6 Rules to write optimized code for embedded systems

www.embeddedshiksha.com

1 Minimize Memory Usage

- Use smaller data types where possible (e.g., uint8_t instead of int).
- Avoid dynamic memory allocation (e.g., malloc) and prefer static allocation or memory pools.
- Be mindful of data alignment and packing to minimize memory waste.

2 Efficient Data Structures:

- Choose algorithms that are efficient for your specific use case.
- Opt for data structures that minimize memory usage and support fast access and manipulation.

3 Loop Optimization:

- Optimize critical loops for speed by minimizing branching, reducing loop overhead, and minimizing memory access latency.
- Consider loop unrolling and loop reordering to improve instruction cache utilization.

4

Reduce Function Calls

Minimize function calls, especially in tight loops, as function call overhead can be significant in embedded systems.

5 Interrupt Handling

- Minimize the time spent in interrupt service routines (ISRs).
- Use appropriate interrupt priorities to handle critical tasks promptly.

6 Power Efficiency

- Implement power-saving strategies, such as using low-power modes when the CPU is idle.
- Be mindful of peripherals and sensors that consume power and deactivate them when not in use.