Batch: A2	Roll No.:16010322014
Experiment / assignment / tutorial No. 8	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: FreeRTOS debugging using SEGGER System View tools
AIM: 1. Set up System SEGGER tool in IDE
2 Analyse Runtime behaviour of Embedded System

OUTCOME: Understand the concepts of co-design and co-simulation and testing methodology for embedded system
Explore advances and recent trends in embedded systems .

Overview on SEGGER System View:

SystemView is a toolkit for visual analysis of any embedded system. System View gives complete insight into an application, to gain a deep understanding of the runtime behaviour, going far beyond what a debugger is offering.

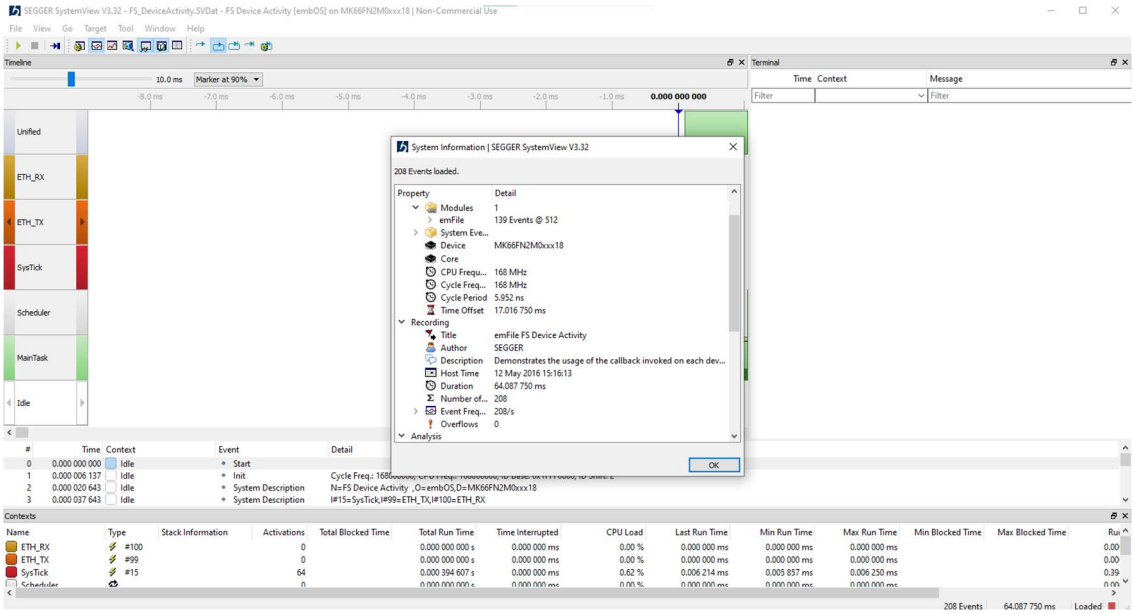
SystemView consists of two parts:
- The PC visualization SystemView Application,
- Code that gathers telemetry data on the target system

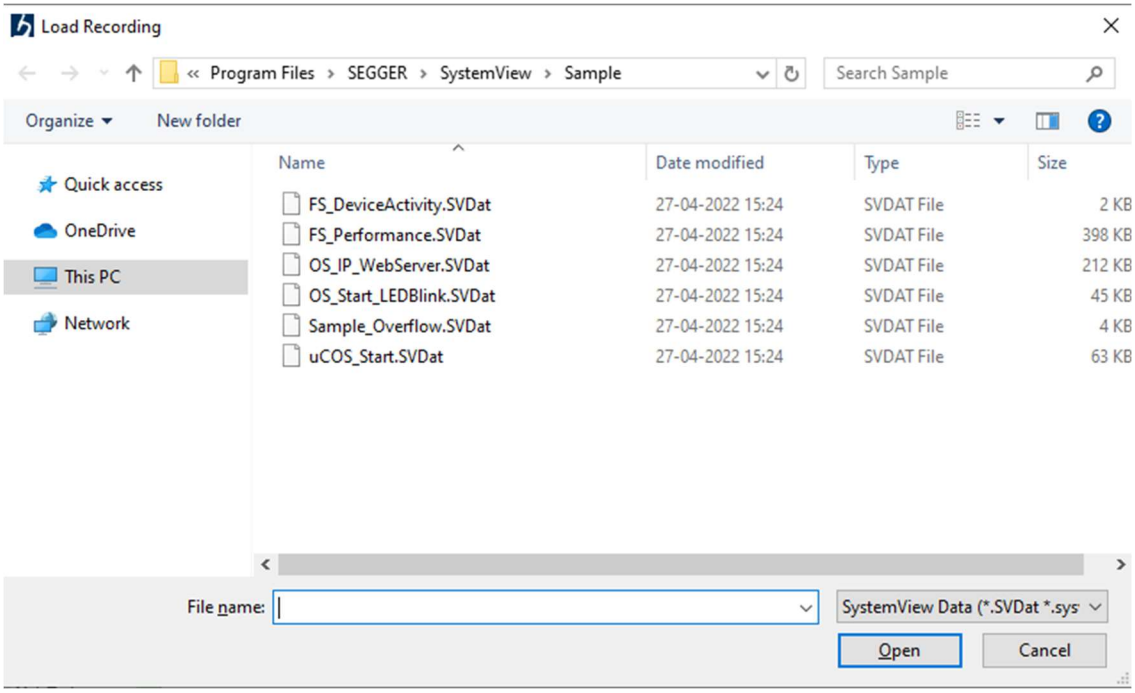
The SystemView application allows analysis and profiling of the behaviour of an embedded system. It records the telemetry data generated by the embedded system and visualizes that information in a variety of ways. The recording can be saved to a file for later analysis or for documentation of the system. The telemetry data is recorded through the debug interface, through a network connection, or over a serial line. SystemView can be used to verify that the embedded system behaves as expected and can be used to find problems and inefficiencies, such as superfluous and spurious interrupts, unexpected task changes, or badly-chosen task priorities. It can be used with any (RT)OS which is instrumented to call SystemView event functions, but also in systems without an instrumented RTOS or without any RTOS at all, to analyze interrupt execution and to time user functionality like time-critical subroutines.

Working:
On the target side a small software module, containing SYSTEMVIEW and RTT, must be included. The SYSTEMVIEW module collects and formats the monitor data and passes it to RTT. The target system calls SYSTEMVIEW functions in certain situations, such as interrupt start and interrupt end, to monitor events. SystemView stores these events together with a configurable, high-accuracy timestamp. Timestamps can be as accurate as 1 CPU cycle, which equates to 5 ns on a 200 MHz CPU. The RTT module stores the data in the target buffer, which enables continuous recording, as well as single-shot recording and post-mortem analysis. The recorder interface reads the data from the RTT buffer and sends it to the SystemView Application

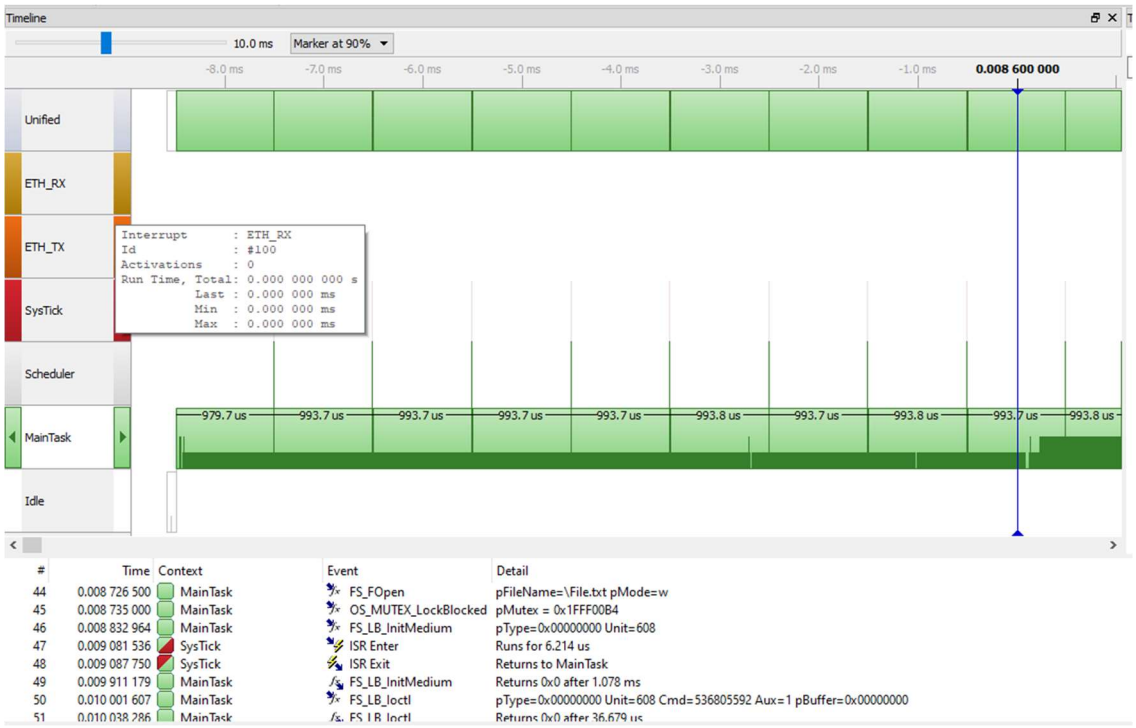
Observations: - (Students are expected to write /screen shots of results obtained.)

1) **Starting System View and loading data**

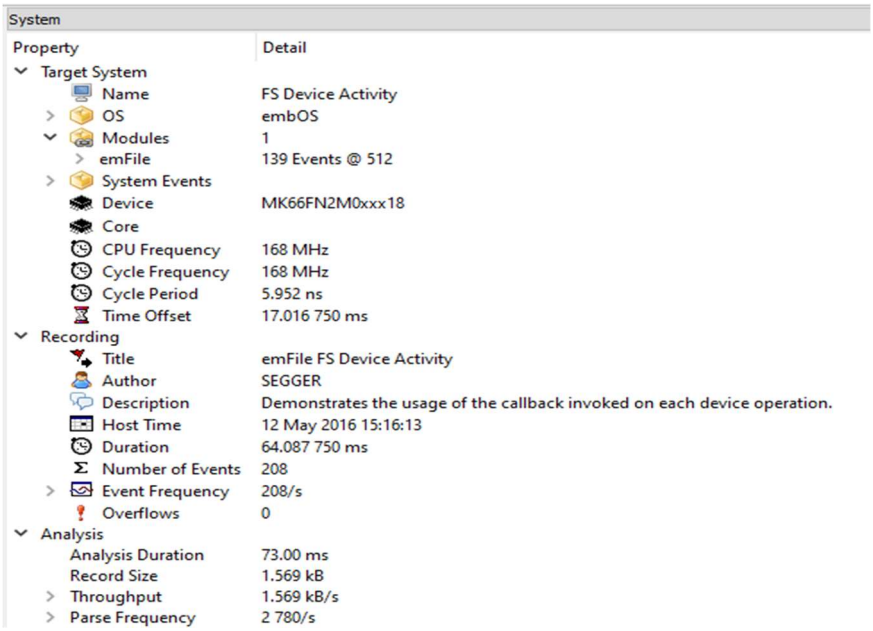


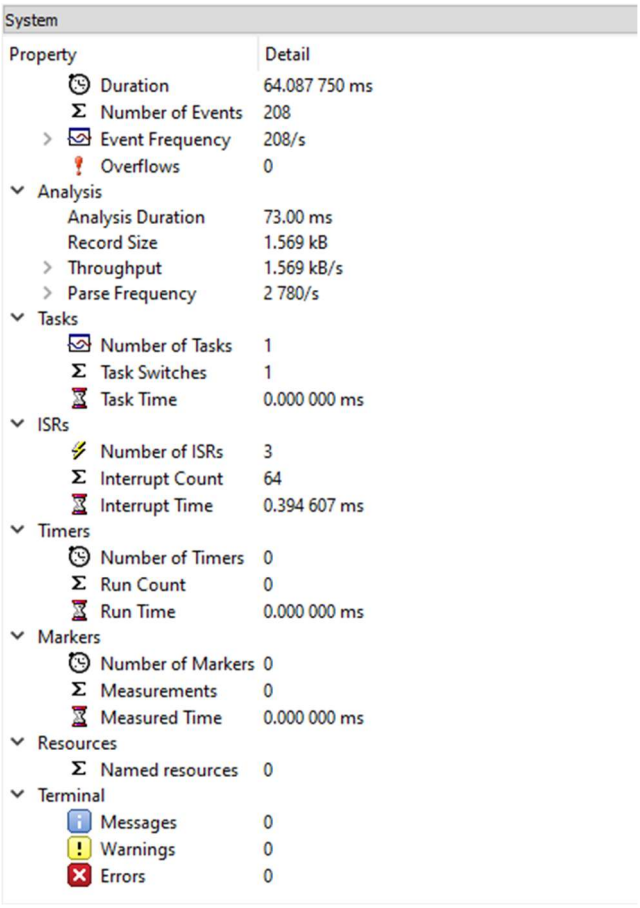


2) Acquire system information(timeline)



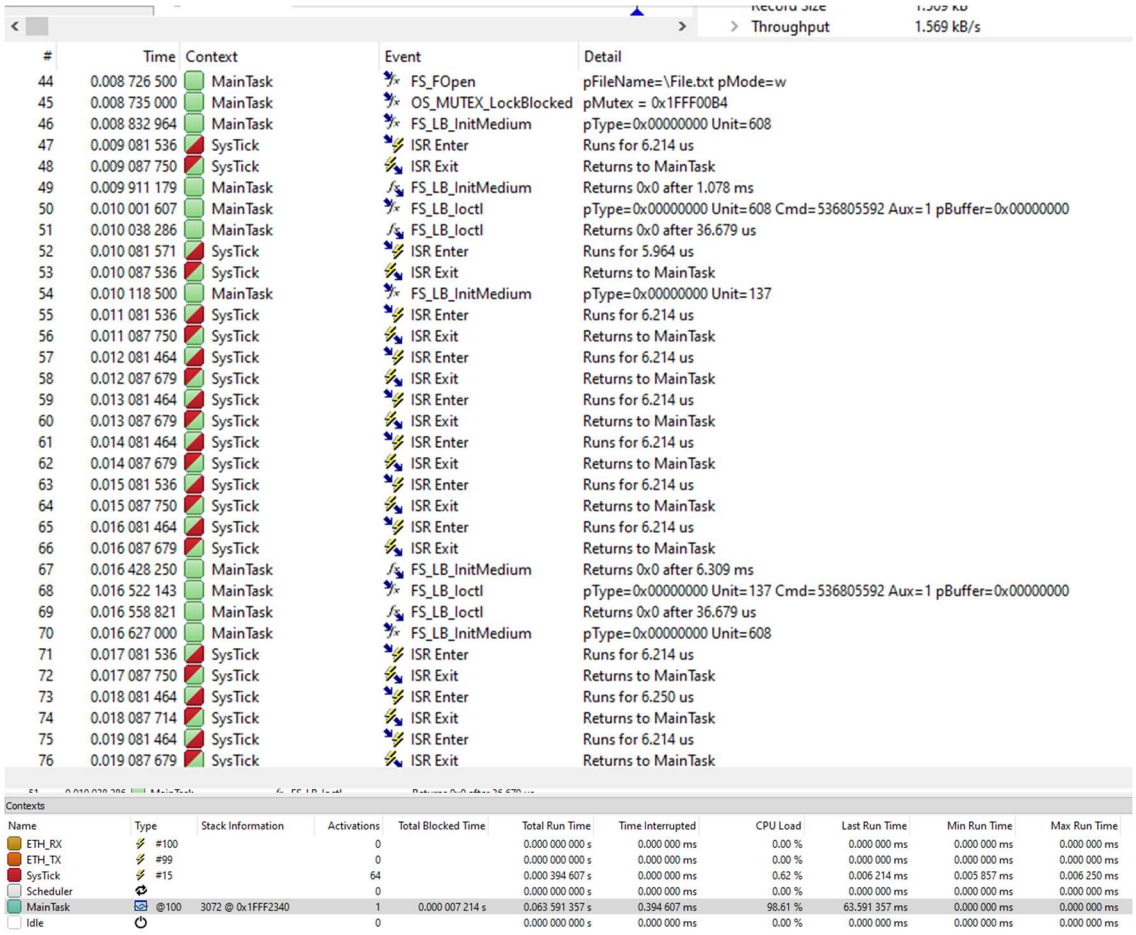
3) Analyzing System Activity: (Timeline, Events list, Terminal and Contexts window)





System	
Property	Detail
Duration	64.087 750 ms
Number of Events	208
Event Frequency	208/s
Overflows	0
Analysis	
Analysis Duration	73.00 ms
Record Size	1.569 kB
Throughput	1.569 kB/s
Parse Frequency	2 780/s
Tasks	
Number of Tasks	1
Task Switches	1
Task Time	0.000 000 ms
ISRs	
Number of ISRs	3
Interrupt Count	64
Interrupt Time	0.394 607 ms
Timers	
Number of Timers	0
Run Count	0
Run Time	0.000 000 ms
Markers	
Number of Markers	0
Measurements	0
Measured Time	0.000 000 ms
Resources	
Named resources	0
Terminal	
Messages	0
Warnings	0
Errors	0

4) **SystemView Application (Timeline, Events, Terminal, CPU load window, Context's window, system information window)**



Throughput 1.569 kB/s

#	Time	Context	Event	Detail
44	0.008 726 500	MainTask	FS_FOpen	pFileName=\File.txt pMode=w
45	0.008 735 000	MainTask	OS_MUTEX_LockBlocked	pMutex = 0x1FFF00B4
46	0.008 832 964	MainTask	FS_LB_InitMedium	pType=0x00000000 Unit=608
47	0.009 081 536	SysTick	ISR Enter	Runs for 6.214 us
48	0.009 087 750	SysTick	ISR Exit	Returns to MainTask
49	0.009 911 179	MainTask	FS_LB_InitMedium	Returns 0x0 after 1.078 ms
50	0.010 001 607	MainTask	FS_LB_Ioctl	pType=0x00000000 Unit=608 Cmd=536805592 Aux=1 pBuffer=0x00000000
51	0.010 038 286	MainTask	FS_LB_Ioctl	Returns 0x0 after 36.679 us
52	0.010 081 571	SysTick	ISR Enter	Runs for 5.964 us
53	0.010 087 536	SysTick	ISR Exit	Returns to MainTask
54	0.010 118 500	MainTask	FS_LB_InitMedium	pType=0x00000000 Unit=137
55	0.011 081 536	SysTick	ISR Enter	Runs for 6.214 us
56	0.011 087 750	SysTick	ISR Exit	Returns to MainTask
57	0.012 081 464	SysTick	ISR Enter	Runs for 6.214 us
58	0.012 087 679	SysTick	ISR Exit	Returns to MainTask
59	0.013 081 464	SysTick	ISR Enter	Runs for 6.214 us
60	0.013 087 679	SysTick	ISR Exit	Returns to MainTask
61	0.014 081 464	SysTick	ISR Enter	Runs for 6.214 us
62	0.014 087 679	SysTick	ISR Exit	Returns to MainTask
63	0.015 081 536	SysTick	ISR Enter	Runs for 6.214 us
64	0.015 087 750	SysTick	ISR Exit	Returns to MainTask
65	0.016 081 464	SysTick	ISR Enter	Runs for 6.214 us
66	0.016 087 679	SysTick	ISR Exit	Returns to MainTask
67	0.016 428 250	MainTask	FS_LB_InitMedium	Returns 0x0 after 6.309 ms
68	0.016 522 143	MainTask	FS_LB_Ioctl	pType=0x00000000 Unit=137 Cmd=536805592 Aux=1 pBuffer=0x00000000
69	0.016 558 821	MainTask	FS_LB_Ioctl	Returns 0x0 after 36.679 us
70	0.016 627 000	MainTask	FS_LB_InitMedium	pType=0x00000000 Unit=608
71	0.017 081 536	SysTick	ISR Enter	Runs for 6.214 us
72	0.017 087 750	SysTick	ISR Exit	Returns to MainTask
73	0.018 081 464	SysTick	ISR Enter	Runs for 6.250 us
74	0.018 087 714	SysTick	ISR Exit	Returns to MainTask
75	0.019 081 464	SysTick	ISR Enter	Runs for 6.214 us
76	0.019 087 679	SysTick	ISR Exit	Returns to MainTask

Contexts

Name	Type	Stack Information	Activations	Total Blocked Time	Total Run Time	Time Interrupted	CPU Load	Last Run Time	Min Run Time	Max Run Time
ETH_RX	#100		0		0.000 000 000 s	0.000 000 ms	0.00 %	0.000 000 ms	0.000 000 ms	0.000 000 ms
ETH_TX	#99		0		0.000 000 000 s	0.000 000 ms	0.00 %	0.000 000 ms	0.000 000 ms	0.000 000 ms
SysTick	#15		64		0.000 394 607 s	0.000 000 ms	0.62 %	0.006 214 ms	0.005 857 ms	0.006 250 ms
Scheduler			0		0.000 000 000 s	0.000 000 ms	0.00 %	0.000 000 ms	0.000 000 ms	0.000 000 ms
MainTask	@100	3072 @ 0x1FFF2340	1	0.000 007 214 s	0.063 591 357 s	0.394 607 ms	98.61 %	63.591 357 ms	0.000 000 ms	0.000 000 ms
Idle			0		0.000 000 000 s	0.000 000 ms	0.00 %	0.000 000 ms	0.000 000 ms	0.000 000 ms

Conclusion:

The integration of SEGGER SystemView with FreeRTOS enables real-time analysis and debugging of embedded system runtime behavior, enhancing system reliability. This approach deepens understanding of co-design and co-simulation techniques, driving advancements in embedded system testing and development.

Signature of faculty in-charge with date