0, 135	0, 90, 90, 0
19	Continuous Dispense	1, 2, 9 & 10	1, 2, 9 & 10	0, 270, 0, 270	0, 90, 90, 0
20	Dual Diluter	1, 2, 9 & 10	1, 2, 9 & 10	0, 270, 0, 270	0, 90, 0, 0

3.2.3 Save and Erase Current Parameters

Command	Description
#SP1	**Command Name:** Save current instrument parameters **Command Description:** This command stores the current left and right side parameters for syringe speed, return steps, back-off steps, valve configuration (type) and valve speed to non-volatile memory.
#SP2	**Command Name:** Erase stored parameters **Command Description:** This command erases stored values for the left and right side for syringe speed, return steps, back-off steps, valve configuration (type) and valve speed from non-volatile memory and resets the values to the instrument default values.

3.3 Status Request Commands

3.3.1 Instrument Information Requests

Request	Response	Descriptions
F	x	Request Name: Instrument Done Request Request Description: This request checks to see if there are commands in the command buffer. Response Values (x = Instrument status): Y = Instrument is idle and command buffer is empty N = Instrument is idle and command buffer is not empty * = Instrument is busy
Z	x	Request Name: Syringe Error Request Request Description: This request checks to see if there was a syringe error. Response Values (x = Syringe status): Y = Syringe overload or initialization error N = No syringe error * = Instrument is busy
G	x	Request Name: Valve Error Request Request Description: This request checks to see if there was a valve error. Response Values (x = Valve status): Y = Valve overload or initialization error N = No valve error * = Instrument is busy

CHAPTER 3: PROTOCOL COMMAND SUMMARY

Request	Response	Descriptions
H	x	Request Name: Instrument Configuration Request Description: This request checks to see if the instrument has one or two syringe drives. Response Values (x = Instrument configuration): Y = Single syringe instrument N = Dual syringe instrument * = Instrument is busy
Q	x	Request Name: Hand Probe/Foot Switch Status Request Description: This request checks for a switch press on the left or right trigger port. Response Values (x = Switch status): Y = Switch is pressed N = Switch is not pressed * = Instrument is busy

3.3.2 Instrument Status Request

Request	Response	Description
E1	x	**Request Name:** Instrument Status Request **Request Description:** This request checks to see if the instrument is busy or if there have been any instrument errors or syntax errors. **Response Values (x = Instrument status byte):** The response is an ASCII character. This character maps to a binary value with 8 bits, where a bit value of 1 means the condition is true and 0 means the condition is false. See Appendix B for ASCII to binary conversion. **Example:** (see table below) x = An ASCII value where the bits are defined as: 0: Instrument idle, command buffer is not empty 1: Syringe drive(s) busy 2: Valve drive(s) busy 3: Syntax error 4: Instrument error (valve or syringe error) 5: Always 0 6: Always 1 7: Always 0 The instrument error bit is reset after an Instrument Error Request and the syntax error bit is reset after the response is sent.

Example:

	Byte							
Bits#	7	6	5	4	3	2	1	0
ASCII value = @	0	1	0	0	0	0	0	0

CHAPTER 3: PROTOCOL COMMAND SUMMARY

Request	Response	Description
E2	abcd	**Request Name:** Instrument Error Request **Request Description:** This request checks for errors with the valve or syringe drives. **Response Values (see E1 for an example):** a and c = These ASCII values show errors for the left (a) and right (c) syringes. The bits are defined as: 0: Not initialized 1: Overload error 2: Stroke too large 3: Initialization error 4: This syringe does not exist 5: Always 0 6: Always 1 7: Always 0 b and d = These ASCII values show errors for the left (b) and right (d) valves. The bits are defined as: 0: Not initialized 1: Initialization error 2: Overload error 3: Always 0 4: This valve does not exist 5: Always 0 6: Always 1 7: Always 0
E3	x	**Request Name:** Timer Status **Request Description:** This request checks to see if the instrument is currently waiting for a timer to complete. **Response Values (see E1 for an example):** x = An ASCII value where the bits are defined as: 0: Timer(s) busy 1: Always 0 2: Always 0 3: Always 0 4: Always 0 5: Always 0 6: Always 1 7: Always 0

Command	Response	Description
T1	x	Request Name: Instrument Busy Status Request Description: This request checks to see if the valve and syringe drives on the instrument are busy. Response Values (see E1 for an example): x = An ASCII value where the bits are defined as: 0: Left valve busy 1: Left syringe busy 2: Right valve busy 3: Right syringe busy 4: Prime/Step active 5: Handprobe/Foot switch active 6: Always 1 7: Always 0
T2	x	Request Name: Instrument Error Status Request Description: This request checks to see if the instrument is currently waiting for a timer to complete. Response Values (see E1 for an example): x = An ASCII value where the bits are defined as: 0: Left valve error 1: Left syringe error 2: Right valve error 3: Right syringe error 4: Always 1 5: Always 1 6: Always 1 7: Always 0

3.3.3 Syringe Parameter Request

Request	Response	Description
YQS	xxxx	Request Name: Syringe default speed Request Description: This requests the current syringe speed setting. Response Value: xxxx = This is the syringe speed parameter. Valid response values are 2–3692 seconds per stroke.
YQN	xxxx	Request Name: Syringe default return steps Request Description: This requests the current return steps setting. Response Value: xxxx = Valid values are 0–1000 steps.
YQP	xxxxx	Request Name: Current syringe position Request Description: This request returns the current location of the syringe drive in steps. Response Value: xxxxx = Valid values are 0–52,800 steps.
YQB	xxxx	Request Name: Syringe default back-off steps Request Description: This requests the current back-off steps setting. Response Value: xxxx = Valid values are 0–1000 steps.

3.3.4 Valve Parameter Request

Request	Response	Description
LQP	xx	**Request Name:** Valve position **Request Description:** This requests the current valve position. The request will return a position name between 1 and 8. Value positioning 7–11 can be mapped to a position between 1–8 which is shown by the table in Section 3.2.2. **Response Value:** xx = Valid values are 1–8 steps.
LQA	xxx	**Request Name:** Valve angle **Request Description:** This requests the current valve angle. **Response Value:** xxx = Valid values are 0–359 degrees.
LQT	xx	**Request Name:** Valve configuration **Request Description:** This requests the current valve configuration. **Response Value:** xx = Valid values are 11–20. See Section 3.2.2 for a configuration cross reference.
LQF	xxx	**Request Name:** Valve speed **Request Description:** This requests the current valve speed. **Response Value:** xxx = Valid values are 15–720 degrees per second.

CHAPTER 3: PROTOCOL COMMAND SUMMARY

3.3.5 Timer Requests

Request	Response	Description
<T	xxxxxxxx	**Request Name:** Timer delay value **Request Description:** This requests the current timer delay value. If a timer delay is in the command buffer, it returns with the value of the timer. If the command has already been executed, the request returns with the number of milliseconds remaining. If no timer is queued or active, the request returns with 0. **Response Value:** xxxxxxxx = Valid values are 0–99999999 milliseconds.

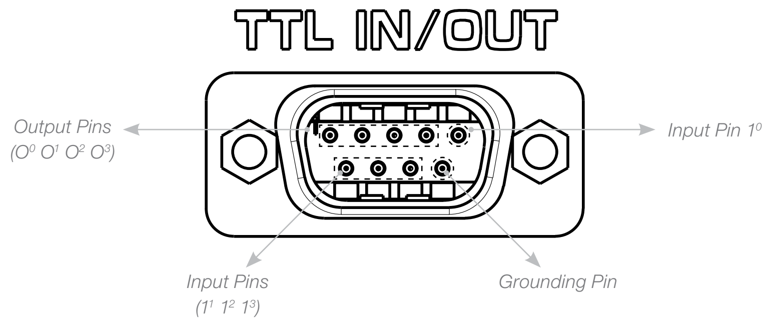
3.3.6 Digital I/O Requests

Request	Response	Description
<D	xx	**Request Name:** TTL data input **Request Description:** This request returns a decimal value that corresponds to the state of four input TTL pins on the back of the instrument. The decimal value maps to a binary value. The last four digits of the binary value indicate the status of each input pin. If nothing is connected, this command will return 15 indicating that all at their default state of on. If the Input 0 pin is shorted to ground, then the command would return 14 indicating a signal is being received on Input 0. **Response Value:** xx = Valid values are 0–15.

Decimal values mapped to input pins

Pin	Function
1	Output 1 (O^0)
2	Output 2 (O^1)
3	Output 3 (O^2)
4	Output 4 (O^3)
5	Input 1 (O^0)
6	Input 2 (O^1)
7	Input 3 (O^2)
8	Input 4 (O^3)
9	Ground

Table 2-4 TTL Pin Configuration



Decimal	Input 0	Input 1	Input 2	Input 3
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

CHAPTER 3: PROTOCOL COMMAND SUMMARY

3.3.7 Firmware Version Request

Request	Response	Description
U	xxii.jj.k	**Request Name:** Firmware Version Request **Request Description:** This requests the current valve configuration. **Response Value:** xx = Product identifier ii = Major version (01–99) jj = Minor version (01–99) k = Revision (A–Z) Product Identifiers/Version OM01 = PSD/2 MV = MVP NV01 = ML600

Appendices

Appendix A — Simple Program Examples

Example program 1:

This series of commands can be used with the Dual Syringe Dispenser setup to fill both syringes and then wait for a trigger to aliquot 25% of the syringe volume in four dispenses from the left and right syringe. When the dispense are complete the Digital TTL signal is sent.

Note: If the instrument was configured with two 10 mL syringes step 4 would fill both syringes with 10 mL (48,000*(10 mL/10 mL) = 48,000) and steps 6, 8, 10, and 12 would dispense 2.5 mL (48000*(2.5 mL/10 mL) = 12,000).

Commands	Description
1. 1a	1. Auto-address (expect response 1b)
2. aXR	2. Initialize the left and right syringe drives
3. aQ	3. Repetitively send this command until the pump is no longer busy from the initialize command. Then the next command can be sent. This is called polling.
4. aBIP48000S10OCIP48000S25OR	4. Fill the left syringe full from the input position at speed 10. At the same time fill the right syringe full from the input position at speed 25. After completing the syringe movements rotate the valves to the output position.
5. aQ	5. Poll until the pump is no longer busy and the hand probe is pressed.
6. aBD12000CD12000R	6. Dispense ¼ of the left and right syringe volume at the default speed.
7. aQ	7. Poll until the pump is no longer busy and the hand probe is pressed.
8. aBD12000CD12000R	8. Dispense another ¼ of the left and right syringe volume at the default speed.
9. aQ	9. Poll until the pump is no longer busy and the hand probe is pressed.
10. aBD12000CD12000R	10. Dispense another ¼ of the left and right syringe volume at the default speed.
11. aQ	11. Poll until the pump is no longer busy and the hand probe is pressed.
12. aBD12000CD12000R	12. Dispense the final ¼ of the left and right syringe volume at the default speed.
13. aQ	13. Poll until the pump is no longer busy.
14. a>D15R	14. Turns on the Digital TTL signal.

Example program 2:

This series of commands can be used with a daisy chain of three Dual Syringe Dispensers. The commands will fill all six syringes and then it will wait for a TTL trigger before using the broadcast command to simultaneously dispense various volumes from syringes across all three pumps.

Commands	Description
1. 1a	1. Auto-address (Expect response 1d)
2. :XR	2. Initialize the left and right syringe drives
3. aQ 4. bQ 5. cQ	3–5. Repetitively send these commands waiting between each command for the response. When all three pumps are no longer busy from the initialize command the next command can be sent. This is called polling.
6. aBIP48000OCIP48000OR 7. bBIP48000OCIP48000OR 8. cBIP48000OCIP48000OR	6–8. Fill the left and right syringe from the input position for each of the three pumps on the daisy chain. Then after completing the syringe movements rotate all valves to the output position.
9. aQ 10. bQ 11. cQ	9–11. Poll all three units one at a time until all pumps are no longer busy
12. aBD12000CD24000	12. Buffers command to dispense ¼ of left syringe and ½ of right syringe
13. bBD48000CD4800	13. Buffers command to dispense all of left syringe and 1/10 of right syringe
14. cBD42000CD42000	14. Buffers command to dispense 7/8 of left syringe and 7/8 of right syringe
15. a<D	15. Poll the Digital TTL Input until the trigger signal is received.
16. :R	16. Broadcasts the Execute command to all pumps to trigger the commands stored in the command buffer.

Example program 3:

This series of commands can be used to determine what the current syringe speeds and valve configuration is for a Single Syringe Dispenser or for the left side of a Dual Syringe Dispenser. Then adjust the syringe speed and valve configuration and finally store the change into non-volatile memory.

Commands	Description
1. 1a	1. Auto-address (expect response 1b)
2. aLQT	2. Requests the configuration for the left valve (expect a numeric response between 11–20)
3. aYQS	3. Requests the current syringe speed (expect a numeric response between 2–3692)
4. aLST19	4. Sets the current valve configuration to 19 which is Continuous Dispenser
5. aYSS25	5. Sets the current syringe speed to 25 seconds per stroke
6. a#SP1	6. Stores the current instrument settings to non-volatile memory so they will remain set after the power is cycled.

Appendix B — ASCII Chart

The following chart shows the relationship between binary numbers, decimal numbers, hexadecimal numbers and their ASCII equivalents for the numbers between 0 and 127.

Binary	Decimal	Hex	ASCII
00000000	0	00	<NUL>
00000001	1	01	<SOH>
00000010	2	02	<STX>
00000011	3	03	<ETX>
00000100	4	04	<EOT>
00000101	5	05	<ENQ>
00000110	6	06	<ACK>
00000111	7	07	<BEL>
00001000	8	08	<BS>
00001001	9	09	<HT>
00001010	10	0A	<LF>
00001011	11	0B	<VT>
00001100	12	0C	<FF>
00001101	13	0D	<CR>
00001110	14	0E	<SO>
00001111	15	0F	<SI>
00010000	16	10	<DLE>
00010001	17	11	<DC1>
00010010	18	12	<DC2>
00010011	19	13	<DC3>
00010100	20	14	<DC4>
00010101	21	15	<NAK>
00010110	22	16	<SYN>
00010111	23	17	<ETB>
00011000	24	18	<CAN>
00011001	25	19	<EM>
00011010	26	1A	<SUB>
00011011	27	1B	<ESC>
00011100	28	1C	<FS>
00011101	29	1D	<GS>
00011110	30	1E	<RS>
00011111	31	1F	<US>

Binary	Decimal	Hex	ASCII
00100000	32	20	
00100001	33	21	!
00100010	34	22	“
00100011	35	23	#
00100100	36	24	$
00100101	37	25	%
00100110	38	26	&
00100111	39	27	‘
00101000	40	28	(
00101001	41	29	)
00101010	42	2A	*
00101011	43	2B	+
00101100	44	2C	,
00101101	45	2D	-
00101110	46	2E	.
00101111	47	2F	/
00110000	48	30	0
00110001	49	31	1
00110010	50	32	2
00110011	51	33	3
00110100	52	34	4
00110101	53	35	5
00110110	54	36	6
00110111	55	37	7
00111000	56	38	8
00111001	57	39	9
00111010	58	3A	:
00111011	59	3B	;
00111100	60	3C	<
00111101	61	3D	=
00111110	62	3E	>
00111111	63	3F	?

Binary	Decimal	Hex	ASCII
01000000	64	40	@
01000001	65	41	A
01000010	66	42	B
01000011	67	43	C
01000100	68	44	D
01000101	69	45	E
01000110	70	46	F
01000111	71	47	G
01001000	72	48	H
01001001	73	49	I
01001010	74	4A	J
01001011	75	4B	K
01001100	76	4C	L
01001101	77	4D	M
01001110	78	4E	N
01001111	79	4F	O
01010000	80	50	P
01010001	81	51	Q
01010010	82	52	R
01010011	83	53	S
01010100	84	54	T
01010101	85	55	U
01010110	86	56	V
01010111	87	57	W
01011000	88	58	X
01011001	89	59	Y
01011010	90	5A	Z
01011011	91	5B	[
01011100	92	5C	\
01011101	93	5D	]
01011110	94	5E	^
01011111	95	5F	_
01100000	96	60	`
01100001	97	61	A
01100010	98	62	b
01100011	99	63	c
01100100	100	64	d
01100101	101	65	e

<table>
<tr><th>Binary</th><th>Decimal</th><th>Hex</th><th>ASCII</th></tr>
<tr><td>01100110</td><td>102</td><td>66</td><td>f</td></tr>
<tr><td>01100111</td><td>103</td><td>67</td><td>g</td></tr>
<tr><td>01101000</td><td>104</td><td>68</td><td>h</td></tr>
<tr><td>01101001</td><td>105</td><td>69</td><td>i</td></tr>
<tr><td>01101010</td><td>106</td><td>6A</td><td>j</td></tr>
<tr><td>01101011</td><td>107</td><td>6B</td><td>k</td></tr>
<tr><td>01101100</td><td>108</td><td>6C</td><td>l</td></tr>
<tr><td>01101101</td><td>109</td><td>6D</td><td>m</td></tr>
<tr><td>01101110</td><td>110</td><td>6E</td><td>n</td></tr>
<tr><td>01101111</td><td>111</td><td>6F</td><td>o</td></tr>
<tr><td>01110000</td><td>112</td><td>70</td><td>p</td></tr>
<tr><td>01110001</td><td>113</td><td>71</td><td>q</td></tr>
<tr><td>01110010</td><td>114</td><td>72</td><td>r</td></tr>
<tr><td>01110011</td><td>115</td><td>73</td><td>s</td></tr>
<tr><td>01110100</td><td>116</td><td>74</td><td>t</td></tr>
<tr><td>01110101</td><td>117</td><td>75</td><td>u</td></tr>
<tr><td>01110110</td><td>118</td><td>76</td><td>v</td></tr>
<tr><td>01110111</td><td>119</td><td>77</td><td>w</td></tr>
<tr><td>01111000</td><td>120</td><td>78</td><td>x</td></tr>
<tr><td>01111001</td><td>121</td><td>79</td><td>y</td></tr>
<tr><td>01111010</td><td>122</td><td>7A</td><td>z</td></tr>
<tr><td>01111011</td><td>123</td><td>7B</td><td>{</td></tr>
<tr><td>01111100</td><td>124</td><td>7C</td><td>|</td></tr>
<tr><td>01111101</td><td>125</td><td>7D</td><td>}</td></tr>
<tr><td>01111110</td><td>126</td><td>7E</td><td>~</td></tr>
<tr><td>01111111</td><td>127</td><td>7F</td><td></td></tr>
</table>

Part No.68559-01 Rev. B — 9/2015

HAMILTON®

Web: www.hamiltoncompany.com
USA: 800-648-5950
Europe: +41-58-610-10-10

Hamilton Americas & Pacific Rim
Hamilton Company Inc.
4970 Energy Way
Reno, Nevada 89502 USA
Tel: +1-775-858-3000
Fax: +1-775-856-7259
sales@hamiltoncompany.com

Hamilton Europe, Asia, & Africa
Hamilton Bonaduz A.G.
Via Crusch 8
CH-7402 Bonaduz, Switzerland
Tel: +41-58-610-10-10
Fax: +41-58-610-00-10
contact@hamilton.ch

To find a representative in your area, please visit hamiltoncompany.com/contacts.