

Centrum Kształcenia Zawodowego i Ustawicznego nr 2
“ELEKTRONIK” w Zielonej Górze

PRACA KONKURSOWA 2022

Inteligentny monitor zużycia energii w oparciu o rozwiązania IoT dostępny w smartfonie i komputerze.



Konsultacja merytoryczna
mgr Mariusz Kmiećkowiak

Wykonawcy
Krystian Wybranowski kl. 4Tc
Jan Szymański kl. 2Td

Wstęp: IoT- Internet Rzeczy	5
Czym jest IoT?	5
Założenia projektu	6
Motywacja	6
Światowa potrzeba matką wynalazków	6
Podwyżki, podwyżki... i jeszcze raz podwyżki!	8
Cele	9
Przystępność	9
Świadomość	9
Historia danych	10
Rozszerzalność	10
Opis projektu	10
Sprzęt - tani, ogólnodostępny, rozszerzalny	10
Wstęp: jakiego sprzętu używamy?	10
Czujnik prądu PZEM-004T	12
Mikrokontroler ESP8266 rozszerzony o Shield WEMOS D1 mini	14
Co wyróżnia ten mikrokontroler?	15
Wysoka trwałość	15
Kompaktowość i wysoka integracja	15
Oszczędna architektura	16
WEMOS	16
Precyzyjna przetwornica AC 220 V → 5 V	16
Oprogramowanie - darmowe, open-source, światowej klasy	17
Wstęp: jakiego programowania używamy?	17
MQTT: protokół komunikacyjny	19
Czym jest MQTT?	19
Dlaczego MQTT?	19
Implementacja protokołu	20
InfluxDB: baza danych dla Internetu Rzeczy	21
Czym jest InfluxDB?	21
Dlaczego InfluxDB?	21
Przepływ danych	22
Instalacja Telegrafa na serwerze Linux	22
Instalacja bazy danych	23
Grafana: wygodna obserwacja	24
Czym jest Grafana?	24

Jak używamy Grafany?	24
Instalacja Grafany	25
Konfiguracja Grafany	25
Autorskie kwerendy InfluxDB do prezentacji danych w Grafanie	26
Zużyta energia (dzisiaj)	26
Zużyta energia (wczoraj)	26
Zużyta energia (7 dni)	26
Zużyta energia (30 dni)	26
Aktualne napięcie	26
Natężenie	26
Moc aktualna	26
Moc bierna	26
Moc pozorna	26
Tasmota: rozbudowany firmware	27
Czym jest Tasmota?	27
Dlaczego Tasmota?	27
Instalacja	27
Ubuntu Server: przyjazna dystrybucja Linuxa	28
Dlaczego Ubuntu?	28
Środowisko uruchomieniowe	28
Instalacja	29
PM2	29
Aplikacja: osobiste i dostosowane użycie	30
Co oferuje aplikacja?	30
Podgląd danych	30
Notyfikacje	30
Jak zbudowana jest aplikacja?	30
React	30
Next.js	30
Progressive Web App	31
Efekt finalny	32
Aplikacja	32
Wersja mobilna	33
Notyfikacje	34
Wersja desktopowa	35
Notyfikacje	35

Grafana	35
Zastosowanie	36
Efekt ekonomiczny	36

Wstęp: IoT- Internet Rzeczy



Czym jest IoT?

“Jeśli myślisz, że Internet zmienił Twoje życie,
pomyśl jeszcze raz. **IoT wkrótce zmieni je na
nowo!**” – Brendan O’Brien

IoT (Internet Rzeczy, Internet of Things) opisuje sieć fizycznych rzeczy, zaczynając od **zwyczajnych, domowych** urządzeń do **przemysłowej** klasy systemów. Urządzenia IoT te, w porównaniu do standardowego typu urządzeń, wyposażone są w **sensory, oprogramowanie** oraz inne **technologie**, umożliwiające zdobywanie danych.

Założenia projektu

Motywacja

Światowa potrzeba matką wynalazków

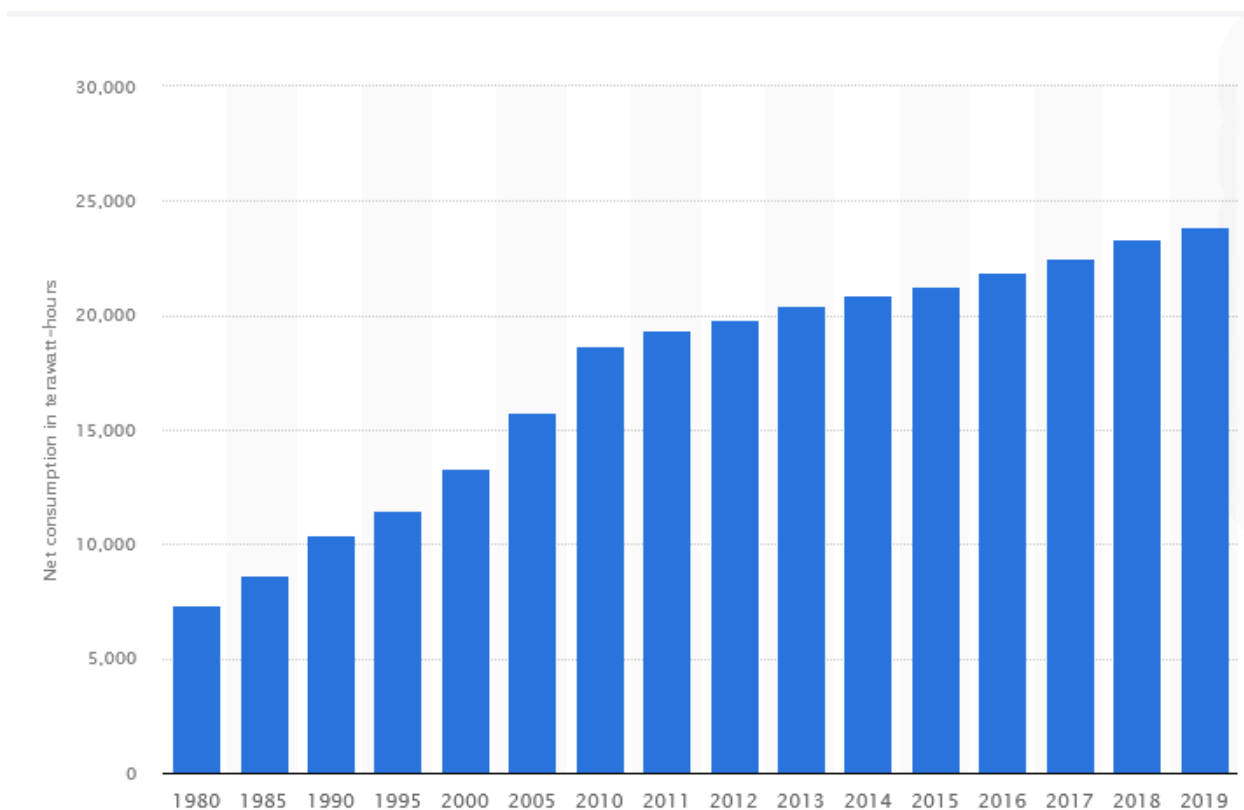
Niska świadomość konsumentów używanej przez nich energii, przyczynia się do **rozwoju efektu cieplarnianego**. Dostrzegamy potrzebę stworzenia urządzenia, które pomoże konsumentom **łatwo i w przyjacielski sposób** oszczędzać energię, aby ratować planetę przed stwarzanymi przez cywilizację zagrożeniami.

“Aby skutecznie ograniczyć skalę globalnego ocieplenia, świat musi jak najszybciej zacząć korzystać z energii w **oszczędny sposób**” –
Europejska Agencja Środowiska¹

Niestety, na rynku **nie odnajdujemy urządzeń**, które byłoby tanie, rozszerzalne, proste i dostępne dla przeciętnego „Kowalskiego” - a jednocześnie przyczyniało się do rozwiązania tego **światowej skali** problemu. Uważamy, że cena pełni tutaj istotną funkcję i powinniśmy ograniczać wszystkie koszty do możliwego minimum, zachowując przy tym jednak wysoką jakość.

Na naszą niekorzyść, statystyki jednoznacznie pokazują, że zużycie energii **dramatycznie wzrasta** z roku na rok.

¹ Energia a zmiany klimatu
<https://www.eea.europa.eu/pl/sygna142y/sygnały-2017/artykuły/energia-a-zmiany-klimatu>



Zużycie energii elektrycznej na świecie w wybranych latach od 1980 do 2019 r.²

Dostrzegamy także, że świadomy konsument może podjąć kroki **we własnym zakresie**, ponieważ według danych Unii Europejskiej:

“Eurostat szacuje, że w 2019 r. całkowity ślad węglowy w UE-27 wynosił **6,7 tony CO₂ na osobę**”³

² Net electricity consumption worldwide in select years from 1980 to 2019

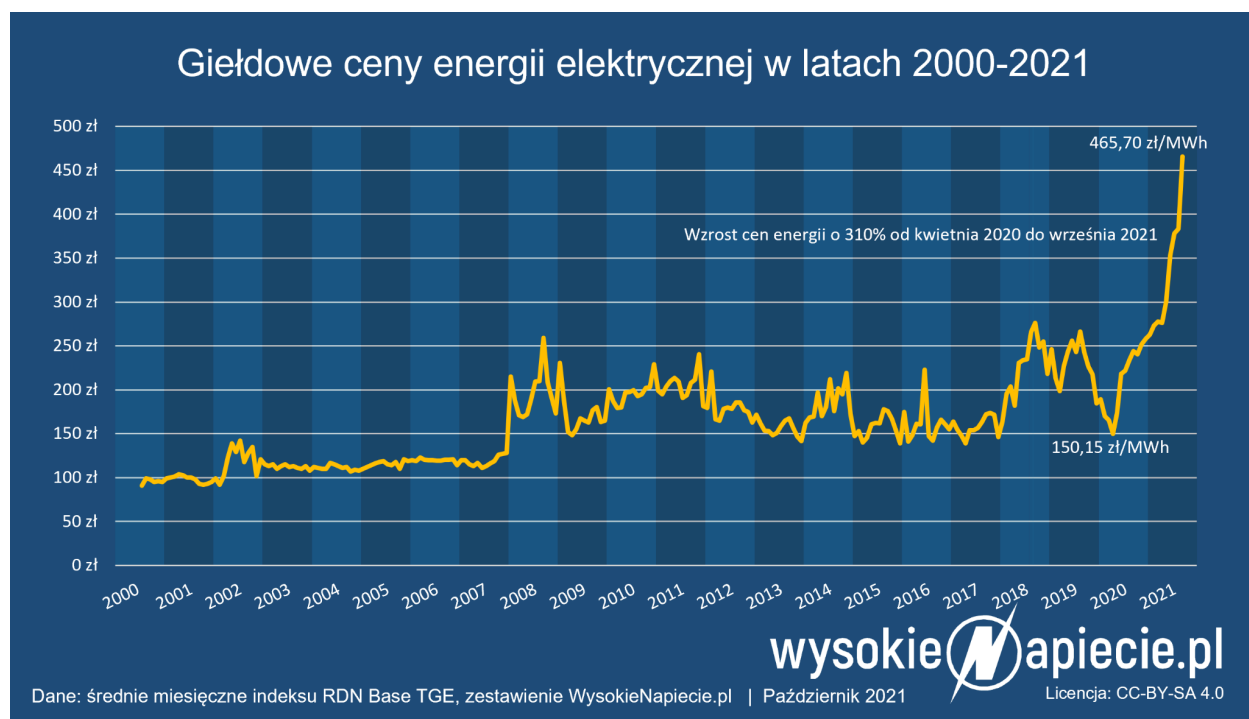
<https://www.statista.com/statistics/280704/world-power-consumption/>

³ Eurostat Greenhouse gas emission statistics - carbon footprints

https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Greenhouse_gas_emission_statistics_-_carbon_footprints

Korzystając z naszego urządzenia, konsument może z lekkim sercem powiedzieć, że robi co w jego mocy, aby być **świadomym i ekologicznym** obywatelem naszej planety.

Podwyżki, podwyżki... i jeszcze raz podwyżki!



Prąd i gaz najdroższe w historii polskiej giełdy⁴

Podwyżki cen energii elektrycznej dla gospodarstw domowych nieprzerwanie wzrastają - mądre gospodarowanie prądem jest nie tylko **ekologiczne**, ale także **oszczędne** w kwestii finansowej. Nie jest to jednak łatwe zadanie, ponieważ większość konsumentów nie posiada urządzeń, dzięki którym mogliby **mądrze analizować, co i kiedy** zużywa najwięcej prądu. Ponadto, świadomy konsument chcący oszczędzać energię, może się zniechęcić wysokimi cenami i skomplikowaniem specjalistycznych urządzeń. Dlatego chcemy stworzyć **przyjazne, tanie i skalowane** rozwiązanie, które użytkownik **z przyjemnością** mógłby wykorzystywać we

4

<https://wysokienapiecie.pl/40034-czekasz-na-5000-zl-z-moj-prad-sprawdz-czy-dostales-ten-e-mail-z-nfosigw/>

własnym domu (choć nic nie stoi na przeszkodzie, aby używać urządzenia w komercyjnym środowisku) - aby płacenie rachunków było **odczuwalnie mniej uciążliwe**. Dzięki naszemu urządzeniu konsument już teraz mógłby zacząć żyć **bardziej ekologicznie i oszczędnie** - jak się okazuje, z naszym urządzeniem **oszczędność i ekologia** stają się o wiele **łatwiejsze**.

Gwarantujemy, że dzięki naszemu rozwiązaniu, zwiększy się nie tylko **komfort finansowy** użytkownika, ale także jego samopoczucie ze względu **ograniczanie indywidualnej szkodliwości** w stosunku do środowiska naturalnego - przy minimalnym wysiłku! Aby sprawić, by urządzenie mogło być powszechne, **powinno kosztować jak najmniej**, jak jest to możliwe. Nie powinno także generować wysokich kosztów.

Cele

Przystępność

Naszym celem jest zaprojektowanie i zbudowanie profesjonalnego monitora energii, który byłby przystępny nawet **dla nietechnicznych użytkowników** oraz **ogólnodostępny i niekosztowny**. Chcemy to osiągnąć, używając **taniego**, ale **spełniającego wymagania** użytkowników sprzętu, który uświadomiłby konsumentom, co zużywa najwięcej energii, aby mogli **wydawać mniej pieniędzy na rachunki** oraz być **przyjaźniejszym dla środowiska**. Aby osiągnąć nasz cel, musimy stworzyć **wygodny interfejs pomiędzy człowiekiem a maszyną**.

Świadomość

Zakładamy stworzenie **autorskiej** aplikacji, która pomoże oszczędzać energię, poprzez:

- **podgląd i monitoring** zużycia energii oraz innych przydatnych informacji
- **powiadomienia** dla użytkownika o przekroczeniu **ustalonego przez niego limitu** zużytej energii

Historia danych

Internet Rzeczy generuje duże ilości danych, które użytkownik może chcieć **analizować i wykorzystywać** według własnych potrzeb. Dlatego poza podglądem rzeczywistym, projekt korzysta ze specjalistycznego oprogramowania przechowującego dane.

Rozszerzalność

Uważamy, że urządzenie powinno być **rozszerzalne i możliwe do dostosowania** według **preferencji** oraz **potrzeb** użytkownika. Aby spełnić ten warunek, musimy użyć **oprogramowania opartego na ogólnodostępnych licencjach**, takich jak:

- **GPL** (General Public License)
- **MIT** (Massachusetts Institute of Technology License)
- **EPL** (Eclipse Public License)

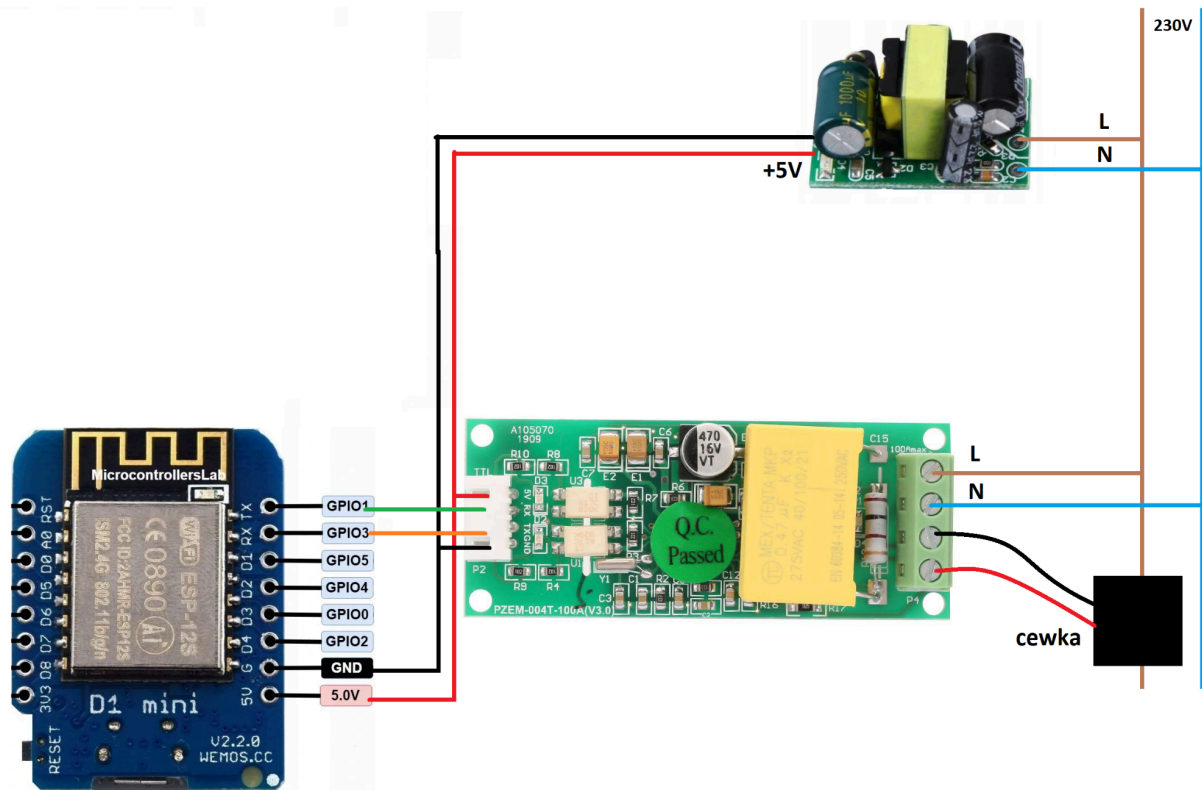
Opis projektu

Sprzęt - tani, ogólnodostępny, rozszerzalny

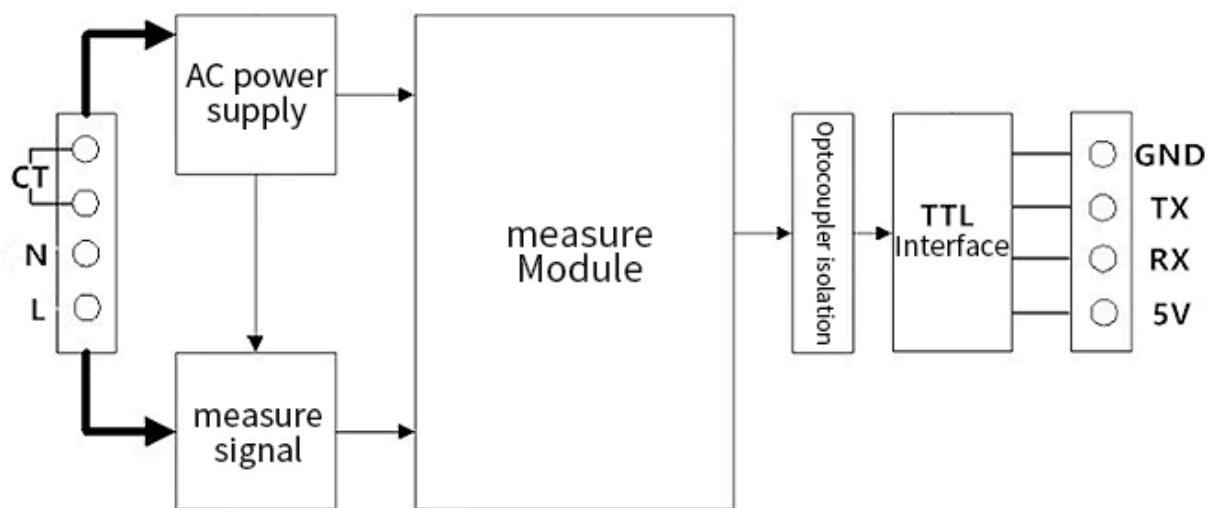
Wstęp: jakiego sprzętu używamy?

1. Z myślą o projekcie korzystaliśmy tylko z **ogólnodostępnego** i relatywnie **niezwykle taniego** sprzętu.
2. Inne urządzenia na rynku kosztują od **kilkuset do kilku tysięcy złotych**. Koszt naszego urządzenia nie przekracza **20 dolarów**. Jest to zgodne z naszą misją stworzenia urządzenia najtańszego, jak to jest możliwe - aby mogło być dostępne w każdym gospodarstwie domowym.
3. Inne urządzenia nierzadko również wymagają dodatkowego abonamentu od użytkownika oraz zmuszają go do używania urządzenia, którego sam nie jest w stanie rozszerzać. W przypadku naszego projektu **użytkownik może dowolnie dostosowywać hardware** (np. wymiana modelu shielda) **i software** (np. użycie preferowanego przez siebie systemu operacyjnego).

4. Urządzenia nie są tak łatwe do aktualizacji, jak jest to np. W przypadku aplikacji. Dlatego staramy się wybrać jak **najodpowiedniejsze komponenty**, aby nie zmuszać użytkownika do ich ewentualnej zmiany. Używamy jednak takiego oprogramowania, które może wykonywać **kwerendy i operacje** nawet **na danych historycznych**.
5. **Bezpieczeństwo** naszego urządzenia jest **priorytetem**. Spełniamy nasz cel poprzez zachowanie zasad bezpieczeństwa i używanie sprawdzonego sprzętu. Nawet przy kontakcie z urządzeniem, zdrowie konsumenta nie zostanie narażone.
6. Projekt powinien działać metodą **Just-in-Time**, czyli operować w **czasie rzeczywistym**. Używając wystarczająco **optymalnego** (i jednocześnie nie nadzbyt kosztownego) sprzętu, jest to możliwe do realizacji, nawet przy **nieprzerwanym korzystaniu z urządzenia**.



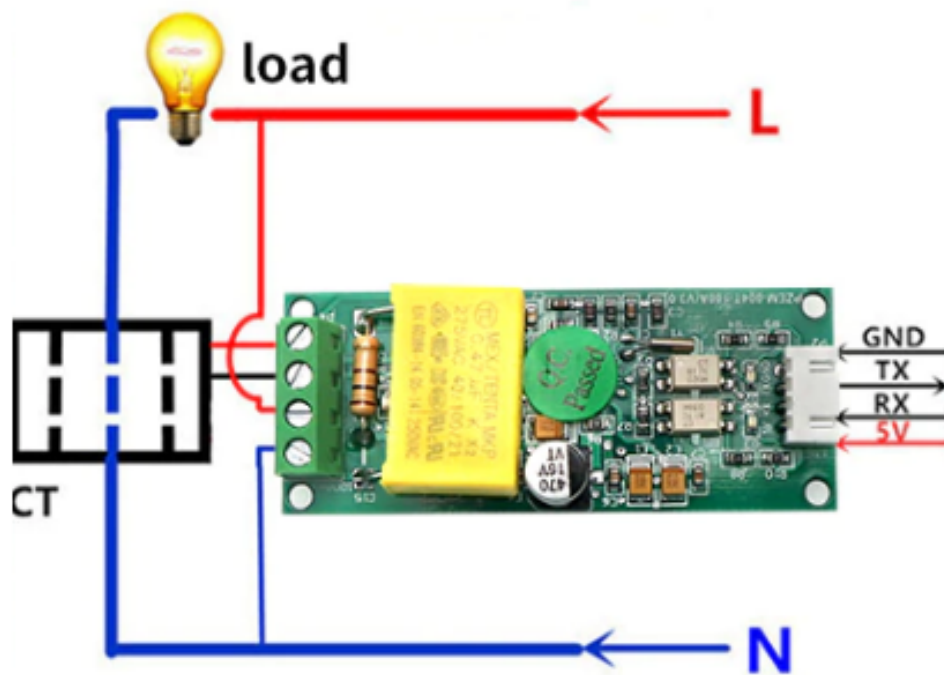
Czujnik prądu PZEM-004T



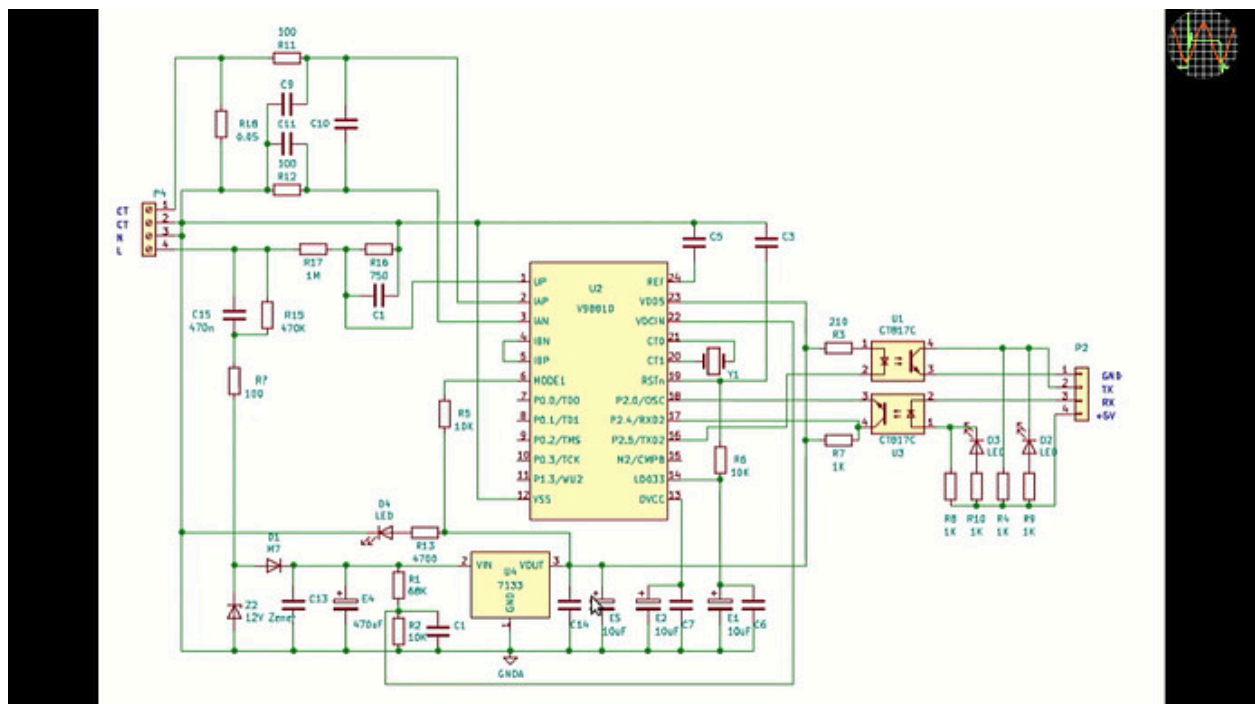
Schemat blokowy funkcji⁵

⁵ Instrukcja PZEM-004

<https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf>



Schemat połączeń

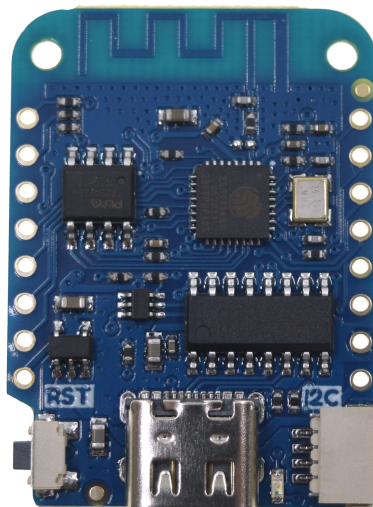


Oficjalny schemat

Ten moduł to niezbędna część projektu. Jest **oficjalnie wspierane** przez używany przez nas firmware, Tasmota⁶. Pozwala on na **łatwy pomiar**:

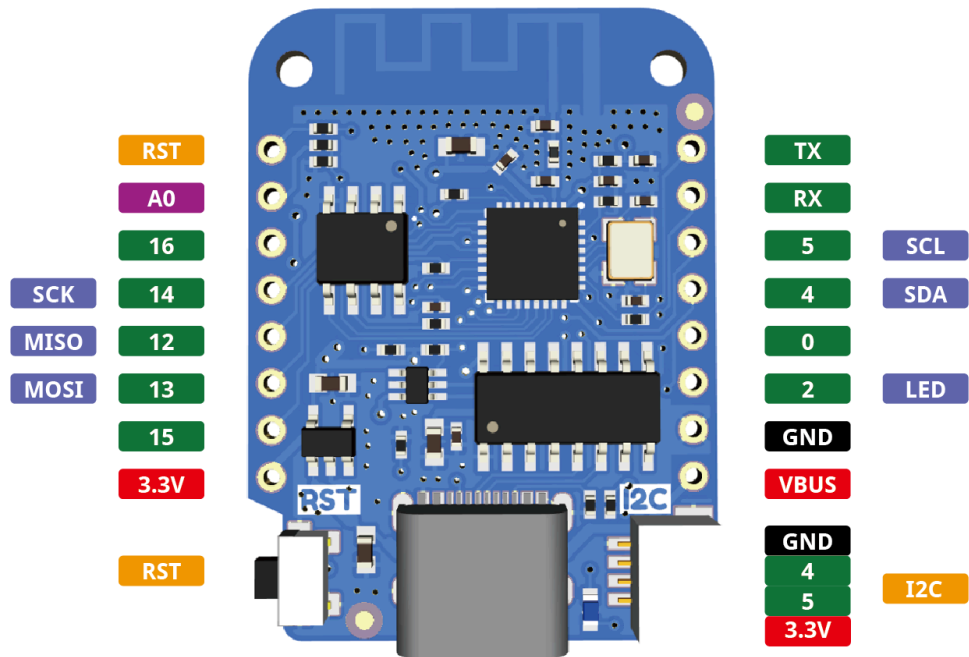
- napięcia
- częstotliwości
- natężenia
- mocy
- zużywanej energii

Mikrokontroler ESP8266 rozszerzony o Shield WEMOS D1 mini

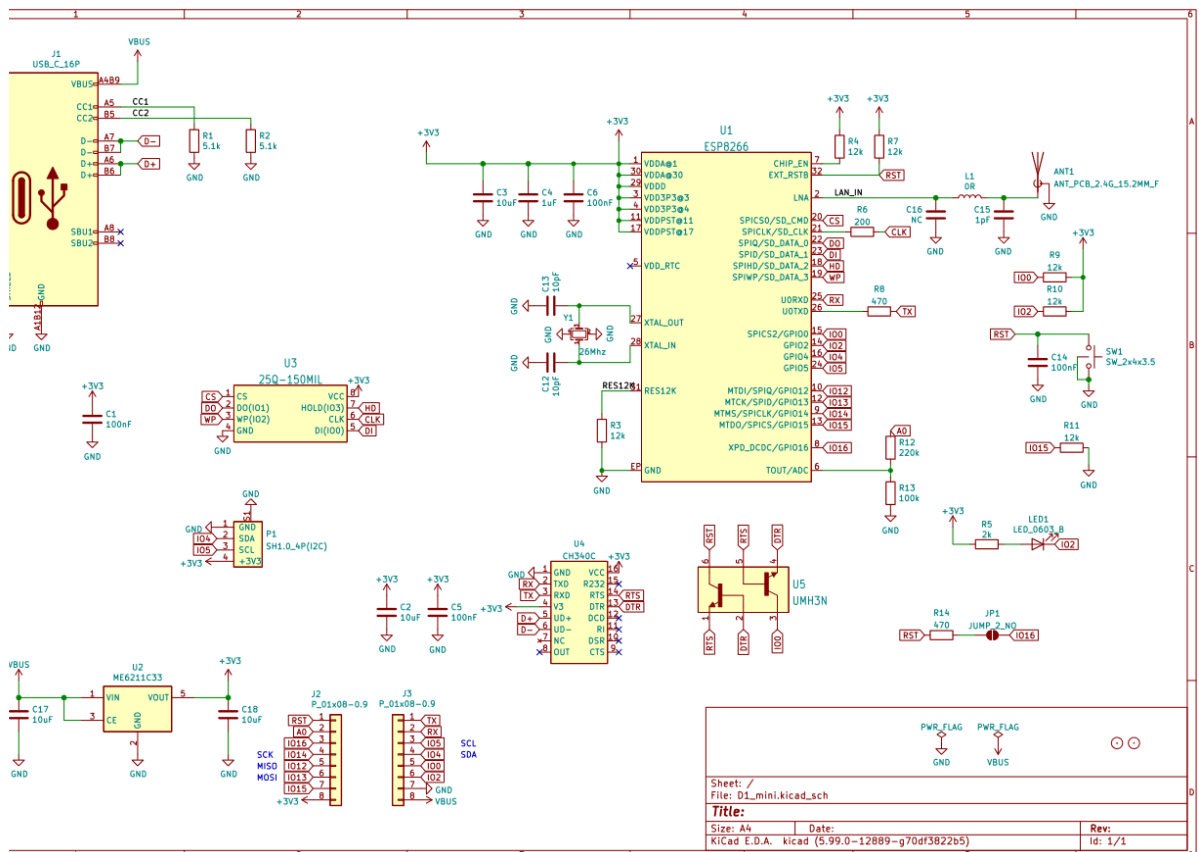


Ekonomiczny i wysoce zintegrowany mikrokontroler Wi-Fi do zastosowań IoT.

⁶ Dokumentacja Tasmota / Peryferia / PZEM-0XX: <https://tasmota.github.io/docs/PZEM-0XX/>



Piny



Co wyróżnia ten mikrokontroler?

Wysoka trwałość

Dzięki szerokiemu zakresowi temperatur pracy urządzenie może pracować **bezwaryjnie**, oferując **niezawodność i wytrzymałość**.

Kompaktowość i wysoka integracja

Małej wielkości urządzenie zawiera⁸:

- 32-bitowy procesor Tensilica
- standardowy interfejs peryferyjny
- przełączniki antenowe
- symetryzator RF (tzw. balun)
- wzmacniacz mocy
- niskoszumowy wzmacniacz odbiorczy
- filtry i moduły zarządzania zasilaniem

Oszczędna architektura

Mikrokontroler charakteryzuje się **niskim zużyciem energii**, dzięki użyciu **procesora Tensilica** oraz **trzem trybom pracy**:

1. tryb aktywny
2. tryb uśpienia
3. tryb głębokiego uśpienia

WEMOS

WEMOS D1 Mini⁹ to obecnie **najtańsza** i bardzo **ogólnodostępna** płytką kompatybilna z Arduino z opisanym wcześniej modułem ESP8266. Oferuje:

- 11 cyfrowych interfejsów IO, obsługa przerwań/pwm/I2C/one-wire (oprócz D0)

⁷ Oficjalny schemat WEMOS https://www.wemos.cc/en/latest/_static/files/sch_d1_mini_v4.0.0.pdf

⁸ Oficjalna strona producenta Espressif <https://www.espressif.com/en/products/socs/esp8266>

⁹ Oficjalna strona WEMOS https://www.wemos.cc/en/latest/d1/d1_mini.html

- 1 wejście analogowe (maks. 3,2 V)
- Port USB typu C
- Port LOLIN I2C
- Kompatybilny z MicroPython, Arduino, nodemcu
- 4 MB pamięci flash

Nic jednak nie stoi na przeszkodzie, aby użytkownik użył innego shielda, **wedle własnych preferencji**. Zachęcamy do zapoznania się z pozostałymi modelami na oficjalnej stronie internetowej producenta: <https://www.wemos.cc/>

Precyzyjna przetwornica AC 220 V → DC 5 V

Przetwornica zamienia prąd zmienny z gniazdka 230 V na prąd stały 5 V.

Oprogramowanie - darmowe, open-source, światowej klasy



Wstęp: jakiego programowania używamy?

W trakcie tworzenia projektu używaliśmy jedynie **darmowego, otwartoźródłowego, światowej klasy** oprogramowania, rozwijanego jednocześnie przez wielkie firmy oraz

utalentowanych wolontariuszy. Skupiliśmy się na **wygodnych i nowoczesnych**, ale zarazem **lekkich** rozwiązaniach, które można uruchomić bezproblemowo na **niedrogich urządzeniach**. Dzięki temu nasze środowisko wpisuje się w **aktualne trendy informatyczne**.

Uważamy, że to istotne, aby używać najnowszych rozwiązań open-source, ponieważ chcemy, aby nasze urządzenie było **rozszerzalne i najtańsze**, jak to jest możliwe.

Priorytetem jest to, aby urządzenie było **bezpieczne**. Używając zarządzanego przez społeczność, sprawdzonego oprogramowania o otwartym kodzie źródłowym, **jesteśmy w stanie to zapewnić**.

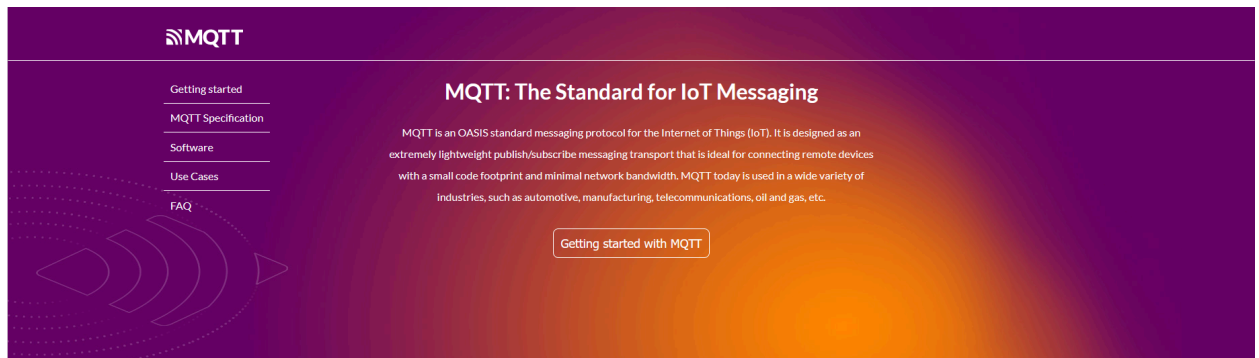
Korzystając szybkiego i dopasowanego software'u, jesteśmy w stanie zapewnić **ekspresowe operacje w czasie rzeczywistym**.

Urządzenia typu Internet of Things produkują i wymagają operowania na **ogromnej ilości danych**. Dlatego starannie przemyśleliśmy użyte technologie, by w jak najlepszy sposób dane:

1. **przechowywać** (InfluxDB),
2. **przesyłać** (MQTT, Mosquitto),
3. **operować** (Grafana),
4. **wizualizować** (Grafana, autorska aplikacja oparta na technologii PWA).

Wierzimy, że dzięki temu użytkownik może czerpać korzyści z **wygodnego i bezbłędnego** produktu.

MQTT: protokół komunikacyjny



Sprawdzony standard dla komunikacji IoT



Czym jest MQTT?

MQTT to protokół używany zarówno komercyjnie, jak i hobbistycznie, **w wielu różnych branżach**, m.in. automotive, logistyka, przemysł, smart home, energetycznych, transportowych.

Dlaczego MQTT?

Wybraliśmy to rozwiązanie, ze względu na:

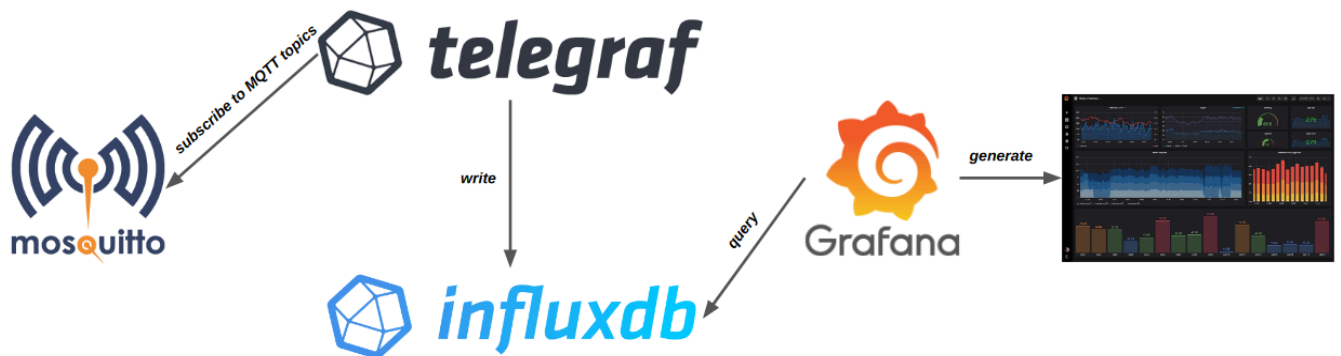
- **lekkość i wydajność:** niewielkie oraz efektywne użycie zasobów i minimalna przepustowość sieci, co jest priorytetem w tego typu rozwiązaniach
- **bezpieczeństwo:** automatyczne szyfrowanie wiadomości przy użyciu nowoczesnych, korporacyjnej klasy protokołów TLS i OAuth, pozwalających na użycie urządzenia nawet w profesjonalnych rozwiązaniach

- **niezawodność**: przemyślana i solidna architektura oraz wsparcie dla zawodnych sieci, zwiększających przystępność technologii, nawet w przypadku zakłóceń połączenia sieci LAN - co sprawia, że użytkownik może korzystać z **domowych, ogólnodostępnych** urządzeń sieciowych

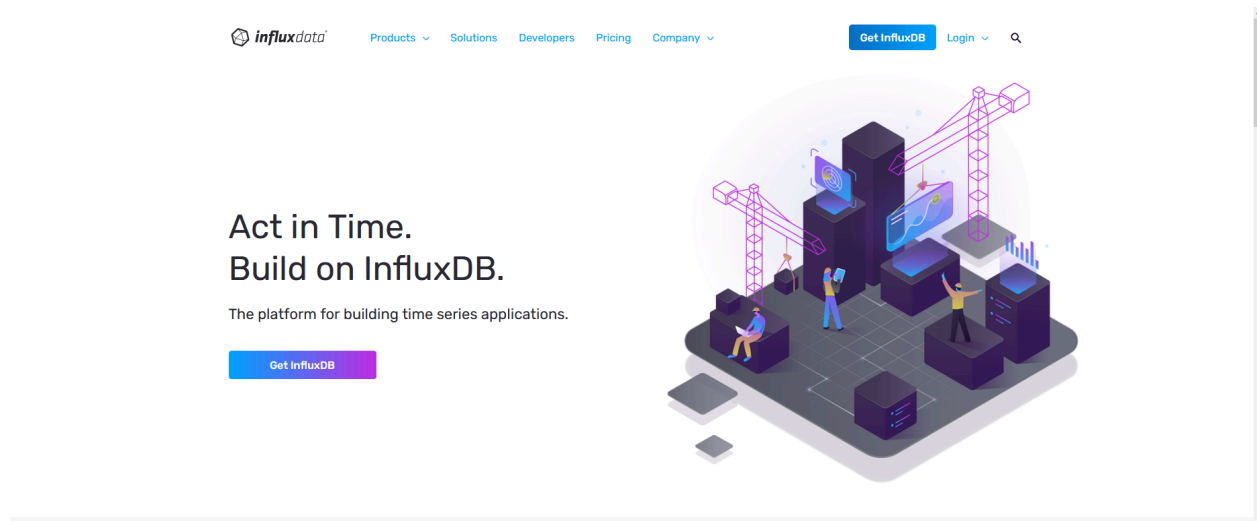
Implementacja protokołu



Mosquitto to **wydajny, otwartoźródłowy** i w pełni **darmowy** serwer, napisany w języku C, implementujący protokół MQTT, rozwijany przez firmę Eclipse. Używamy go w tym projekcie, aby urządzenia mogły się **bezproblemowo komunikować**.



InfluxDB: baza danych dla Internetu Rzeczy



Platforma do budowania aplikacji z uporządkowanymi szeregami czasowymi

Czym jest InfluxDB?



InfluxDB to **otwartoźródłowy, nowoczesny i darmowy** zestaw narzędzi do analizy szeregów czasowych (TSDB) - w bazie danych, każda dana ma zapisany znacznik czasu.

Dlaczego InfluxDB?

InfluxDB, w porównaniu ze standardowymi bazami MySQL, pozwala na operacje i przechowywanie w **wydańiejszy, wygodniejszy** i bardziej **adekwatny** sposób, dlatego często jest używany w nowoczesnych projektach typu IoT. Używanie standardowego typu baz danych,

po dłuższym czasie byłoby niemożliwe ze względów praktycznych. Dlatego uważamy, że InfluxDB to **najlepsza technologia** w sektorze generujących **olbrzymie ilości danych**, takim jak Internet of Things.

Ponadto, ta baza danych pozwala nam operować na danych historycznych (dzięki systemie **TSDB**), co sprawia, że pole do analizy jest nieograniczone.

Przepływ danych

Mosquitto ⇔ Telegraf ⇔ InfluxDB

Telegraf to **otwartoźródłowy** agent, oparty na **tworzonych przez społeczność** wtyczkach - działa jako most pomiędzy Mosquitto a InfluxDB. Pozwala na **wiele źródeł** wejściowych i wyjściowych, umożliwia także **transformacje danych**.



Instalacja Telegrafa na serwerze Linux

```
sudo apt install telegraf -y # Zainstaluj pakiet Telegraf
sudo systemctl start telegraf # Włącz usługę Telegraf
sudo systemctl enable telegraf # Uruchamiaj Telegraf przy
starcie systemu
sudo nano /etc/telegraf/telegraf.conf # Edytuj plik
konfiguracyjny

[[outputs.influxdb]]
    database = "telegraf"
    urls = [ "http://127.0.0.1:8086" ]
    username = "telegraf" # Nazwa użytkownika
    password = "zaq1@WSX" # Hasło

[[inputs.mqtt_consumer]]
```

```
servers = ["tcp://127.0.0.1:1883"]
topics = [
    "tele/energia/SENSOR", "tele/energia/STATE"
] # Tematy, na które nasłuchiwać ma MQTT
data_format = "json"
```

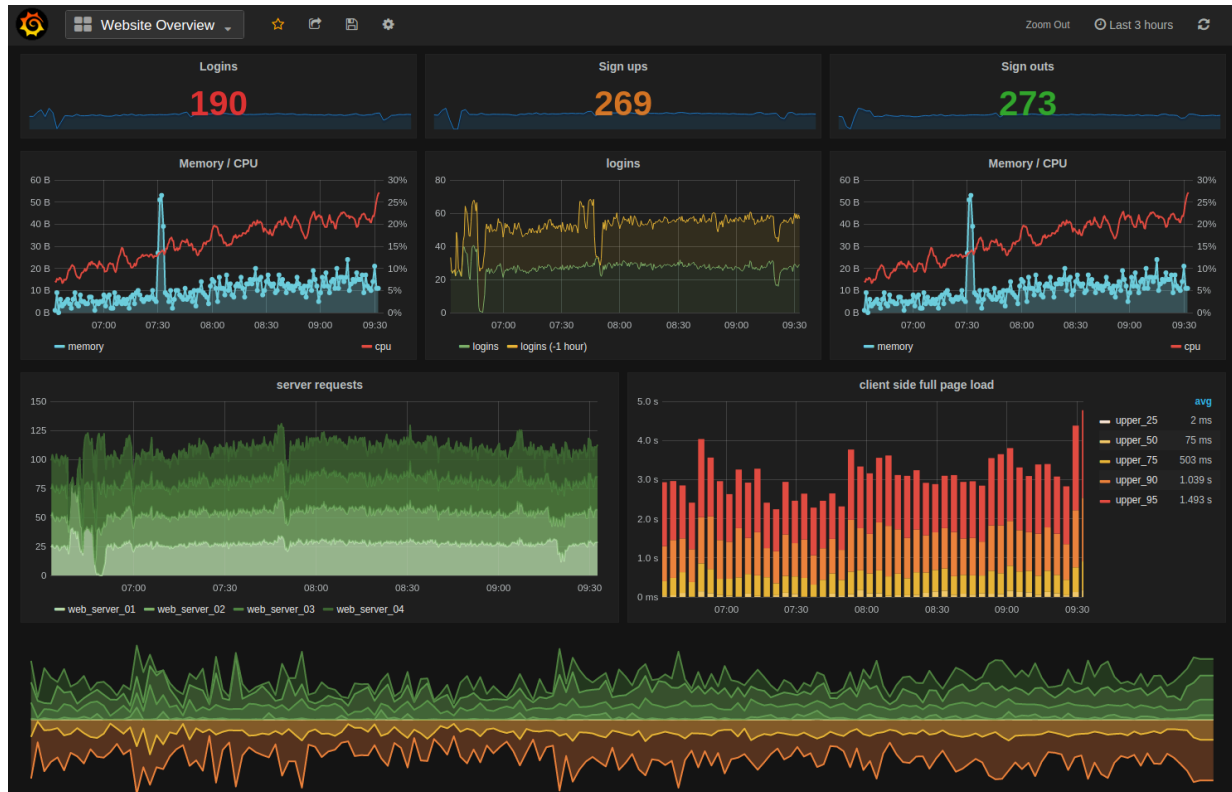
```
sudo systemctl restart telegraf # Zrestartuj usługę, aby
zaktualizować plik konfiguracyjny
sudo systemctl status telegraf # (opcjonalne) Pokaż status
usługi
```

Instalacja bazy danych

Oficjalną dokumentację, która przeprowadzi nas **w łatwy sposób** przez proces instalacji i konfiguracji, możemy znaleźć pod adresem: <https://docs.influxdata.com/>

```
sudo apt-get install influxdb # Zainstaluj pakiet InfluxDB
sudo service start influxdb # Włącz usługę InfluxDB
sudo service enable influxdb # Uruchamiaj bazę danych przy
starcie systemu
```

Grafana: wygodna obserwacja



Otwartoźródłowa platforma do monitorowania i obserwacji

Czym jest Grafana?

Grafana to **innowacyjna** aplikacja internetowa służąca do obserwacji. Pozwala na **wyszukiwanie, wizualizację, ostrzeganie i zrozumienie** danych w **łatwy i dostępny** sposób, również **dla nietechnicznych użytkowników**.

Jak używamy Grafany?

Grafana to **alternatywny sposób przeglądania danych**, poza naszą aplikacją. Odczytujemy i prezentujemy dane **z bazy InfluxDB**.

Instalacja Grafany

Szczegółową dokumentację projektu można znaleźć pod adresem: <https://grafana.com/docs/>

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key
add - # Dodaj klucz GPG, aby móc instalować pakiety Grafany
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb
stable main" # Dodaj repozytorium zawierające pakiety Grafany
sudo apt update # Zaktualizuj lokalną listę pakietów
sudo apt install grafana # Zainstaluj pakiet Grafana
sudo systemctl start grafana-server # Włącz usługę Grafana
sudo systemctl enable grafana-server # Uruchamiaj bazę danych
przy starcie systemu
```

Konfiguracja Grafany

Aby umożliwić poprawne działanie aplikacji mobilnej, należy włączyć opcję wstawiania.

```
sudo nano /etc/grafana/grafana.ini # Edytuj plik
konfiguracyjny
    allow_embedding: true # Ustawiamy wartość z false na true
    auth.anonymous
        enabled = true
        org_name = Main Org.
        org_role = Viewer
sudo systemctl restart grafana-server.service # Zrestartuj
usługę Grafany, aby odświeżyć konfigurację
```

Autorskie kwerendy InfluxDB do prezentacji danych w Grafanie

Zużyta energia (dzisiaj)

```
SELECT mean("ENERGY_Today") FROM "mqtt_consumer" WHERE time ≥ now() - 1m and  
time ≤ now() GROUP BY time(1d) fill(null)
```

Zużyta energia (wczoraj)

```
SELECT mean("ENERGY_Yesterday") FROM "mqtt_consumer" WHERE time ≥ now() - 1m  
and time ≤ now() GROUP BY time(200ms) fill(null)
```

Zużyta energia (7 dni)

```
cumulative_sum(mean("ENERGY_Today")) FROM "mqtt_consumer" WHERE time ≥ now()  
- 7d and time ≤ now() GROUP BY time(1d) fill(null)
```

Zużyta energia (30 dni)

```
SELECT cumulative_sum(mean("ENERGY_Today")) FROM "mqtt_consumer" WHERE time ≥  
now() - 30d and time ≤ now() GROUP BY time(1d) fill(null)
```

Aktualne napięcie

```
SELECT ENERGY_Voltage FROM mqtt_consumer ORDER BY DESC LIMIT 1
```

Natężenie

```
SELECT ENERGY_Current FROM mqtt_consumer ORDER BY DESC LIMIT 1
```

Moc aktualna

```
SELECT ENERGY_Power FROM mqtt_consumer ORDER BY DESC LIMIT 1
```

Moc bierna

```
SELECT ENERGY_ReactivePower FROM mqtt_consumer ORDER BY DESC LIMIT 1
```

Moc pozorna

```
SELECT ENERGY_ApparentPower FROM mqtt_consumer ORDER BY DESC LIMIT 1
```

Tasmota: rozbudowany firmware



Potężny firmware dla urządzeń opartych na ESP8266 i ESP32

Czym jest Tasmota?

Tasmota to alternatywny firmware dla urządzeń opartych na ESP8266 i ESP32 z **łatwą konfiguracją** za pomocą strony internetowej, **zdalnymi** aktualizacjami, **automatyzacją** za pomocą timerów lub reguł, możliwością **rozbudowy** i całkowicie **lokalną kontrolą** przez MQTT, HTTP, Serial lub KNX.

Dlaczego Tasmota?

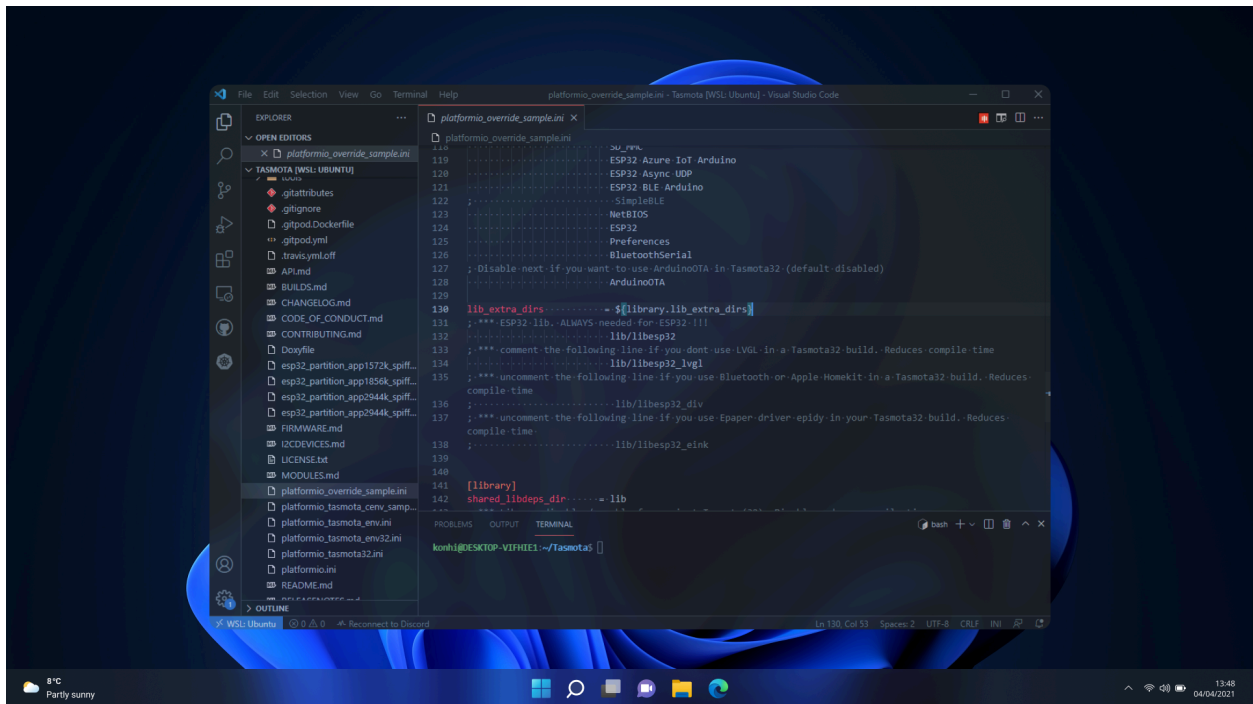
Dzięki społeczności bogata w funkcje Tasmota oszczędza nam wiele czasu, przez **wbudowane rozwiązania najczęstszych problemów**.

Instalacja

Pomoc i instrukcję można znaleźć pod adresem: <https://tasmota.github.io/docs/>

Konfiguracja

Konfiguracja buildu Tasmoty jest opcjonalna, ale pozwala nam uzyskać **maksymalną optymalizację** według potrzeb urządzenia. Osiągamy to dzięki potężnemu edytorowi kodu **Visual Studio Code**.



Ubuntu Server: przyjazna dystrybucja Linuxa

ubuntu[®]

Najpopularniejszy serwerowy Linux

Dlaczego Ubuntu?

Ubuntu Server zapewnia **ekonomiczną i techniczną skalowalność** serwerów. To jedna z najbardziej **przyjaznych** i **powszechnych** dystrybucji Linuxa. W naszym projekcie uznaliśmy Ubuntu za najlepszy wybór, ale nie ma żadnych przeciwności, aby bardziej doświadczony użytkownik wybrał inną dystrybucję, która bardziej będzie odpowiadała jego potrzebom,

preferencjom i możliwościom (np. w przypadku posiadania procesora o architekturze 32-bitowej)

Środowisko uruchomieniowe

Użytkownik może wybrać, czy serwer będzie zainstalowany na jego lokalnym urządzeniu, czy w chmurze. Przetestowaliśmy oba, aby upewnić się, że **Monitor Energii będzie działać bezzmiennie w każdym środowisku.**

Ubuntu Server Edition zapewnia minimalistyczną podstawę dla wielu aplikacji serwerowych. Ta wersja obsługuje cztery architektury **64-bitowe**:

- amd64 (Intel/AMD 64-bit)
- arm64 (64-bitowa architektura ARM)
- ppc64el (POWER8 i POWER9)
- s390x (IBM Z i LinuxONE).

Zalecane wymagania systemowe to:

- CPU: 1 GHz lub lepszy
- RAM: 1 GB lub więcej
- Dysk: co najmniej 2,5 GB

W przypadku braku urządzenia, spełniającego parametry, użytkownik może zdecydować o bezkosztowym użyciu darmowego planu, oferowanego przez najpopularniejszych dostawców usług chmurowych, na przykład:

- Google Cloud,
- Microsoft Azure,
- Amazon Web Services,
- Oracle Cloud.

Instalacja

Zachęcamy do skorzystania z **oficjalnej dokumentacji** wybranej dystrybucji (w przypadku Ubuntu: <https://ubuntu.com/server/docs>), aby proces instalacji systemu operacyjnego przebiegł **bezbłędnie**.

PM2



Zaawansowany, otwartoźródłowy menedżer procesów, gotowy do użycia na produkcji. Pomaga **zarządzać aplikacją i utrzymywać ją w trybie online 24/7**.

Dokumentacja dostępna jest pod adresem: <https://pm2.keymetrics.io/docs/usage/quick-start/>.
PM2 wykonuje nasz **autorski skrypt, który zarządza** brokerem Mosquitto.

```
#!/bin/sh  
mosquitto_sub -u olimpiada -P zaq1@WSX -t "tele/energia/SENSOR"  
mosquitto_sub -u olimpiada -P zaq1@WSX -t "tele/energia/STATE"
```

Aplikacja: osobiste i dostosowane użycie

Co oferuje aplikacja?

Podgląd danych

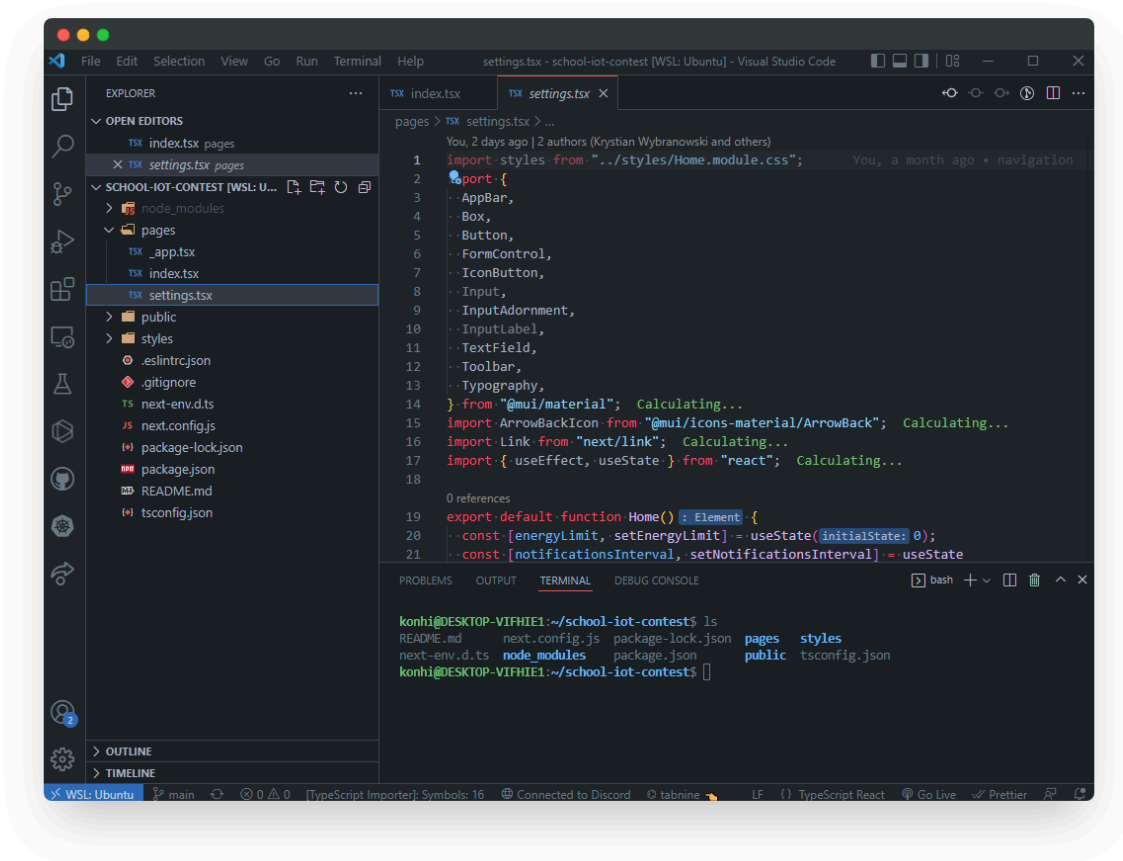
Wygodny, szybki i autorski sposób na podgląd danych z mikrokontrolera. Użytkownik ma dostęp do widoku dashboardu Grafany **bez przekomplikowanych funkcji**.

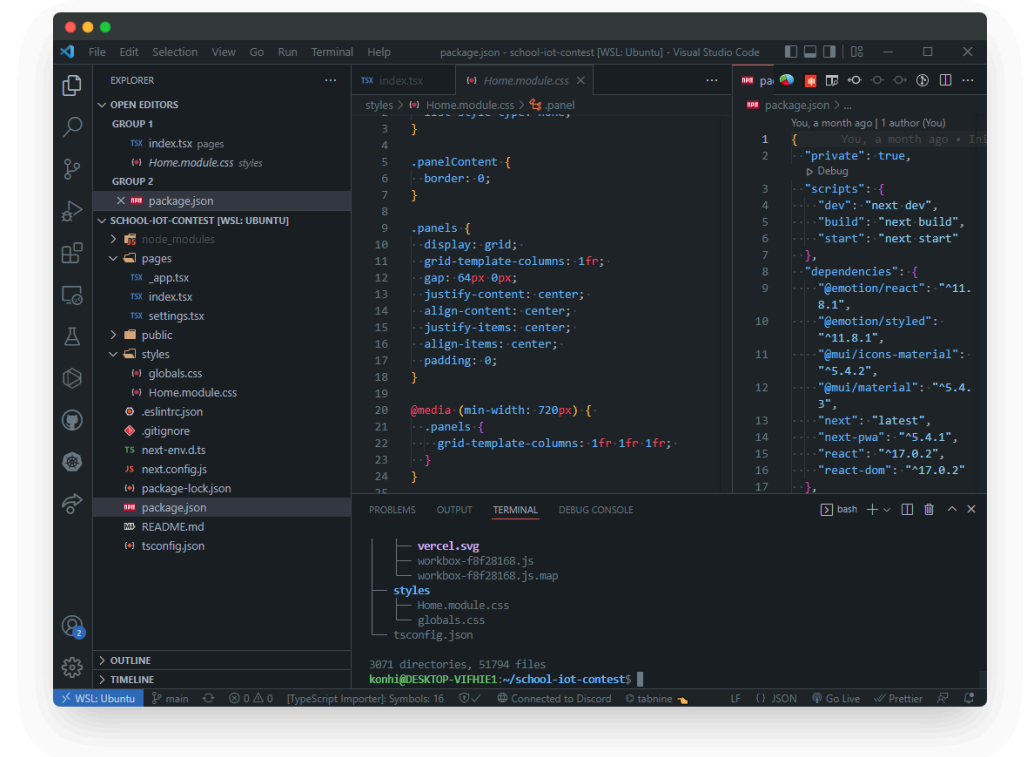
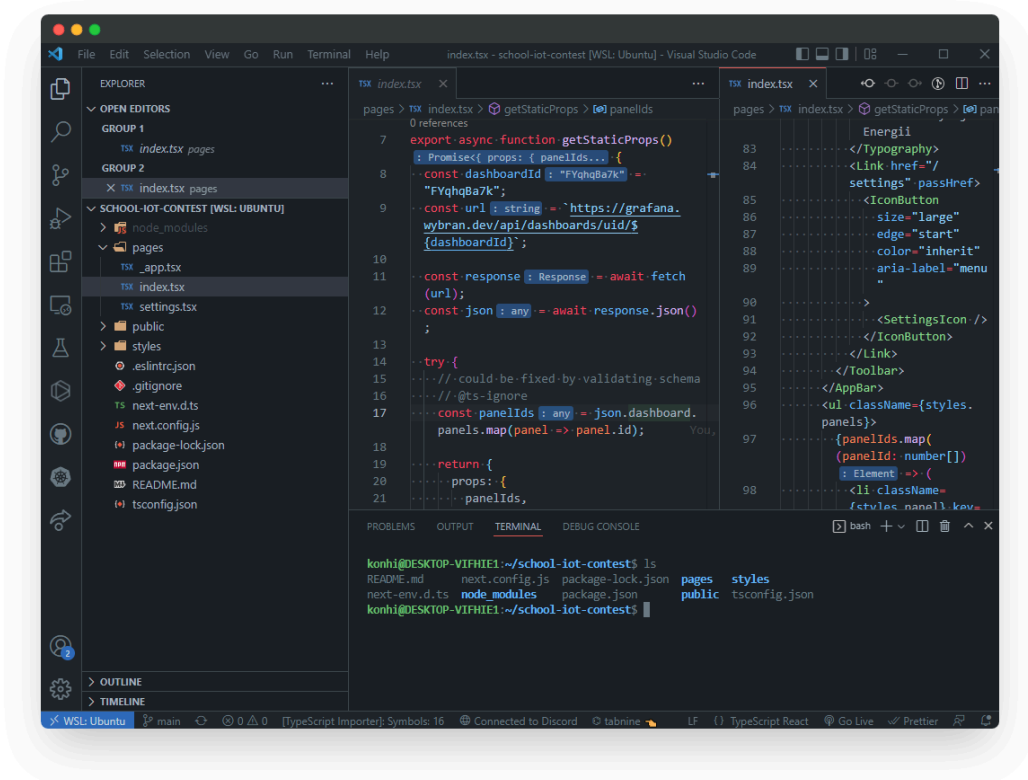
Notyfikacje

Aplikacja implementuje **autorski system spersonalizowanych alertów** w przypadku przekroczenia zużycia energii.

Jak zbudowana jest aplikacja?

Pełny kod źródłowy aplikacji jest dostępny pod adresem: github.com/konhi/school-iot-contest





React



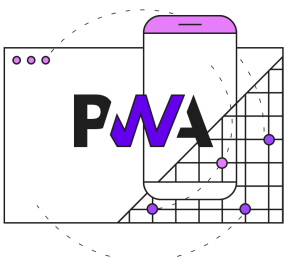
Wykorzystuje światowego standardu bibliotekę front-endową **React**, która umożliwia szybkie budowanie pięknych, własnych interfejsów - co pomaga zapewnić dobre doświadczenia użytkownika. W taki sposób staramy się tworzyć nowoczesny interfejs pomiędzy człowiekiem a maszyną.

Next.js



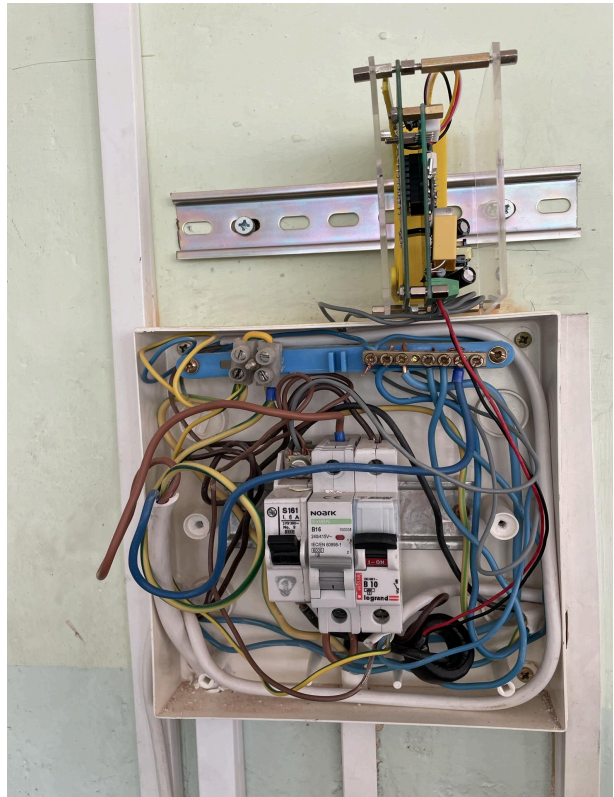
Aplikacja jest oparta na minimalistycznym frameworku webowym **Next.js** (opierającym się na potężnie wzrastającym runtimem **Node.js**), który pozwolił nam zbudować **nowoczesną, optymalną i autorską** aplikację.

Progressive Web App



Progressive Web Apps to w istocie po prostu **aplikacje internetowe**. Dzięki **progresywnemu ulepszaniu** nowe możliwości są dostępne w nowoczesnych przeglądarkach. Dzięki nowoczesnej technologii przeglądarkowej **“Service Workers”** aplikacja staje się **niezawodna** nawet w przypadku braku dostępu do internetu oraz możliwa do zainstalowania na telefonie i komputerze, bez publikowania jej do standardowego sklepu aplikacji, np. Google Play lub Apple Store.

Efekt finalny



Aplikacja

Nasza aplikacja dostępna jest pod adresem
<https://olimpiada.wybran.dev>

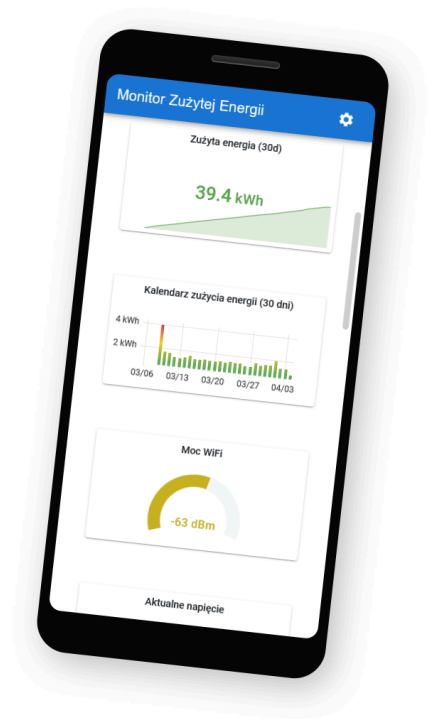


Wersja mobilna

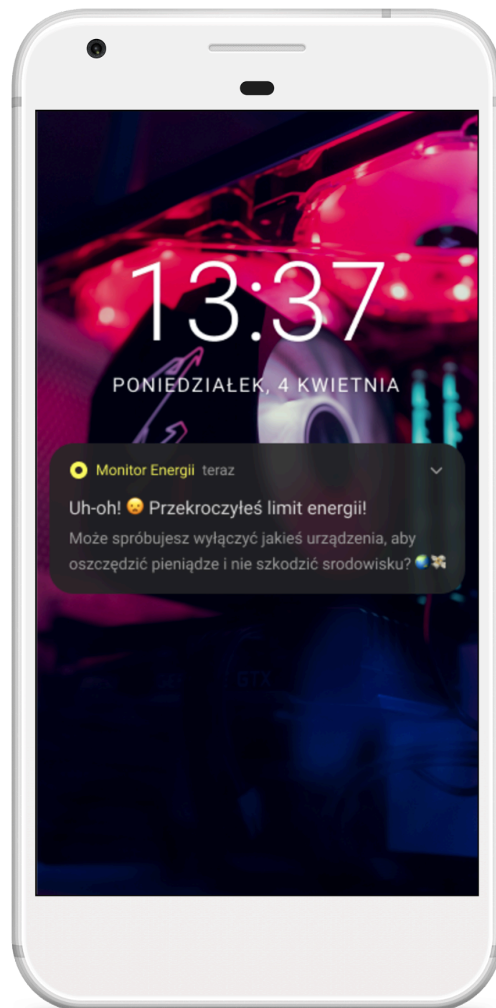


APLIKACJA IOT

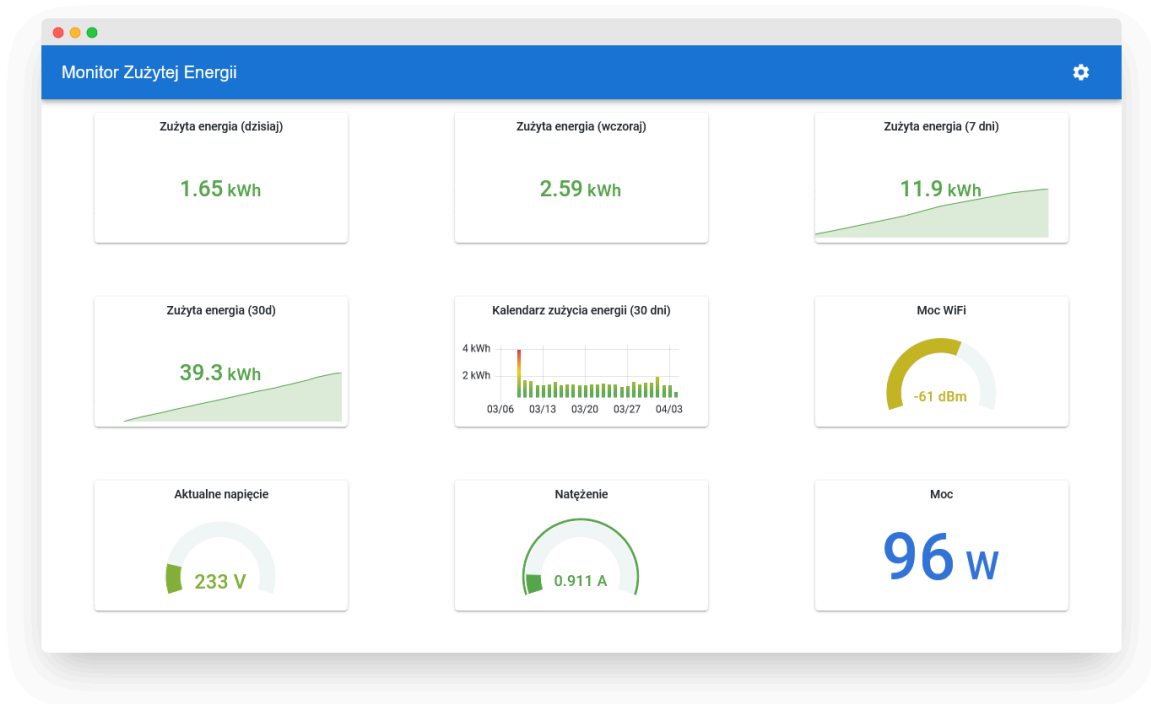
Monitor Zużytej Energii



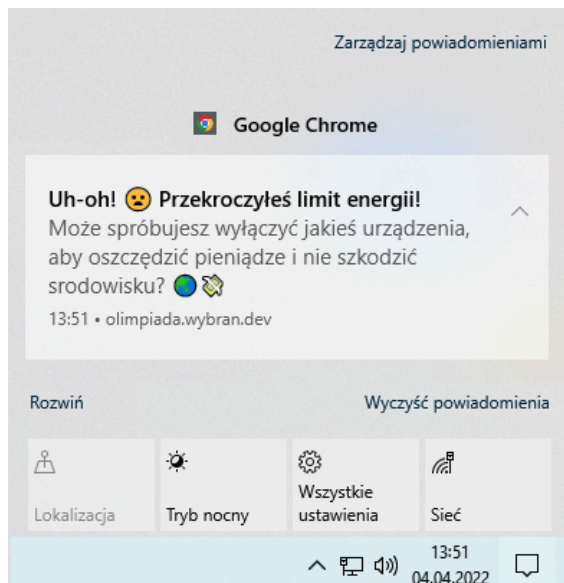
Notyfikacje



Wersja desktopowa



Notyfikacje



The settings screen has a blue header bar with a back arrow and the title "Ustawienia". Below the header, the section "Spersonalizuj aplikację" (Customize the app) contains two settings:

- Limit zużycia energii:** A text input field containing "kWh 2".
- Interwał powiadomień:** A text input field containing "minuty 10".

At the bottom of the settings section is a large blue button labeled "ZMIEN" (Change).

Grafana

Nasz dashboard Grafany jest publicznie dostępny pod adresem <https://grafana.wybran.dev>. Zachęcamy do przekonania się, jak proste jest korzystanie z serwisu!



Zastosowanie

Nasze narzędzie może być używane, **niezależnie od skali użycia** - w warunkach zarówno **domowych**, jak i **korporacyjnych**.

Efekt ekonomiczny

Dzięki naszemu urządzeniu użytkownik **uświadamia** sobie, co zużywa najwięcej energii, przez co w niekompleksowy sposób **oszczędza pieniądze**.

Poza oszczędnością pieniędzy konsument, użytkując energię w oszczędny sposób, **minimalizuje zagrożenia stwarzane dla planety**.