



Communication library SPIDCOMM

Library includes protocols for communication between the PC and the controllers of SPID Elektronik.

GENERAL	3
ERROR LIST	3
0 – MD_NO_ERROR	3
1 – MD_WRONG_PARAMETERS	3
2 – MD_COM_PORT_NOT_OPEN	3
3 – MD_DATA_SEND_ERROR	3
4 – MD_DATA_RECV_ERROR	3
5 – MD_HOST_NOT_EXISTS	3
6 – MD_NOT_READY	3
7 – MD_BAD_RESPONSE_FROM_DEVICE	3
-1 – MD_UNEXPECTED_ERROR	3
STRUCTURES	4
TMD_DEVICE_ID	4
TMD_APP_ID	5
TMD_LOGIN_PARAMS	6
FUNCTION LIST	7
MD_COM_INIT	7
MD_ETH_INIT	8
MD_FREE	9
MD_LOGIN	10
MD_LOGOUT	10
MD_STOP_ALL_ROTORS	10
MD_STOP_ROTOR	10
NOTES	10

General

Library includes protocols for communication between the PC and the controllers of **SPID Elektronik** as **MD01**, **ROT1PROG** and **ROT2PROG** via RS232 and Ethernet (**MD01** only).

All strings in parameters are AnsiStrings.

Error list

0 – MD_NO_ERROR

Function succeeds.

1 – MD_WRONG_PARAMETERS

Wrong parameters passed to function.

2 – MD_COM_PORT_NOT_OPEN

Selected COM port not opened.

3 – MD_DATA_SEND_ERROR

Data could not be sending.

4 – MD_DATA_RECV_ERROR

No data received. The device does not respond.

5 – MD_HOST_NOT_EXISTS

The server does not exist.

6 – MD_NOT_READY

The device is not ready.

7 – MD_BAD_RESPONSE_FROM_DEVICE

Bad response from device: wrong check sum, wrong answer.

-1 – MD_UNEXPECTED_ERROR

There was an unexpected error.

Structures

TMD_DEVICE_ID

Description

All commands to the MD01 include this structure.

Delphi (Pascal) syntax

```
PMD_DEVICE_ID = ^TMD_DEVICE_ID;  
TMD_DEVICE_ID = record  
    group: word;  
    number: word;  
end;
```

C++ syntax

```
struct TMD_DEVICE_ID {  
    unsigned short group;  
    unsigned short number;  
};
```

Fields

group

Device group. If group is equal 0, then command is received from all devices in all groups with number equal to the field "*number*".

number

Device number in group. If number is equal 0, then command is received from all devices in this group.

TMD_APP_ID

Description

Present as parameter in all commands, that change the device settings (include changes the angles of rotors). Generate with "*md_new_app_id*" function.

Delphi (Pascal) syntax

```
PMD_APP_ID = ^TMD_APP_ID;  
TMD_APP_ID = record  
    hiPart: cardinal;  
    loPart: cardinal;  
end;
```

C++ syntax

```
struct TMD_DEVICE_ID {  
    unsigned int hiPart;  
    unsigned int loPart;  
};
```

TMD_LOGIN_PARAMS

Description

This structure is used by functions MD_LOGIN and MD_LOGOUT as parameter.

Delphi (Pascal) syntax

```
PMD_LOGIN_PARAMS = ^TMD_LOGIN_PARAMS;  
TMD_LOGIN_PARAMS = record  
    dev: TMD_DEVICE_ID;  
    app: TMD_APP_ID;  
end;
```

C++ syntax

```
struct TMD_DEVICE_ID {  
    TMD_DEVICE_ID dev;  
    TMD_APP_ID     app;  
};
```

Fields

dev

Device ID to which you want to login or logout.

app

The application ID generated with “*md_new_app_id*” function.

Function list

MD_COM_INIT

Description

Prepare MD protocol to communicate via RS232.

Delphi (Pascal) syntax

```
function md_com_init(  
    aComNumber,  
    aBaudrate: integer  
): integer; stdcall;
```

C++ syntax

```
int __stdcall md_com_init(  
    int aComNumber  
    , int aBaudrate  
);
```

Parameters

aComNumber [in]

COM port number (1..255)

aBaudrate [in]

COM port communication speed (600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400¹, 460800²)

Return value

If the function succeeds, the return value is **MD_NO_ERROR**.

If the function fails, possible errors include the following:

- MD_BAD_PARAMETERS,
- MD_COM_PORT_NOT_OPENED,
- MD_UNEXPECTED_ERROR.

¹ Virtual SPID USB COM port only

² Virtual SPID USB COM port only

MD_ETH_INIT

Description

Prepare MD protocol to communicate via ETHERNET (TCP/IP).

Delphi (Pascal) syntax

```
function md_eth_init(  
    aHost: pChar;  
    aPort: integer  
): integer; stdcall;
```

C++ syntax

```
int __stdcall md_eth_init(  
    char *aHost  
    , int aPort  
);
```

Parameters

aHost [in]

Host name or IP address.

aPort [in]

Port number - default port number in MD01 controller is 23.

Return value

If the function succeeds, the return value is **MD_NO_ERROR**.

If the function fails, possible errors include the following:

- MD_BAD_PARAMETERS,
- MD_HOST_NOT_EXISTS,
- MD_UNEXPECTED_ERROR.

MD_FREE

Description

Close all communication port - ETHERNET and COM port.

Delphi (Pascal) syntax

```
function md_free: integer; stdcall;
```

C++ syntax

```
int __stdcall md_eth_init(void);
```

Parameters

No parameters.

Return value

If the function succeeds, the return value is **MD_NO_ERROR**.

If the function fails, possible errors include the following:

- MD_BAD_PARAMETERS,
- MD_UNEXPECTED_ERROR.

MD_LOGIN

Description

Syntax

Parameters

Return value

MD_LOGOUT

Description

Syntax

Parameters

Return value

MD_STOP_ALL_ROTORS

Description

Syntax

Parameters

Return value

MD_STOP ROTOR

Description

Syntax

Parameters

Return value

Notes
