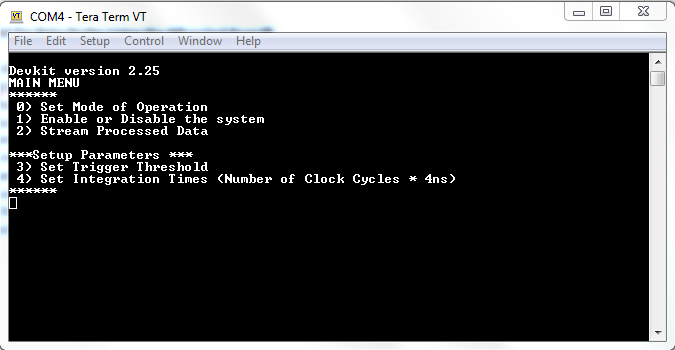
**Command List for Beta Probe Using the MicroZed Board**

This document will instruct the user about the commands and the proper syntax for controlling the MicroZed (uZ) board for the Beta Probe project. The commands will be listed along with the syntax for writing the command, what the command is used for, what the latency is for the command, and what the return value is.

Communication to and from the uZ is done serially and all commands need to be terminated by ‘\r’, carriage return, or ‘\n’, newline, for them to be recognized by the system. The system is constantly polling for user input, but unless it is properly terminated, the input will be rejected.

The user will interact with the uZ primarily through the “main menu”, see Figure 1, which is the access point for any of the functions of the uZ. From the main menu, the user will enter an integer to choose a function from a list. That list includes five different functions of the system, with one of them being data acquisition, and the other four being used to set system parameters.

**Figure 1: the main menu of the Beta Probe Emulator as seen via Tera Term.**

Before streaming processed data, it is important to set the uZ system parameters correctly, otherwise data may not be available. The first system parameter to set is what mode of operation is desired. This parameter determines what type of data will be streamed by the system and the user can choose from a list of data streaming modes including:

* AA Waveforms
* LPF Waveforms
* DFF Waveforms
* TRG Waveforms
* Processed Data

As this project progresses, more of these modes may be made available, but for now, always choose option 4: processed data mode. The other modes are primarily used for diagnostics when running the system, but may be useful in the future.

The next system parameter to set is the trigger threshold, option 3 from the main menu. This sets the value that a pulse must have to successfully trigger the system and register an event. The value set for the trigger threshold must be an integer between 0 – 10240. After sending a value to the system, it will echo the value that was set back to the user.

The other system parameters that must be set before streaming processed data are the integration times. These control how long a triggered event will be integrated for and directly govern the values of the processed data. There are four integration times which must be set: baseline, short, long, and full. The default values are below:

* Baseline = -52
* Short = 88
* Long = 472
* Full = 6000

These values will change as the system develops, but for now just use the defaults.

The last thing to do before streaming processed data is to enable the system. This enables the readout of data from the ADC and tells the system to begin receiving data. There are two options, enable and disable, where enable gets the system ready to start taking data and disable turns off that capability.

**System Commanding**

This section has the syntax, parameters, and return values of each command listed. Either a ‘\r’ or ‘\n’ may be used to tell the system to receive a command, but for this section only ‘\r’ is shown.

**Set Mode of Operation**

Syntax: *0\r*

This allows the user to change the mode of operation. To select a mode of operation, use one of the following inputs:

|  |  |
| --- | --- |
| Mode | Input Syntax |
| AA Waveforms | 0/r |
| LPF Waveforms | 1/r |
| DFF Waveforms | 2/r |
| TRG Waveforms | 3/r |
| Processed Data | 4/r |

Return: *Transfer Processed Data\n\* or *Invalid Command\n\*rr

The system will echo back the mode that was chosen or if the input was not 0-4, the system will send invalid command.

**Enable/Disable the System**

Syntax: 1\r

This allows the user to enable or disable the ADC. The following input is allowed:

|  |  |
| --- | --- |
| Mode | Input Syntax |
| Disable System | 0/r |
| Enable System | 1/r |

Return: *DAQ Enabled\n\r* or *Invalid Command\n\r*

The system will echo back the choice of the user or if the input was not 0 -1, the system will send invalid command.

**Stream Processed Data**

Syntax: 2\r

This command immediately begins streaming processed data. To stop streaming data, the user enters a ‘q’ to the terminal. The system is constantly checking for ‘q’ and will stop streaming immediately after receiving it. This means that streaming can be stopped in the middle of reading out an event.

Return: Processed Data

Data will be printed to the serial port as ASCII characters followed by a space then both return characters, ‘\r’ and ‘\n’. A sample event is shown below:

|  |  |
| --- | --- |
| Data Product | Example Serial Output |
| Identifier | 111111 \r\n |
| Time | 16562707 \r\n |
| Total Events | 3807 \r\n |
| Event Number | 3807 \r\n |
| PSD: Baseline Integral | 656432 \r\n |
| PSD: Short Integral | 3809776 \r\n |
| PSD: Long Integral | 18243584 \r\n |
| PSD: Full Integral | 35294992 \r\n |
| PH: Channel 0 | 6586 \r\n |
| PH: Channel 1 | 1526 \r\n |
| PH: Channel 2 | 5660 \r\n |
| PH: Channel 3 | 9017 \r\n |
| PH: Channel 4 | 4978 \r\n |
| PH: Channel 5 | 12620 \r\n |
| PH: Channel 6 | 47309 \r\n |
| PH: Channel 0 | 21335 \r\n |
| PH: Channel 0 | 4094 \r\n |
| PH: Channel 0 | 2328 \r\n |
| PH: Channel 0 | 6767 \r\n |
| PH: Channel 0 | 5086 \r\n |
| PH: Channel 0 | 2026 \r\n |
| PH: Channel 0 | 7914 \r\n |
| PH: Channel 0 | 570 \r\n |
| PH: Channel 0 | 1721 \r\n |
| Temperature | 2090 \r\n |

**Set Trigger Threshold**

Syntax: 3\r

This command allows the user to change the event trigger threshold. After choosing this function from the main menu, the user will be prompted to enter an integer value for the threshold. This value must be between 0 – 10240 or the input will be rejected and the user will be returned to the main menu.

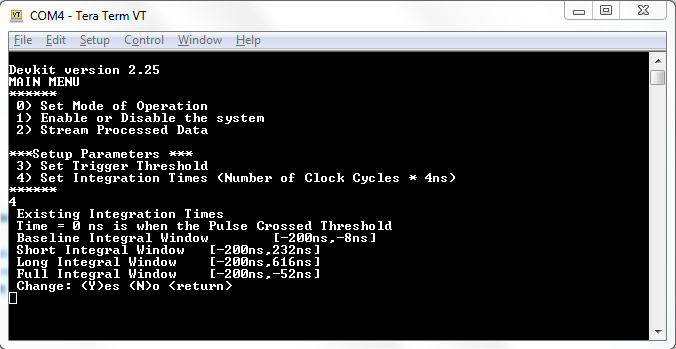
Return: *New Threshold = \_x\_\r\n* or *Invalid command\r\n*

In the above, *\_x\_* is the value that the system now has in memory. The system will echo back the new threshold set or invalid command if the value was not between 0 – 10240.

**Set Integration Times**

Syntax: 4\r

This command allows the user to check and/or change the four integration times. After entering this menu, the current integral windows will be printed to the screen, see Figure **2**, and the user will be prompted to enter either yes, to change the integration times, or no, to leave them unchanged.

**Figure 2: the Set Integration Times Menu**

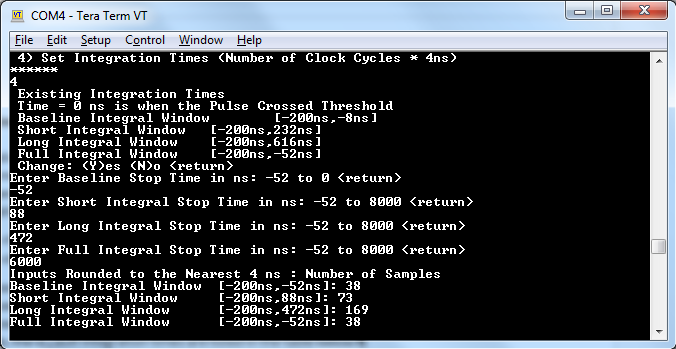
The syntax is listed below:

* *y\r* or *Y\r*
* *n\r* or *N\r*

If the user chooses to leave the integration times unchanged, they are directly returned to the main menu. If the user chooses to change the integration times, they will be prompted to enter new times, in nanoseconds, for each of the integrals. Each time must be greater than the previous one and each time must be within the values given. As an example, the default integration times are listed in the table below:

|  |  |  |
| --- | --- | --- |
| Integral | Input Syntax | Input Range |
| Baseline | -52\r | -52 – 0 |
| Short | 88\r | -52 - 8000 |
| Long | 472\r | -52 – 8000 |
| Full | 6000\r | -52 – 8000 |

After changing the integration times successfully, the system echoes back the new values, but they are in samples, rather than nanoseconds (each sample is 4 nanoseconds). The output is shown in Figure **3**.

**Figure 3: the Set Integration Times Echo Value**

Return:

* No changes were requested
  + No return value
* Integration times were updated
  + The number of samples are echoed back via five lines of text:
  + *\n\rInputs Rounded to the Nearest 4 ns : Number of Samples\n\r*
  + *Baseline Integral Window [-200ns,\_x1\_ns]: \_y1\_ \n\r*
  + *Short Integral Window [-200ns,\_x2\_ns]: \_y2\_ \n\r*
  + *Long Integral Window [-200ns,\_x3\_ns]: \_y3\_ \n\r*
  + *Full Integral Window [-200ns,\_x4\_ns]: \_y4\_ \n\r*
  + *\_x1\_* - *\_x4\_* are the values that the user entered in units of nanoseconds
  + *\_y1\_* - *\_y4\_* are the values entered in units of samples