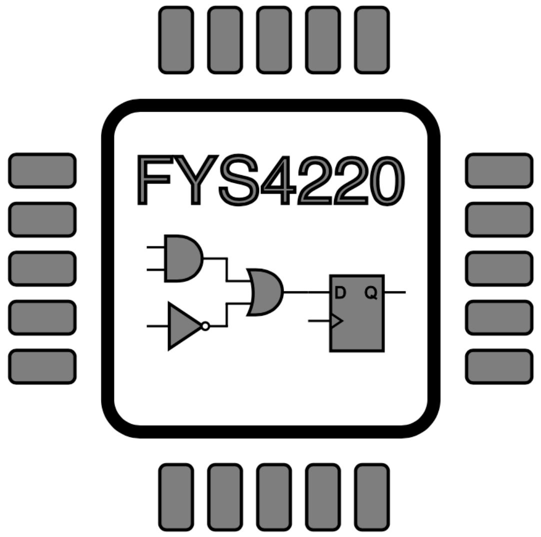
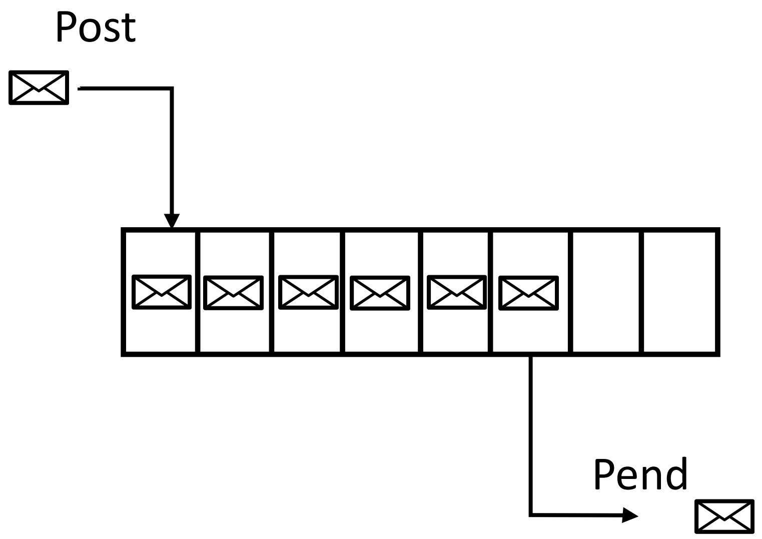


RTOS: Queues

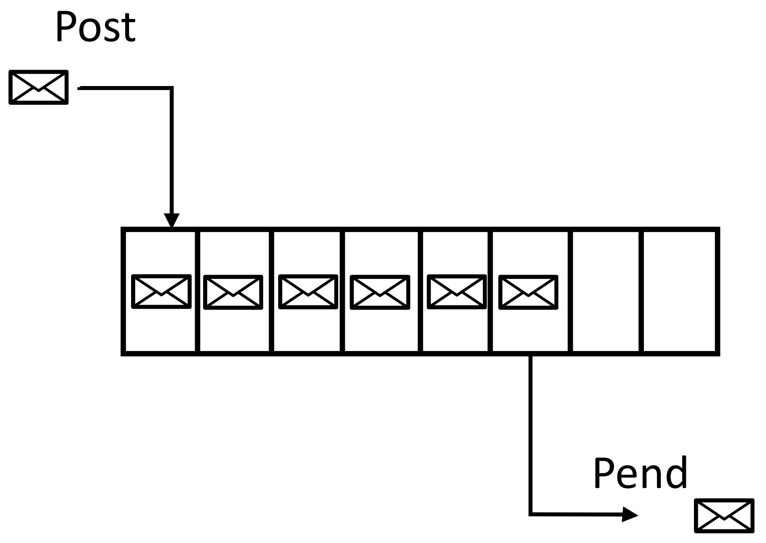


Message queue

- Tasks can also communicate by sending messages via message queues

- While a mailbox is limited to one meassage, a message queue can contain several messages

- FIFO*: First message inserted in the queue is the first message extracted from the queue



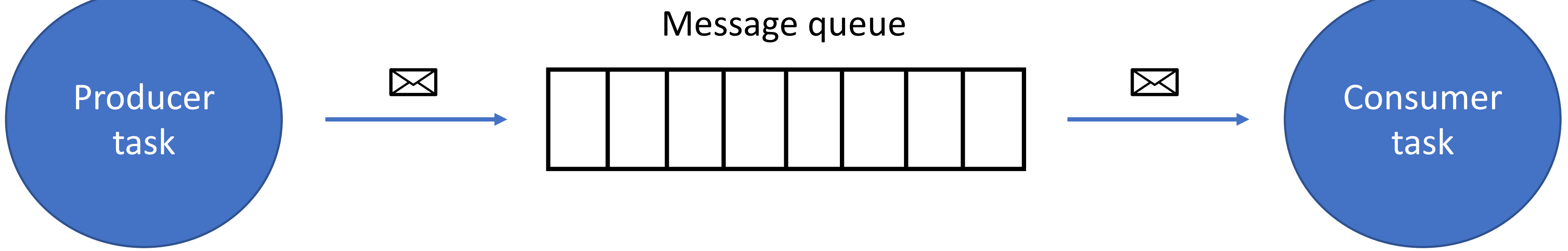
Mutual exclusion of queues is handled by the operating system

**FIFO: First In First Out.*
uC/OS-II can also be configured to support Last In First Out (LIFO)

Message queues

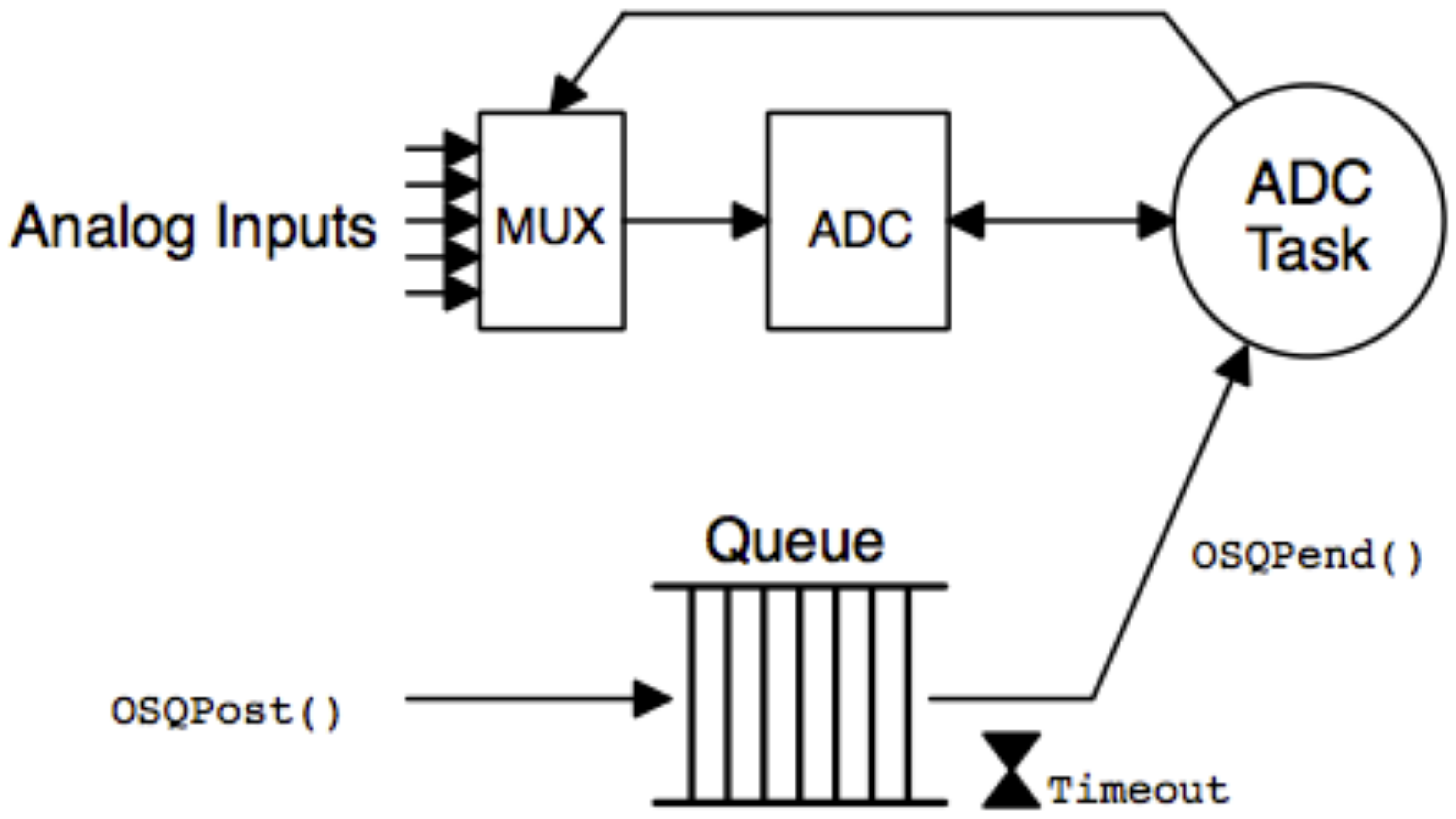
Message queues (and mailboxes) provide an asynchronous communications protocol

- the sender and receiver of the message do not need to interact with the message queue at the same time.
- Messages placed onto the queue are stored until the receiver retrieves them.



A waiting list is associated with each message queue
If queue is empty, requestion task blocked and put on waiting list

Using message queue for DAQ



Timeout of pend for queue corresponds to sampling frequency

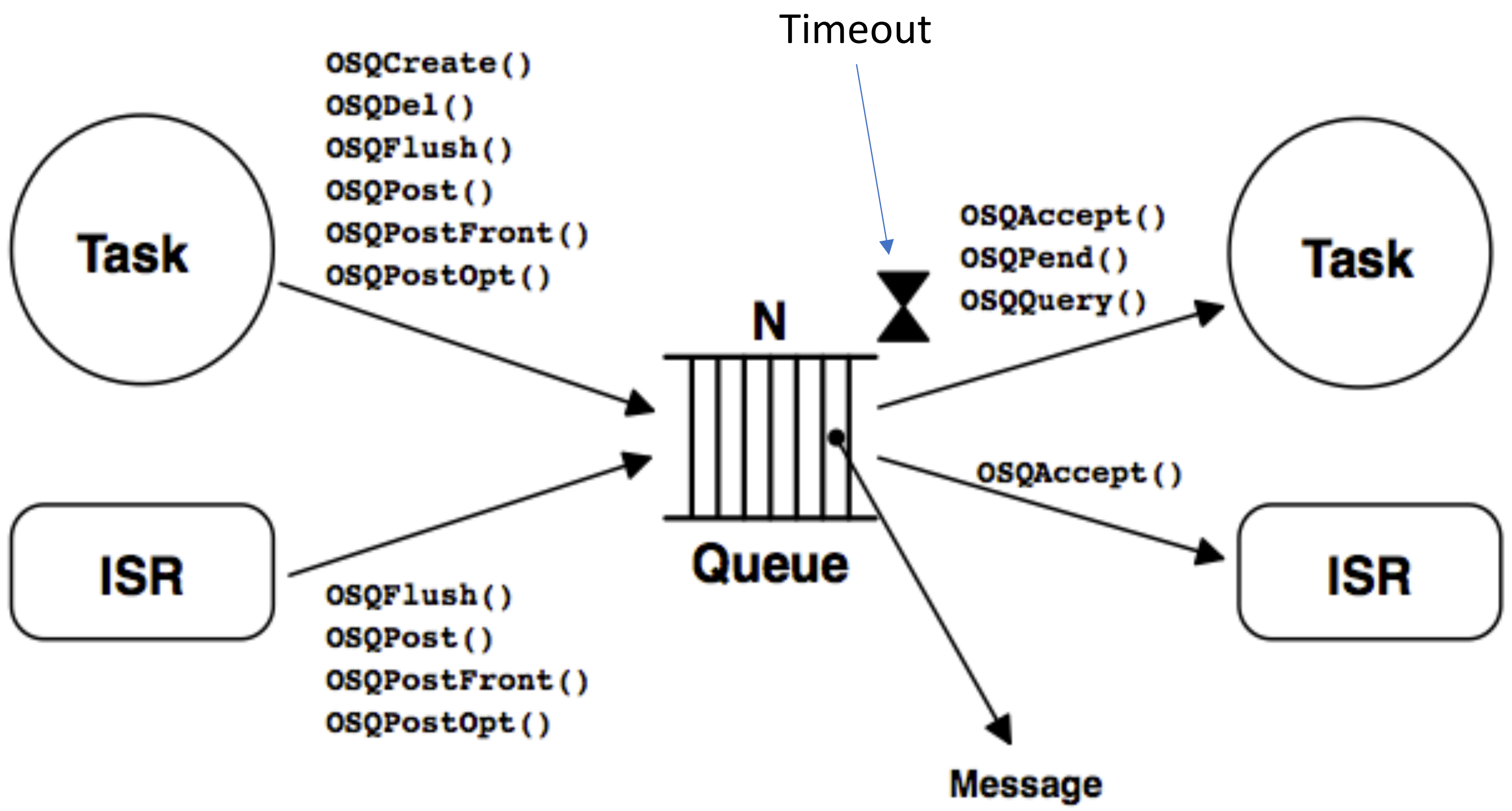
Message could contain information about
- Updated sampling frequency
- Which channel to read

And force an immediate conversion from another task if necessary

Message queues

- Generally, three types of operations can be performed on a message queue
  - Initialize (CREATE; always assumed to be empty)
  - Deposit a message into the queue(POST)
  - Wait for a message (PEND)

Relationship between tasks, ISR, and queues

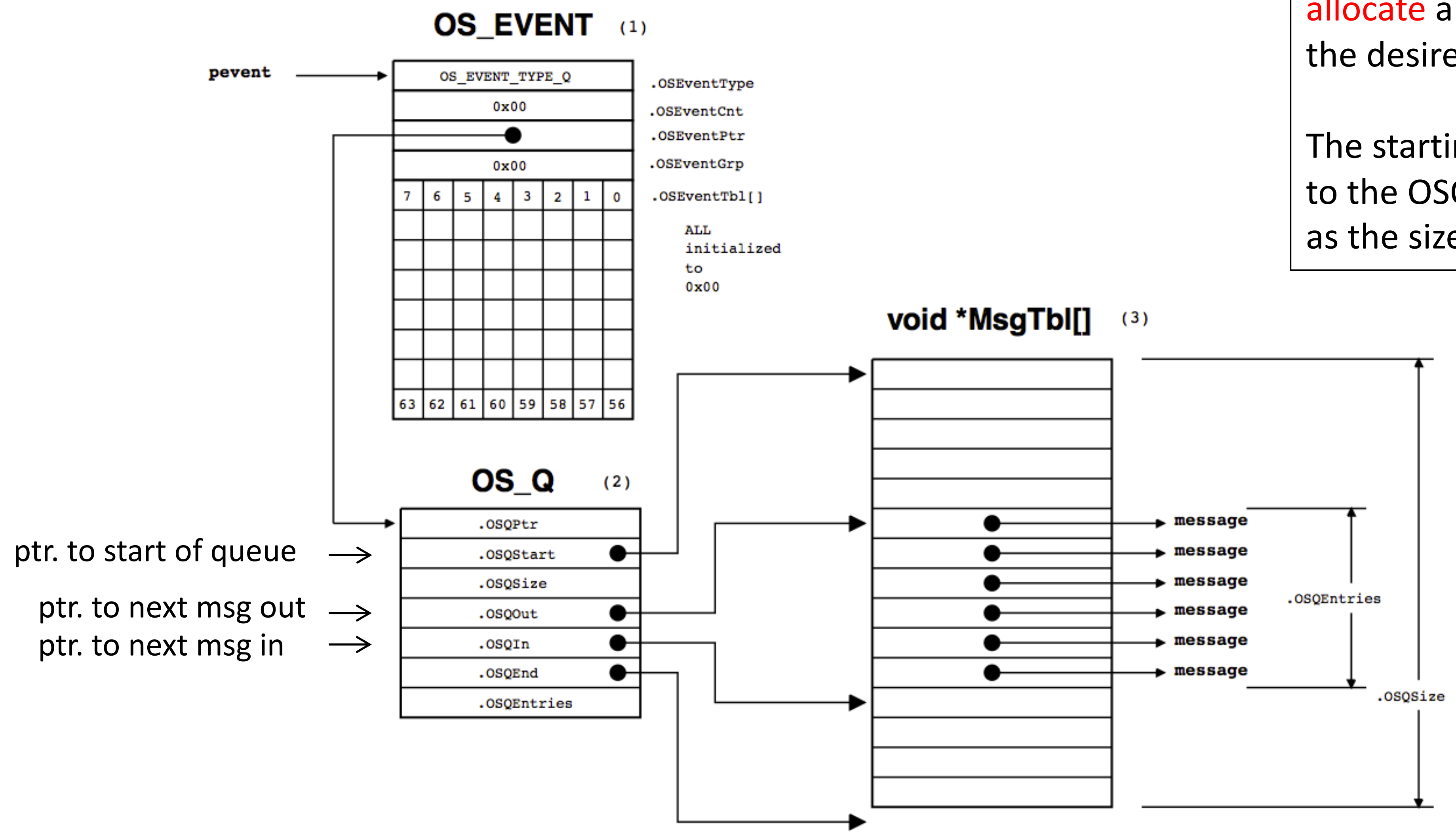


Source and documentation: https://doc.micrium.com/display/osiidoc/Message+Queue+Management

Message queue functions in uC/OS-II

- OS_EVENT *OSQCreate(void **start, INT16U size)
  - **start: pointer to an array that holds the messages
  - Array must be declared as an array of pointers to void
    - Void *MyArrayOfMsg[size]
  - Pass the address of MyArrayOfMsg[] to OSQCreate()
- void *OSQPend(OS_EVENT *pevent, INT16U timeout, INT8U *err)
  - Timeout: integral #ticks (0: wait forever)
- INT8U OSQPost(OS_EVENT *pevent, void *msg)
  - FIFO
- INT8U OSQPostFront(OS_EVENT *pevent, void *msg)
  - LIFO
- INT8U OSQPostOPT(OS_EVENT *pevent, void *msg, INT8U opt)
  - Allows posting of a message to all tasks (i.e. broadcast) waiting on the queue
  - Supports both LIFO and FIFO

Data structure for message queues



Before you create a queue, you need to allocate an array of pointers that contains the desired number of queue entries.

The starting address of the array is passed to the OSQCreate() as an argument, as well as the size of the array

```
//Message queue OS_EVENT structure
OS_EVENT *MSG_Q;
#define QSIZE  10
//Storage for message pointers
void *queue_data[QSIZE];
```