LoRa-internet Interface

LoRa® is a **wireless modulation technique** derived from **Chirp Spread Spectrum** (CSS) technology. CSS uses wideband linear frequency modulated chirp pulses to encode information.

The goal of this project is to connect a network of two LoRa devices to internet. LoRa devices are SX1278 boards that transmitt and receive data at 433MHz using the LoRa modulation protocol. The SX1278 boards are transceivers manufactured by SEMTECH that feature the LoRa® long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption. These boards work as slave devices controlled by Arduino-type boards acting as master devices. Communication between master and slave is via SPI.

One of the LoRa terminals can be placed in a remote location without an internet connection. The master device can be any microcontroller like an Arduino board. The other terminal plays the role of an internet interface, so the master board must be able of connection to an acces point, such as an ESP8266-based board. The sketch to be loaded into the master device of the remote terminal is LoraTerminal.ino, and the sketch to be loaded into the master device of the interface terminal is LoraInternet.ino.

The sketchs loaded into the master devices use the Lora.h library, which has the functions of sending AT commands to the SX1278 board to receive and send data packets using the LoRa modulation protocol (see https://github.com/sandeepmistry/arduino-LoRa/tree/master). Communication with the SX1278 board is implemented by the SPI.h library. Connection between the master device and the SX1278 is (see details in https://how2electronics.com/lora-sx1278-esp8266-transmitter-receiver/)

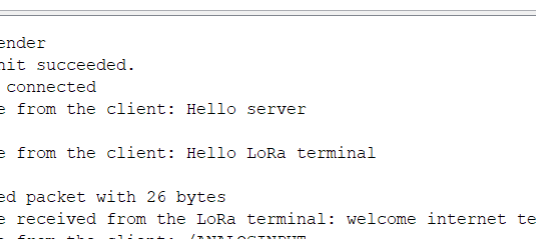| ESP8266 Pins	or Arduino pins	SX1278 Pins
GND	GND	GND
3.3V	3.3V	VCC
D8	GPIO15	NSS
D7	GPIO13	MOSI
D6	GPIO12	MISO
D5	GPIO14	SCK
D0	GPIO16	RST
D2	GPIO4	DIO0

Have into account that Arduino boards handle 5V on their IO terminals, while SX1278 works at 3.3V, so voltage levels must be adapted for communication.

Connection of the SX1278 to a nodeMCU v1 board

In this project the interface terminal is a Wemos D1 R2 board which connects to an internet access point via WiFi. It is configured as a server that waits for a connection from a client. After connection, the interface simply forwards the messages received from the client and from the LoRa remote terminal to the other device. The messages exchanged between the devices can be simply sentences, which are displayed on the serial monitor of the Arduino IDE in the case that the remote LoRa terminal is connected to a computer (not mandatory), or specific commands to perform an action. The commands considererd in this project are

/LED=ON : puts on the built-in led and sends the message "led is on"
/LED=OFF : puts off the built-in led and sends the message "led is off"
/STATEBUTTON   : sends the messages "button pressed" or "button unpressed" according to the state of the built-in flash button.
/ANALOGINPUT : sends the message "analog input=xxx", where xxx is the value of the analog input (0-1023).
/STOPCLIENT : message sent by the internet client to the interface to stop the internet connection. This command is not sent to the remote terminal.
The client in this project is implemented through a VCL application which uses Indy components to connect to the server as a client (see the project https://github.com/quini27/IdClientApp). Of course, other internet communication configurations can be implemented. For example, the interface terminal can connect as a client to a MQTT broker, which delivers received messages to other clients subcribed to specific topics, or it may serve an interactive web page available through any browser.

Serial monitor of the LoRa terminal

Serial monitor of the LoRa-Internet interface
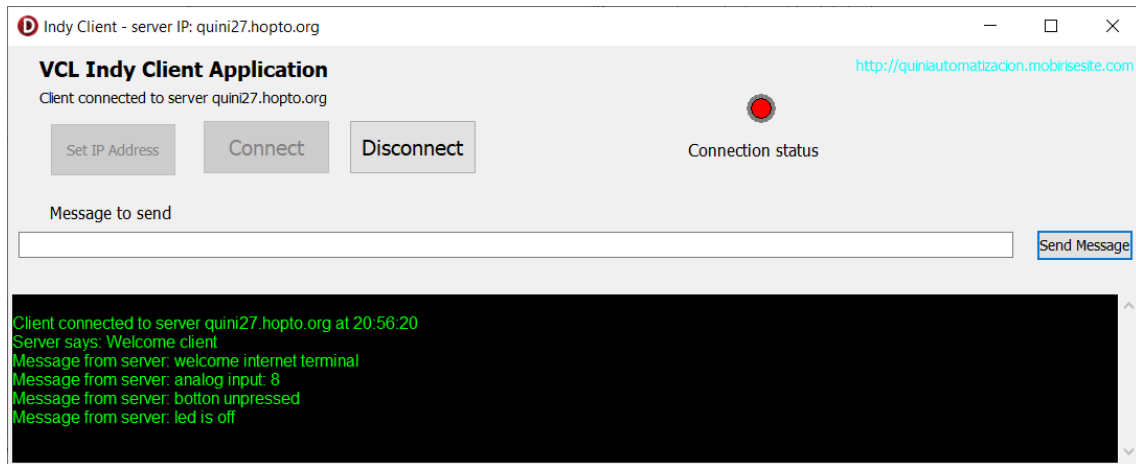
Indy Client VCL Application