

CSP4CMSIS Quick Reference (v1.0)

Formal Process Networks for ARM Cortex-M & FreeRTOS

1. Process Management

```
class MyProc : public CSProcess {  
public:  
    void run() override {  
        while(true) { /* Logic */ }  
    }  
};
```

Mapped to persistent RTOS tasks.

2. Channel Communication

Rendezvous (Sync):

```
static Channel<type> c;  
Chanin<type> in = c.reader();  
Chanout<type> out = c.writer();
```

Buffered (Async):

```
static BufferedOne2OneChannel<type, N> c;
```

Operators: `in >> msg;` (Read) | `out << msg;` (Write)

3. External Choice (\square)

```
Alternative alt(inA | msgA, inB | msgB);  
int idx = alt.fairSelect();
```

Guard: channel | variable populates the variable automatically.

4. Interrupt Handling (ISR)

```
chan.writer().putFromISR(val);  
portYIELD_FROM_ISR(pdTRUE);
```

5. Network Orchestration

```
Run(InParallel(p1, p2, p3),  
     ExecutionMode::StaticNetwork);
```

6. Formal Mappings

CSP Math	C++ Syntax	Semantics
$P \parallel Q$	InParallel(p, q)	Parallel
$c!x \rightarrow P$	out << x	Output
$c?x \rightarrow P$	in >> x	Input
$P \square Q$	Alternative alt	Ext. Choice
$P \sqcap Q$	Internal Logic	Int. Choice

7. Safety & Zero-Heap

- Static Memory:** Use `static` for all Channels and Processes.
- POD Data:** Send only simple structs.
- Stack ALT:** Alternative lives on the task stack.
- Determinism:** Fixed-size buffers prevent OOM errors.

8. Troubleshooting

- Deadlock:** Cycle of processes waiting on Rendezvous.
- ISR Jitter:** Use `BufferedOne2OneChannel`.
- Starvation:** `fairSelect()` ensures fairness.