ttpventus
Performance. Precision.

Technical Note TN003:
Serial Communications Guide

1 INTRODUCTION

1.1 Disc Pump

Disc Pump is a silent, high-performance piezoelectric micropump.

Owing to its operating mechanism, Disc Pump can be controlled with unmatched precision, yet at the same time respond to full-scale set point changes in a matter of a few milliseconds. The compact form factor means it can be tightly integrated into products, increasing portability.

Figure 1. A Disc Pump

1.2 Communication Interface

This Application Note provides details of the commands recognized by the Disc Pump Evaluation Kit Drive PCB via a serial-over-USB connection. For more general operating instructions for the Evaluation Kit, please refer to its user manual.

The information in this Application Note is believed to be accurate and complete, however TTP Ventus does not accept any liability whatsoever for any consequential damage or losses arising from any errors, inaccuracies, or omissions.

2 DISCLAIMER

This Application Note is provided "as is" and without any warranty of any kind, and its use is at your own risk. TTP Ventus does not warrant the performance or results that you may obtain by using this Application Note. TTP Ventus makes no warranties regarding this Application Note, express or implied, including as to non-infringement, merchantability, or fitness for any particular purpose. To the maximum extent permitted by law TTP Ventus disclaims liability for any loss or damage resulting from use of this Application Note, whether arising under contract, tort (including negligence), strict liability, or otherwise, and whether direct, consequential, indirect, or otherwise, even if TTP Ventus has been advised of the possibility of such damages, or for any claim from any third party.

3 COMMUNICATIONS INTERFACE

The Disc Pump evaluation PCB implements a serial-over-USB interface, accessible via a USB mini-B receptacle.

3.1 Attaching the PCB to your computer

The PCB is attached to your computer via a USB A to USB mini B lead. A suitable lead is supplied with our evaluation system. Please note that similar leads supplied (e.g. with bicycle lights) for battery charging tend not to have the data lines connected and are therefore not suitable.

Connecting the PCB to your computer should result in a COM port being installed by the operating system software. If this does not happen automatically, please download and install the appropriate device driver from the FTDI website:

https://www.ftdichip.com/FTDrivers.htm

Once the COM port is installed, make a note of the port number. In Windows this can be identified by looking in the “Ports” section of the “Device Manager” window. With the Ports section open, unplug the USB cable connecting your computer to the PCB and note which COM port disappears. Reconnect the USB cable once the port is identified.

3.2 Opening the COM port

The PCB’s COM port runs at a baud rate of 115,200, 8 bits, no parity, and one stop bit.

Please refer to the documentation supplied with your chosen software development environment for details of how to open a serial port.

3.3 Reading and Writing to the PCB

The operation of the PCB is controlled by a number of registers, which are either “read” or “read/write”.

To write to a PCB register, a command is written to the appropriate COM port (which must first be opened within your software). The command to write to a register takes the form:

#W<REGISTER_NUMBER>,<VALUE>\n

Please refer to the documentation supplied with your chosen software development environment for details of how to write to a COM port. Please note that the new-line character ‘\n’ must be appended to all commands.

The PCB responds to “write” commands by echoing the command back. This response should be read and checked by the controlling software to confirm that the command has been received correctly. If the command causes an error, or isn’t received at all, the PCB doesn’t respond.

To read from a PCB register, a command is written to the appropriate COM port (which must first be opened within your software), and the response is then read from the COM port. The command to read from a register takes the form:

#R<REGISTER_NUMBER>\n

The PCB responds to “read” commands by echoing the command back, followed by a comma and the value held in the register. The board is ready to receive another command after a response is received. If no response is received after 100ms, the command should be considered lost and a repeat command issued as necessary. In any event, do not send another command to the PCB until either (i) a response has been received to the previous command, or (ii) 100ms has elapsed.

3.4 Stream Mode

The PCB has a streaming mode, in which a comma separate list of useful variables is regularly sent (about 60Hz). This mode can be activated with register 2, described in section 3. Streamed variables take the form:

#S<PUMP_ENABLED>,<VOLTAGE>,<CURRENT>,<FREQUENCY>,<ANA1>,<ANA2>,<ANA3>,<FLOW>,<CHK>\n

The <CHK> field contains a simple 1-byte checksum, used to validate the rest of the streamed message. The checksum is computed by taking the ascii value of each character in the line, before <CHK> appears, and adding them together. This sum is then limited to 0-255 by taking the modulo of the sum to 255:

SUM % 255 = expected checksum value

4 COMMANDS

4.1 Pump Enable

This register controls whether the pump is enabled or not and overrules all other register settings.

Register ID	Read/Write	Name	Values
0	R/W	Pump enabled	0 = disabled 1=enabled

Example:

#W0,1\n Enables the pump

4.2 Power Limit

The maximum power to the pump, under any circumstance is limited with this register value, which overrules all of the control modes.

Register ID	Read/Write	Name	Values
1	R/W	Power limit	0 to 1400 milliwatts

Example:

#W1,1000\n Limits the pump input power to 1000mW

4.3 Stream Mode

Allows the streaming mode to be enabled.

Register ID	Read/Write	Name	Values
2	R/W	Enable stream mode	0=disabled 1=enabled

Example:

#W2,1\n Enables the streaming mode

4.4 Read Only Registers

Several read only registers are provided, so that useful information about the PCBs current condition may be read.

Register ID	Read/Write	Name	Values
3	R	Drive Voltage	0 to 60 volts
4	R	Drive Current	0 to 150 milliamps
5	R	Drive Power	0 to 2000 milliwatts
6	R	Drive Frequency	20000 to 23000 Hertz
7	R	Analog 1 (dial on eval kit)	Value range dependent on channel gain and offset
8	R	Analog 2 (pressure on eval kit)	Value range dependent on channel gain and offset
9	R	Analog 3 (analog in on eval kit)	Value range dependent on channel gain and offset

Example:

#R3\n	Requests the current drive voltage
#R3,25123\n	Response from the PCB, for a drive voltage of 25.123 volts

4.5 Control Mode

The PCB offers three control modes:

- Manual mode, where the drive power can be set directly.
- PID mode, where the output of a PID controller, sets the pumps driving voltage. Can be used for closed loop control of a parameter, such as pressure.
- Bang Bang mode.

Register ID	Read/Write	Name	Values
10	R/W	Control mode	0=Manual 1=PID 2=Bang Bang

Example:

#W10,1\n	Set the PCB to use the PID control mode

4.6 Manual Mode Settings

One of four system inputs can be used to set the target drive power for the pump, in milliwatts.

Register ID	Read/Write	Name	Values
11	R/W	Manual mode source	The source of the target power (in milliwatts) when in manual mode: 0=Set val (register 23) 1=Analog 1 2=Analog 2 3=Analog 3

Examples:

#W11,2\n Use the Analog 2 value (after gain and offset) as the target power

4.7 PID Mode Settings

Several registers are available to flexibly configure the PCBs internal PID controller.

Register ID	Read/Write	Name	Values
12	R/W	PID setpoint source	0=Set val (register 23) 1=Analog 1 2=Analog 2 3=Analog 3
13	R/W	PID input source	0=Set val (register 23) 1=Analog 1 2=Analog 2 3=Analog 3 4=External Flow Sensor
14	R/W	PID proportional coeff	Unbounded but values within -2000 to 2000 are recommended
15	R/W	PID integral coeff	Unbounded but values within -100 to 100 are recommended
16	R/W	PID integral limit coeff	The PID mode output controls the voltage used to drive the pump in millivolts. Therefore, setting this value to the peak drive voltage the pump might use is recommended. Typically, this can be left at 55,000.
17	R/W	PID differential coeff	Unbounded, but rarely useful in practise. Leaving this at 0 is recommended.

Example:

#W12,0\n	Set the manual set value as the PID setpoint source
#W13,2\n	Use Analog 2 as the PID input source
#W14,100\n	Use a proportional coefficient of 100
#W15,10\n	Use an integral coefficient of 10

4.8 Bang Bang Mode Settings

Several registers are available to configure the PCBs internal bang bang controller.

Register ID	Read/Write	Name	Values
18	R/W	Bang Bang input source	0=Set val (register 23) 1=Analog 1 2=Analog 2 3=Analog 3
19	R/W	Bang Bang lower threshold	Unbounded
20	R/W	Bang Bang upper threshold	Unbounded
21	R/W	Bang Bang lower power mW	The drive power in milliwatts when the lower threshold is reached
22	R/W	Bang Bang upper power mW	The drive power in milliwatts when the upper threshold is reached

Example:

#W18,2\n	Use analog 2 as the input to the bang bang controller
#W19,10\n	Set the lower threshold to 10
#W20,100\n	Set the upper threshold to 100
#W21,1000\n	Set the power at the lower threshold to 1 watt
#W22,0\n	Turn the pump off when the upper threshold is reached

4.9 Input Settings

Several registers are available to configure the system inputs. Each input can be routed to a control mode, through the registers for that mode.

The three analog inputs provide a raw value between 0 and 1. A gain and offset is applied to each analog input, before the value is routed to the areas where it is used. For example, a gain of 500, and offset of 250, could be applied to analog 3, and then analog 3 used as the input to the manual mode. This would allow the analog 3 input to control the drive power between 250mW and 750mW.

Register ID	Read/Write	Name	Values
23	R/W	Set Value	Unbounded. This value can be used as the input for the different modes.
24	R/W	Analog 1 Offset	Offset applied to the analog 1 input after gain is applied.
25	R/W	Analog 1 Gain	Gain applied to the raw analog 1 input, which is between 0 and 1.
26	R/W	Analog 2 Offset	Offset applied to the analog 2 input after gain is applied.
27	R/W	Analog 2 Gain	Gain applied to the raw analog 2 input, which is between 0 and 1.
28	R/W	Analog 3 Offset	Offset applied to the analog 3 input after gain is applied.
29	R/W	Analog 3 Gain	Gain applied to the raw analog 3 input, which is between 0 and 1.
30	R/W	Store current settings	Writing a 1 to this register, causes the current settings to be stored in flash. These are then retrieved when the board powers up.

Example:

```
#W23,500\n      Set the “Set Value” to 500
#W28,250\n      Set the analog 3 offset to 250
#W29,500\n      Set the analog 3 gain to 500
```

5 FURTHER SUPPORT

5.1 Code snippet library

The TTP Ventus code snippet library, hosted on GitHub (https://github.com/TTP-Ventus/), provides serial communication and control examples in Python for common functions, including turning the pump on and off, setting drive power, closed loop control of pressure and reading back and plotting data. The code snippet library implements the aspects of the communication protocol set out in this Application Note and is intended to support customers after their initial evaluation of our pump technology, as they move on to developing prototypes and products.

5.2 Additional Support

The support section of TTP Ventus website (https://www.ttpventus.com/support) provides advice on:

- Getting Started
- Applications
- Development Process
- Downloads (including datasheets, application notes, case studies and 3D models)
- Frequently Asked Questions

TTP Ventus is happy to discuss next steps beyond prototyping, including system design. If you would like to discuss this with us, or for any other additional support, please contact us at support@ttpventus.com.

6 REVISION HISTORY

Date	Revision	Change
03 August 2021	r210803	Update to TN and new document format.
23 April 2021	r210423	Add setting 4 to Register 13 and Code Snippet Library to Further Support.
19 June 2020	r200619	Corrected typing errors.
28 May 2020	r200528	Added <FLOW> to streamed values in 3.4.
29 January 2020	r200129	Correct wrong USB receptable specification in 3.0.
28 May 2019	r190528	Reissue as AN003.
31 January 2019	r190131	Updated the documentation to match the new evaluation kit commands.
28 September 2018	r180928	Initial revision.