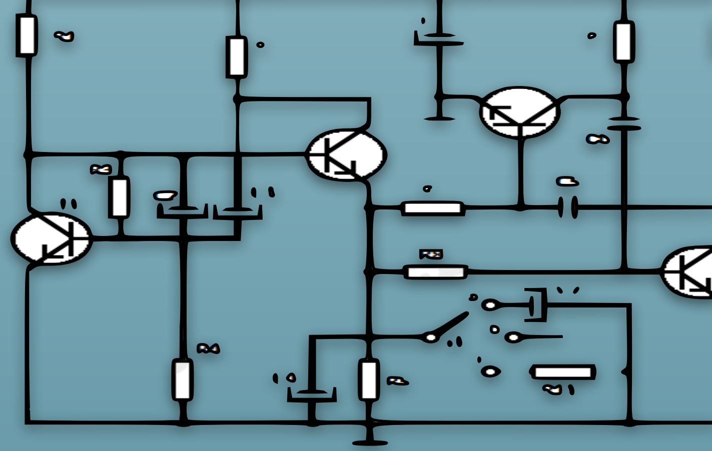


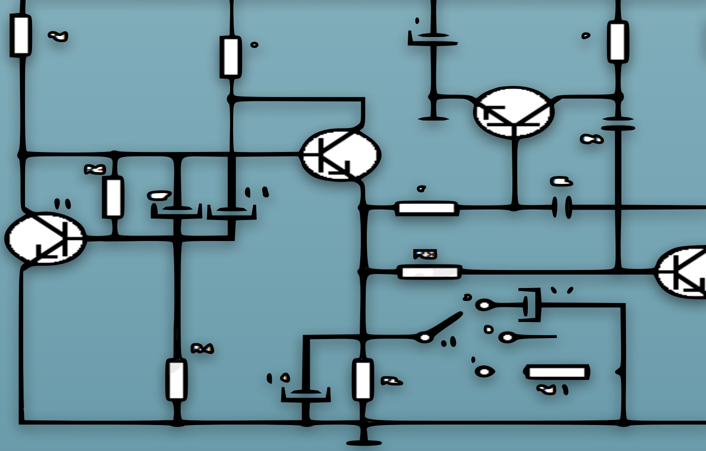
Implementation of FIFO queue for generic data type

THREAD-SAFE QUEUE API FOR EMBEDDED SYSTEMS

RTES

Mannone Vito

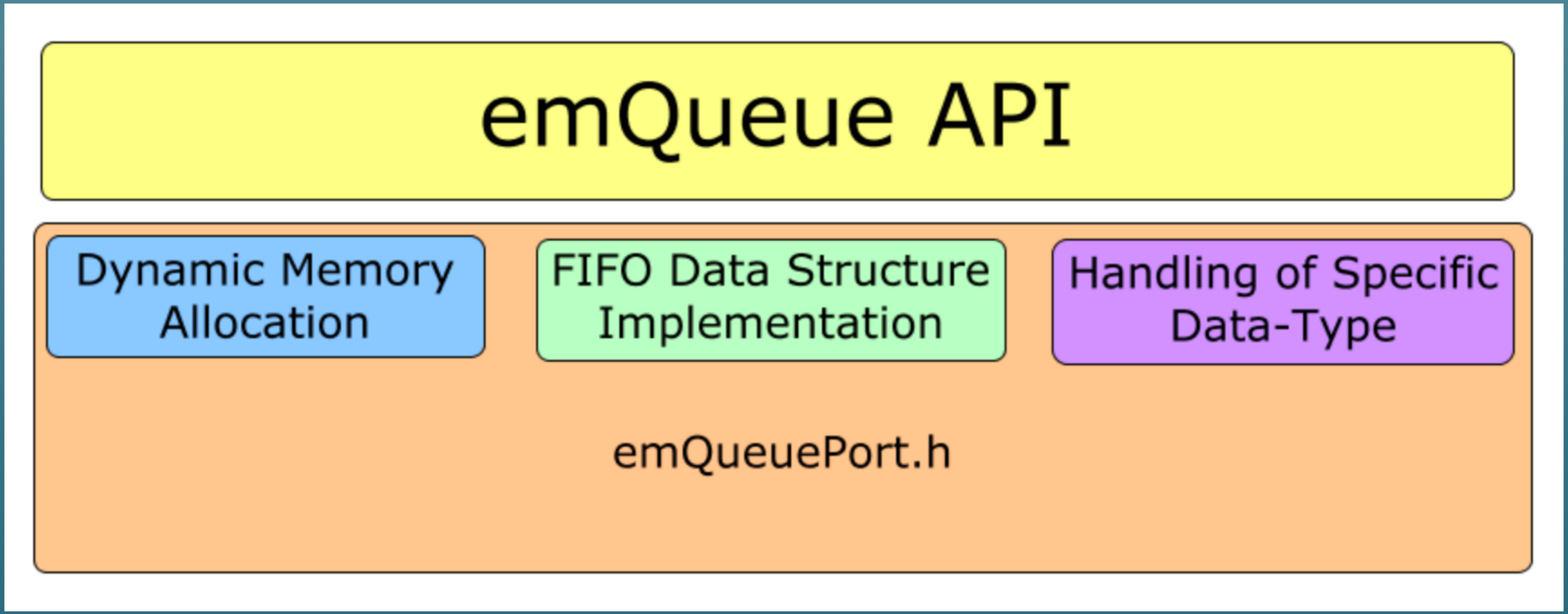
BOLOGNA

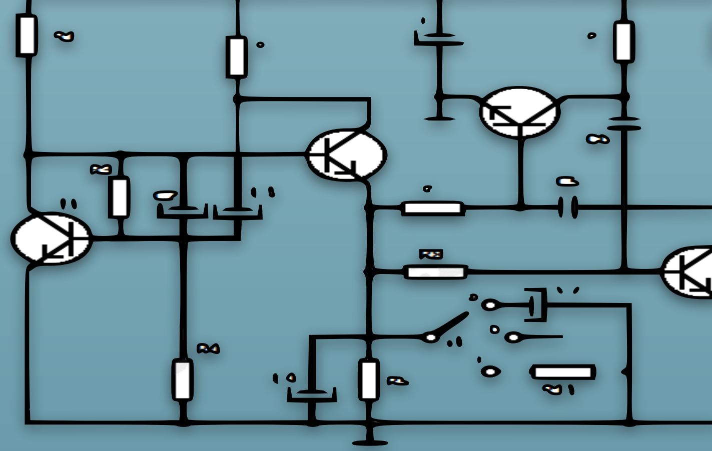


EMQUEUE API

Directory organisation

- **emQueue.c —> functions implementation**
- **emQueue.h —> functions declaration**
- **emQueuePort.h —> declaration of necessary porting functions**

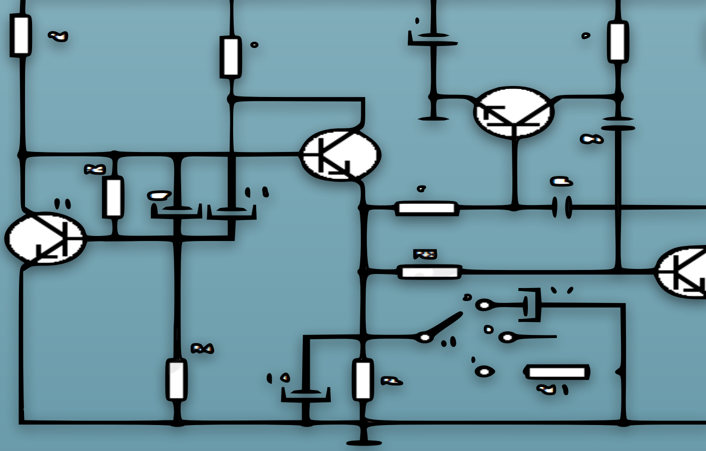

emQueue API
Dynamic Memory Allocation
FIFO Data Structure Implementation
Handling of Specific Data-Type
emQueuePort.h



EMQUEUE API

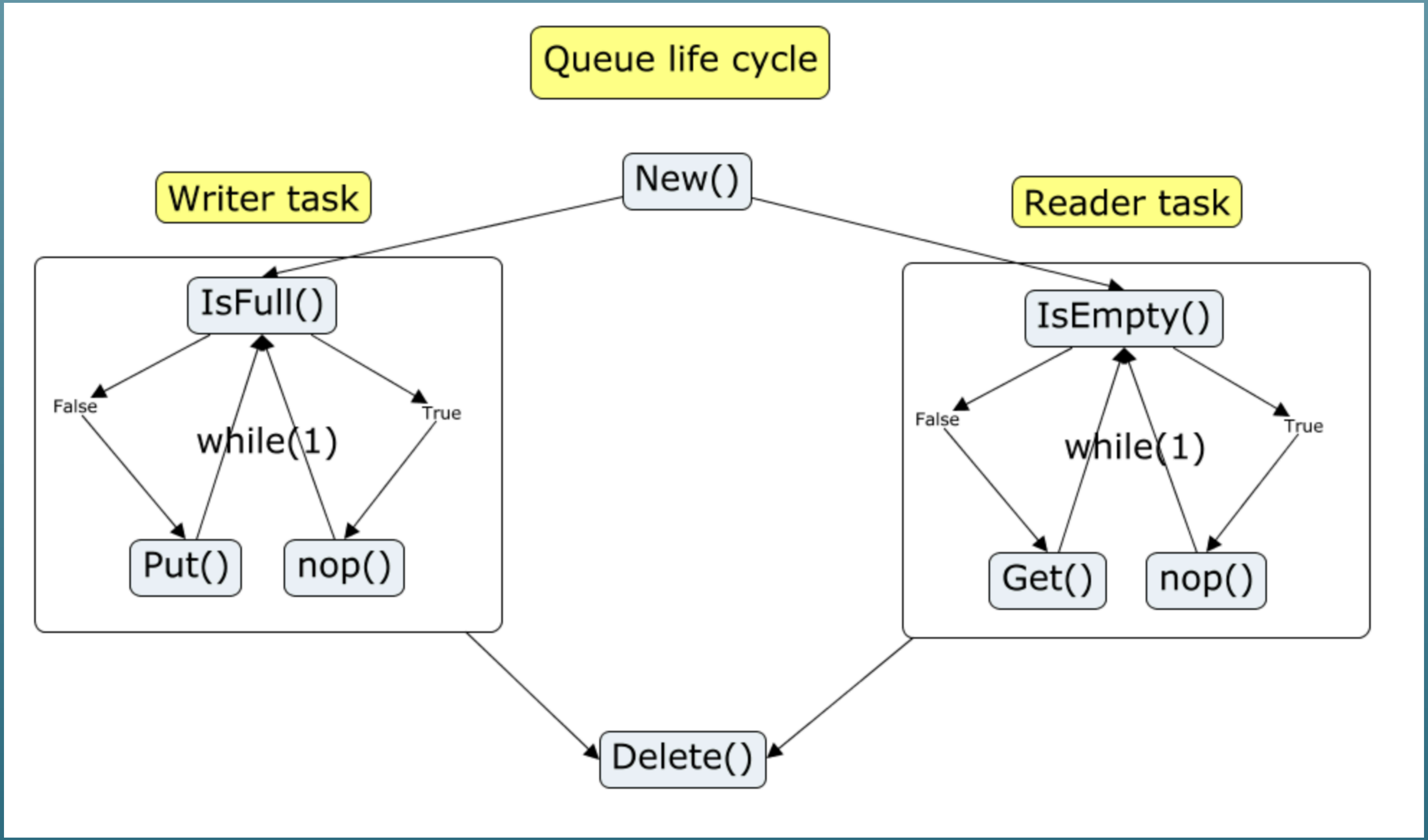
API functions *emQueue_[function]*

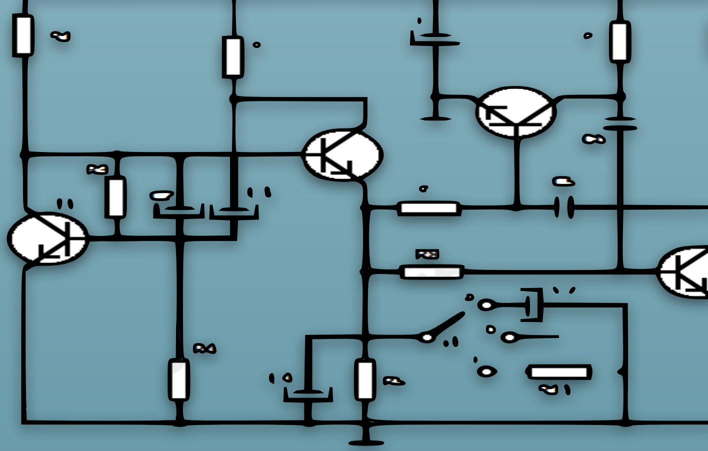
- New() —> initialises the queue, allocating the necessary pool of memory
- IsFull() / IsEmpty() —> return the queue status: whether is or is not empty/full of elements
- Put() —> inserts an element of known size at the head of the queue
- Get() —> extracts an element and puts it in a given memory address
- Delete() —> de-allocates all the memory needed for the queue



EMQUEUE API

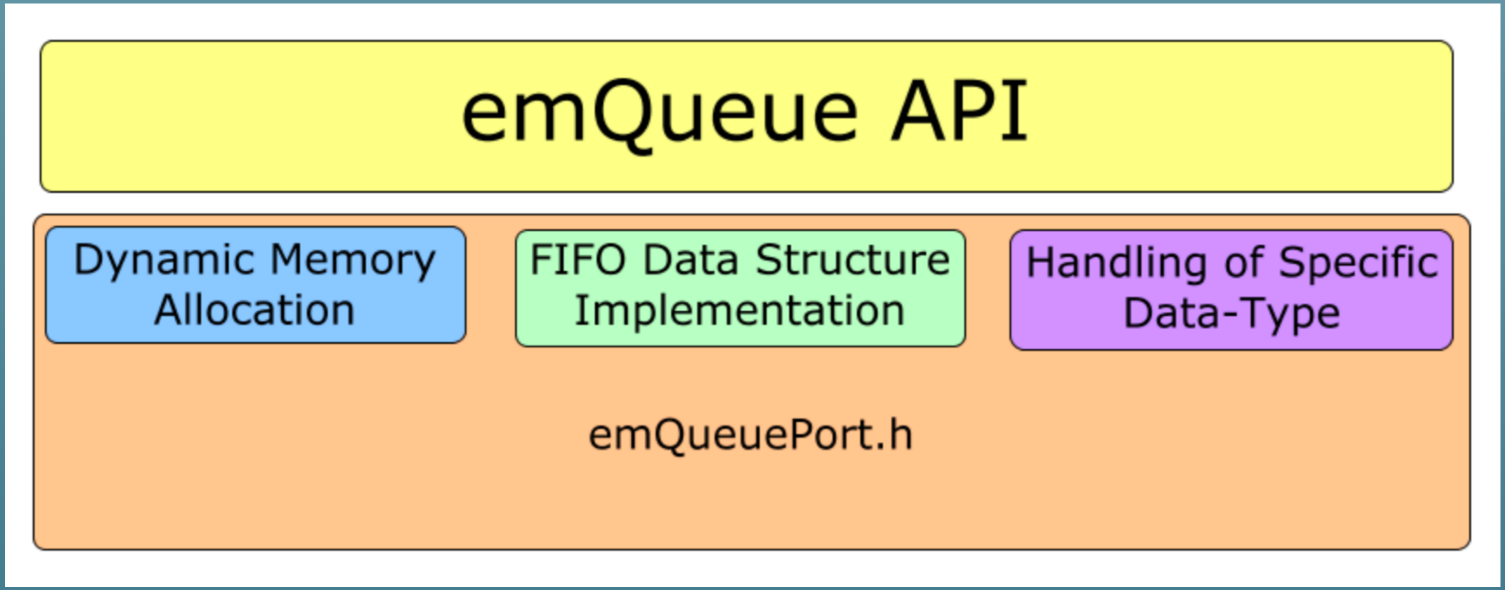
API functions *emQueue* **life cycle**

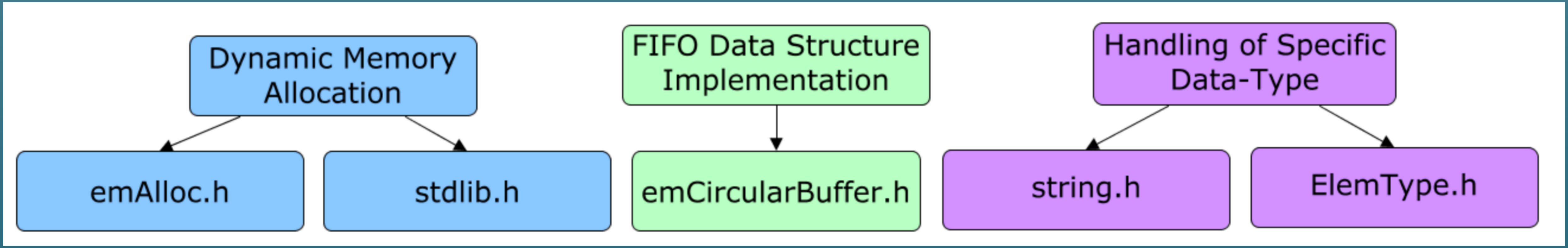


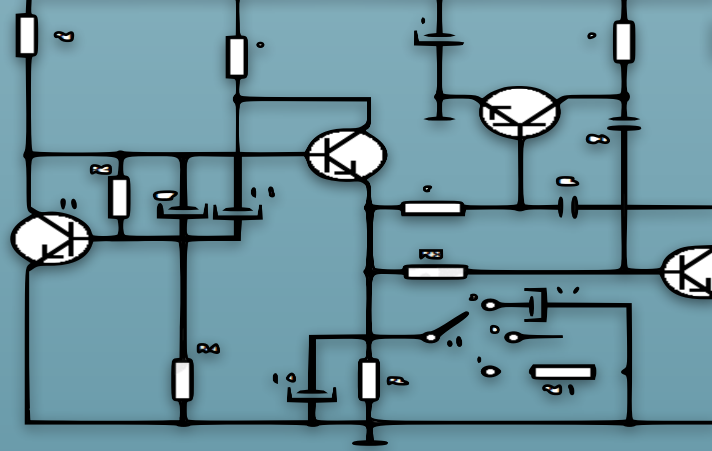


EMQUEUE API

Porting functions implementation

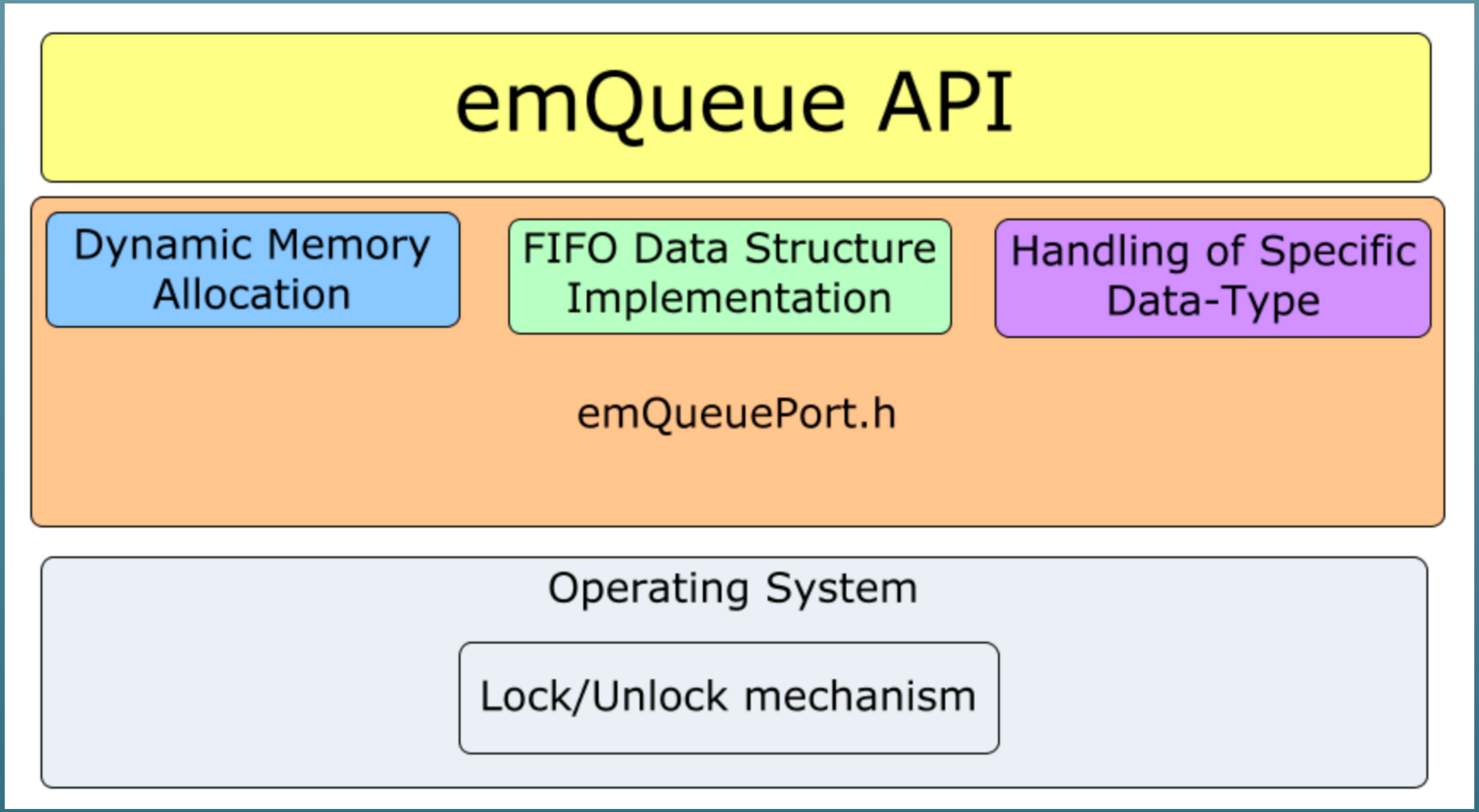

emQueue API
Dynamic Memory Allocation
FIFO Data Structure Implementation
Handling of Specific Data-Type
emQueuePort.h


Dynamic Memory Allocation
FIFO Data Structure Implementation
Handling of Specific Data-Type
emAlloc.h
stdlib.h
emCircularBuffer.h
string.h
ElemType.h

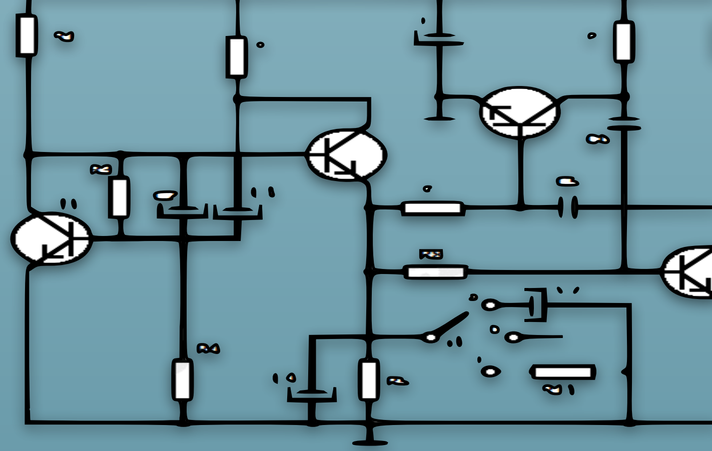


EMQUEUE API

Thread-safe implementation


emQueue API
Dynamic Memory Allocation
FIFO Data Structure Implementation
Handling of Specific Data-Type
emQueuePort.h
Operating System
Lock/Unlock mechanism

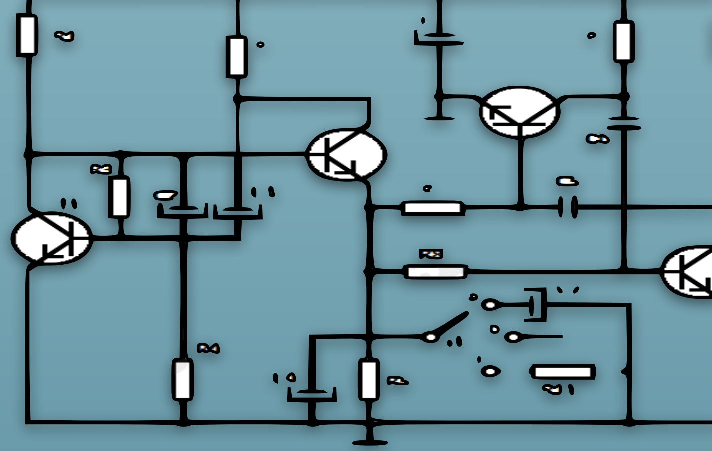
- **In porting headers it is possible to configure the library to use lock/unlock mechanisms such as semaphores and mutexes**
- **CMSIS-FreeRTOS and pthread can be used based on the actual machine the code will be running on**



EMQUEUE API

CMSIS-FreeRTOS, Nucleo F303RE example

- **Static allocation of memory**
- **User-defined data-type**
- **Queue implemented as ring buffer**
- **Queue disposal during runtime**
- **1 reader and 1 writer task**



EMQUEUE API

pthread, MacOs example

- Static or Dynamic allocation of memory
- User-defined data-type or generic data type handling
- Ring buffer implementation of the queue
- Multiple readers and multiple writers threads