1) Navigate to Zcu_102_varun.
2) Open Vivado. I was running it on the 2019.1 version.
3) Changes to the block design can be done accordingly in the Vivado folder. In case the plain text and key data needs to be changed, that can be done accordingly in the trigger.v file.
4) Generate the bitstream and export the hardware file by including the bitstream with it.
5) Launch SDK.
6) The range of data points can be adjusted in the for loop but needs to be changed in 2 different files. One is sysmon.c in the sysmon_ported_zcu102 folder which sends messages via UART and the other one is the helloworld.c.
7) Go to Run configurations -> Run as debugger-> Create a new one -> Select Hardware Platform 0 -> Select the bitstream -> Go to application and check the boxes for psu_cortexa53_1 and psu_cortexa53_2. You will be able to see that the sysmon_ported_zcu102 and ZCU102_trigger projects and their corresponding elf files are selected. You can select the FPGA option along with it. Select program FPGA on Target Setup Tab.
8) Click on RUN.
9) Navigate to src/data_retrieval which is outside the vivado designs folder. Run measure.py($python3 measure.py) corresponding text files will be generated. This will take time depending upon the number of iterations but you can see the data is being sent through UART. The UART port is set to /dev/ttyUSB1 at 115200 baud. You can change accordingly if needed. Incase baud rate is being changed. Change it in the sysmon.c too.
10) You can also get the plots. Run plot.py. Edit the range in the code accordingly. This will plot the data accordingly.

Points to note:
1) Sometimes UART doesn't work. Plugging out the UART cables switching off the FPGA and connecting it back will do the job.
2) Don't have any print statements(in any of the application which are simultaneously running) other than the asiprintf statements in sysmon.c. This will result in corrupting the data being sent through UART.
3) Check linker script in the application projects. The ddr_0_MEM0 address shouldn't overlap.