Implementing EDF Scheduler

Verifying the system implementation:

Using analytic methods:

Calculate the system hyperperiod:

H = LCM(Pi) = LCM (50,50,100,20,10,100) = 100 ms

Execution time of the tasks:

Task Name	Execution time	Frequency
Button_1_monitor	24us	2
Button_2_monitor	24us	2
Periodic_Transmitter	100us	1
Uart_Receiver	24us	5
Load_1_Simulation	5ms	10
Load_2_Simulation	12ms	1

- ○

Calculate the CPU load:

CPU_Load = (total busy time in hyperperiod / hyperperiod)*100 = (24us *2 + 24us*2 + 110us + 24us *5 + 5ms *10 + 12ms) = (62.326ms / 100ms)*100 = 62.326%

Check system schedulability:

Using Rate Monotonic utilization bound:

- If $U \leq n(2^{1/n} -1)$ then system is schedulable
- $U = 62\%$
- $Urm = 6*(2^{1/6} -1) = 0.735$
- System is schedulable.

Time demand analysis:

$$w_i(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil e_k \quad for \ 0 < t \leq p_i$$

- Load_1_Simulation: (P = 10ms, E = 5ms, D = 10ms)
    - W(10) = 5ms + 0 = 5ms < D
    - Load_1_Simulation is schedulable.

- UART_Receiver: (P = 20ms, E = 24us , D = 20ms)
    - W(20) = 24us + (20/10)*5ms = 10.024ms < 20ms
    - UART_Receiver is schedulable
- Button_1_Monitor: (P =50ms , E = 24us ,D =50ms)
    - W(50) = 24us + (50/10)*5ms + (50/20)*24us = 25.096ms < D
    - Button_1_Monitor is schedulable.
- Button_2_Monitor: (P =50ms , E = 24us ,D =50ms)
    - W(50) = 24us + (50/10)*5ms + (50/20)*24us + (50/50)*24us = 25.12ms < D
    - Button_2_Monitor is schedulable.
- Periodic_Transmitter: (P= 100ms , E = 100us , D =100ms)
    - W(100) = 100us + (100/10)*5ms + (100/20)*24us + (100/50)*24us + (100/50)*24us = 50.316ms < D
    - Periodic_Transmitter is schedulable.
- Load_2_Simulation: (P= 100ms , E = 12ms , D =100ms)
    - W(100) = 12ms + (100/10)*5ms + (100/20)*24us + (100/50)*24us + (100/50)*24us + (100/100)*100us = 62.31ms < D
    - Load_2_Simulation is schedulable.

Simulation of tasks on simso:
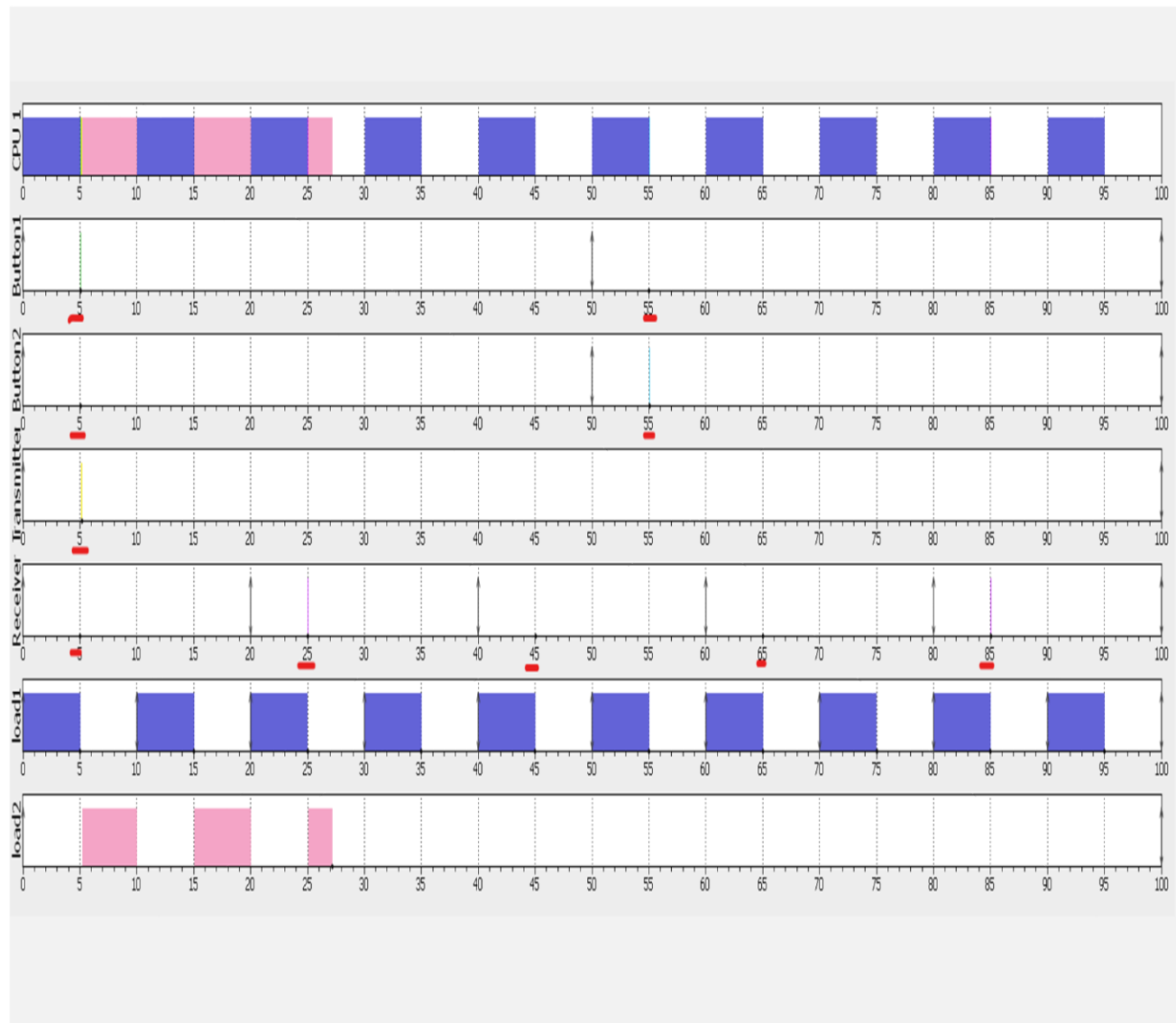
Keil simulation for tasks and CPU load: