

ASU ID: 1213366971
SUSHANT TRIVEDI
Team Name: 20

PROGRAM RUNNING

1. First the Build_code.sh present in the folder needs to be edited. Please edit the SD_CARD, PROGRAM_LOCATION and ZEPHYR_BASE locations. The -Wall flag has been added in this script.
 2. Next paste the “measure_20” folder at the appropriate location /samples/measure_n
 3. Use the shell file to configure the SD Card with Zephyr files
 - /boot
 - /efi
 - /kernel
 4. Once this is done, Source code need to be configured to run with comments or without also we need to choose how many samples need to be collected for measurements. This has been set on the line 11 & 12. Please refer below for further details
 5. Now run the build_code.sh shell or the set of instructions (you have) required to compile Zephyr code on the SD Card.
 6. Place the SD Card in the slot of Galileo board and power on.
 7. When select Zephyr kernel it will run the code for as many iterations defined and store the data in Time[][] variable.
 8. Press enter key once the samples have finished collecting to show the shell
 9. Here use Shell module (Results) to select which set of data to display from the following option set.
 1. Interrupt Latency Without Background Task
 2. Interrupt Latency With Background Task
 3. Context Switching Overhead
- This can be done with the help of Following commands built into shell
1. help
 2. select.
9. It will display the dataset for that function. Each row will have 15 measurements printed out for the sake of aesthetics.
 10. Please note that IO6 → Output (Configured as PWM or GPIO as required) and IO5 → Input (Interrupts triggered here on rising edges – Active High)
 - 11.

UNDERSTANDING THE CODE

1. The code has been made scalable so that you can run each functionality for as many times as you want. This has been defined at the start #define LIMIT macros
2. Also the functionality to run the code with no comments vs with comments has been provided.
Comments=1 → Will display comments
Comments=0 → No comments displayed
3. The required functionalities has been implemented as follows

measure_context_switching_overhead() → Context Switching measurements
calculate_interrupt_latency() → Interrupt Latency Measurements

gpio_callfunc() → Callback attached to Input for interrupts. Triggered when Input is received
(Rising Edge)
threadsend() → Used for message queue sending data, Message queue threads are cooperative
threadreceive() → Used for message queue receiving data, Message queue threads are cooperative

FILES ENCLOSED

- | | |
|----------------------------|--|
| 1. HW-3 Report.pdf | → REPORT |
| 2. Readme.doc | → README |
| 3. Histogram-1.xlsx | → DATA SET COLLECTS AND GRAPH PLOTTED |
| 4. CS.txt | → DATA VALUES COLLECTED AND FILTERED USING LOG & SHELL |
| 5. IL - No Computing.txt | → DATA VALUES COLLECTED AND FILTERED USING LOG & SHELL |
| 6. IL - With Computing.txt | → DATA VALUES COLLECTED AND FILTERED USING LOG & SHELL |