

Stacja pogodowa ESP8266 z OLED

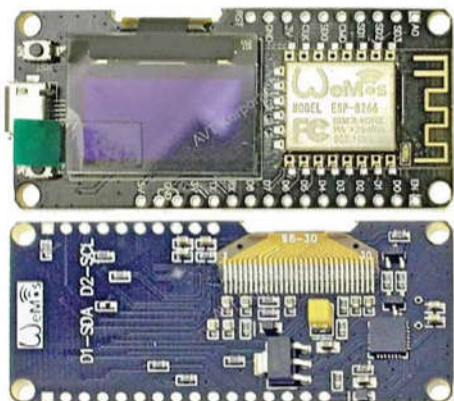


Popularne i tanie moduły Wi-Fi z ESP8266 pozwalają łatwo i szybko budować stosunkowo skomplikowane konstrukcje. ESP8266 poza podstawową pierwotną funkcją mostka Wi-Fi UART, po wgraniu własnego programu do modułu, może realizować przeróżne funkcje, np. bezprzewodowych czujników czy sterowników. Możliwość programowania z poziomu ArduinoIDE i bogactwo przykładów w Internecie ułatwiają budowę własnych urządzeń.

W artykule przedstawiono konstrukcję prostej stacji pogodowej mierzącej ciśnienie, wilgotność i temperaturę z kilku czujników. Wyniki są prezentowane na wyświetlaczu OLED oraz wysyłane na serwer *thingspeak*. Trwający kurs Arduino i zakończony C niewątpliwie ułatwią zrozumienie zasady działania programu stacji pogodowej.

Zadaniem stacji pogodowej jest pomiar temperatury przez maksymalnie 10 czujników temperatury, ciśnienia i wilgotności. Wyniki na bieżąco są prezentowane na monochromatycznym wyświetlaczu OLED 128x64. Wyświetlacz jest zamontowany na płycie modułu

Fot. 1 ESP-12F (fotografia 1), który



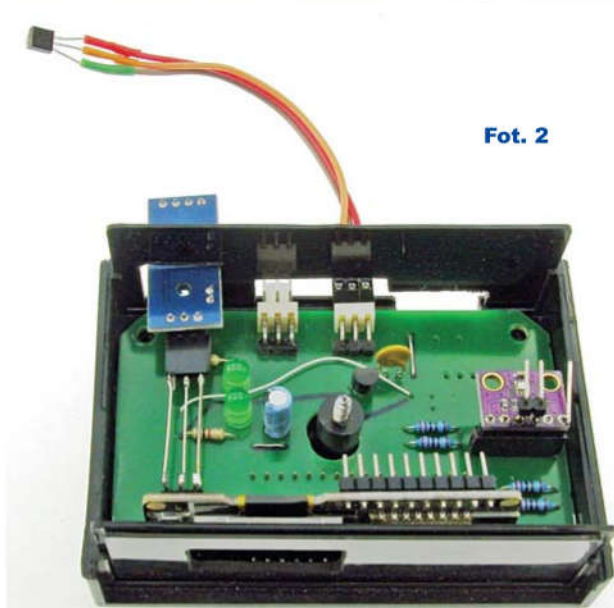
można nabyć w sklepie AVT, kod towaru: ARD-9534 i link – <https://sklep.avt.pl/modul-nodemcu-wifi-esp8266-esp-12f-z-wyswietlaczem-oled-0-96-cala-arduino.html>, w skrócie: <https://bit.ly/2HSLriy>.

Ciśnienie mierzy czujnik BMP280, kod towaru ARD-7206, link – <https://sklep.avt.pl/product/search?query=bmp280>, w skrócie: <https://bit.ly/2HRLPtX>.

Wilgotność mierzy moduł z czujnikiem DHT11 ARD-8608, link – <https://sklep.avt.pl/czujnik-wilgotnosc-i-temperatury-dht11-modul-arduino.html> w skrócie: <https://bit.ly/2EOuCQ8>.

Temperaturę mierzy popularny termometr cyfrowy DS18B20, link – <https://sklep.avt.pl/product/search?query=ds18b20> w skrócie: <https://bit.ly/2WJBXuB>.

Ze względu na to, że moduły te były opisywane na łamach EdW, nie ma sensu powielać tych informacji.

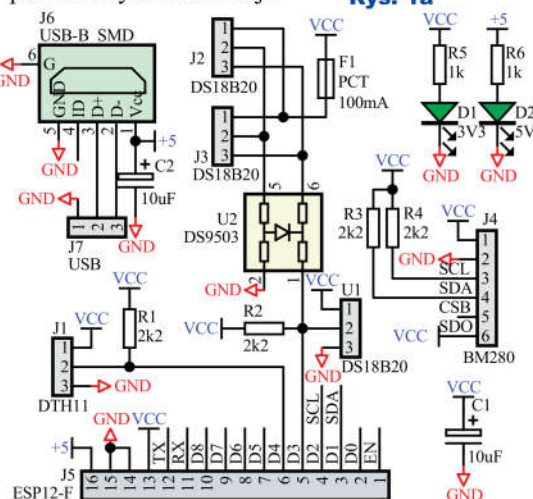


Fot. 2

Opis układu

Schemat stacji pogodowej pokazany jest na rysunku 1a. Zasilanie układu czerpane jest z dowolnego zasilacza z wtykiem mini USB. Złącze J5 łączy płytę bazową z modułem ESP8266. Na płycie bazowej znajdują się złącza J2, J3 dla zewnętrznych termometrów DS18B20. Na płycie jest miejsce dla jednego termometru (U1) oraz barometru BM280 (J4) i higrometru DH11 (J1). Układ U2 zabezpiecza mikrokontroler przed uszkodzeniem przez ładunki ESD i zakłócenia mogące pojawić się w przewodach łączących czujnik z mikrokontrolerem. Niestety ten element jest często pomijany w amatorskich konstrukcjach albo zastępowany przez diodę Zenera, która nie jest w stanie zapewnić dobrej ochrony, a wprowadza niepotrzebnie dużą pojemność.

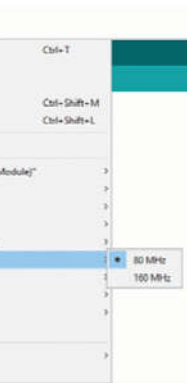
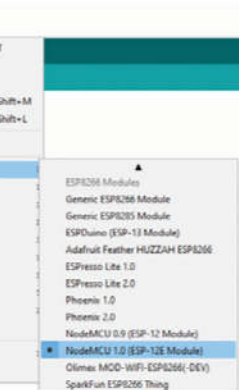
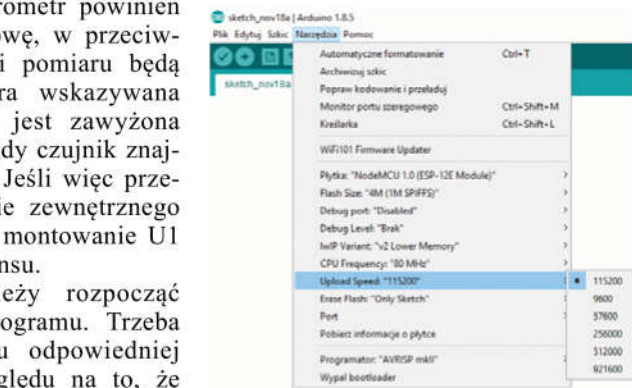
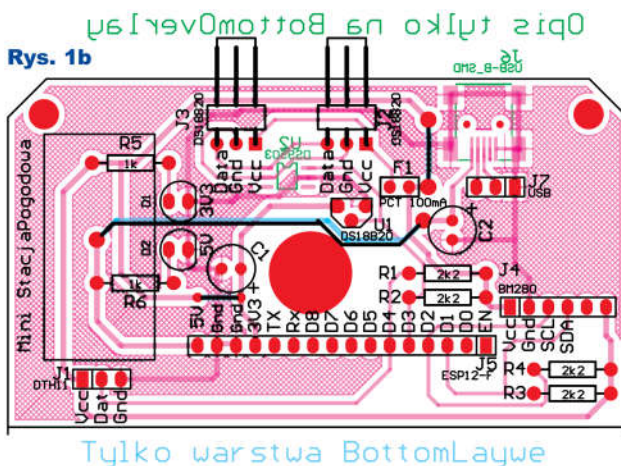
Rys. 1a



Montaż i uruchomienie

Podstawą układu jest gotowy moduł zawierający procesor i wyświetlacz (fotografia 2). Montaż całości możliwy jest więc na wiele sposobów. Można wykorzystać płytkę drukowaną, pokazaną na rysunku 1b, która będzie bazą dla pozostałych elementów i modułów (dokumentacja dostępna w Elportalu). Higrometr powinien wystawać poza obudowę, w przeciwnym wypadku wyniki pomiaru będą zaniżone. Temperatura wskazywana przez termometr U1 jest zawyżona o jakieś dwa stopnie gdy czujnik znajduje się w obudowie. Jeśli więc przewiduje się podłączenie zewnętrznego czujnika (czujników), montowanie U1 w obudowie nie ma sensu.

Uruchomienie należy rozpocząć od skompilowania programu. Trzeba pamiętać o wybraniu odpowiedniej wersji płytki. Ze względu na to, że wersji ESP jest kilka, na poniższych zrzutach ekranowych pokazano prawidłowy wybór.



Naturalnie muszą być zainstalowane wymagane biblioteki. Procesu instalacji bibliotek nie będę opisywał, ponieważ zajęłoby to kilka stron a ponadto na łamach EdW jest prowadzony kurs, w którym wszystko jest szczegółowo opisane i nie ma sensu powielać tych informacji. Jeśli program kompiluje się prawidłowo, można go wgrać do modułu ESP. Po uruchomieniu powinien pojawić się napis „Restart”, po chwili ekran główny:



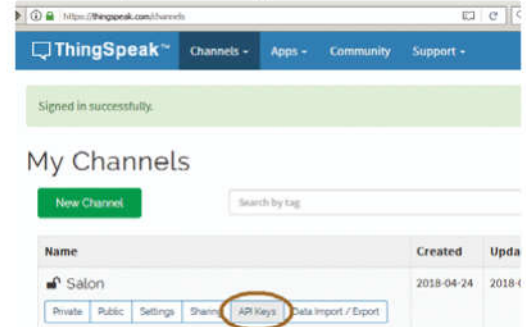
Ze względu na brak interfejsu użytkownika w postaci np. klawiatury, dane sieci Wi-Fi (SSID, hasło) i serwera thingspeak należy umieścić w kodzie źródłowym, po czym skompilować i wgrać do ESP. Aby zmienić dane sieci Wi-Fi, należy odszukać (CTRL+F) ciągu znaków „password”, po czym wpisać dane swojej sieci:

```
//parametry logowania do sieci WiFi
const char* ssid = "nazwaSieci";
const char* password = "haslo";
```

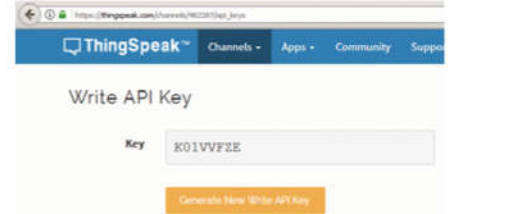
Jeśli dane mają być wysyłane na serwer thingspeak, należy odszukać:

```
//dane konta na thingspeak
const char * apiKey = "klucz";
unsigned long myChannelNumber = 1;
```

 i zmodyfikować apiKey. Skąd wziąć apiKey? Najpierw trzeba zarejestrować się na stronie thingspeak.com. Po rejestracji, potwierdzonej e-mailem i zalogowaniu na stronie, w zakładce „Channels”



klikamy na „Api Keys” i zobaczymy



gdzie można odczytać lub zmienić klucz.

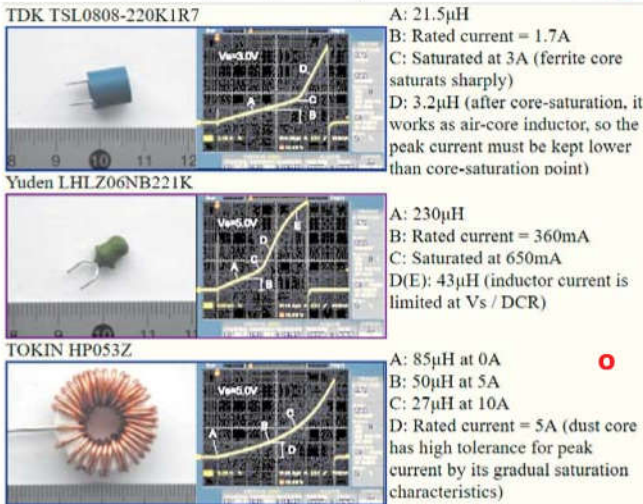
Program

Program, oczywiście w postaci szkicu .ino, jest dość prosty, zawiera komentarze, nie ma sensu omawiać go szczegółowo. W pętli „loop()” nie ma żadnych „delay” (co jednak wcale nie dowodzi, że funkcje biblioteczne z „delay” nie korzystają). Wszystkie opóźnienia realizowane są przy użyciu „millis()”, co oznacza kłopoty co ok. 49 dni. Najprostszym rozwiązaniem problemu jest reset np. co 24 godziny. W kodzie realizuje to linia `CzasDoResetu = millis() + 24 * 60 * 60 * 1000UL;`

wykonane pod koniec funkcji „setup()” i `if(millis() > CzasDoResetu){ Serial.print("\n*** Abort ***"); abort(); }`

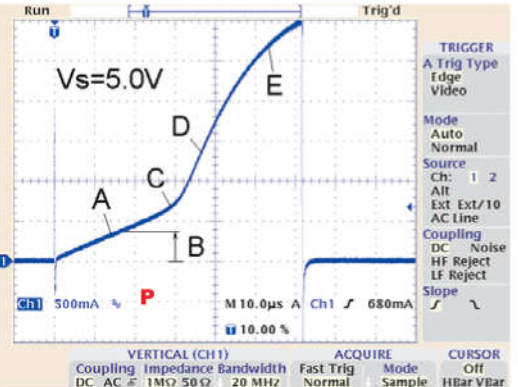
„w loop()”. Nie jest to jedyny sposób rozwiązania problemu z przepełnieniem licznika systemowego i nie zawsze możliwy do zaakceptowania.

Jedynie w całym kodzie polecenie `delay()` jest użyte w „startup()” w funkcji inicjalizującej sieć Wi-Fi. Nie było sensu, aby w tej sytuacji używać `millis()`, bo program kręci się w pętli do czasu nawiązania połączenia z siecią Wi-Fi. Jeśli to się nie uda w ciągu 30 sekund, realizowana jest dalsza część programu, bez komunikacji Wi-Fi.



(...) podłączone do zacisków DUT. Dioda zabezpiecza MOSFET przed przepięciem.

W obwodzie drenu MOSFET-a włączony jest rezystor 0,01Ω, z którego odbiera się przebieg do oscyloskopu w celu obserwacji płynącego prądu przez uzwojenie (...). Jednocześnie z tego rezystora pobierane jest napięcie sterujące tranzystorem Q2, który podaje sygnał zerujący wyjściowy układ czasowy 555 w przypadku zbyt dużych prądów (...)



Zadanie marcowe (omyłkowo oznaczone Jak7) nie było trudne i większość odpowiedzi była prawidłowa.

Nagrody – książki wydawnictwa BTC: Tomasz Jabłoński – **Graficzne wyświetlacze LCD w przykładach**.

otrzymują:

- Bogdan Jakubik – Warszawa,
- Jarosław Tarnawa – Zasole,
- Krzysztof Zych – Grabin.

Wszyscy uczestnicy zostają dopisani do listy kandydatów na bezpłatne prenumeraty.



Ta oryginalna wersja pokazana jest na rysunku N. Jeden ze stałych uczestników napisał: Układ składa się z układu sterującego zbudowanego z dwóch układów czasowych na kościach 555, pierwszy multiwibrator wytwarzający impulsy o częstotliwości ok. 655Hz (dla R1=2.2M i C6=1nF), oraz przerzutnika monostabilnego, który wytwarza impulsy o długości zależnej od wartości VR1+R2 i C4/C5=4.7nF/470pF. W rezultacie są to powtarzające się impulsy 655Hz. W takt tych impulsów włączany jest MOSFET N, który steruje uzwojenie

Układ czasowy przestaje podawać sygnał sterujący i zapobiega uszkodzeniom.

Na stronie: <http://elm-chan.org/works/lchk/report.html> podane są dodatkowe informacje, w tym zmierzone tym przyrządem charakterystyki szeregu cewek – rysunek O. Jeden z wykresów pokazany jest w powiększeniu na rysunku P. Kształt przebiegu prądu pozwala określić indukcyjność, maksymalny prąd pracy oraz daje wyobrażenie o innych parametrach badanej cewki.

Ciąg dalszy ze strony 22

Aby funkcja ta działała, musi zostać uruchomiony serwer www, realizowana obsługa zapytań itp. Wszystkie te funkcje można odnaleźć, szukając w kodzie źródłowym ciągu „webServer”. Strona nie jest zbyt okazała, a jeśli miałaby zawierać grafikę, należałoby posłużyć się systemem plików. Dokładny opis można znaleźć w EP 12/2018.

Stacja pogodowa może być punktem wyjścia do podobnych, bardziej rozbudowanych projektów.

SaS
sas@elportal.pl

Wykaz elementów

R1,R2,R3,R42,2kW THT	J2,J3NS25-G6
R5,R61kW THT	J4BM280 moduł
C1,C2	10uF/6.3V ce6.3/2.5 elektrolit	J5ESP12-F moduł
D1,D2THTDioda LED zielona 5mm	J6MUSB-B5-S-RA-SMT
U1DS18B20 TO-92		gniazdo USB-B Mini
U2DS9503 SO-6		
F1PCT 100mA THT		
J1DTH11 Moduł		
	+ gniazdo Goldpin 2,54mm		

Płytki drukowane są dostępne w Sklepie AVT jako AVT3249

Ciąg dalszy ze strony 69

Napięcie pozyskane diodą D632 odpowiada wartości wejściowego napięcia sieci. Napięcia pozyskane podczas obu faz podane są na dzielnik rezystancyjny, którego wartości ustalają wagi, z jaką oba te napięcia mają wpływ na sygnał Protect-2. Nietrudno przeliczyć, że dla oporów odpowiednio 12kΩ i 22kΩ napięcie w węzle X przyjmie wartość dodatnią dla współczynnika wypełnienia PWM większego od ok. 35%. Nie bardzo wiadomo, jaki poziom sygnału zabezpieczenia (Protect-2) uznany

jest za aktywny. Uwzględniając jeszcze obecność diody D650, należy uznać, że to proste zabezpieczenie pozwala przetwornicy na pracę z PWM do ok. 40%, powyżej którego układ logiki nakaże wyłączenie odbiornika. Zmierzone, że w nominalnych warunkach przetwornica kluczuje z wypełnieniem zaledwie 20%, a Protect-2 stanowi ochronę przed pracą przy zbyt niskim napięciu sieci energetycznej, jak również może pełnić funkcję zabezpieczenia typu

brown_out (wstrzymania kluczowania przetwornicy, gdy napięcie na kondensatorze za mostkiem Graetza opadnie poniżej krytycznej wartości), które jest często wbudowane w strukturę scalonych sterowników, angażując sporą liczbę elementów (co jest jednak bezbolesne, jeśli wszystko zamknięte jest w układzie scalonym).



Karol Świerc
rtv@silnet.pl