Standalone IOT sensor system

System Overview

The objective of the system is to provide a complete standalone sensor telemetry system using the LoRa radio technology which is totally independent of third party (network) software. In this project an EU compliant 868 MHz LoRa module was used but can obviously be adapted for your specific region.

The components of the system are:

1. Node - a battery powered which takes readings from the connected sensors and transmits them over the air-interface to the gateway using the LoRa system.

2. Gateway - connects to the node using the LoRa system and to the web based User Interface using WiFi.

3. Web based User Interface and Data Base.

A functional diagram of the system is shown in Fig. 1 which also serves as a point of reference.

Hardware

The end node uses a bare bones ATM328P microcontroller and an RFM95W LoRa module, which consists of the Semtech SX1276 Lora transceiver IC with ancillary components such as voltage regulator and crystal.
The bootloader and code can be uploaded using the Arduino IDE and a separate Arduino board which is well documented elsewhere. Simply ensure that the internal 8MHz RC oscillator is selected from the bootloader options so that no crystal or any other components are required, except for decoupling capacitors.
See Fig. 2 for the circuit diagram.
The sensors used here are a DHT22 temperature and humidity device, which should be useful for most applications and a simple potentiometer connected via an ADC input, but this could be substituted for any sensor which provides an analog output.
Provision is made for downlink control of a (green) LED, although again this could be substituted by a relay for example, to permit control of mains powered equipment. The red LED was used for debugging purposes and may be omitted if not required.
There are numerous spare GPIO and ADC pins to facilitate the addition of extra sensors, and the software has been written to easily accommodate the extra data generated.

The Gateway uses an ESP32 Dev Module and a RFM95W module which can simply be 'piggy-backed' via the SPI interface wiring. Again, the only additional components are the decoupling capacitors. The circuit diagram is shown in Fig. 3.

Software

The Arduino IDE is used for developing all micrcontroller software and the ESP32 framework for Arduino must first be installed which is amply described elsewhere.
The node requires the following additional libraries:
SPI, included with IDE, LowPower from Arduino, LoRa from Sandeep Mistry, and DHT and Unified Sensor Library from

Adafruit.
The gateway requires the following additional libraries:
SPI, included with IDE, WiFi library from Arduino and included with ESP32 framework from Arduino, LoRa by Sandeep Mistry, and HttpClient by Adrian McEwen.
The code itself is well commented, and should be self-explanatory.

Web Files

The web based User Interface consists of a simple MySQL Data Base to store sensor values and 3 functional files:

a. Index.php
Renders the main user screen and displays sensor values retrieved from the DB.
A slider switch permits the user to control the remote LED at the node.

b. Loaralink.php. Provides the panel permits the user to read any of the registers located in the remote SX1276 LoRa chip located on the node, and is useful for debugging and optimisation purposes.

c. Webcomms.php. This file does not display any data but is the communication link with the gateway. Each sensor value is received via a dedicated POST request, and the message echoed back to the gateway for confirmation. The received sensor value is then stored in the DB.
Using the same process the gateway interrogates for user commands, and the command data is echoed back to gateway for transfer to the node.

The code is heavily commented and with the help of Fig.1 should be easy to understand.

Operational notes

Keep the connections for the SPI interface between the microcontroller and LoRa module as short as possible to avoid pickup.
If possible, incorporate a ground plane under the LoRa module to improve effective range and reliability.
Ensure decoupling capacitors are used.
A simple wire antenna of length 8.6 cm will suffice, but greater range will be obtained with a dipole antenna.
Bear in mind the node always initiates the data transfer and is in low-power mode as much as possible. Both index.php and loralink.php use page refresh timers which are independent. It is likely that the node loop will need to execute twice before the LoRa module registers are updated in the UI.

Credit
The database and http communication features are based on an article originally published by Elecronoobs.com

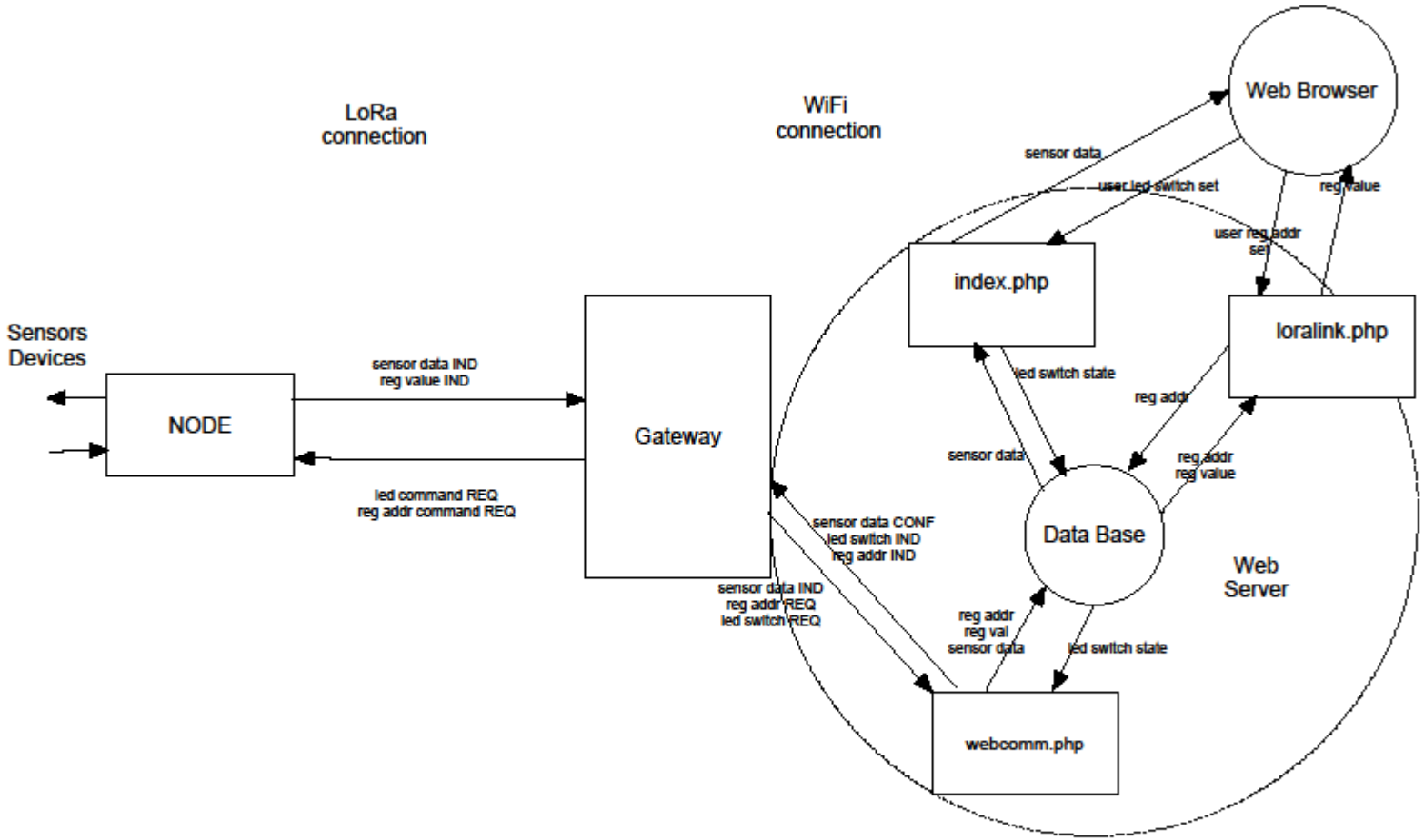

Fig 1. System Functional Diagram

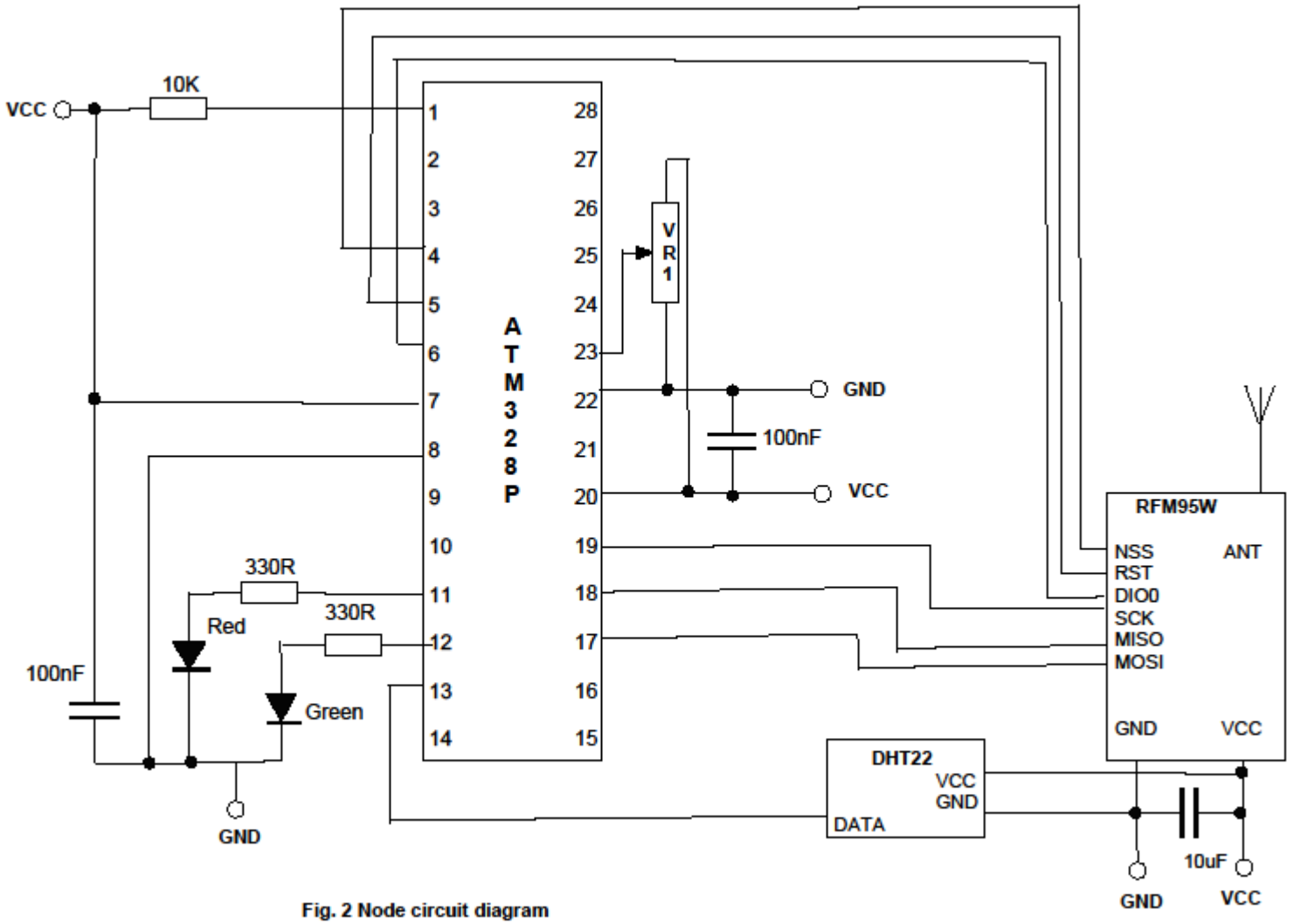

Fig. 2 Node circuit diagram

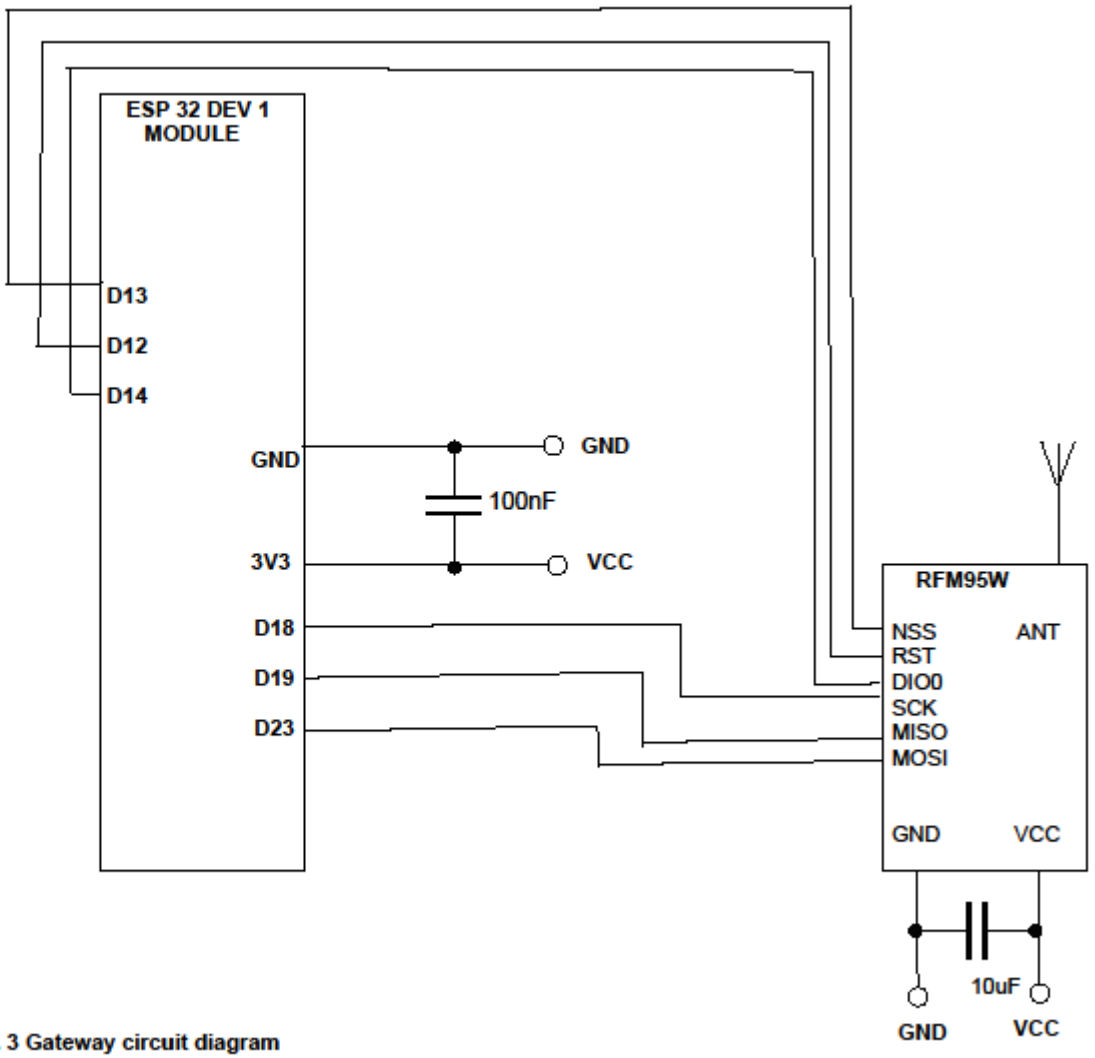

Fig. 3 Gateway circuit diagram

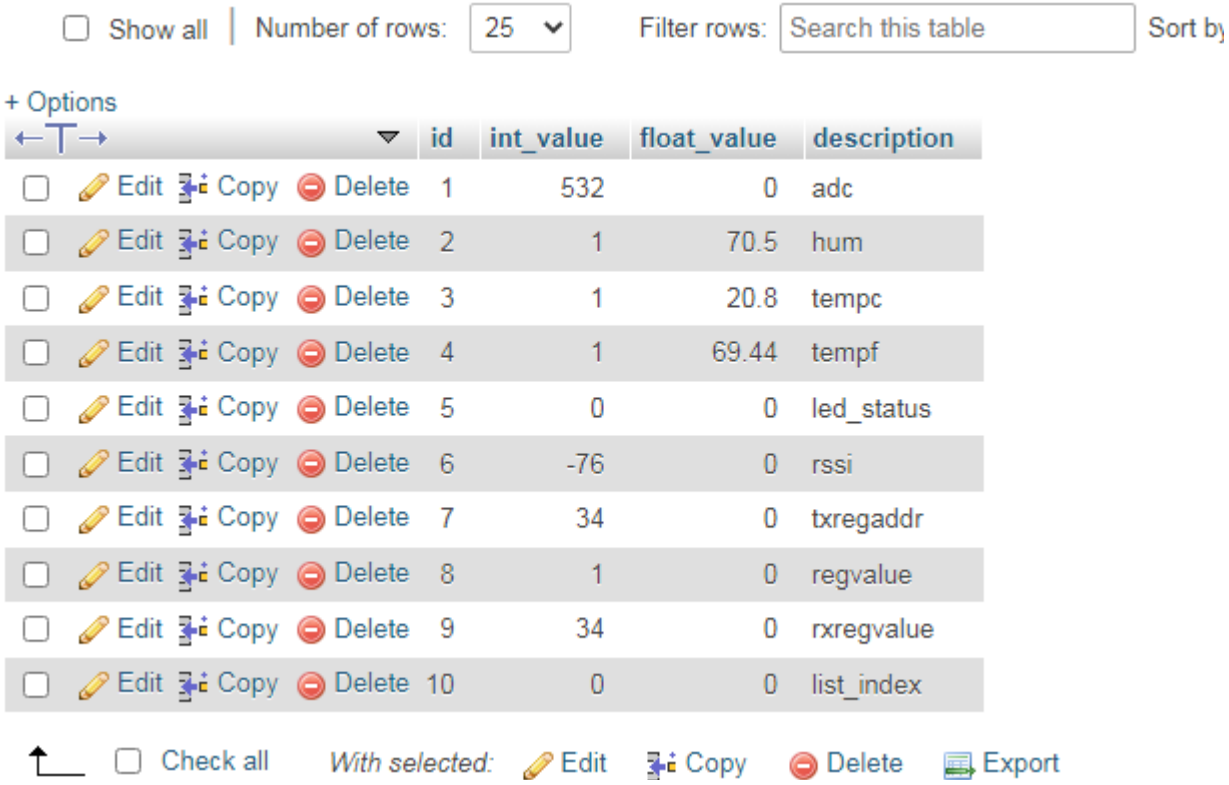

Fig 4. Example of database