

MWMMR

General presentation

The MWMMR 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.

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

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

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

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

More Information

The MWMMR 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 MWMMR protocol. They can be used by a software task to read from(or write into) one or several MWMMR channels. The code is a MutekH library available as libmwmr
- The hardware part is a generic MWMMR controller. This hardware component has a DMA capability, and implement the 5 steps MWMMR protocol. It and can be used by any hardware coprocessor that has one or several simple FIFO interfaces.