

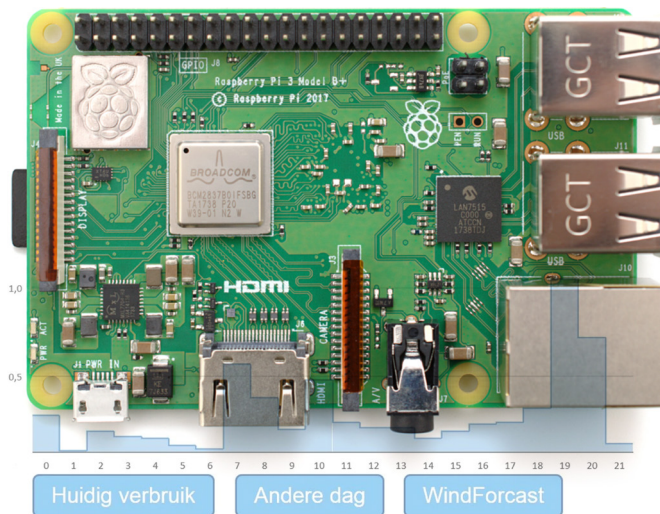
# Miernik zużycia energii elektrycznej na bazie ESP8266 i Raspberry Pi



Monitorowanie ilości zużywanej energii elektrycznej jest podstawą do jej oszczędzania. Jeśli na bieżąco wiemy, ile nasze gospodarstwo domowe pochłonęło energii, jesteśmy w stanie ukierunkować nasze starania w celu redukcji tej ilości. Energooszczędność to nie tylko mniejsze rachunki, ale także mniejsza emisja CO<sub>2</sub> do atmosfery.

Jest to pierwszy projekt elektroniczny użytkownika didier\_windey z portalu Instructables.com. Wymagał od niego sporo nauki, jako że zastosowano Arduino (moduł z układem ESP8266, oprogramowany w Lua) oraz Raspberry Pi z bazą danych MySQL do zbierania i prezentacji danych. Początkowym celem tego projektu było tylko monitorowanie zużycia energii elektrycznej w domu, rejestrowanie tych danych i możliwość ich późniejszego analizowania. Finałnie został wzbogacony o szereg dodatkowych funkcji.

W przypadku domu autora projektu głównym dostawcą energii elektrycznej jest lokalna farma wiatrowa. Oczywiście dom zasilany jest także z normalnej sieci energetycznej, ale jak wskazuje sam autor – głupio byłoby mu włączać większe, zbędne obciążenie (jak np. zmywarka czy pralka) w bezwietrzny dzień. Dlatego miernik zużycia energii nie tylko pokazuje aktualną jej konsumpcję w domu, ale także potencjał energetyczny lokalnej farmy wiatrowej, jaki jest podawany w czasie rzeczywistym i dostępny przez przeglądarkę internetową. Taka integracja była możliwa w miejscu zamieszkania autora projektu (Elia w Belgii) – czy taka integracja możliwa jest także w Polsce, zależy od operatorów konkretnych instalacji energetyki odnawialnej.



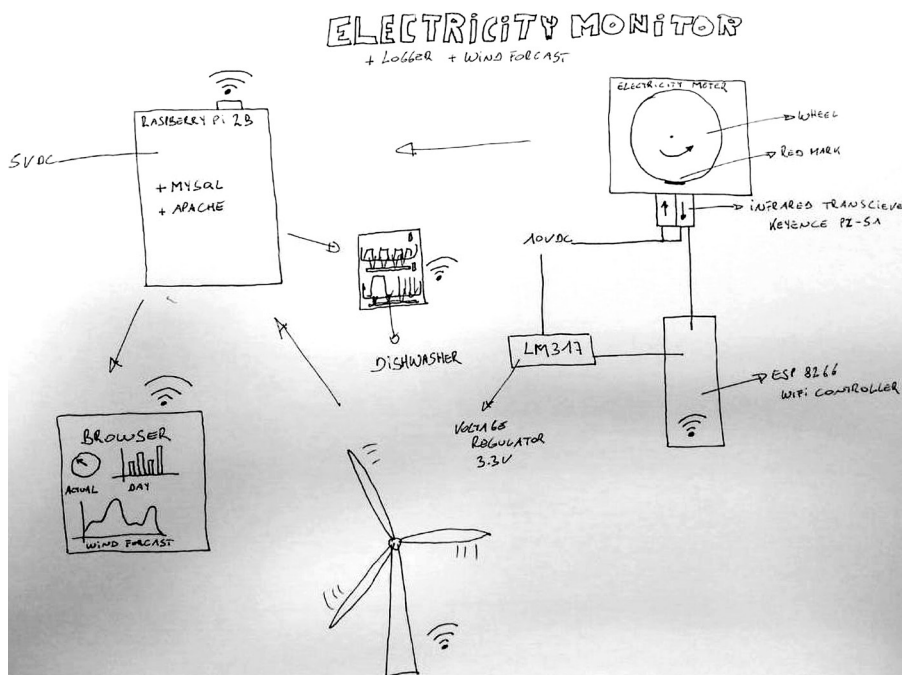
Wyjście czujnika jest podłączone do pinu GPIO modułu z układem ESP8266. Kiedy pin zmienia swoją wartość z jedynki na zero, to kompletuje wiadomość dla bazy danych, a przy najbliższym zloczcu narastającym wysyła ją do bazy danych MySQL, która działa na Raspberry Pi. Co 5 minut Raspberry Pi pobiera najnowszą prognozę uzysku energii wiatrowej w formacie XML ze strony internetowej Grid Managera. XML jest tłumaczony na polecenie SQL, a dane są przechowywane w tabeli MySQL. Pierwsza strona w przeglądarce prezentuje rzeczywiste zużycie energii elektrycznej. Na następnej stronie można sprawdzić

## Potrzebne elementy i zasada działania

Projekt został zbudowany z kilku zasadniczych modułów programowych i sprzętowych. System zaprezentowano schematycznie na rysunku 1. Składa się z:

- miernika zużycia energii z modulem do komunikacji poprzez Wi-Fi,
- komputera jednopłytkowego Raspberry Pi z odpowiednim oprogramowaniem,
- oprogramowania do pobierania danych (w formacie XML) ze strony operatora sieci przesyłowej,
- przeglądarki WWW, która służy za interfejs użytkownika do zarządzania i używania systemu.

Czujnik licznika energii elektrycznej wykorzystuje prosty sensor podczerwieni, monitorujący obrót koła miernika zużycia energii elektrycznej. Śledzi on ruch czerwonej linii na tym kole, co pozwala na zliczanie obrotów.



Rysunek 1. Schemat blokowy budowy i działania urządzenia

dzienne zużycie energii na godzinę. Trzecia strona to prognoza ilości produkowanej energii wiatrowej i rzeczywista jej produkcja w danym momencie, wyświetlane na wykresie Google. Z poziomu przeglądarki sterować można także odbiornikami – autor podłączył do systemu sterowanie zasilaniem swojej zmywarki, gdyż jest to urządzenie, które, jego zdaniem, może poczekać na uruchomienie na odpowiedni moment. Dzięki temu, gdy dostępne jest dostatecznie dużo „zielonej” energii, można jednym kliknięciem uruchomić energochłonny odbiornik energii elektrycznej. Docełowo, jak pisze autor, chciałby i ten aspekt zautomatyzować, by włączała się ona automatycznie, gdy dostępne jest odpowiednio dużo prądu ze źródeł ekologicznych.

**Listing 1. Skrypt dla ESP8266 do monitorowania miernika zużycia energii elektrycznej**

```
function start()
if (x == nil) then
print("do setup")
tmr.alarm(1,1,0, setup)
else
tmr.alarm(1,1,0, puls)
print("do puls")
end
end

function setup()
t= 0 -- toer
x= 1 -- start
t0 = tmr.now()

gpio.mode(4, gpio.OUTPUT)
gpio.write(4, gpio.HIGH)
print("do wifcon")
tmr.alarm(1,1,0, wifcon)
end

function wifcon()
wifi.setmode(wifi.STATION)
wifi.sta.config("login","password")
print("do pulse")
tmr.alarm(2, 1, 0, puls)
end

function puls()
p = gpio.read(3) -- p to impuls

if (p == 0) then
a = 1
u = 0
end

if (p ==1) then
u = 1
end

if (a == 1) and ( u == 1) then
t= t+1
u=0
a=0
t1= tmr.now()
t2= (t1-t0)
t2= (t2/1000)
t2= (9600000/t2)
t0 = t1
print (t2)
print ("do check")
tmr.alarm(4, 1, 0, check)
else
tmr.alarm(1,1,0,puls)
end
end

function check()
if (wifi.sta.getip() == nil) then
tmr.alarm(3, 1, 0, wifcon)
else
print("do wifsend")
print(wifi.sta.getip())
tmr.alarm(4, 1, 0, wifsend)
end
end

function wifsend()
conn=net.createConnection(net.TCP, 0)
conn:on("receive", function(conn, payload) print(payload) end )
conn:connect(80, "192.168.0.156")
local s=("GET /electriciteitsmeter.php?toer="..t.."&KwH="..
t2.."&stand=1\r\n HTTP/1.1\r\nHost: 192.168.0.122\r\n" ..
"Connection: keep-alive\r\nAccept: */*\r\n\r\n")
conn:send(s)
t=0

print("do start")
print(t0)
tmr.alarm(4, 200, 0, start)
end

print ("do start")
tmr.alarm(1,1,0, start)
```

## Sensor obrotu tarczy miernika zużycia energii elektrycznej

Konstrukcja czujnika jest zaskakująco prosta. Emiter podczerwieni wysyła sygnał, który następnie odbija się od metalowego koła licznika energii elektrycznej. Kiedy czerwona linia mija odbijającą się wiązkę, światło podczerwone jest pochłaniane nieco bardziej niż normalnie. Czujnik odbitej podczerwieni jest ustawiony na wykrywanie zmiany jasności i włącza się w momencie, gdy intensywność zmienia się. ESP8266 uruchamia w takiej sytuacji proste programy do „ładowania” i „wysyłania” wiadomości do bazy danych.

Do konstrukcji czujnika potrzebne są:

- Transceiver podczerwieni (transceptor odbiciowy). Autor zastosował sensor Keyence PZ-51, pochodzący z rozbiórki innego urządzenia. W mierniku energii elektrycznej kółko znajduje się za szybką z pleksi, około 50 mm od okienka i miejsca, gdzie można ustawić sensor. Zastosowany sensor idealnie pasuje do tej lokalizacji.
- Stabilizator obniżający napięcie. ESP8266 zasilany jest napięciem 3,3 V. Trzeba ostrożnie budować system, by nie uszkodzić tego mikrokontrolera zbyt wysokim napięciem zasilania. Można zastosować liniowy stabilizator napięcia i drobne elementy dyskretnie (scalony stabilizator napięcia LM317, oporniki 400 i 240 Ω oraz kondensatory 1 i 0,1 μF).
- Fotoopornik i rezystor pozwala na konstrukcję dzielnika napięcia. Został on wykorzystany do odczytywania stanu sensora Keyence. Sygnał wyjściowy z samego sensora był bardzo zakłócony i słabo czytelny dla portów GPIO mikrokontrolera. Na szczęście sensor ma także dwie diody LED – zieloną i czerwoną, które sygnalizują jego stan. Fotoopornik przyczepiony jest do zielonej diody LED i śledzi jej stan.
- Moduł z ESP8266 – mikrokontroler z zainstalowanym NodeMCU.
- Arduino IDE – środowisko programistyczne służące m.in. do ładowania oprogramowania do mikrokontrolera. System ten oprogramowano w Lua (**listing 1**).
- Zasilacz prądu stałego 10 V do zasilania całego urządzenia (max. 24 V – tyle maksymalnie przyjmuje sensor oraz stabilizator).
- Taśma klejąca, potrzebna między innymi do przyłączenia sensora do miernika zużycia energii elektrycznej, oraz lutownica, cyna, itd., potrzebne do zmontowania systemu.

Montaż sensora jest łatwy. W pierwszej kolejności dobrze jest przetestować cały układ na płytce stykowej, szczególnie jeśli mamy inny sensor odbiciowy lub inaczej skonstruowany licznik zużycia energii elektrycznej (niektóre liczniki mają wyjście impulsowe, które daje jeden impuls na Wh, które można wykorzystać analogicznie jak omawiany czujnik optyczny). Po przetestowaniu systemu możemy go zmontować, jednakże zanim zainstalujemy moduł z ESP8266, trzeba go zaprogramować. W tym celu korzystamy ze skryptu *init.lua*, który znaleźć można na **listingu 1**. Po zaprogramowaniu modułu możemy go włutować na płytkę uniwersalną, do której lutujemy następnie wyjścia z poszczególnych sensorów, zasilanie, itp.

## Baza danych i prezentacja wyników

Do działania systemu potrzebna jest również baza danych, do której moduł będzie mógł zapisywać wykonywane pomiary. Autor miał już zainstalowane Apache i MySQL na komputerze, używał ich m.in. do testowania stron www. Jednakże laptop nie jest najlepszym rozwiązaniem do pracy ciągłej. Dużo lepiej sprawi się tutaj komputer jednopłytkowy, taki jak Raspberry Pi. Nie potrzebuje on ekranu i pobiera maksymalnie około 7 W. Sam komputer jest także mały i może być zabudowany w niewidocznym miejscu.

Na Raspberry Pi można bez problemu zainstalować Apache i MySQL, jeśli nie ma ich tam domyślnie (zależne jest to od wykorzystanej dystrybucji Linuxa, jednakże większość z nich ma je domyślnie zainstalowane. Jeśli nie, to w Internecie znajdziemy wiele poradników, jak to zrobić). Po skonfigurowaniu Raspberry Pi i podłączeniu

go do sieci Wi-Fi można utworzyć bazę danych i rozpocząć jej zapełnianie.

Baza danych MySQL przechowuje dane w 2 bardzo prostych tabelach. Jedna tabela dla danych z liczników energii elektrycznej, druga dla prognoz energii wiatrowej. Moduł z ESP8266, po wykryciu każdego pełnego obrotu koła wskaźnika w mierniku zużycia energii elektrycznej, łączy się z serwerem i wchodzi na stronę serwowaną przez skrypt *electriciteitsmeter.php*. Skrypt ten zapisuje dane w tabeli *ElectMeterMeter* w bazie danych *SensiHome*.

### Prezentacja danych

Korzystając z przeglądarki internetowej, można wejść na jedną z czterech podstron, hostowanych na Raspberry Pi. Każda z nich przeznaczona jest do innych informacji:

- zużycie energii elektrycznej (w czasie rzeczywistym),
- strona konfiguracyjna, gdzie można ustawić m.in. zakres dat wyświetlania zużycia prądu,
- dzienne zużycie prądu (podzielone na godziny),
- aktualna siła wiatru na lokalnej farmie wiatrowej.

Przyjrzyjmy się im bliżej. Aktualne zużycie energii elektrycznej – *meter.php* + *meterstand.php*. Skrypt *meterstand.php* wywołuje dwa polecenia SQL na tabeli *ElectMeterMeter* w bazie danych *SensiHome*, która znajduje się na Raspberry Pi. Polecenia te zwracają dwie wartości:

- ostatni wpis w tabeli, czyli rzeczywiste zużycie energii elektrycznej w danym momencie,
- łączną liczbę obrotów koła pomnożoną przez 375 (375 obrotów to 1 kWh).

Z kolei skrypt *meter.php* uruchamia polecenie Ajax w celu odświeżenia strony. Główną zaletą tej metody jest to, że tak naprawdę realnie nie odświeżasz strony, a jedynie same dane. Tak więc strona *meter.php* pobiera dane z *meterstand.php* i aktualizuje je co 5 sekund. To, że są one aktualizowane co 5 sekund, nie oznacza, że są nowe. W przypadku systemu autora 1 kWh to 375 obrotów koła. Czujnik wysyła wiadomość do bazy danych dopiero po pełnym obrocie. Tak więc przy wysokim zużyciu nowe dane są dostępne w mniej niż 5 sekund, jednakże przy niskim zużyciu może minąć kilka minut, zanim nastąpi

Listing 2. Skrypt PHP odpowiedzialny za aktualizowanie bazy danych

```
<?php
$servername = "127.0.0.1";
$username = "user";
$password = "password";
$dbname = "SensiHome";

parse_str( html_entity_decode( $_SERVER["QUERY_STRING"] ), $out);

if ( array_key_exists( "toer" , $out ) ) {
// Utwórz połączenie
$conn = new mysqli($servername, $username, $password, $dbname);
// Sprawdź połączenie
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$toeren = $out["toer"];
$ActGeb = $out["Kwh"];
$stand = $out["stand"];

$sql = "INSERT INTO ElectMeterMeter (toer, Kwh, stand) VALUES ($toeren, $ActGeb, $stand)";

if ($conn->query($sql) === TRUE) {
echo "ok";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

aktualizacja. Za pomocą 2 przycisków na dole strony można przejść do kolejnej strony – prognozowania wiatru lub dziennego zużycia.

Strony konfiguracji dat i dziennego zużycia – *datepicker.php* + *dagverbruik.php*. Na stronie konfiguracyjnej (*datepicked.php*) wybiera się datę, dla której chcemy zobaczyć dokładniejsze dane. Stamtąd skrypt przekierowuje użytkownika do Google Chart, który pokazuje zużycie energii elektrycznej w wybranym dniu z podziałem na godziny.

Siła wiatru – *WindFor.php*. Strona, która prezentuje dane, pobrane z lokalnej farmy wiatrowej. *Windfo.php* prezentuje aktualną produkcję energii elektrycznej przez farmę oraz trzydniową prognozę wietrzności i wynikającej z niej produkcji energii elektrycznej. Strona ta pokazuje też wykres produkcji energii elektrycznej oraz prognozę na najbliższe 3 dni.

### Podsumowanie

Wystarczy tylko wejść na stronę serwowaną przez Raspberry Pi, aby mieć dostęp do szerokiej możliwości – zbierania danych na temat zużycia prądu w swoim domu, dostępie do energii z odnawialnych źródeł i do sterowania wybranymi odbiornikami prądu w domu.

Nikodem Czechowski

Źródło:

<https://bit.ly/34x5U4C>

REKLAMA

# Wstąp do Klubu AVT Elektronika

będziesz miał prawo do korzystania z szeregu przywilejów:

- do 50% zniżki w Sklepie AVT
- darmowe prenumeraty Wydawnictwa AVT
- do 50% zniżki w Ulubionym Kiosku
- Zapraszamy do zapoznania się z zasadami Klubu!

<http://bit.ly/2GaDwtQ>

