MWMR

General presentation

The MWMR communication middleware implements a generic inter-task communication mechanism for shared memory multi-processors architectures. This protocol has been designed to support both communication between *software tasks* (running on a programmable processor), and *hardware tasks*, implemented as dedicated hardware coprocessors.

MWMR stands for *Multi Writers, Multi-Readers*. The MWMR channel itself is implemented as a software FIFO, that can have several producers, and several consumers. Each MWMR communication channel is protected by a dedicated *lock*, for exclusive access.

Any access to a shared MWMR channel respect the following five stages protocol :

- get the lock protecting the MWMR (READ/WRITE access).
- test the status of the MWMR (READ access).
- transfer a burst of data between a local buffer and the MWMR (READ/WRITE access).
- update the status of the MWMR (WRITE access).
- release the lock (WRITE access).

The MWMR middleware can be used with both the MutekH and MutekS operating systems.

More Information

The MWMR middleware has two main components :

- The software part is a library of C functions. Those functions are build on top of the POSIX API and implement the 5 steps MWMR protocol. They can be used by a software task to read from(or write into) one or several MWMR channels. You can get the code ?here
- The hardware part is a generic MWMR controller. This hardware component has a DMA capability, and implement the 5 steps MWMR protocol. It and can be used by any hardware coprocessor that has one or several simple FIFO interfaces.

More Information

You can obtain more detailed information, and download the code