

# Documentazione libreria Helper\_TRDP

DEFINIZIONE VARIABILI, OGGETTI, FUNZIONI E METODOLOGIE

Luca Gargiulo | Mediamobile | 12/11/2020

## Informazioni generali sulla libreria

Lo scopo della libreria è quello di creare uno standard per la gestione delle comunicazioni basate sul TRDP dal C, orientate a oggetti, semplificate e rielaborate per il C++. La libreria sfrutta le funzioni già preesistenti TCNOpen<sup>[1]</sup> concepite per il C. Helper\_TRDP non è una libreria di fatto, cioè un namespace, ma è stata realizzata e lo è ideologicamente.

## File della libreria

La libreria è composta da un oggetto header:

*TRDP\_Handler.h*

Un header per funzioni:

*funzioni.h*

## Criteri funzionamento TRDP\_Handler.h

La classe TRDP\_Handler.h è utilizzata per l'inizializzazione della connessione, il settaggio di variabili per la sua riuscita e l'invio dei messaggi. Manca una funzione di ricezione dei messaggi, questo perché la classe non è pensata per essere la soluzione finale per la gestione dei dati TRDP, bensì solo una classe da cui ereditare attributi e funzioni in cui aggiungere una funzione di ricezione in base ai criteri ritenuti necessari dipendentemente dal suo scopo.

La classe è mirata alla gestione dei dati da un solo indirizzo IP, ergo se si vogliono gestire X connessioni differenti, si potranno usufruire di X istanze di classe differenti.

Ogni messaggio, ricevuto o mandato, ha un primo byte che ne identifica la natura, il tutto gestito automaticamente dalla classe.

È necessario specificare per ogni oggetto il tipo di protocollo da utilizzare per la sola ricezione di messaggi, ciò significa che si può ricevere solo dai messaggi con il primo byte dichiarato, ma si può inviare un messaggio con il primo byte di qualsiasi tipo si voglia.

## Invio dei pacchetti sulla rete

Un dato da inviare sulla rete, usufruendo del protocollo TRDP, può contenere un numero elevato di informazioni di diversa origine. Esempio, si vuol poter inviare in un unico pacchetto X numeri, Y stringhe, K chars. È chiaro che nel pacchetto c'è la possibilità che una delle informazioni possa contenere un byte uguale a zero, valore che starebbe ad indicare la fine del pacchetto, quando questo sarebbe del tutto involontario. Per ovviare al problema bisogna convertire a dovere i diversi dati da integrare nel pacchetto da spedire. Le funzioni sono definite nell'header *funzioni.h*.

## Criteri funzionamento della comunicazione

Per l'integrazione di un intero nel pacchetto da inviare, sono necessarie manipolazioni per convertirlo in un array di char l'array do char a sua volta deve essere strutturato in modo tale da evitare gli 0 in ogni byte del pacchetto (Protocollo NOCHARARRAY, spiegato più avanti).

Per l'integrazione di una stringa/char ogni byte di valore 0 viene convertito in 1.

Ogni conversione subisce una conversione inversa alla ricezione del messaggio.

È necessario chiamare le funzioni per la conversione manualmente ogni qualvolta si deve mandare un messaggio.

Maggiori documentazioni la si può trovare nel file *funzioni.h*

## Protocollo NO0CHARARRAY

Ogni byte di informazione utile deve essere preceduto da un altro byte di controllo che indicherà se il byte successivo ha valore zero. Il byte che precede è 1 se indica che il prossimo byte ha valore 0, 2 se il byte ha valore proprio.

Ex:

char array da inviare: 231,0,32,0

Risultato: 2,231 / 1,1 / 2,32 / 1,1

## Riferimenti

1) Fonte: <http://www.tcnopen.eu/>