modbus\_read\_registers(3)

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

NAME

----

modbus\_read\_registers - read many registers

SYNOPSIS

--------

\*int modbus\_read\_registers(modbus\_t \*'ctx', int 'addr', int 'nb', uint16\_t \*'dest');\*

DESCRIPTION

-----------

The \*modbus\_read\_registers()\* function shall read the content of the \_nb\_

holding registers to the address \_addr\_ of the remote device. The result of

reading is stored in \_dest\_ array as word values (16 bits).

You must take care to allocate enough memory to store the results in \_dest\_

(at least \_nb\_ \* sizeof(uint16\_t)).

The function uses the Modbus function code 0x03 (read holding registers).

RETURN VALUE

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

The function shall return the number of read registers

if successful. Otherwise it shall return -1 and set errno.

ERRORS

------

\*EMBMDATA\*::

Too many registers requested

EXAMPLE

-------

[source,c]

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

modbus\_t \*ctx;

uint16\_t tab\_reg[64];

int rc;

int i;

ctx = modbus\_new\_tcp("127.0.0.1", 1502);

if (modbus\_connect(ctx) == -1) {

fprintf(stderr, "Connection failed: %s\n", modbus\_strerror(errno));

modbus\_free(ctx);

return -1;

}

rc = modbus\_read\_registers(ctx, 0, 10, tab\_reg);

if (rc == -1) {

fprintf(stderr, "%s\n", modbus\_strerror(errno));

return -1;

}

for (i=0; i < rc; i++) {

printf("reg[%d]=%d (0x%X)\n", i, tab\_reg[i], tab\_reg[i]);

}

modbus\_close(ctx);

modbus\_free(ctx);

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

SEE ALSO

--------

linkmb:modbus\_write\_register[3]

linkmb:modbus\_write\_registers[3]

AUTHORS

-------

The libmodbus documentation was written by Stéphane Raimbault

<stephane.raimbault@gmail.com>