Design document

project title: Design a Healthcare system using RTOS

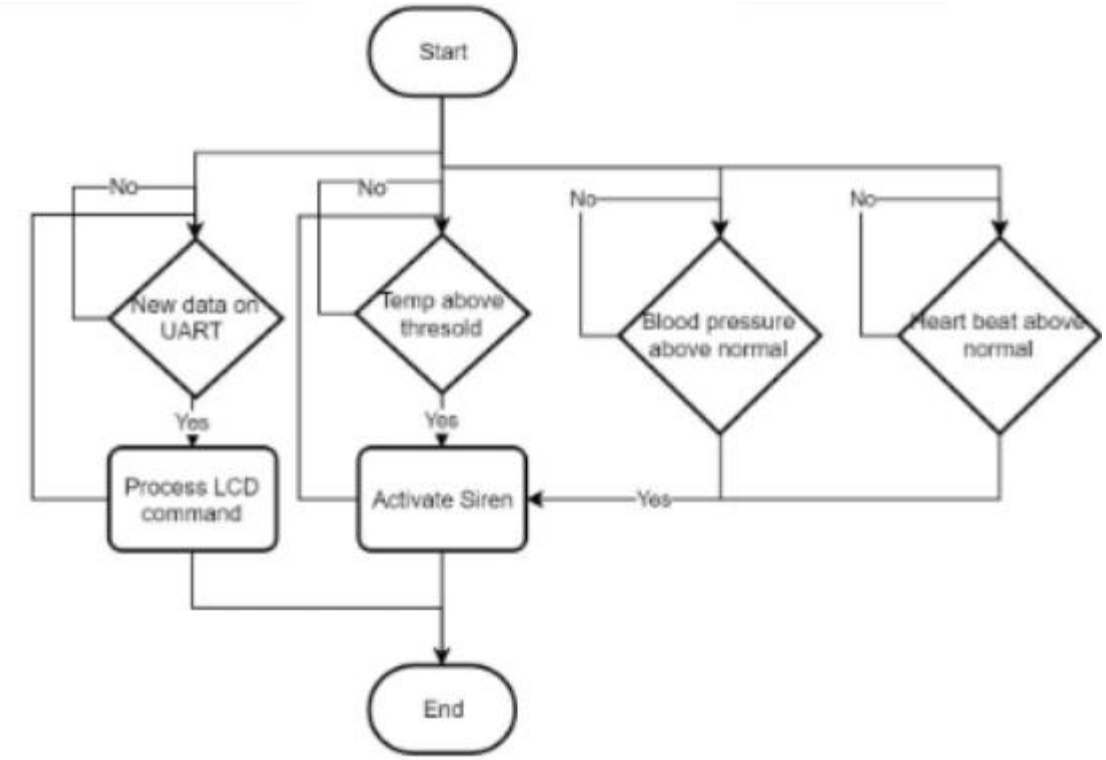
Represented by: Hazem Ashraf

Team 3

• Project Description

Design a healthcare system using RTOS with the following requirements

- A touch LCD as input that can control the system and give commands. Every command is represented in 4 bytes. LCD is connected to the microcontroller through USRT with speed 9600 bps [Reading 4 bytes and processing the command takes 2ms]
- Blood pressure sensor with new data every 25ms [Reading the sensor and processing its data takes 3ms]
- Heart beat detector with new data every 100ms [Reading the sensor and processing its data takes 1.5ms]
- Temperature sensor with new data every 10ms [Reading the sensor and processing its data takes 2.5ms]
- Alert siren [Activate or Deactivate the siren takes 1ms]



• Project Requirements

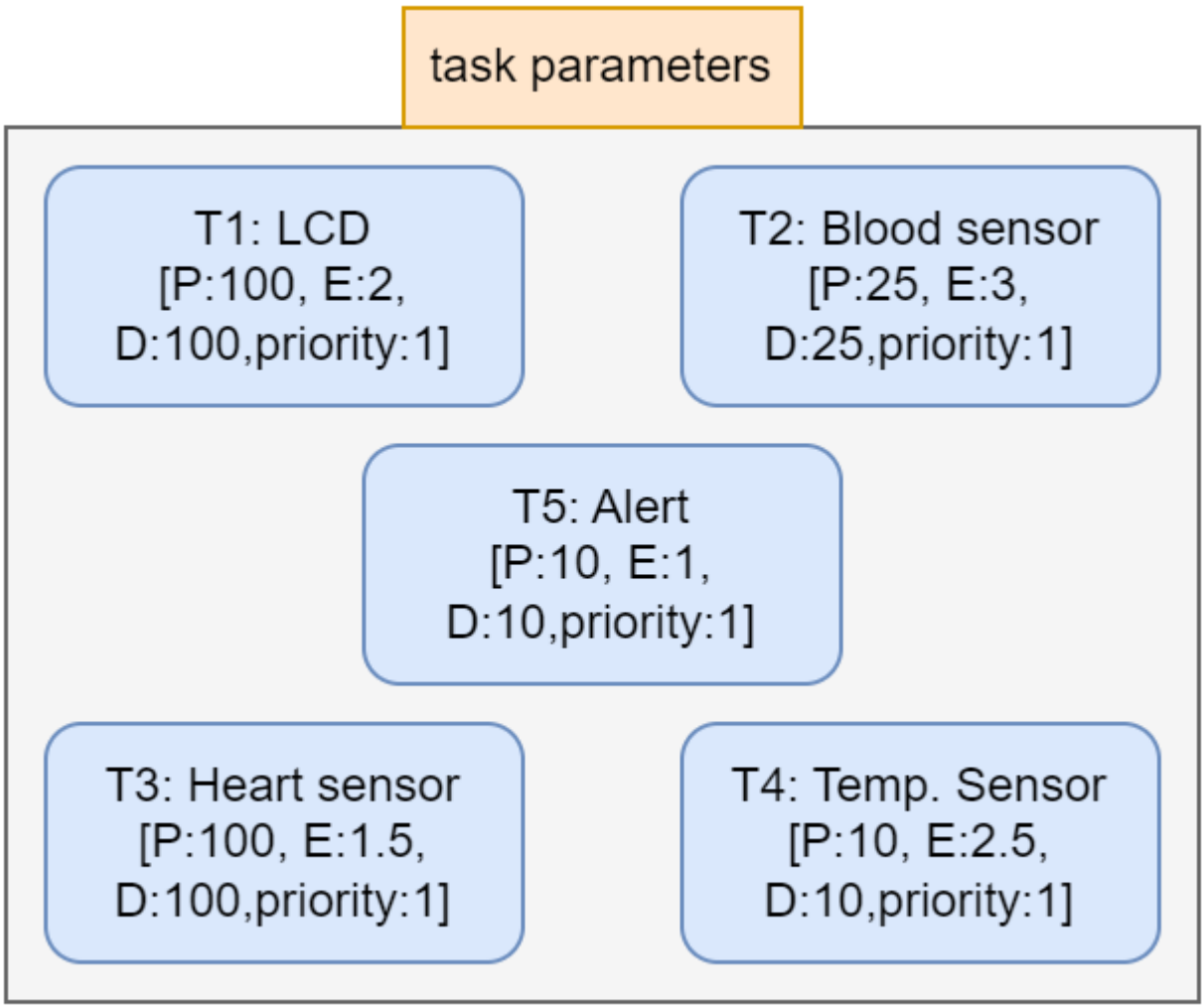
- Decide How many tasks are needed
- Decide the task parameters (priority – periodicity – Deadline)
- Decide the system tick rate
- Calculate Hyperperiod
- Calculate CPU Load
- Draw the timeline manually and analyze system schedulability
- Model the system in Simso and verify that your design is schedulable

- Decide how many tasks are needed

From system perspective we can divide the tasks to 5 tasks

1. Task to get input from LCD and send command to controller, Let call it T1
2. Task to get read from Blood pressure sensor, Let call it T2
3. Task to get read from Heart beat detector, Let call it T3
4. Task to get read from Temperature sensor, Let call it T4
5. Task to set Alert siren, Let call it T5

- Decide the task parameters (priority – periodicity – Deadline)



System Design

- Decide the task parameters (priority – periodicity – Deadline)

- Regarding Tasks Deadline are assumed to be equal to the task periodicity (P=D).
- Regarding LCD Task periodicity. Its given that LCD is connected to the microcontroller through USRT with speed 9600 bps and Every command is represented in 4 bytes, then every 3.3ms LCD send the command But this period is too fast in comparison to the time to get input from human So to be realistic we select the task periodicity = 100ms.
- We assume the Alert siren task to be periodic task and its periodicity = 10ms.so it could work with any of the three tasks and respect their periodicity.

- Decide the system tick rate

Systick can be decided using the following simple general rule

Systick value > Total execution time of all tasks

Task name	periodicity ms	Execution time ms	Deadline ms
T1: LCD	100	2	100
T2: Blood sensor	25	3	25
T3: Heart sensor	100	1.5	100
T4: Temp sensor	10	2.5	10
T5: Alert	10	1	10
Total execution time		10	

Systick rate = **12ms**

- Calculate Hyperperiod

To calculate Hyperperiod = LCM(all tasks periodicity)

= LCM(100,25,10) = 100

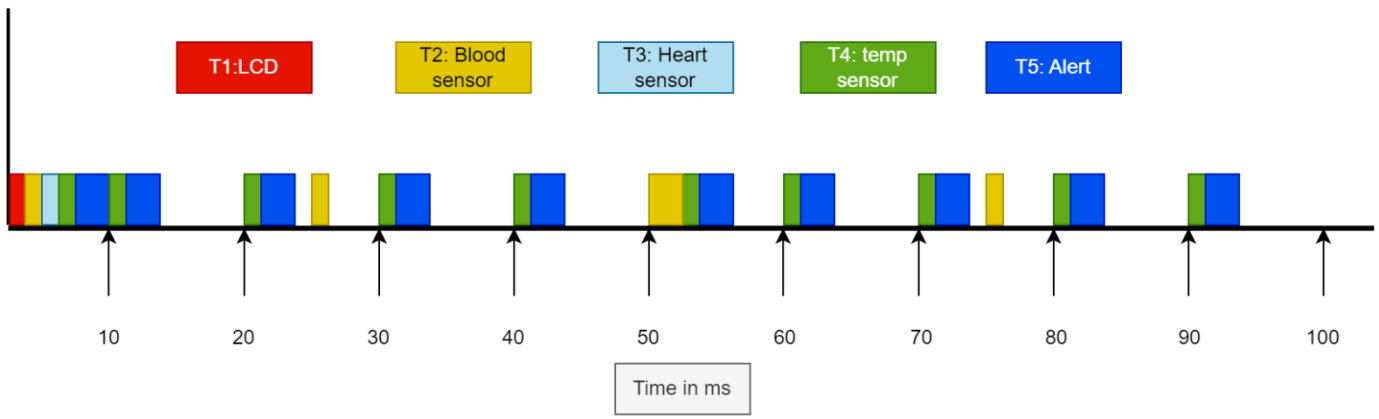
- Calculate CPU Load

To calculate CPU Load, simply calculate Utilization = R/C
R: Requirements which in simple terms is the BUSY TIME
C: Capacity which in simple terms is BUSY TIME + IDLE TIME
BUSY TIME = Execution time * (Hyperperiod / periodicity)

Task Name	periodicity ms	Execution time ms	BUSY TIME ms
T1: LCD	100	2	2x1 = 2
T2: Blood sensor	25	3	3x4=12
T3: Heart sensor	100	1.5	1.5x1=1.5
T4: Temp sensor	10	2.5	2.5x10=25
T5: Alert	10	1	1x10=10
TOTAL BUSY TIME			50.5

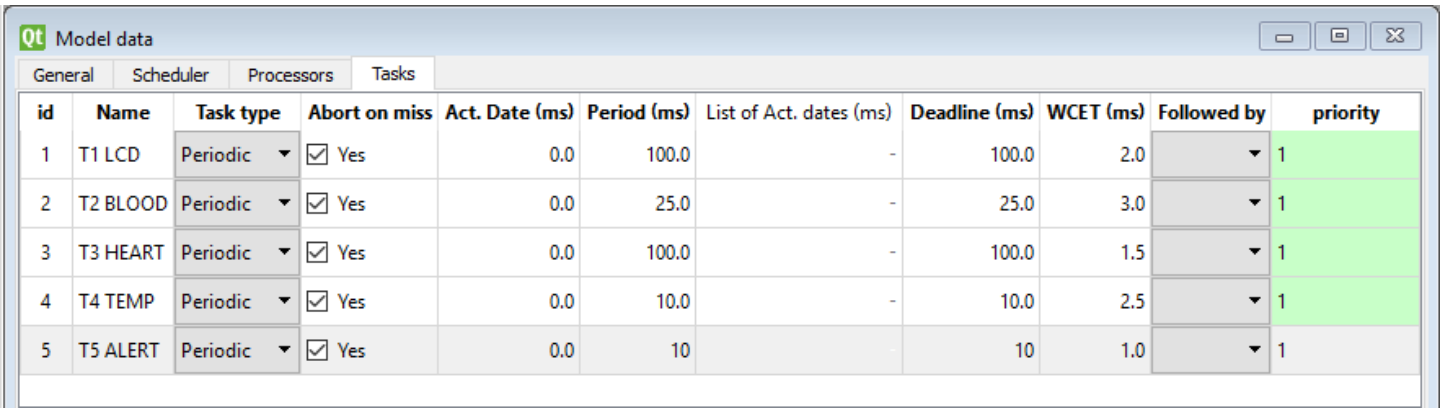
CPU Load = **(50.5/100) * 100 = 50.5 %**

- the timeline and system schedulability



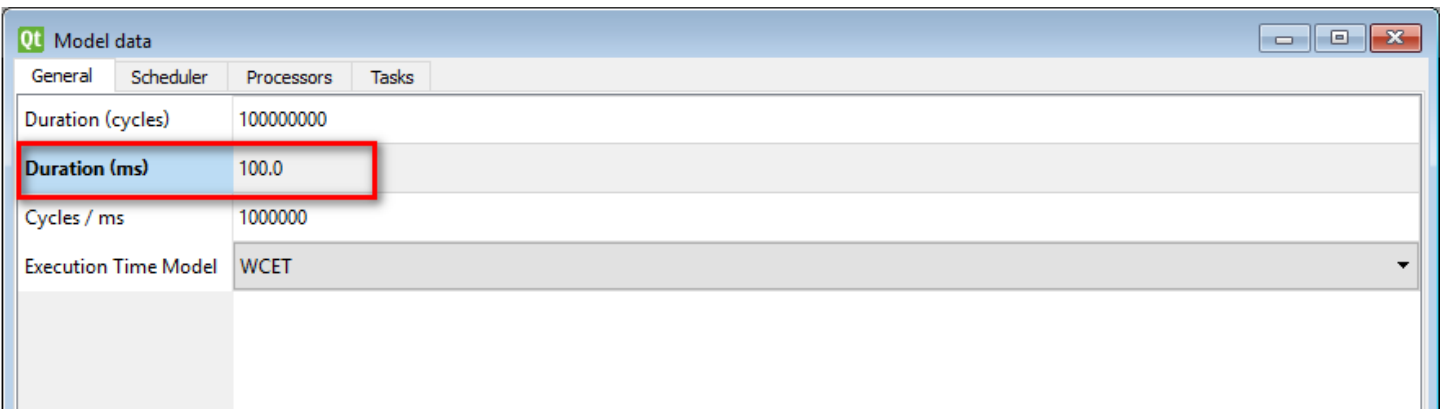
Model the system in Simso

- Tasks setup

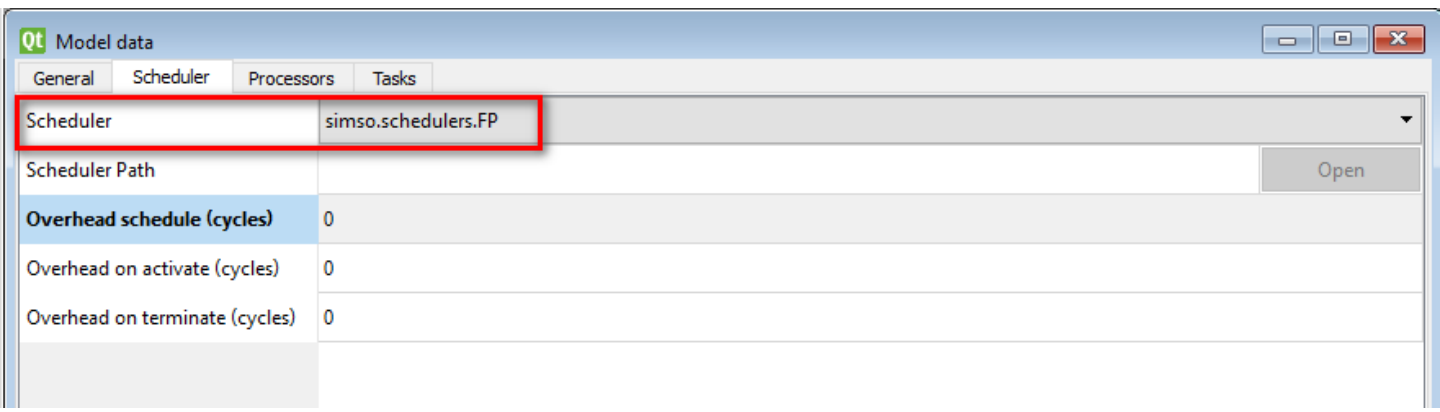


id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
1	T1 LCD	Periodic	Yes	0.0	100.0	-	100.0	2.0		1
2	T2 BLOOD	Periodic	Yes	0.0	25.0	-	25.0	3.0		1
3	T3 HEART	Periodic	Yes	0.0	100.0	-	100.0	1.5		1
4	T4 TEMP	Periodic	Yes	0.0	10.0	-	10.0	2.5		1
5	T5 ALERT	Periodic	Yes	0.0	10	-	10	1.0		1

- Model setup: set the Hyperperiod

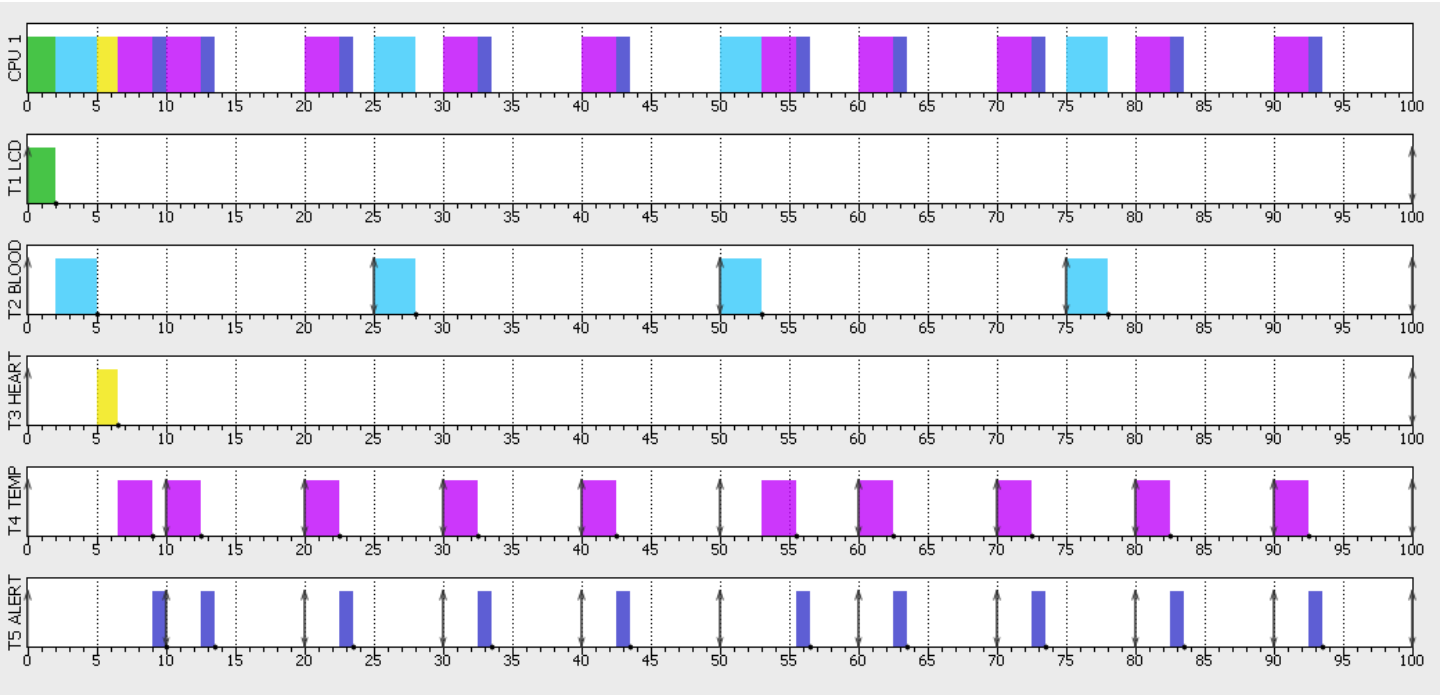


- Scheduler setup: set the scheduler type to fixed priority type

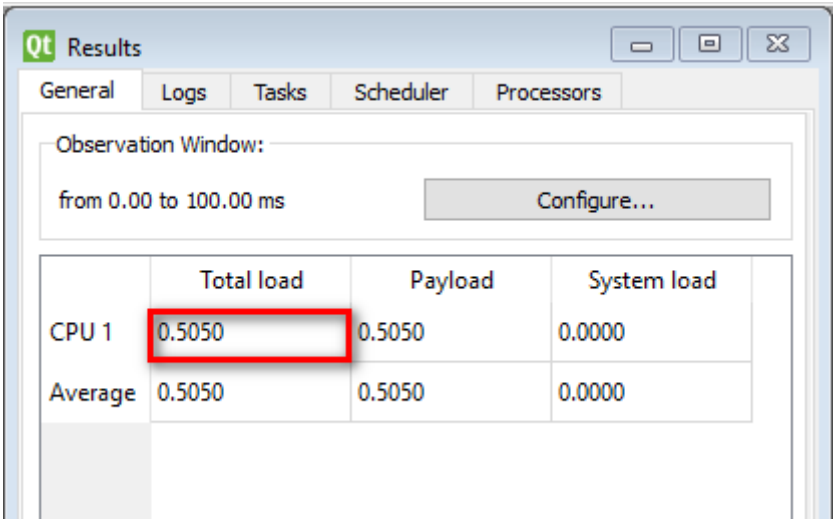


Results

- Task Timeline



- Timeline statistics and CPU Load



- Conclusion

From the results, the task set according to the design parameters is schedulable as all tasks meet their deadline and execute as expected and as their periodicity