Voyager3-1Z Version 0.5 Updates

Version 0.5

Created: 30/7/2021

Author: Jack McCarthy

Contents

Summary of New Features

- Configurable low power sleep mode after one frame of data is collected per axis to enable more realistic use cases.
- Control the sampling frequency of the ADCs, with 1Hz resolution.
- Control which axes Voyager collects data from, via the GUI.
- Control the number of samples that are collected per frame.
- Voyager's 1GB external flash chip has been utilised. Up to 1/3GB of data per axis can be sampled continuously, at the desired sampling frequency, with no time gaps.
    - Note, if the number of samples per frame exceeds 1024 then FFT is not calculated.
- Additional means of removing ADC offset. Assumes the mote is in the vertical position i.e., G in X and Y axes = 0. G in Z axis = 1.
- Terminal (non-GUI) mode.
- Handshaking between Voyager and Manager keeps the two in sync and stops Voyager collecting data that will not be observed.
- Fixed a scaling issue within the GUI which caused G values to be misrepresented.

High Level Operation

A summary of the high-level operation of Voyager 3 can be seen in Figure 1 and Figure 2. Note, the sequence of events shown in Figure 1 are collectively known as a Stage.

The state machine has been expanded to incorporate the new features. While the user now has more control over the use case that they would like to employ through the ability to vary the sampling frequency, the number of samples collected and for how long the microcontroller should sleep (if at all).

When a mote is connected to the manager via the GUI, Stages are repeated until the GUI is disconnected.

If the mote is connected to the manager through the Terminal Mode, the user passes in the argument NUM_STAGES to control how many times to repeat the sequence.

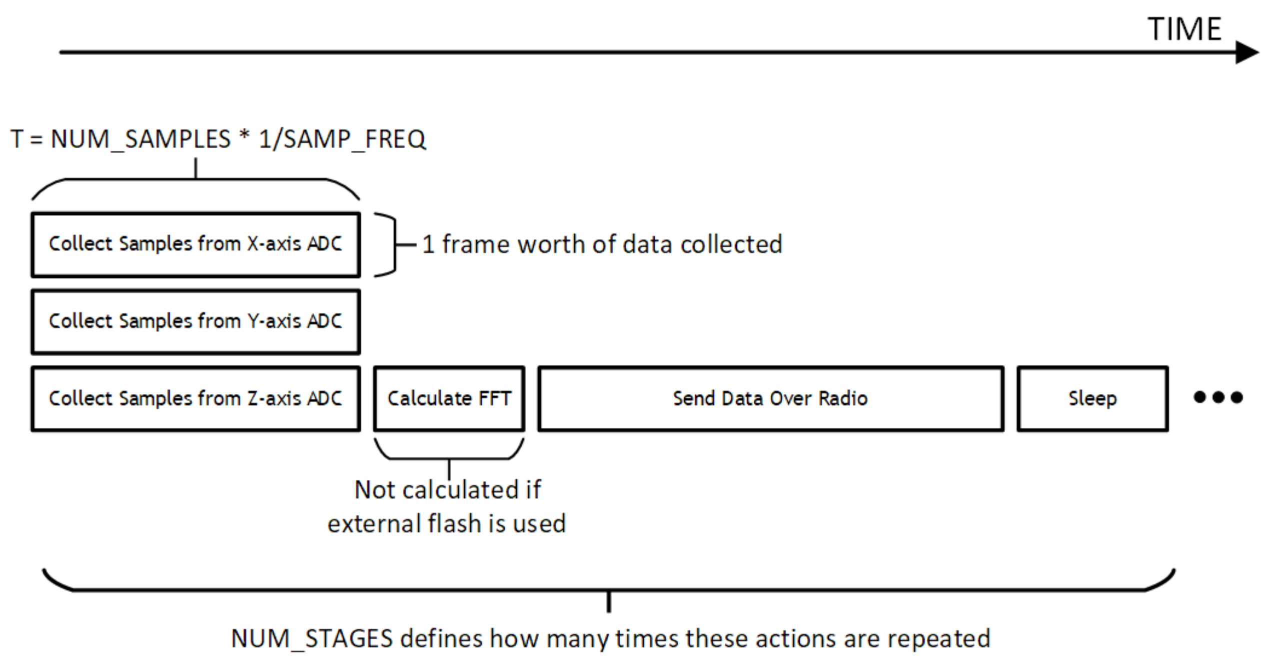

TIME
T = NUM_SAMPLES * 1/SAMP_FREQ
Collect Samples from X-axis ADC
1 frame worth of data collected
Collect Samples from Y-axis ADC
Collect Samples from Z-axis ADC
Calculate FFT
Send Data Over Radio
Sleep
Not calculated if external flash is used
NUM_STAGES defines how many times these actions are repeated

FIGURE 1: VOYAGER 3 TIMELINE OF EVENTS. THIS COLLECTION OF EVENTS IS CALLED A STAGE

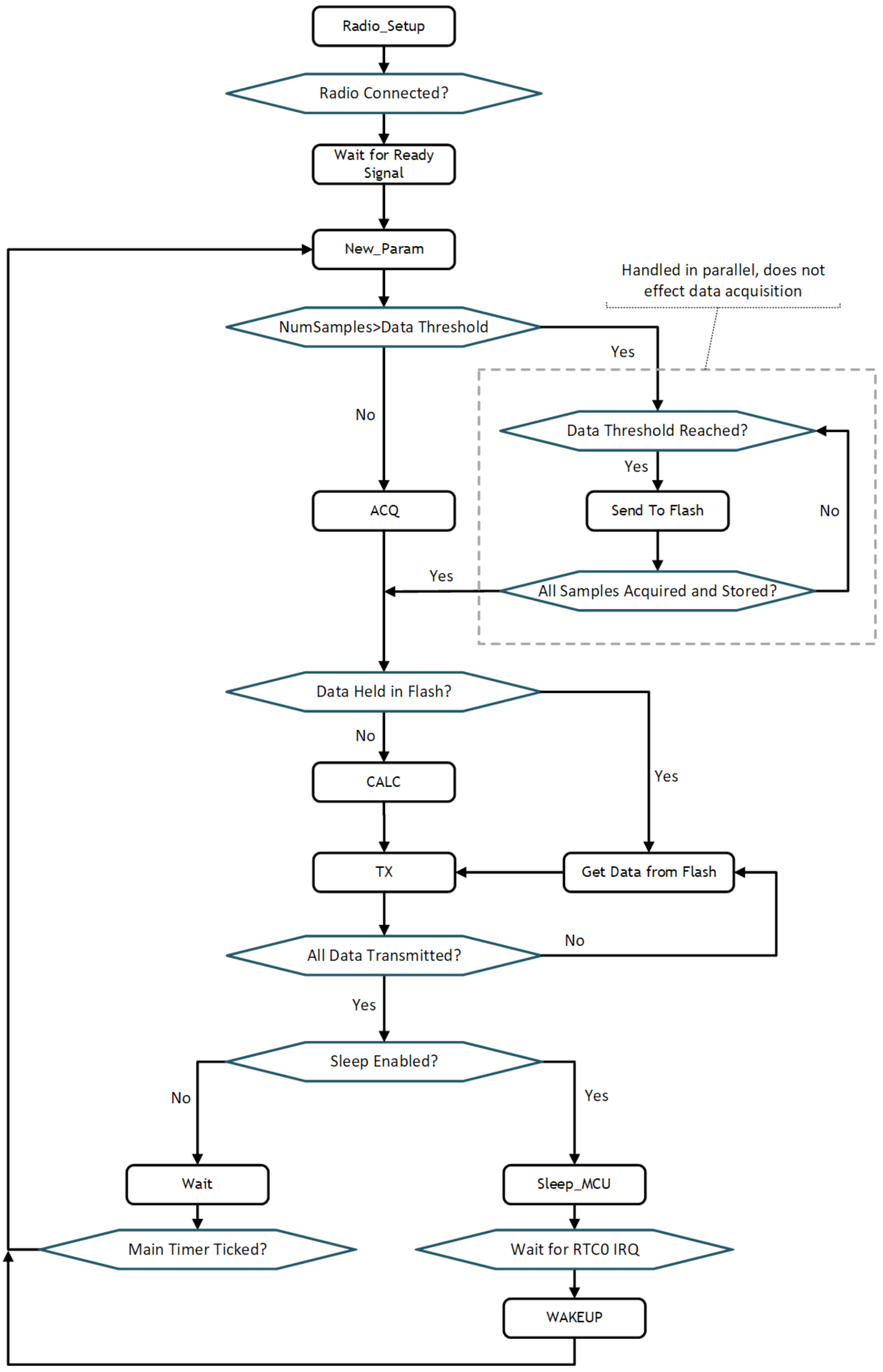


FIGURE 2: VOYAGER 3 VERSION 0.5 MAIN STATE MACHINE

Terminal Mode

NOTE: A prerequisite for Terminal Mode is that the user has followed the python setup outline in the “Python Setup” section in the “CBM_setup.pdf” file that can be found in the local github repository.

In version 0.5 a new feature has been added to the python manager software called Terminal Mode (aka non-GUI mode).

Terminal Mode allows the user to pass in parameters that control the mote when calling the python script, without the need to launch the GUI. This gives the user more control when compared to the predetermined values in the dropdown menus in the GUI. It also enables the user to define how Stages should be performed. When the defined number of Stages have been performed the terminal is disconnected from the manager and the collected data is saved to an excel file (and SQL database). Note, the mote will perform one extra stage before it realises the manager software has been disconnected.

The expected input to the terminal is as follows:

```
python CBM_app.py com<PORT_NUM> <SAMP_FREQ> <NUM_SAMPLES>
<NUM_STAGES> <EN_AXES_XYZ> <SLEEP_DURATION_S> <OFFSET_REMOVAL>
```

For example:

```
python CBM_app.py com9 500 1024 2 110 60 0
```

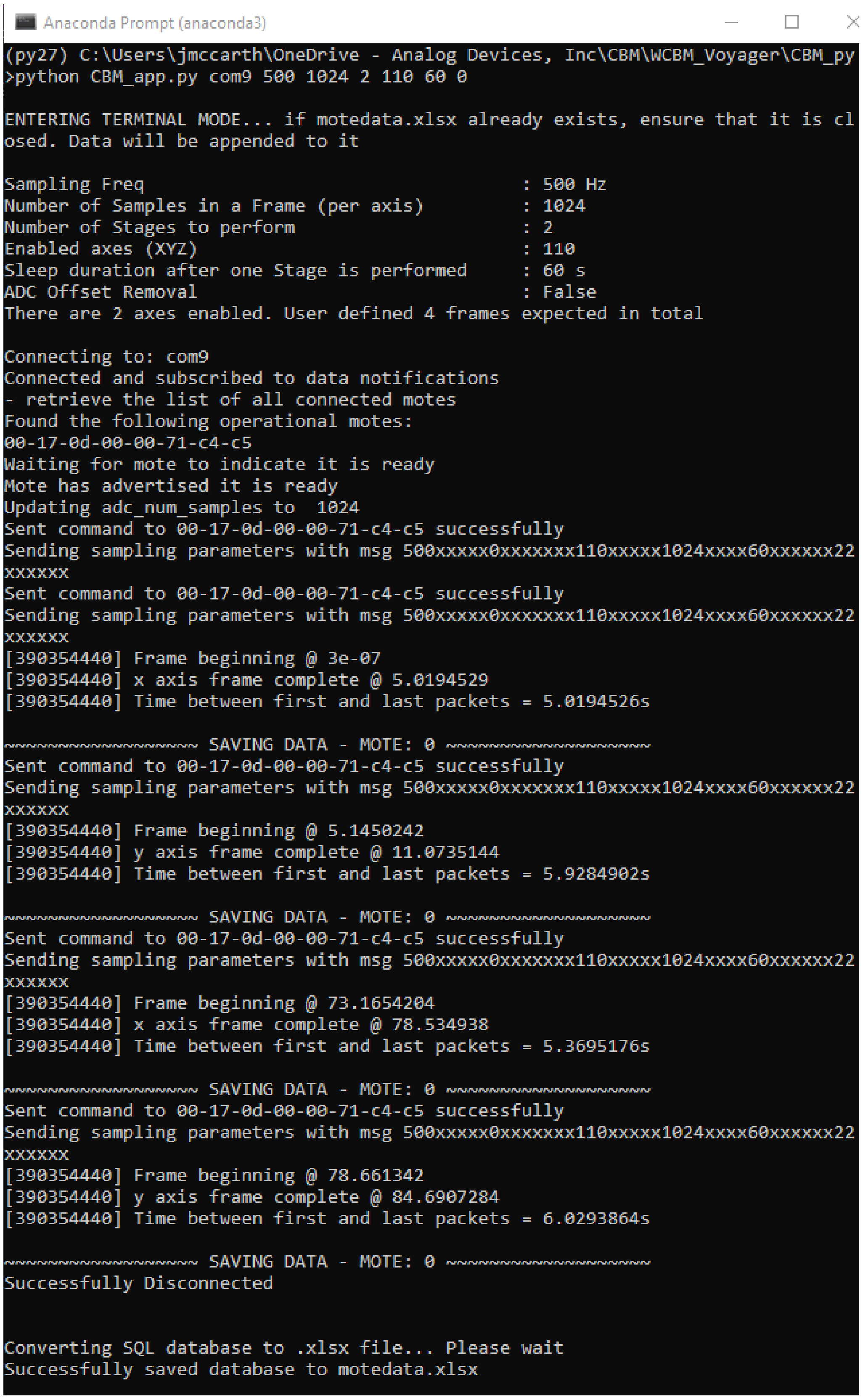
If the user inputs unacceptable parameters the script will detect this and advise the user on which input needs to be changed.

Figure 3 shows a screenshot of terminal mode in operation.

NOTE: Terminal Mode currently only works with a single mote connected.

NOTE: If an error is received that states the data could not be saved to the excel file try to enter terminal mode without using the python keyword in the command. E.g.:

```
CBM_app.py com9 500 1024 2 110 60 10
```



```
(py27) C:\Users\jmccarth\OneDrive - Analog Devices, Inc\CBM\WCBM_Voyager\CBM_py
>python CBM_app.py com9 500 1024 2 110 60 0

ENTERING TERMINAL MODE... if motedata.xlsx already exists, ensure that it is cl
osed. Data will be appended to it

Sampling Freq                                    : 500 Hz
Number of Samples in a Frame (per axis)          : 1024
Number of Stages to perform                      : 2
Enabled axes (XYZ)                               : 110
Sleep duration after one Stage is performed      : 60 s
ADC Offset Removal                               : False
There are 2 axes enabled. User defined 4 frames expected in total

Connecting to: com9
Connected and subscribed to data notifications
- retrieve the list of all connected motes
Found the following operational motes:
00-17-0d-00-00-71-c4-c5
Waiting for mote to indicate it is ready
Mote has advertised it is ready
Updating adc_num_samples to  1024
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 500xxxxx0xxxxxxx110xxxxx1024xxxx60xxxxxx22
xxxxxx
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 500xxxxx0xxxxxxx110xxxxx1024xxxx60xxxxxx22
xxxxxx
[390354440] Frame beginning @ 3e-07
[390354440] x axis frame complete @ 5.0194529
[390354440] Time between first and last packets = 5.0194526s

~~~~~~~~~~~~~~~~~~~ SAVING DATA - MOTE: 0 ~~~~~~~~~~~~~~~~~~~~
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 500xxxxx0xxxxxxx110xxxxx1024xxxx60xxxxxx22
xxxxxx
[390354440] Frame beginning @ 5.1450242
[390354440] y axis frame complete @ 11.0735144
[390354440] Time between first and last packets = 5.9284902s

~~~~~~~~~~~~~~~~~~~ SAVING DATA - MOTE: 0 ~~~~~~~~~~~~~~~~~~~~
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 500xxxxx0xxxxxxx110xxxxx1024xxxx60xxxxxx22
xxxxxx
[390354440] Frame beginning @ 73.1654204
[390354440] x axis frame complete @ 78.534938
[390354440] Time between first and last packets = 5.3695176s

~~~~~~~~~~~~~~~~~~~ SAVING DATA - MOTE: 0 ~~~~~~~~~~~~~~~~~~~~
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 500xxxxx0xxxxxxx110xxxxx1024xxxx60xxxxxx22
xxxxxx
[390354440] Frame beginning @ 78.661342
[390354440] y axis frame complete @ 84.6907284
[390354440] Time between first and last packets = 6.0293864s

~~~~~~~~~~~~~~~~~~~ SAVING DATA - MOTE: 0 ~~~~~~~~~~~~~~~~~~~~
Successfully Disconnected


Converting SQL database to .xlsx file... Please wait
Successfully saved database to motedata.xlsx
```

FIGURE 3: TERMINAL MODE EXAMPLE OUTPUT

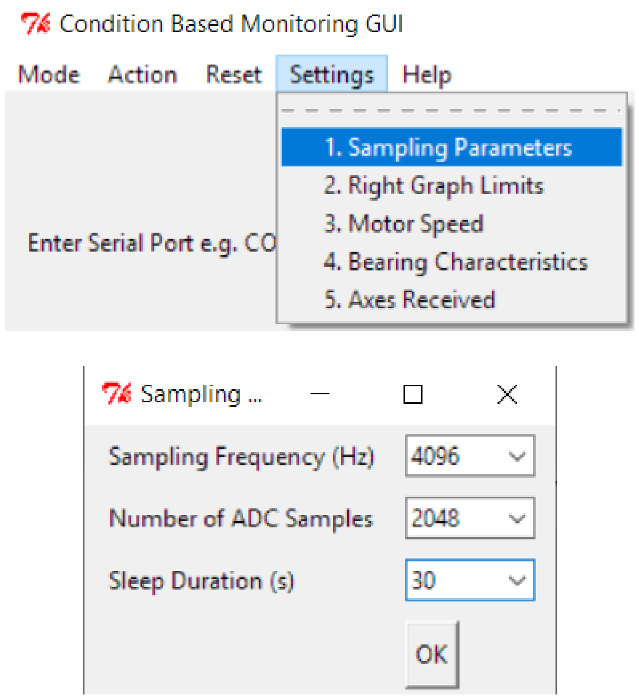
Sleep Mode

The user can optionally put the microcontroller into sleep mode after performing a Stage. The duration in sleep is programmed in units of seconds. After the defined number of seconds have elapsed the one of the microcontroller's real-time clocks (RTC0) is generates an interrupt to wake the core up and operation begins from the NEW_PARAM state. Power consumption by the

The radio remains operational during this sleep phase, but it will naturally be in a low power state and only sending messages intermediately to maintain the SmartMesh network.

Controlling Sampling Parameters From GUI

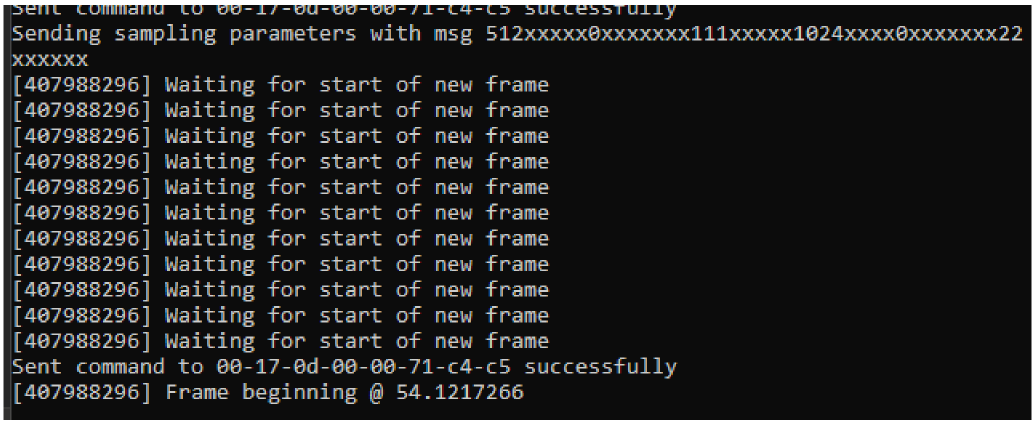
The Settings->Sampling Parameters dropdown menu open up another window from where the sampling parameters can be controlled. The three boxes must be filled in before the 'OK' button is clicked. Once the 'OK' button is clicked the manager will send a message to all connected motes. When the motes reach the NEW_PARAM state these new settings will be applied for the next stage



Sampling Frequency (Hz) 4096
Number of ADC Samples 2048
Sleep Duration (s) 30

FIGURE 4: CONTROLLING SAMPLING PARAMETERS FROM GUI

NOTE: When the number of samples to collect are changed the current Stage will need to finish before the Motes and manager are in sync again. Messages like the below will be seen in the meantime:



```
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
Sending sampling parameters with msg 512xxxxx0xxxxxxx111xxxxx1024xxxx0xxxxxxx22
xxxxxx
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
[407988296] Waiting for start of new frame
Sent command to 00-17-0d-00-00-71-c4-c5 successfully
[407988296] Frame beginning @ 54.1217266
```

External Flash

Voyager has a 1GB flash memory device connected to the microcontroller via SPI2. This firmware update makes use of this memory which allows up to 1GB to be collected in a single Stage, without interruption to the selected sampling frequency.

If the user selects that more than 1024 samples should be collected per frame, then the firmware automatically stores and retrieves data to/from flash.

NOTE: If external flash is used then FFT will not be calculated.

The flash chip is divided into 64 pages. Each page contains 1024 blocks. Each block contains 2048B of data and 128B of error correction codes which are automatically enabled.

Each axis has 1/3 of the flash memory dedicated to it, as shown in Figure 5:

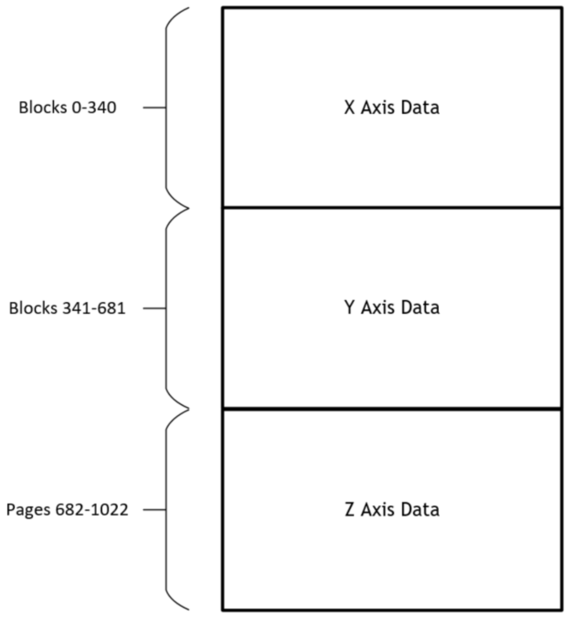


FIGURE 5: FLASH DIVISION

ADC Offset Removal

There are now two options when removing the ADC offset. They are:

1. This method assumes that the mote is stationary and in the upright position, i.e. G in X and Y axes = 0 and G in Z axis = 1. The distance from these ideal values shall be calculated during one frame and subtracted, for each respective axis, from all subsequent frames. To use this method in Terminal mode pass "1" in for the OFFSET_REMOVAL argument.

2. The second method calculates the average value of each axis in a frame and subtracts that value, for each respective axis, from all subsequent frames. This must be performed when the mote is stationary and was already in place in previous versions of the software. To use this method in Terminal mode pass "2" in for the OFFSET_REMOVAL argument.

NOTE: If ADC offset removal is invoked in terminal mode then one extra stage will be performed at the start. During this stage the offsets will be calculated. The user defined number of stages shall then be performed with the offsets removed from the data.