AppIDClient application for Windows

AppIdClient for Windows is a Delphi-based application designed using IDE RAD Studio Tokyo 10.2. It presents a user-friendly interface which communicates with a nodeMCU 1.0 board via a WiFi connection. The nodeMCU board acts as a server, whereas the application is a client. With this application, IoT (internet of things) projects can be easily designed.

Before the using of this app, the sketch test\_indy.ino must be downloaded to the nodeMCU module using the Arduino IDE. This sketch is based on the library esp8266wifi.h. It configurates the internet connection using a local fixed IP address which must be previously known. This address must be a free one available by the router (see intructions in the sketch). The state of the connection and the WiFi signal strength are reported to the user via the serial monitor. In the loop routine, it waits for a client to connect, reads a command sent by the client (if no command is sent, it reads an empty string), and prints the command on the serial monitor. If the command is /LED=ON, or /LED=OFF, the builtin led is put on or off and its state is reported to the client. If the command is /STATEBUTTON, it reports the state of the builtin button to the client. If a string is introduced by the user through the serial monitor, it is sent to the client. Finally, if the command is /STOPCLIENT, it closes the connection.

The AppIdClient for Windows is based on the use of the IdTCPClient component and the IdThreadComponent. The Indy (internet direct) components are open source easily found in the internet and they are installed in the standard version of the RAD Studio IDE. These components allows to state an internet connection easily. The AppIdClient application presents a simple interface which has three buttons. The first one is a Set IP Address button, which opens a little form to introduce the host IP address. The other two buttons allows the user to connect and disconnect the client application with the nodeMCU server. The interface presents an edit box to write a string and a fourth button permitts to send this string to the server. This string can be a command as those described above or simply a sentence which will be printed on the serial monitor by the server. Messages sent by the server, as well as messages describing the state of the connection are reported in a black memo log window.

