Measurement in Zephyr RTOS

Assignment 3: Report

Submitted for the Subject

Real Time Embedded Systems

CSE 522

By:

Nisarg Trivedi (1213314867)

Nagarjun Chinnari (1213287788)

Under the guidance of:

Prof. Yann-Hang Lee

Shiksha Patel
Shivani Oza

Arizona State University

Tempe Campus, AZ

March, 2018

ACKNOWLEDGEMENT

We owe a great thanks to many people who helped and supported us during the completion of this project.

Our deepest thanks go to **Prof. Yann-Hang Lee**, Professor for the subject Real Time Embedded Systems who acted as our mentor and guided us throughout this project. We thank him for giving ideas and briefs for the work that had to be done.

A special thanks to **Shivani Oza** and **Shiksha Patel**, for constant help and guidance in the labs.

We also thank all our friends for helping us out during the downs. We respect everyone for giving their time when we were in need.

Thank You.

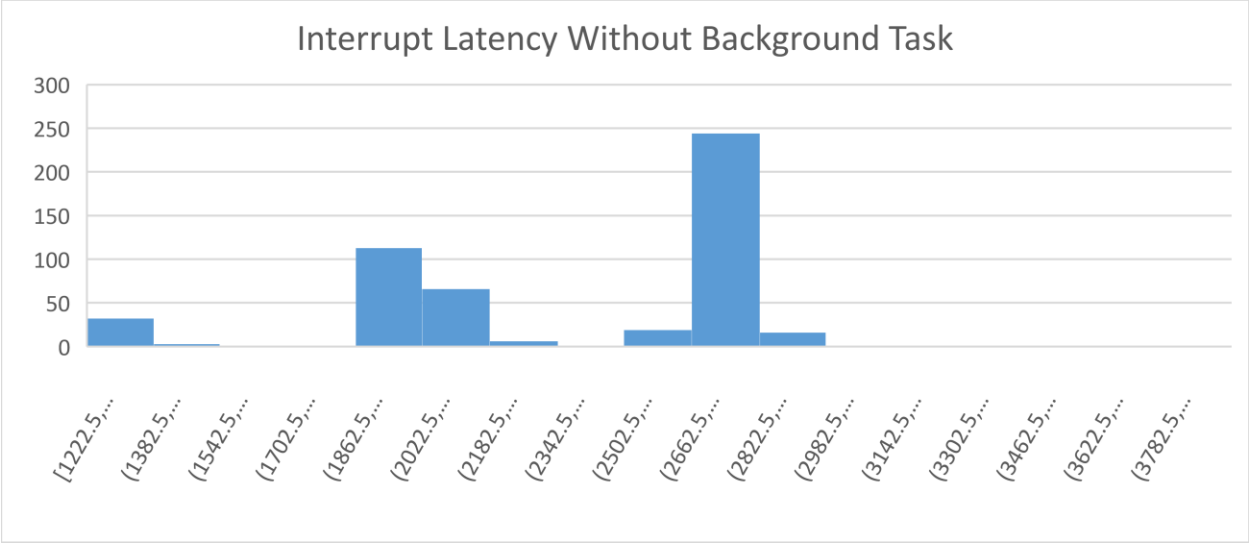
This project is about measuring the interrupt latency and the context switching overhead for Zephyr OS. Interrupt latency is the total delay between the interrupt signal being asserted and the start of the interrupt service routine execution. Context switching overhead is the delay of context switching process of saving the context of the executing thread, restoring the context of the new thread, and starting the execution of the new thread. Here we measure the interrupt latency without a background task and once with the background task. The difference is observed and shown in the form of a histogram. Histogram of Interrupt Latency without background task Histogram of Interrupt Latency with background task Context Switching Overhead.

We used the GPIO pin configuration table to read the data on the Galileo pins. The pinmux.c file was used to configure the multiplexing pins and producing pwm signals.

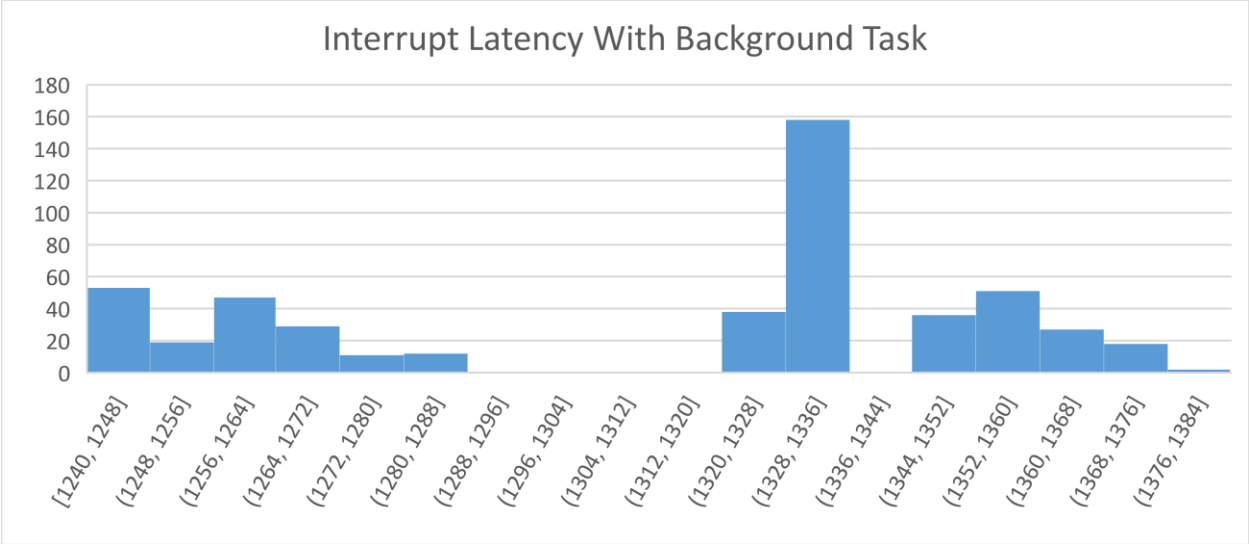
3 histograms have been used to show the different tasks that have been done in the code:

- The first two graphs are about the interrupt latency without a background task and interrupt latency with a background task. When we are computing without a background task, a time stamp is continuously taken and a difference is measured in the call back function to measure the latency.
- The second graph is about the interrupt latency with a background task running. For this, PWM is providing a continuous interrupt in the background and a time stamp is used to take the ticks reading before each instruction. When the interrupt is called, the threads are pre-empted and the value of the tine stamp is stored. The difference of these time stamps is used to calculate the latency (with background task).
- The third graph is about the context switch overhead. Here two threads have been used having different priorities. The thread with a higher priority tries to take a mutex but gets blocked as the thread with lower priority has locked it. The time stamp is used to take the time count when the context switch happens. When the thread with lower priority releases the mutex, the higher priority thread starts running and the time stamp can again be taken. The difference between these time stamp gives us the context switch overhead for the Zephyr OS.

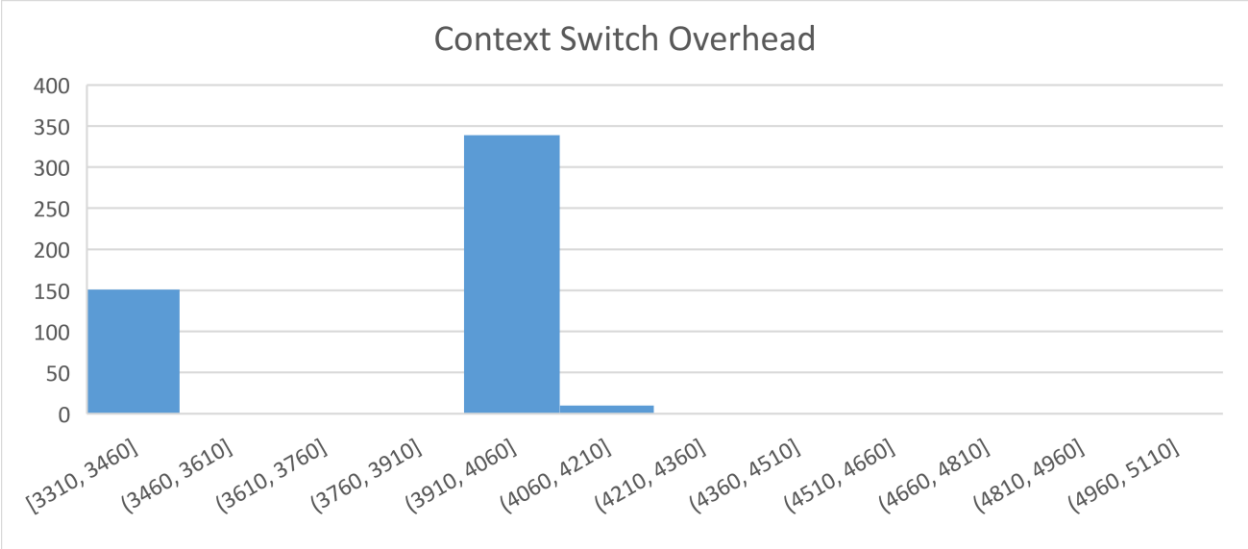
Histogram 1



Histogram 2



Histogram 3


Context Switch Overhead
400
350
300
250
200
150
100
50
0
[3310, 3460]
(3460, 3610]
(3610, 3760]
(3760, 3910]
(3910, 4060]
(4060, 4210]
(4210, 4360]
(4360, 4510]
(4510, 4660]
(4660, 4810]
(4810, 4960]
(4960, 5110]