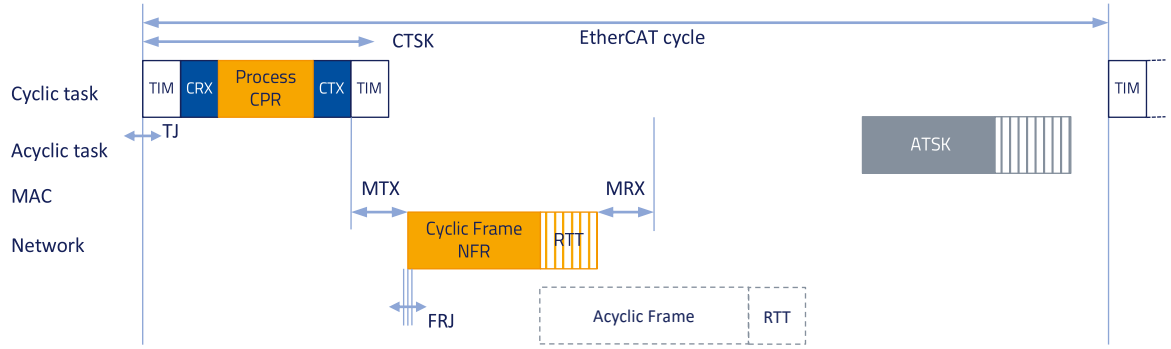
icECAT. EtherCAT Master Stack

Benchmark

Target System: NXP i.MX RT1064

- CPU: NXP i.MX RT1064 @ 528 MHz, ARM Cortex-M7
- MAC: Internal MAC (ENET)
- Operating System: FreeRTOS v10.1.1, execution from ITCM [FRTOS1], execution from SRAM [FRTOS2]
  Bare-metal (no OS), execution from SRAM [NOOS1], execution from flash [NOOS2]
- EtherCAT Software: icECAT EtherCAT Master Stack, v1.7, acyclic frame chained to one cyclic task
  icNET Optimized Link Layer Driver for i.MX RT1050/64, v2.2
- EtherCAT Network: Beckhoff EK1100 + EL2008 + EL1018 + EL4132 + EL3102 + EL9011
- EtherCAT Cyclic Frame: PDUs: 2x DC + MBX IRQ + Process Data (LWR + LRD + LRW) + BRD => Σ 183 bytes

EtherCAT cycle timing diagram



			[FRTOS1]	[FRTOS2]	[NOOS1]	[NOOS2]
Cyclic Task (with acyclic communication in parallel)						
Receive last cyclic frame (icECAT Master)	CRX	µs	7,2	7,9	5,7	11,6
Process cyclic data by application (example: 50x read access, increment data, 50x write access)	CPR	µs	0,6	1,0	0,6	1,4
Send next cyclic and pending acyclic frame (icECAT Master)	CTX	µs	1,5	1,8	1,2	2,7
Cyclic task overall, incl. management of cycle timer	CTSK	µs	9,8	11,0	7,9	19,7
Acyclic Task						
Acyclic task, with async command handling (icECAT Master), average value	ATSK	µs	6,4	6,8	3,2	9,2
MAC, Network						
MAC DMA time between send until start of frame (SOF) on the network	MTX	µs	2,4	2,4	2,4	2,4
Cyclic frame length time on the network (183 bytes)	NFR	µs	14,6	14,6	14,6	14,6
Network round trip time (5 slaves)	RTT	µs	3,4	3,4	3,4	3,4
MAC DMA time between end of frame (EOF) on network until frame can be received by software	MRX	µs	2,5	2,7	2,4	3,0
Jitter (with cyclic and acyclic load)						
Timer jitter (interrupt latency, thread scheduling)	TJ	µs	1,3	1,5	0,9	2,9
Resulting frame jitter incl. DC timer adjustment in master, measured with DC reference clock (slave)	FRJ	µs	2,4	2,9	3,1	
CPU Load (Cyclic task and idle acyclic task called @ 1ms)						
Cycle time @ 100 µs			13%	14%	13%	---
Cycle time @ 500 µs			3%	3%	6%	8%
Cycle time @ 1 ms			2%	2%	1%	2%

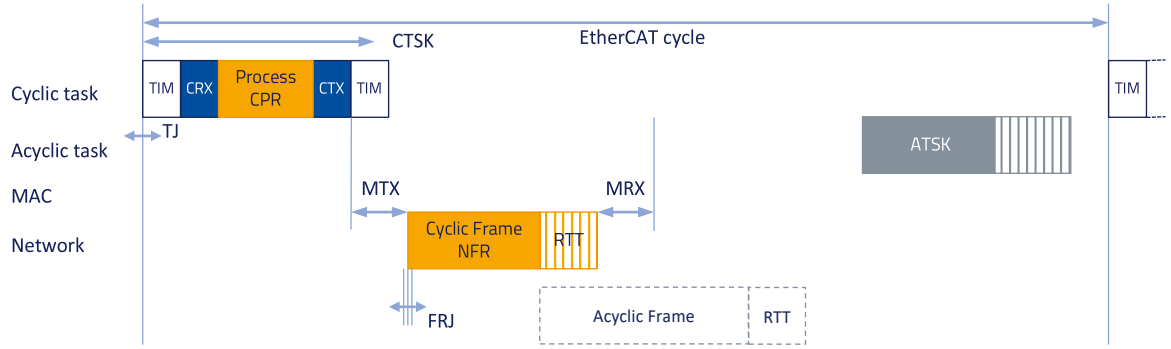
icECAT. EtherCAT Master Stack

Benchmark

Target System: Intel x86

- CPU: Intel Core i3-10100 CPU @ 3.60GHz (4 core, 8 threads) x86_x64
- MAC: Intel I225
- Operating System: Linux (with Preempt-RT Patch), #1 SMP PREEMPT_RT Debian 5.10.13-1 (2021-02-06)
- EtherCAT Software: icECAT EtherCAT Master Stack, v1.7, acyclic frame chained to one cyclic task
  icNET Optimized Link Layer Driver for Intel I225 [I225]
  icNET Standard Link Layer Driver for Linux RAW socket [RAW] (supporting I225)
- EtherCAT Network: Beckhoff EK1100 + EL2008 + EL1018 + EL4132 + EL3102 + EL9011
- EtherCAT Cyclic Frame: PDUs: 2x DC + MBX IRQ + Process Data (LWR + LRD + LRW) + BRD => Σ 183 bytes

EtherCAT cycle timing diagram



			[I225]	[RAW]
Cyclic Task (with acyclic communication in parallel)				
Receive last cyclic frame (icECAT Master)	CRX	µs	0.7	1.4
Process cyclic data by application (example: 50x read access, increment data, 50x write access)	CPR	µs	0.1	0.1
Send next cyclic and pending acyclic frame (icECAT Master)	CTX	µs	0.1	2.0
Cyclic task overall, incl. management of cycle timer	CTSK	µs	0.8	3.4
Acyclic Task				
Acyclic task, with async command handling (icECAT Master), average value	ATSK	µs	0.5	0.5
MAC, Network[1]				
MAC DMA time between send until start of frame (SOF) on the network	MTX	µs	6.5	---
Cyclic frame length time on the network (183 bytes)	NFR	µs	14,6	14,6
Network round trip time (5 slaves)	RTT	µs	5.0	---
MAC DMA time between end of frame (EOF) on network until frame can be received by software	MRX	µs	5.0	---
Jitter (with cyclic and acyclic load)				
Timer jitter (interrupt latency, thread scheduling)	TJ	µs	1.8	1.7
Resulting frame jitter incl. DC timer adjustment in master, measured with DC reference clock (slave)	FRJ	µs	5.3	22.0
CPU Load (Cyclic task and idle acyclic task called @ 1ms)[2]				
Cycle time @ 100 µs			2.2%	---
Cycle time @ 500 µs			0.5%	1.6%
Cycle time @ 1 ms			0.3%	1.0%

1 With support of hardware timestamping enabled
2 Measured with "top" (100% == 1 hyper thread) incl. the load of the kernel IRQ task for raw socket driver

Benchmark

- Target System: STM32H7, Optimized Link Layer Driver
- Target System: NXP i.MX 8, Optimized Link Layer Driver
- Target System: …

Benchmark information for further target systems is coming soon ...