MITOS P-PUMP BASIC

PROGRAMMING INSTRUCTIONS

Contents

1 Summary

This document contains details of how to communicate with the Mitos P-Pump Basic and Remote Basic gas pumps from your own external PC applications. Your code can be written in any language appropriate for your system that can use a standard PC serial ("com") port.

2 Introduction

This device uses an embedded 8 bit microcontroller (MCU) board for all pump control functions. All communications takes place via RS232 using a fixed length data packet for both send and receive. Multiple boards on a single RS232 bus are supported - each device on the line must have a unique ID. Communications are always one of:

- Set a mode
- Read a value
- Write a value
- Control a data stream
- Read firmware version

Values and modes/status information are written to and read from a number of variables which are all addressed by a location number. The module requires a single 24 volt DC supply – see user instructions.

3 Safety

It is responsibility of the user to ensure that all pressure equipment used with the Mitos P-Pump complies with all relevant directives and guidelines. Always use suitable personal protective equipment – e.g. safety glasses and shields etc. – when dealing with pressurised gases.

4 RS232 protocol

The RS232 protocol is a multi-drop implementation on a single RS232 bus. Comms settings for all devices are:

115200 baud, 8 data bits, 1 stop bit, no parity, no handshaking.

Each device on a single bus has a unique address in the range 0...15 (0xF), manually set by the small rotary address switch. Commands are sent to specific device addresses, though address 0 acts as a *broadcast* address – all devices will respond to commands sent to address 0 – *so you should never set a device to address 0*. All control over the device is achieved by reading from and writing to variables stored at particular addresses on the device – these addresses are simple numerical values in the range 0 to 127.

All data is sent to and received from the device as a 12 byte packet with the following structure.

Byte Number	Means
0	STX – Always STX (0x02, character 2) denoting the start of the packet.
1	ID – This byte is split into two 4 bit "nibbles". The top four bits are used as a "Packet ID" to tie responses to requests - range is

Byte Number	Means
	0...0xF (0...15). This feature may be ignored if not required. The lower 4 bits form the "Device ID" – this is the value set on the rotary setting on the board. Range is 0...0xF (0...15) and a board will only respond to messages sent to its own Device ID or ID = 0. Note that as all boards respond to packets sent to ID = 0, so you should not explicitly set your board to that address.
2	MSG – this is the *message* type to be sent or the *response* type received. There are five sent message types and four returned response types. See section 3.1 and 3.2 below.
3	Bytes 3...6 can be interpreted as four 8 bit bytes, two 16 bit words or a single 32 bit integer depending on the message being sent or received. Byte 3 is the high byte of the integer; bytes 3 and 5 are the high bytes of the words.
4	
5	
6	
7	Bytes 7...10 are normally interpreted as a 32 bit integer value – byte 7 is the high byte.
8	
9	
10	
11	CS – This is a checksum byte calculated for each message and is the bitwise exclusive OR of all preceding 11 bytes.

All packets sent to the device are acknowledged either with a simple "OK" packet, a "Requested Info" packet or an "Error" packet. When a value is written to a variable, you should always read it back (sometimes from a different location) to check it was received and executed correctly.

4.1 PC to Device Messages

The five types of messages that can be sent to the module are:

MSG Type	Meaning	Explanation
1	Write	Request to write a new value. The first 2 bytes (bytes 3...4) define a variable location (0...127) to be written, bytes 5 and 6 are unused. The value to be written to the specified variable is provided as a 4 byte integer in bytes 7...11.
2	Read	Request to read a current value. The first 2 bytes (bytes 3...4) define a variable location (0...127) to be read, bytes 5...11 are unused.
3	Device Mode	Request to set the device mode. The first 4 bytes (bytes 3...7) contain the *mode number*. Allowable modes are numeric 0...5. The second 4 bytes contain any required parameter. See below for more details.
4	Stream	Request to stream variables. Up to 4 variables can be set to stream data continually (without further requests) at a predefined rate. Streaming takes place at the rate set (in milliseconds) in variable 1. The speed at which you can successfully stream data depends on the bus traffic. With multiple bus devices and/or multiple additional data requests, if you stream to fast, you will see communication errors. Under these

MSG Type	Meaning	Explanation
		conditions you should always check the returned packets checksum. First 2 bytes: first and second variable location (0...127). Second 2 bytes: third and fourth variable location (0...127). A value of >127 in any of bytes 3...7 will stop the stream on that variable only. Prompt with all values set to >127 to stop all streaming.
5	Firmware version	Requests the present firmware version.

Message type 3 – “Device Mode” – *mode number* has five possible values.

Mode	Meaning	Parameter	Explanation
1	Go to bootloader mode	-	Bootloader mode is not useful for the user.
2	Go to “safe” state	-	Puts the device in a safe state. This stops control and vents the chamber.
3	“Ignore Comms”	*Time*	Sets the device to ignore all further comms activity for *Time* seconds. This can useful when on a multi drop RS232 system and other nodes are being updated.
4	Reset	-	Performs a soft reset of the MCU and goes “safe”.
5	Preserve variables	-	Copies all static variable (0...63) to the on board flash memory (for permanent storage at power off). This is automatically performed during*Taring*.

4.2 Device to PC Responses

These are generally solicited response packets (except for *streamed* variables) and the format is the same as the PC to device messages. The ID byte will match what was received from the PC (i.e. the response packet will have the sent packets ID), though the Packet ID for streamed responses is meaningless - it will normally reflect the last requests packet ID, *not* the original stream requests packet ID.

All unused bytes in the packet are undefined – they will commonly contain data (*junk*) from previous packets. Note however, the checksum is always calculated using all eleven bytes.

The MSG response type is one of:

MSG Type	Meaning	Details
1	Read data	First 4 bytes (bytes 3..6) give variable address. Second 4 bytes (bytes 7..10) give current variable value
2	OK	No further valid data - that is, 8 bytes of junk from the last command
3	Error	One byte (see below) + 7 bytes of junk
4	Firmware	First 4 bytes (bytes 3..6) give firmware version as integer + 4 bytes of junk. Byte 5 is the major version number, byte 6 the minor version number.

The Error response indicates a problem with the sent message rather than a physical error on the device. The possible values for the error byte are:

Error	Meaning	Details
1	Checksum error	The previously sent message failed the checksum test.
2	Command Unknown	The previously sent message command was not recognised.
3	Data Invalid	The previously sent message contained invalid data e.g. an out of range address or value.
4	Timeout	Communications timeout failure.

5 Variable locations

There are two types of location, though they behave identically from the programming perspective: Variables 0...63 are considered STATIC and variables 64...127 are DYNAMIC. Data held in the static variables will persist over power cycles if explicitly saved back to the flash memory on the device (see “Mode” 5 in 3.1 above.) This is typically used for persistent tare constants etc. and these data are explicitly reloaded back into memory from the flash on boot or reset.

All data are stored as 4 byte integers and variable locations are defined in the table below.

Static Variables	Usage	Comment
0	Application type	The PPB is application type 3
1	Stream rate	Read and write the data streaming rate here – in msecs between streamed packets.
2	Application version	Unsigned decimal. Please do not modify – this is used for recognition and diagnostic purposes only
3	Module serial number	Unsigned decimal. Please do not modify – this is used for recognition and diagnostic purposes only
14	Tare Atmospheric Pressure	Atmospheric pressure at tare. Divide by 10 to get value in mbar absolute - set by *Tare*
15	Tare Supply Pressure	Supply tare offset (mbar) - set by *Tare*
16	Tare Chamber Pressure	Chamber tare offset (mbar) - set by *Tare*
19	Fluid name for fluid calibration	Four letters as four ASCII bytes, use space for blank - see standard fluid list (section 7.2) below for values. Note the order of the characters in the byte are as follows: `const char name[4] = {'0','4','C','F' };`
21	Flow sensor type for fluid calibration	Unsigned long. Set this before set 19. Note that if it does not match the real flow sensor type, then default calibration (water) will be used to calculate the reported flow rate. (Note the stored calibration data in these static memory locations will not be changed in this circumstance, so external controllers can detect this by comparing the value here with the connected flow sensor type from address 88). Ignored if external flow sensor is used

Dynamic Variables	Usage	Comment
64	Atmospheric sensor	Read the current absolute atmospheric pressure. Divide by 10 to get value in mbar absolute.
65	Supply sensor	Read the supply pressure in mbar gauge. This value is corrected for the most recent tare.
66	Chamber sensor	Read the chamber pressure in mbar gauge. This value is corrected for the most recent tare.
67	Temperature of atmospheric pressure sensor	Divide by 10 to get value in Celsius.
68	Temperature of supply pressure sensor	Divide by 10 to get value in Celsius.
69	Temperature of chamber pressure sensor	Divide by 10 to get value in Celsius.
73	Current flow rate (pl/sec)	Integer. Read-only variable that contains the flow rate (if a flow sensor is present)
74	External flow rate (pl/sec)	Long. Write-only variable that contains the flow rate from an *external flow sensor*. This is used in flow control mode if no internal flow sensor is connected. Write the value as (flow_rate **OR** 0x80000000), the extra bit signals a new value has been written. This is cleared when the pump uses the flow rate.
75	Low pressure leak rate	Result is 4 bytes. Highest 2 bytes are the leak rate in mbar/bar/minute. Lowest two bytes are a flag and the pressure at which the test was done. Bit 15 is the flag – 0 = pass and 1 = fail, bits 0...14 is the pressure at which the test was done.
76	High pressure leak rate	
77	Control Type and External flow sensor type	Integer. The lowest byte specifies the type of control 0 – PRESSURE control 1 – FLOW control On a 0⇨1 transition, the pump automatically sets the target flow rate to the current flow rate (i.e. effectively 73 is written into 79). On a 1⇨0 transition, the pump automatically sets the target pressure to the current chamber pressure (i.e. effectively 66 is written into 79). The second lowest byte is the flow sensor type (see 7.1) when a flow sensor with display is connected to the pump.
78	Control Mode	Write this variable location to specify the '*Control Mode*': 0 - *IDLE* - halt current operation. 1 - *CONTROL* pressure or flow at the current target. 2 - *TARE* - performs a system tare and then reverts to idle. The second lowest byte is the tare options: 0 any; 1 pressure only; 2 flow only. [3 - *ERROR* (read only, see variable 81 below)] 4 – LEAK TEST – performs a system leak test

Dynamic Variables	Usage	Comment
79	Target (Pressure or flow – set by 77)	Integer. Write the required target (in mbar gauge or pl/sec) to this variable location. When control mode is 1, the value in here is used as the target pressure or flow. If the target is set to 0, the pump reverts to IDLE if in pressure control mode but NOT in flow control mode, as in a multiple pump system, maintaining a zero flow rate may require the pump has to actively do something with the pressure control to prevent back flow. Acceptable pressure targets are the range between min pressure target (address 89) and max pressure target (address 90) and 0. Acceptable flow targets are between 20% of the min flow rate range and the max flow rate range for the current flow sensor and 0.
80	Current target (pressure or flow – depends on 77)	Read-only variable location. This value reflects the current target pressure or flow of the pump (in mbar gauge or pl/sec). Read this back after writing to 79 to confirm the change has taken effect.
81	Control mode/Status	Read-only variable location. This is the current pump control mode and uses the same numeric codes as variable 78 above. You should logically AND this location with 0xFF to remove other information. After writing a new control mode to 78, read this location to monitor the pumps present state and when the pump has completed any operation (e.g. tare). If an error occurs, this variable will respond with '3 – *ERROR*'. The exact error type can be read from location 82. To clear an error, you need to set control mode to '*IDLE*' (Set variable 78 to 0).
82	Error Number	Read the last error type from here. This value is only meaningful when 81 reads back as 3 – ERROR. Values/meanings are: 0 – No error. 1 – Supply > maximum pressure. 2 – Tare: timed out. 3 – Tare: supply still connected. 4 – Control start timed out. Opening valves, but chamber pressure unchanged. 5 – Target too low. 6 – Target too high. 7 – Leak test: supply pressure too low 8 – Leak test: time out if the pressure cannot reach target 9 - Flow sensor lost during flow control.
88	Flow sensor type	Integer. Read-only variable. If this value differs from the sensor type for the stored calibration data in memory location 21 then the "default" calibration for water will be used to calculate the reported flow rates. See flow sensor types below.

Dynamic Variables	Usage	Comment
89	Minimum Target Pressure (mbar)	Integer. Read-only variable. Current minimum settable target pressure. This may be negative or positive!
90	Maximum Target Pressure (mbar)	Integer. Read-only variable. Current maximum settable target pressure. This may be negative or positive!

6 Controlling the pump

This section deals primarily with pressure control but flow control is handled in exactly the same way – see section 7 for more information on flow control.

6.1 General

You control the pump by writing appropriate values to the control registers. After each command is sent to the pump, you should wait for a response. This may be 'OK', a 'Read' response (with data) or an 'Error' response. When writing new values (e.g. pressure targets), its good practice to read the value back to check the write was successful.

If the pump goes into error whilst running, its control mode will change to 3 (*ERROR*). You can only detect this by regularly poling the control mode (location 81) and you must clear the error state by re-setting the control mode to 0 (*IDLE*) *after* any physical error condition has been corrected. When in error, you can then read the exact error type from location 82.

6.2 Initialise

Powering up the board performs an initialisation. It's also possible to re-initialise by performing a soft reset (see 3.1 above). This isolates the supply, vents the chamber and re-reads all the static variables into dynamic memory. All streaming rates, tare etc. will be retained.

6.3 Taring

The three pressure sensors will typically all show a slightly different reading when attached to the same pressure. This difference is an offset than can be removed by *taring* the pump. This should be done with the supply line removed and any chamber vented, by setting the pumps control mode to 2 (*TARE*) – (see section 4 above).

After successfully taring, the pump will revert to control mode 0 (*IDLE*), the pressure sensor readings (variables 64..66) should all read the same pressure and the tare variables (memory locations 14..16) will contain the atmospheric pressure at the time of taring, and the chamber and supply pressure sensors offsets.

The tare will change over time as the sensors age and so should be periodically repeated.

6.4 Venting

You can vent the chamber by setting the pump state to 0 (*IDLE*). The pump target pressure (location 80) is not reset.

6.5 Controlling

To control at a fixed pressure setpoint you must send the target pressure to the system and then set the control mode (location 78) to 1 (*CONTROL*). The set target cannot be read back from location 80 until the control mode is set to *CONTROL*. The pump will now open and close its valves as required to maintain the requested target pressure in the chamber. If you set the target too close to or above the supply pressure, the pump will not be able to reach it.

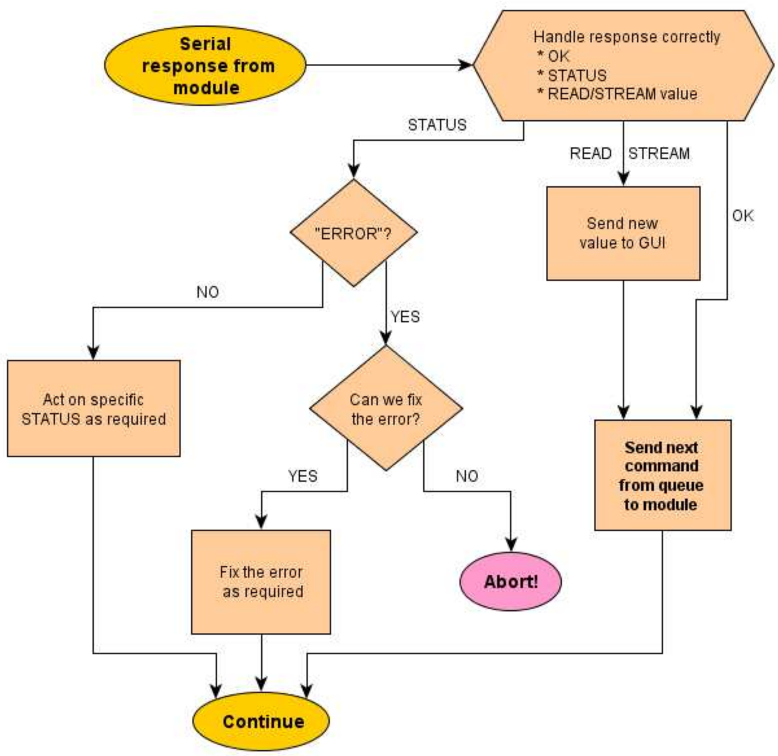
Stop control by either explicitly setting the control mode to *IDLE* or by setting the pump target pressure back to 0 (the pump control mode will change to *IDLE*). In each case, control will stop and the pump will vent.

6.6 Leak test

You can leak test the system by connecting a standard sealed chamber, connecting a gas supply and setting the control mode to 4 – LEAK TEST. The test takes about 2 minutes and performs a leak test at two pressures, close to the supply and close to atmospheric. The system is deemed leak tight if neither test shows a change in pressure of more than ±5mbar/bar/minute. Leak test results can be found in locations 75 (low pressure test) and 76 (high pressure test). After completion the pump will return to IDLE or error if an error occurs.

6.7 Control strategy

This section assumes the pump has previously been properly tared. Exactly what information you require from the pump depends on your particular task – e.g. you may or may not be interested in the sensor temperatures.



Typically you will always need at least the chamber pressure and the status continually, so if you set these up as a *stream*[1] (message type 4) with a 500msec update you will have one new value for each every second. You need the status frequently to check for the pump going into *ERROR*. Other variables of lower interest can be read as frequently as required.

It's convenient to build a circular "queue" of periodic commands that can be issued sequentially after a response is received from the module. You can then insert individual mode or target change commands into the loop which will be executed in sequence.

As communication is asynchronous, it's safest to write commands to the pump immediately following a complete response from the pump to avoid collisions on the bus.

The schematic below shows a simple way of handling the data flow to and from the pump. The command queue handling is not explicitly shown.

Bear in mind you must handle any error conditions explicitly – the *ERROR* can only be cleared by re-setting the control mode to *IDLE* once the physical condition that caused the error has been resolved.

[1] You can stream up to four variables of interest. See section 3.1.

For example, the maximum permitted supply pressure is fixed at 11.5 bar and exceeding this will result in *ERROR* and the chamber venting. You can only set the control mode back to *IDLE* once the supply pressure is reduced below 11.5 bar.

7 Flow control

Flow control is inherently more complex than pressure control and requires additional flow control hardware and sensors attached to the P-Pump. If you are only using pressure control, you can ignore this section. The information here applies to the Mitos sensor interface (#3200200) when connected to a P-Pump basic and fitted with an appropriate flow sensor.

Controlling by flow requires knowledge of the flow sensor type and the fluid being pumped. There are five standard fluids calibrations defined and it's possible to define further ones – contact Dolomite Support for details.

It generally good practice to establish the flow you need using pressure control and then write 1 to location #77 to bumpblessly transfer to flow control at the current flow target. Similarly, writing 0 to #77 will cause a smooth transition back to pressure control, and setting the target to 0 will then switch the pump off.

Note that when in flow control mode, setting a "flow" target of 0 will normally **not** result in the pump moving to IDLE in pressurised systems.

7.1 Types of Flow Sensor

This is an enumeration used to identify Sensirion flow Sensors. 0 implies no flow sensor is connected.

Type number	0	1	2	3	4	5
Type	n/a	LG16-0025	LG16-0150	LG16-0480	LG16-1000	LG16-2000
Range (water)	none	0.07-1.5 µl/min	0.4-7µl/min	1-50µl/min	30-1000 µl/min	200-5000 µl/min
Flow sensor units/ dp		µl/min / x.xxx or µl/min / -x.xx	µl/min / x.xx or µl/min / -x.xx	µl/min / xx.x or µl/min / -xx.x	µl/min / xxxx or µl/min / -xxx	ml/min / x.xx or ml/min / -x.xx

Note when using a vacuum pump to provide a negative pressure supply, it is possible to control flow "backwards", that is the negative version the flow rates above are possible. However different flow sensors have different accuracies for positive and negative flow rates as can be seen above, so when converting these values for display you need to convert the units from pl/s to the units for the flow sensor in the table above and then pay attention to the number of decimal places it supports.

7.2 Fluid names

Note the minimum flow rate actually means the minimum *calibrated* flow rate, lower values can measured but will be less accurate.

Fluid	Short Name
Water	<SPACE>H2O
FC-40	FC40
Novec 7500	NOVE
Hexadecane	HEXA
Mineral Oil	<SPACE>OIL

8 Example Messages and Responses

The examples below all show the packets as 12 bytes in hexadecimal. For simplicity in all cases, the device ID is assumed to be 1 (this is the standard setting for a P-Pump basic) and the packet ID is set to 0: the checksum is correct. Each set is the sent message followed by the (typical) received response, junk in the response is *italicised in grey* [NB. junk will vary].

STX	ID	MSG	B1	B2	B3	B4	B5	B6	B7	B8	CS
Request Reset – performs a soft reset											
02	01	03	00	00	00	04	00	00	00	00	04
02	01	02	*02*	*40*	*01*	*04*	*51*	*F9*	*C0*	*25*	0B
Response MSG is 'OK'. Everything else is *junk*.											
Write variable 1 – stream rate in msecs. (0x000001F4 is 500)											
02	01	01	00	01	00	00	00	00	01	F4	F6
02	01	02	*00*	*00*	*00*	*01*	*00*	*00*	*01*	*F4*	F5
Response MSG is 'OK'. Everything else is *junk*.											
Read Variable 1 – Stream rate in msecs. (Response of 0x01F4 is 500msecs)											
02	01	02	00	01	00	00	00	00	00	00	00
02	01	01	00	00	00	01	00	00	01	F4	F6
Response MSG is 'READ'. 0x00000001 is variable and 0x000001F4 is the value (500).											
Change Control Mode variable 78 (0x4E) - start Control, Tare or Leak test.											
02	01	01	00	4E	00	00	00	00	00	0X	??
Response should be 'OK'. You should regularly read the control mode status (variable 81) to monitor the PPBs current control state. TARING will automatically drop back to IDLE on completion. X values: 0 = IDLE, 1 = CONTROL, 2 = TARE, 4 = LEAK TEST											
Read control mode status variable 81 (0x51)											
02	01	02	00	51	00	00	00	00	00	00	50
02	01	01	00	00	00	51	00	00	00	0X	??
Response MSG is 'READ'. 0x00000051 is variable and 0x000000XX is the value. X values: 0 = IDLE, 1 = CONTROLLING, 2 = TARING, 3 = ERROR, 4 = LEAK TEST											
Stream variable 64, 65, 79 and 81 (0x40, 0x41, 0x4F and 0x51)											
02	01	04	40	41	4F	51	00	00	00	00	1A
02	01	02	*00*	*00*	*00*	*51*	*00*	*00*	*00*	*00*	50
02	01	01	00	00	00	40	00	00	26	E9	8D
etc.											
First response MSG is 'OK'. Stream of 'READ' responses follows, for each variable - first variable shown above is value 0x000026E9 (9961) in variable 0x40 (64 – atmospheric pressure)											
Stop stream of variables 64 and 65 (keep 79 and 81)											

STX	ID	MSG	B1	B2	B3	B4	B5	B6	B7	B8	CS
02	01	04	F0	F0	4F	51	00	00	00	00	19
02	01	02	*00*	*00*	*00*	*51*	*00*	*00*	*00*	*00*	50
02	01	01	00	00	00	4F	00	00	07	D0	9A
etc.											
First response MSG is 'OK'. Stream of 'READ' responses follows, but now only variables 79 and 81 are streamed. Variable shown above is 0x4F (79 – target pressure) - value 0x000007D0 (2000)											

9 Default Settings

As supplied, the unit has been initially tared and a total leak rate measured[2]. The leak test was performed attached to a 30ml sealed chamber, using 6 bar supply pressure. Maximum supply pressure is set to 11.5 bar.

10 Support Software

This document provides instruction on how to interface with the device from your own software. You can use any programming language that can directly use a standard PC serial ("com") port and any operating system - Windows, Linux, Mac OS etc.

11 Further Information

For more information about Dolomite and our range of other microfluidic products, please visit:

http://www.dolomite-microfluidics.com/

Contact us at:

Tel: +44 (0)1763 242491 (Rest of World)

Tel : 617 803 6655 (USA)

Fax: +44 (0)1763 246125

Email: sales@dolomite-microfluidics.com

Support : support@dolomite-microfluidics.com

[2] An acceptable leak rate of <5mbar/bar/minute maximum is specified. Typical values are <3mbar/bar/minute.

The Dolomite Centre Ltd.
Unit 1, Anglian Business Park, Royston,
Hertfordshire, SG8 5TW, United Kingdom

T: +44 (0)1763 242491
F: +44 (0)1763 246125
E: info@dolomite-microfluidics.com
W: www.dolomite-microfluidics.com

Dolomite Microfluidics
29 Albion Place
Charlestown, MA 02129

F: 617 848 1211
F: 617 500 0136
E: salesus@dolomite-microfluidics.com
W: www.dolomite-microfluidics.com