

**NAME**

trlo\_setup\_map\_hardware, trlo\_setup\_check\_version, trlo\_unmap\_hardware – Setup access to a TRLO II module.

**SYNOPSIS**

```
#include "trlo_functions.h"
```

```
volatile trlo_register_map *
```

```
trlo_setup_map_hardware(int vmvk, void **unmapinfo);
```

```
void trlo_setup_check_version(volatile trlo_register_map *hardware);
```

```
void trlo_unmap_hardware(void *unmapinfo);
```

Link with `-ltrlo_ctrl`.

**DESCRIPTION**

The **trlo\_setup\_map\_hardware()** function maps the direct access hardware memory to be able to communicate with a VULOM/TRIDI running the TRLO II firmware. *vmvk* holds the address of the module as given by the two HEX switches. *unmapinfo* is a pointer to a pointer to optionally store unmap information. It may be NULL.

The returned pointer is used by all other read and control functions. Can also be used directly with the definitions in **trlo\_defs.h**. Note that this function calls **trlo\_setup\_check\_version()**.

The **trlo\_setup\_check\_version()** function checks that the module/firmware accessible at the virtual address *hardware* has the same version as the library. This can be used if the memory mapping has been provided elsewhere, without invoking **trlo\_setup\_map\_hardware()**

**trlo\_unmap\_hardware()** is used to release memory mapped by **trlo\_setup\_map\_hardware()**

**RETURN VALUE**

Virtual address pointer to access the module/firmware. (They may terminate the program via error reporting routines.)

**AUTHOR**

Håkan T. Johansson <f96hajo@chalmers.se>

**SEE ALSO**

**trlo\_read\_config\_file(3)**, **trlo\_read\_event(3)**, **trlolib(7)**, **trloii(7)**,