

# Obiekty Internetu Rzeczy

(Laboratorium 3)

Krzysztof Pierczyk

grudzień 2020

## 1 Wstęp

Ostatnie z zadań laboratoryjnych polegało na zwizualizowaniu danych pomiarowych w środowisku **Node-RED**. Dane pochodziły z symulowanych przez program `/opt/obir_services/obir_exe` węzłów IoT komunikujących się za pośrednictwem brokera MQTT działającego na maszynie wirtualnej. Po skonfigurowaniu i uruchomieniu maszyny wirtualnej (MAC interfejsu sieciowego: 080027bb8047) możliwe było odczytanie danych z czterech dostępnych tematów:

- `sensor/3338/temperature` (zakres wartości od 32 do 117)
- `sensor/3338/humidity` (zakres wartości od 16 do 62)
- `sensor/3338/light` (zakres wartości od 12 do 71)
- `sensor/8735/light` (zakres wartości od 18 do 53)

## 2 Opis rozwiązania

Pierwszym krokiem było skonfigurowanie odpowiedniego przepływu (ang. *Flow*) z wykorzystaniem przeglądarkowego interfejsu serwera Node-RED. Każdy z dostępnych tematów otrzymał własny węzeł subskrybujący. Otrzymywane dane przychodziły w formie bufora znaków ASCII, którego jednym z elementów był zapis faktycznej wartości pomiarowej. Aby go wyekstrahować i przekonwertować do wartości liczbowej użyto bloków `function` skonfigurowanego z poniższym kodem języka JavaScript. Każdy z węzłów subskrybujących otrzymał własną blok konwertujący.

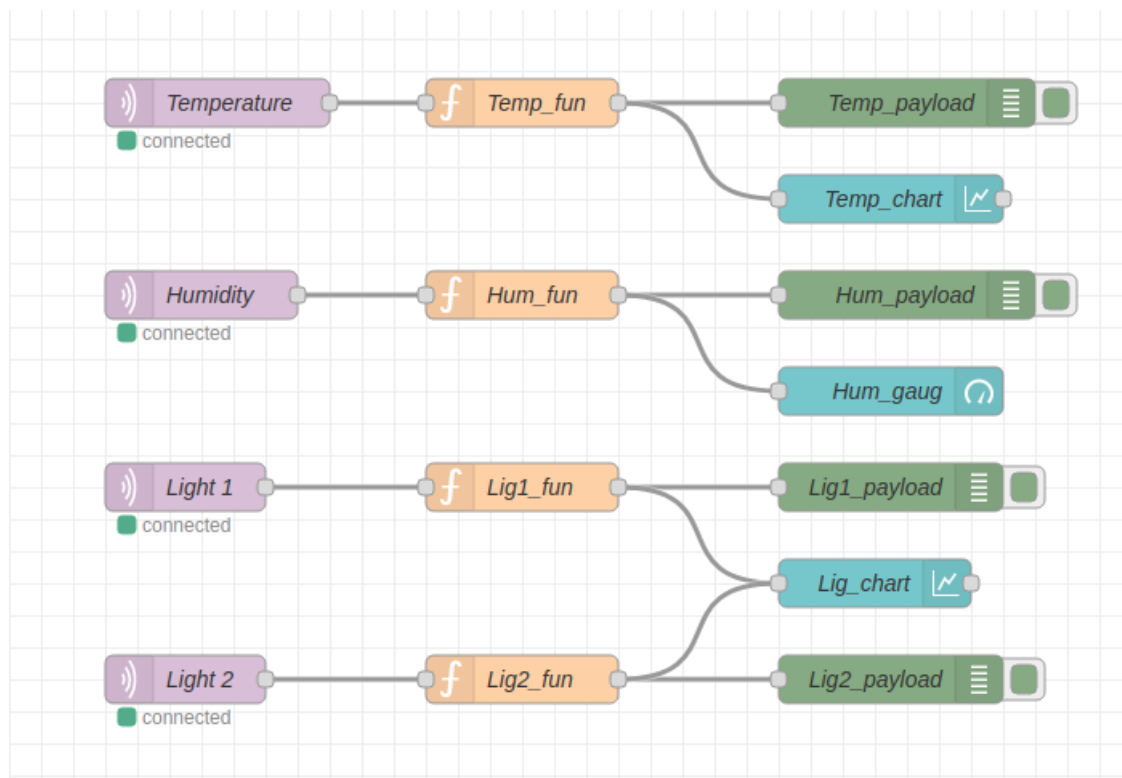
---

```
1 var t=msg.payload.split("{")[1].split("}") [0].split(",") [2].split(":") [1].trim();
2 msg.payload=parseInt(t);
3 msg.topic="Human-friendly topic name"
4 return msg;
```

---

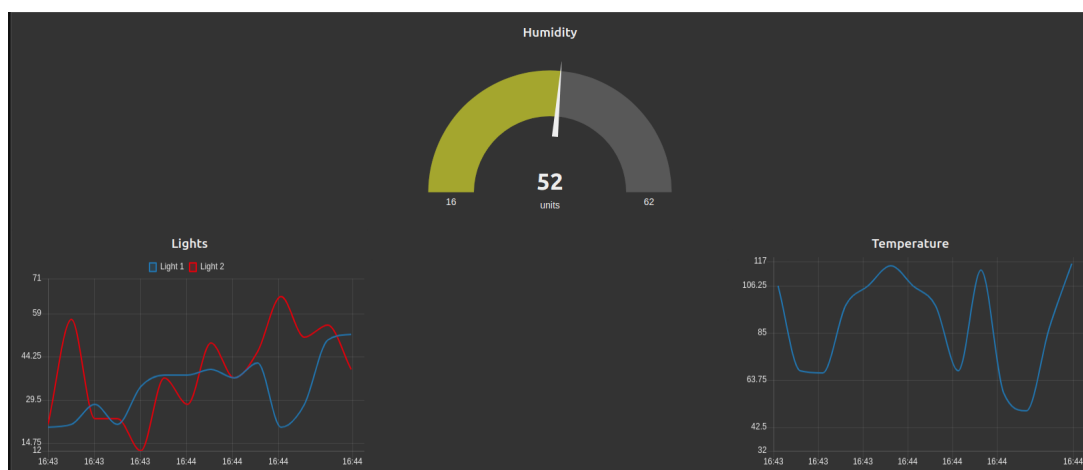
W celu weryfikacji poprawności wprowadzonego kodu wyjście każdego z bloków funkcyjnych zostało wyprowadzone do węzła `debug`, który umożliwia obserwację danych wejściowych w konsoli. Ostatnim krokiem było przygotowanie wizualizacji danych w panelu **NodeRED Dashboard**. Posłużyły do tego trzy bloki: dwa typu `chart` oraz jeden typu `gauge`. Wyjście tematu zawierającego dane o temperaturze zostało podłączone do wejścia pierwszego z wykresów. Wejścia drugiego zestawiono z kolei z wyjściami linii przepływów przetwarzających dane z tematów `*/light`. Ostatnia linia przepływu (skojarzona z tematem `*/humidity`) została skierowana do bloku `gauge`. W ten sposób otrzymano diagram przepływu widoczny na Rys.1.

We wszystkich blokach wizualizacyjnych skonfigurowano także granice wartości zgodne z opisanymi w treści zadania. Ponadto przypisano im również odpowiednie tytuły. Wykresy sformatowano w ten



Rysunek 1: Finalny diagram przepływu środowiska Node-Red

sposób, aby pokazywały one ostatnie 30 próbek, przy czym wartości osi X przyjęto w formacie *HH:mm*. Zastosowano także monotoniczną interpolację wielomianem trzeciego stopnia, aby poprawić wygląd panelu. Należy podkreślić, że dopuszczalność takiego rozwiązania zależy silnie od charakteru obserwowanych danych. Po dodaniu odpowiednich bloków typu **spacer** w menu konfiguracyjnym **Dashboard** otrzymany rezultat ukazany na Rys.2.



Rysunek 2: Panel Dashboard środowiska Node-Red

### 3 Podsumowanie

Aplikacja systemu Node-RED przedstawiona w niniejszym sprawozdaniu należy z pewnością do gatunku trywialnych. Mimo to pokazuje ona w jak szybki i intuicyjny sposób możliwe jest monitoro-

wanie, analizowanie oraz sterowanie pracą wielu działających niezależnie węzłów IoT. Prosty interfejs, dostępność wielu kluczowych bloków przetwarzania, elastyczny panel wizualizacji oraz szerokie zaangażowanie społeczności w rozwijanie możliwości systemu sprawiają, że jest on wyborem odpowiednim zarówno dla prostych rozwiązań typu Smar-Home jak i dla bardziej złożonych projektów. Ponadto możliwość skonfigurowania lokalnego serwera na maszynie klasy PC lub nawet Rasperry-Pi znacząco obniżają próg wejścia dla nowych użytkowników. Osobiście uważam, że Node-RED stanie się moim pierwszym wyborem w przypadku większości projektów DIY zastępując tym samym rozwiązania takie jak OpenHAB czy Home Assistant.