IoT TRAP debug tool

TRAP debug tool for Transiently-Powered sensing systems

Student
Matteo Spadetto
[214352]

Battery-less sensors network

With **battery-less wireless sensors** the communication is guaranteed only if, in the transmission instant, both the sending node and the receiving one have **enough energy** to share data between each other

TRAP approach

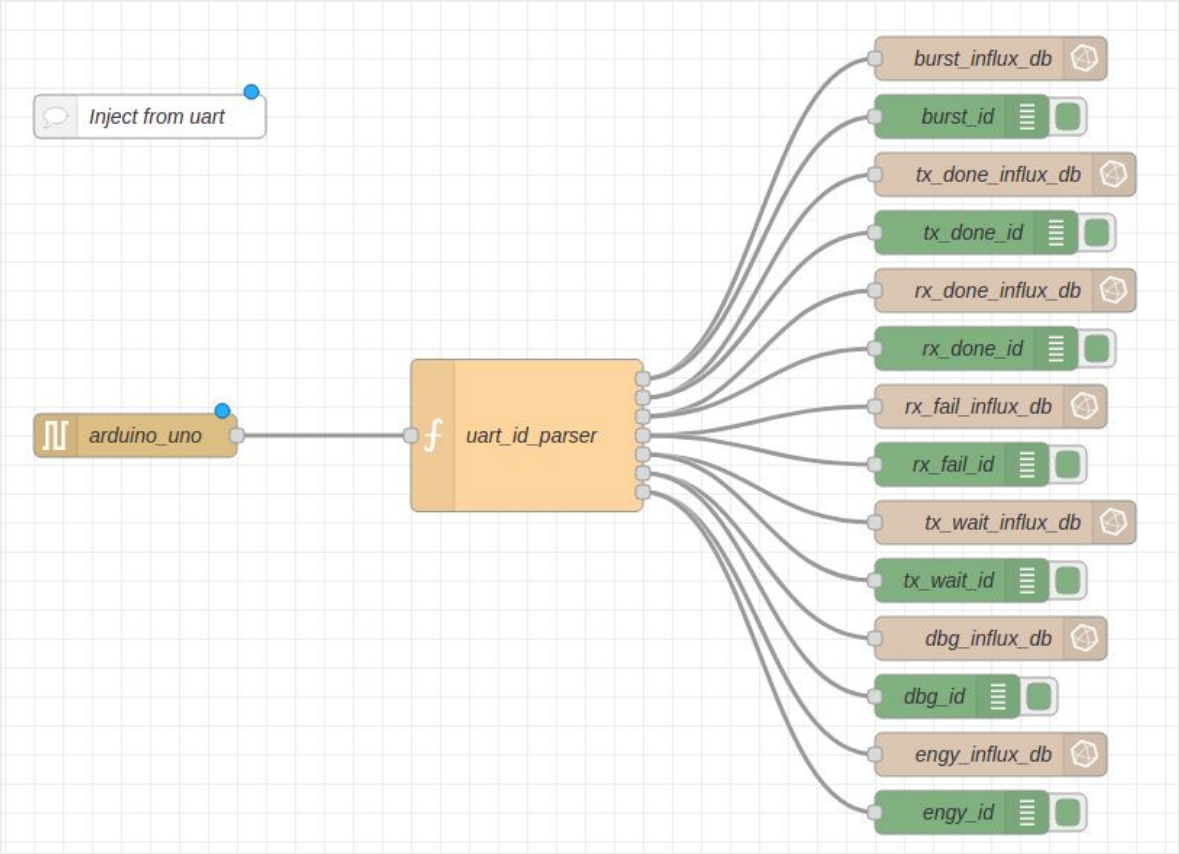
Thanks to the energy status information, **TRAP** verifies that both sender and receiver have enough energy before the transmission starts in order to ensure a successful communication **without loss of data**.

Why the TRAP debug tool

Several sensors are connected in the same **network** so, if there is not a proper tool, it is not so easy to **test** the TRAP protocol itself.
The key words are: **monitoring, remote and scalable**

Basic test

Design an entire application, acquiring from an **Arduino UNO**, parsing and storing messages with **Node-RED** and showing the data saved in the **InfluxDB** database thanks to the **Grafana** visualiser



InfluxDB database	iot_tool						
Mearuements	engy	burst	tx_done	rx_done	rx_fail	tx_wait	debug
Fields	engy_0	burst_0	tx_done_0	rx_done_0	rx_fail_0	tx_wait_0	debug_0

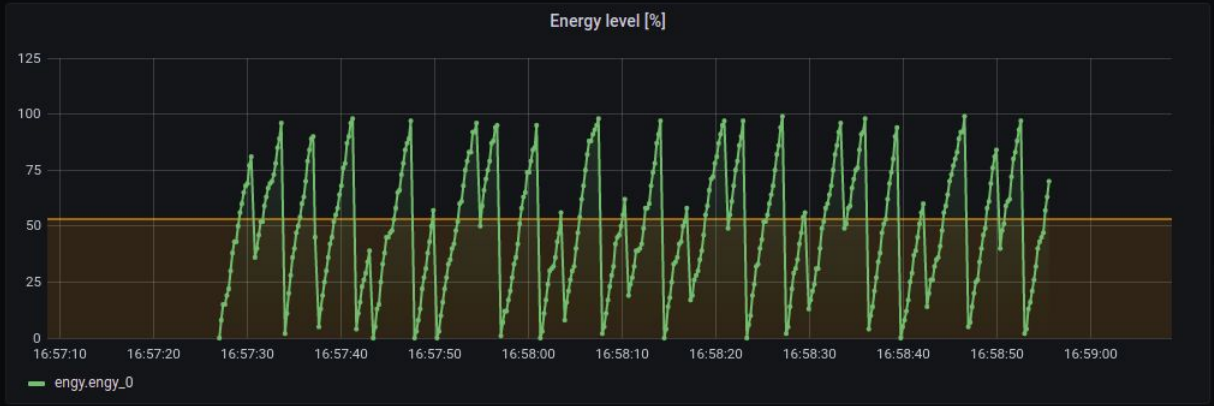
General / Iot_TRAP_08/06/2021_16:52:07

Last 2 minutes | 5s

Total statistics

total_tx_done	total_rx_done	total_tx_wait	total_rx_fail	total_burst	tot_dbg
13	12	No data	1	No data	No data

Device_0



tx_done	rx_done
13	12

tx_wait	rx_fail
No data	1

burst	dbg
No data	No data

Scalability level

Several sensors should be tested simultaneously, for this reason, an upgrade to **increase scalability** is necessary.
In TRAP debug tool, the **“constructor.py”** and **“manage_influx_db.sh”** scripts were implemented to be able to **automatically** add new devices to Node-RED, InfluxDB and Grafana with one command

Constructor.py

Python3 script responsible to add new devices to Node-RED and Grafana json files.

```
/usr/bin/python3 constructor.py  '/dev/ttyACM3 /dev/ttyACM4'
```

manage_influx_db.sh

Linux **bash** script that manages the entire application:

```
./manage_influx_db.sh -r '/dev/ttyACM3 /dev/ttyACM4'
./manage_influx_db.sh -w /path/to/folder/
```

Remotely?

The TRAP debug tool can run on every **Linux based OS**, also remotely, just importing all the necessary files and dependencies. Then, when the environment is set, just **ssh** commands are needed and the Grafana output is reachable on **port 3000** of the IP address of the remote device.

Thank you for your attention

TRAP debug tool for Transiently-Powered sensing systems

Student
Matteo Spadetto
[214352]