**Memory Manager.**

Its a singleton class that’s a wrapper on top of the FreeRTOS’s memory handling APIs. For now, the heap for the memory allocation is initialized by the freeRTOS core itself. No wrapper has been implemented to control it from the wrappers. This can give a very granular control over the memory allocation and maybe we can have a general separation of the memory for different components.

The current implementation has 7 API’s thru which we can manage the heap memory.

* MemoryManager& get\_Instance();
* eMemoryResult get\_block(void\*\* pMemHolder, size\_t iNumBytes)
* void release\_block(void\* pMemHolder);
* eMemoryResult get\_stack(rtos\_thread\_stack\_t& ppStack, rtos\_stack\_size\_t StackSize);
* void release\_stack(rtos\_thread\_stack\_t ppStack);
* eMemoryResult get\_TCB(rtos\_thread\_cb\_t& ppTCB);
* void release\_TCB(rtos\_thread\_cb\_t ppTCB);