

TCP Client/Server

from tcpcom import *
TCPServer

server = TCPServer (port, port, stateChanged, isVerbose = False)	crée une requête TCP pour se connecter avec un client avec le port port (entier). Les changements d'état appellent stateChanged (). Pour isVerbose = True, les messages de débogage sont écrits dans la fenêtre de sortie.
stateChanged (state, msg)	Rappel, qui est appelé lors d'un changement d'état. state: TCPServer.PORT_IN_USE, msg: port state: TCPServer.CONNECTED, msg: adresse IP du client state: TCPServer.LISTENING, msg: vide state: TCPServer.TERMINATED, msg: vide state: TCPServer.MESSAGE, msg: Message reçu du client (String)
server.isConnected ()	True si le client est connecté au serveur
server.terminate()	Clôture de la connexion, fin de l'état LISTENING. Le port IP est libéré.
server.isTerminated ()	renvoie True si le serveur est dans l'état TERMINATED
server.isConnected ()	renvoie True si le client est connecté à un serveur
server.disconnect ()	met fin à la connexion au serveur. state=LISTENING
server.sendMessage (msg)	envoie les informations au client (string, le caractère \ 0 (ASCII 0) sert indique la fin du msg, automatiquement ajouté et supprimé de manière transparente).
TCPServer.getVersion ()	renvoie la version du module (String)

TCPClient

client = TCPClient (ipAddress, port, stateChanged, isVerbose = False)	crée une requête TCP pour se connecter avec un serveur TCP avec l'adresse IP ipAddress (String) et le port (entier). Les changements d'état appellent stateChanged (). Pour isVerbose = True, les messages de débogage sont écrits dans la fenêtre de sortie.
stateChanged (state, msg)	Rappel, qui est appelé lors d'un changement d'état. state: TCPClient.CONNECTING, msg: adresse IP du serveur state: TCPClient.CONNECTION_FAILED, msg: adresse IP du serveur state: TCPClient.CONNECTED, msg: adresse IP du serveur state: TCPClient.SERVER_OCCUPIED, msg: adresse IP du serveur state: TCPClient.DISCONNECTED, msg: vide state: TCPClient.MESSAGE, msg: message reçu du serveur (String)
client.connect ()	crée une connexion au serveur (bloquant jusqu'à expiration du délai). Renvoie True si la connexion a été établie, sinon False sera renvoyé.
client.connect (timeout)	idem, mais en spécifiant le délai d'attente (en s) pour la tentative de connexion
client.isConnecting ()	renvoie True lorsque le client tente de se connecter
client.isConnected ()	renvoie True si le client est connecté à un serveur
client.disconnect ()	met fin à la connexion au serveur
client.sendMessage (msg, responseTime)	envoie les informations au serveur (string, le caractère \ 0 (ASCII 0) sert indique la fin du msg, automatiquement ajouté et supprimé de manière transparente). Pour responseTime> 0, la méthode bloque et attend une réponse dans responseTime (en s). La réponse est livrée en tant que valeur de retour. Si la réponse n'arrive pas, aucun n'est retourné
TCPClient.getVersion ()	renvoie la version du module (String)