modbus\_mapping\_new\_start\_address(3)

===================================

NAME

----

modbus\_mapping\_new\_start\_address - allocate four arrays of bits and registers accessible from their starting addresses

SYNOPSIS

--------

\*modbus\_mapping\_t\* modbus\_mapping\_new\_start\_address(int 'start\_bits', int 'nb\_bits',

int 'start\_input\_bits', int 'nb\_input\_bits',

int 'start\_registers', int 'nb\_registers',

int 'start\_input\_registers', int 'nb\_input\_registers');\*

DESCRIPTION

-----------

The \_modbus\_mapping\_new\_start\_address()\_ function shall allocate four arrays to

store bits, input bits, registers and inputs registers. The pointers are stored

in modbus\_mapping\_t structure. All values of the arrays are initialized to zero.

The different starting adresses make it possible to place the mapping at any

address in each address space. This way, you can give access to values stored

at high adresses without allocating memory from the address zero, for eg. to

make available registers from 10000 to 10009, you can use:

[source,c]

-------------------

mb\_mapping = modbus\_mapping\_offset\_start\_address(0, 0, 0, 0, 10000, 10, 0, 0);

-------------------

With this code, only 10 registers (`uint16\_t`) are allocated.

If it isn't necessary to allocate an array for a specific type of data, you can

pass the zero value in argument, the associated pointer will be NULL.

This function is convenient to handle requests in a Modbus server/slave.

RETURN VALUE

------------

The \_modbus\_mapping\_offset\_new()\_ function shall return the new allocated structure if

successful. Otherwise it shall return NULL and set errno.

ERRORS

------

ENOMEM::

Not enough memory

EXAMPLE

-------

[source,c]

-------------------

/\* The first value of each array is accessible at the defined address.

The end address is ADDRESS + NB - 1. \*/

mb\_mapping = modbus\_mapping\_offset\_start\_address(BITS\_ADDRESS, BITS\_NB,

INPUT\_BITS\_ADDRESS, INPUT\_BITS\_NB,

REGISTERS\_ADDRESS, REGISTERS\_NB,

INPUT\_REGISTERS\_ADDRESS, INPUT\_REGISTERS\_NB);

if (mb\_mapping == NULL) {

fprintf(stderr, "Failed to allocate the mapping: %s\n",

modbus\_strerror(errno));

modbus\_free(ctx);

return -1;

}

-------------------

SEE ALSO

--------

linkmb:modbus\_mapping\_new[3]

linkmb:modbus\_mapping\_free[3]

AUTHORS

-------

The libmodbus documentation was written by Stéphane Raimbault

<stephane.raimbault@gmail.com>