

Taskit Modbus Library

TML V2.02

Overview

The Taskit Modbus Library provides an open source, lightweight and easily portable framework supporting Modbus ASCII as well as Modbus RTU protocol. It was designed to guarantee access to the gpio.NET modules made by taskit GmbH.

The library splits into two parts – a client side (Modbus Master) which creates requests and cares for the received answer packets; and a device side that can be used for creating your own Modbus devices.

This document is targeted at the client part. Much of the information given here also applies to the device side as well.

Providing Input/Output functionality

As said, TML is a cross-platform library. To keep the structure and development of the library simple, there are no – or almost no – code providing platform specific code for hardware support. The whole code is prescinded to work on input characters, output characters and some timing information.

The library expects the program to provide user-specific functions for reading/writing bytes and for timing issues such as getting the number of milli- or nanoseconds.

The user-specific functions are (defined in `inc/tml-def.h`):

```
tml_user_nb_read,  
tml_user_nb_write,  
tml_user_get_msec,  
tml_user_get_usec,  
tml_user_pause,  
tml_user_rtu_frame_end,  
tml_user_setup_rtu
```

Most of them are quite simple to implement. Useful implementations for Linux and Windows can be found at `sys/*.c`.

Simple Example

An example writing digital data to a gpio.DIO or gpio.Relais module is located at `example/io-test/io-test.c`.

User defined information in struct `tml_conn_t`

Each library function uses a `tml_conn_t` struct to distinguish between different connections. This allows the application to handle transactions from and to several sources. Sometimes it is required to store some information needed to access e.g. files or sockets.

The sample programs and the basic implementations of the user-specific functions handle this by passing a pointer to a file descriptor.

The `tml_conn_t` struct defines the void pointer `user_data` for such purpose. This struct also selects the whether the connection is RS232 or RS485, the timeout used a data buffer allocated by the user (see example) and the Modbus mode (RTU or ASCII). It is defined in header `inc/tml-def.h`.