

Screen shot for memset result (KL25z)

```
^09 DEADBEEF 06000000600000
^09 DEADBEEF 06000000F80000
^09 DEADBEEF 06000000780000
^09 DEADBEEF 06000001720100
^09 DEADBEEF 06000001710200
^09 DEADBEEF 06000001640400
^09 DEADBEEF 06000002D80400
^09 DEADBEEF 06000006F40400
^09 DEADBEEF 06000002F20400
^09 DEADBEEF 06000002240100
^09 DEADBEEF 06000001FA0200
^09 DEADBEEF 06000001D20400
^09 DEADBEEF 0600001B740400
^09 DEADBEEF 06000042B80400
^09 DEADBEEF 0600001B8C0400
^09 DEADBEEF 06000009300100
^09 DEADBEEF 06000007400200
^09 DEADBEEF 060000063B0400
^09 DEADBEEF 06000088DA0400
^09 DEADBEEF 0600014C5B0400
^09 DEADBEEF 06000088F10400
^09 DEADBEEF 06000028730100
^09 DEADBEEF 0600001EB50200
^09 DEADBEEF 06000019C60400
```

Interpretation of memset results (KL25z)

	Library function	my_memset		Dma memset		
		-O0	-O3	1byte	2byte	4byte
10	96	248	120	370	369	356
100	728	1780	754	548	506	466
1000	7028	17080	7052	2352	1856	1595
5000	35034	85083	35057	10355	7861	6598

Screenshot of memmove result (KL25Z)

```
^09 DEADBEEF 06000000890401
^09 DEADBEEF 06000001350401
^09 DEADBEEF 060000009B0401
^09 DEADBEEF 060000019C0101
^09 DEADBEEF 060000019D0201
^09 DEADBEEF 060000018E0401
^09 DEADBEEF 06000003B30401
^09 DEADBEEF 06000008970401
^09 DEADBEEF 06000003C50401
^09 DEADBEEF 06000002500101
^09 DEADBEEF 06000002240201
^09 DEADBEEF 06000001FE0401
^09 DEADBEEF 060000235A0401
^09 DEADBEEF 060000526D0401
^09 DEADBEEF 06000023690401
^09 DEADBEEF 060000095A0101
Nordic test...
Enter data at the console.
^09 DEADBEEF 060000076C0201
^09 DEADBEEF 06000006650401
^09 DEADBEEF 060000AFFD0401
^09 DEADBEEF 0600019A900401
^09 DEADBEEF 060000B00C0401
^09 DEADBEEF 060000289D0101
^09 DEADBEEF 0600001EDF0201
^09 DEADBEEF 06000019F00401
```

Interpretation of memmove result (KL25z)

	Library function	my_memset		Dma memset		
		-O0	-O3	1byte	2byte	4byte
10	137	309	155	412	413	398
100	947	2199	965	592	548	510
1000	9050	21101	9065	2394	1900	1637
5000	45053	105104	45068	10397	7903	6640

Conclusions from profiler run on KL25z

- All results in the above table are in clock cycles.
- It can clearly be seen that DMA transfers pay off only for a large number of bytes.
- Library routines written by manufacturer (ARM) are highly optimized
- The functions written for the project (my_memset & my_memmove) perform very poorly without optimization and with maximum optimization come close to library routines
- DMA because it uses hardware for transfers is way faster than any software written transfers. Also DMA efficiency depends on byte size. Using the largest byte size that DMA supports (the CPU bus size) results in the most optimized transfer