

# Programowanie ESP8266

## Radioodbiornik sterowany bezprzewodowo

Kolejna część kursu, kolejna szansa, żeby przedstawić niesamowite możliwości niepozornego układu ESP8266 w module ESP-12. Moduł RDA5807 to tanie odbiorniki stereofoniczne FM z RDS. Dzięki temu, z użyciem małej liczby elementów, można wykonać w pełni funkcjonalne radio FM. Dzięki funkcji RDS możemy takie radio wyposażyć w wyświetlacz LCD, na którym będą pokazywane komunikaty (nazwa odbieranej stacji, nazwy utworu lub audycji i inne informacje przesyłane tą drogą).

Typowo radioodbiornik trzeba obsługiwać za pomocą gałek, guzików lub pilotem. A co, gdyby wyposażyć radio w moduł Wi-Fi? Możliwość obsługi powiększa się o telefony, komputery i każde urządzenie wyposażone w ten interfejs.

Cały układ jest oparty o dwa moduły. Moduł sterujący – ESP8266 w wersji ESP12 oraz moduł radia FM RDA5807. Komunikacja pomiędzy układami to I<sup>2</sup>C. Na **rysunku 1** pokazano sposób przyłączenia układów oraz podstawową aplikację układu RDA5807. Rezystory zasilające (R12 i R11) są opcjonalne, ponieważ niektóre z modułów ESP mają wbudowane rezystory podciągające linie danych i zegarową do zasilania. Należy wtedy ich nie montować, ponieważ w przeciwnym wypadku komunikacja może nie działać poprawnie lub zawieszać się w nieoczekiwanych momentach. Oba układy pracują przy zasilaniu napięciem 3,3 V, co pozwala na zastosowanie jednego zasilacza dla całego modułu.

Na schemacie ideowym pokazano podstawową aplikację modułu RDA5807. Pochodzi ona z dokumentacji technicznej producenta. Indukcyjność cewki L1 jest na tyle mała, że można z powodzeniem własnoręcznie wykonać cewkę z wykorzystaniem dostępnych informacji, np. używając opisu z Internetu. Dodatkowo, do modułu ESP12 dodano 2 guziki, przydatne przy rozwijaniu oprogramowania, pozwalają one na szybkie ustawienie modułu w trybie aktualizacji oprogramowania.

Kontynuując trend z wcześniejszych artykułów, język wykorzystany podczas tworzenia kodu to Arduino, natomiast środowisko to Arduino IDE. Cała idea jest oparta o nieskomplikowany model sterowania na zasadzie rozkazów wysyłanych w pakietach UDP jako string. Moduł przyjmuje komunikację w postaci `start:command:value:stop`, gdzie:

- Start – jest wartością rozpoczynającą paczkę danych, to ta wartość jest szukana jako pierwsza w pakiecie.
- Command – komenda wysłana jako string.
- Value – opcjonalna wartość, zależnie od komendy, może zostać wysłana lub nie.

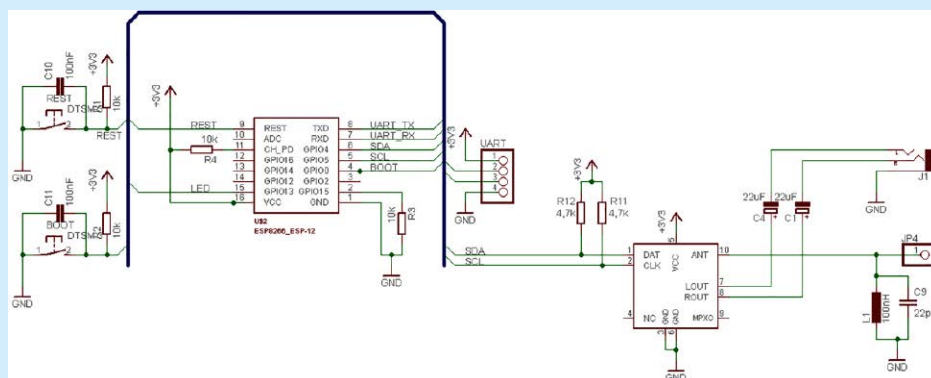
- Stop – Wartość kończąca paczkę danych.  
W obecnej wersji kodu zaimplementowane są 2 możliwe rozkazy. Są to:

- **start:set:[value]:stop**. Jest to rozkaz ustawiający częstotliwość odbiornika. Wartość, która powinna być wysyłana do układu, jest obliczana następująco:  $(Freq * 10) - 870 = value$ . Wartość *Freq* to wartość „normalna” częstotliwości (np. 101,7 MHz lub 92,5 MHz). W tym wypadku argument *value* podczas wysyłania komendy jest wymagany.

- **start:vol:[value]:stop**. Jak wcześniej, również tutaj argument w postaci liczbowej jest wymagany. Według dokumentacji technicznej układu RDA5807 za regulację mocy odpowiadają bity 0:3 w rejestrze 0x05. Skala mocy jest logarytmiczna, niestety jest jedynie 16 poziomów sterowania tą mocą. Ustawiając bity 0:3 w tym rejestrze na 0, wprowadzimy układ w stan MUTE oraz wyjście przełączy się w stan wysokiej impedancji.

Kod jest przygotowany do rozszerzenia funkcjonalności modułu o nowe komendy. W następnej wersji mogłyby to być komendy odczytujące wartość czasu oraz daty pobranej z systemu RDS lub informacji o obecnej stacji.

Na potrzeby kursu przygotowana została aplikacja na telefony z systemem Android (4.4.x i wyższe – **rysunek 2**). Ma ona możliwość zmiany poziomu głośności oraz prostego ustawienia częstotliwości stacji. W przygotowane miejsce do wpisania częstotliwości należy wpisać faktyczną, oczekiwaną częstotliwość stacji (np. 103,0 lub 91,5), obliczenia zostały zaimplementowane w działanie aplikacji. Do sterownika powinny dochodzić jedynie gotowe dane.



Rysunek 1. Schemat ideowy aplikacji (zrzut ekranu z programu Eagle)

W aplikacji nie trzeba już podawać adresu IP modułu. Dzięki wykorzystaniu w tym celu broadcastu moduł sam odnajduje aplikację i rozpoczyna komunikację. W momencie uruchomienia aplikacji, pojawia się tylko jeden przycisk, jest to zabezpieczenie, by nie używać aplikacji bez urządzenia odbiorczego. Po kliknięciu przycisku „połącz” aplikacja wyśle metodą broadcast string „esp\_8266\_radio”, biorąc pod uwagę to, że broadcast rozchodzi się tylko w danej sieci, układ oraz smartfon muszą być, w tym wypadku, podłączone do jednej sieci Wi-Fi. Moduł ESP8266 nasłuchuje na porcie 8080 i gdy otrzymuje pakiet, jest on parsowany na 2 możliwości.

W pierwszej, jest to pakiet od urządzenia, które chce się dopiero połączyć. W drugiej, jest to komenda od urządzenia, które już się połączyło. W takim wypadku na adres nadawcy jest odsyłana wartość *ok* oznaczająca koniec parowania i rozpoczęcie normalnej pracy. Obecnie jest zabezpieczenie, by tylko jedno urządzenie mogło sterować ESP1. Gdy układ odbierze odpowiedni pakiet z informacją *esp\_8266\_radio*, odczyta i zapamięta adres IP urządzenia, które pytało o moduł ESP. Do momentu, aż inne urządzenie nie rozpocznie procedury odszukiwania modułu, sterować radiem może tylko urządzenie, które jako ostatnie z powodzeniem zakończyło proces odszukiwania i parowania. Później pakiety są już nadawane bezpośrednio pomiędzy smartfonem i ESP8266.

## Konfiguracja

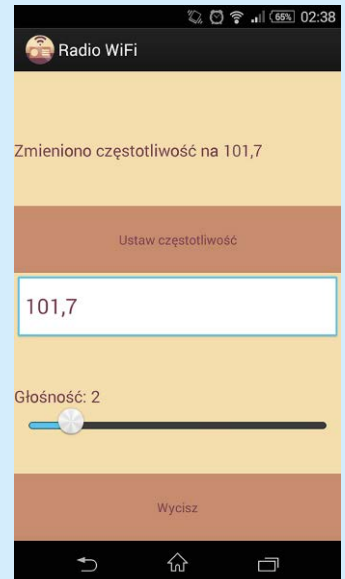
Funkcja *Setup()* inicjalizuje komunikację I<sup>2</sup>C oraz UART z prędkością 9600 b/s. Następnie układ próbuje przyłączyć się do sieci Wi-Fi z użyciem wcześniej podanych *SSID* i *PASSWORD*. Na koniec zostaje skonfigurowany układ RDA5807 oraz ustawiona częstotliwość na 105,7 MHz ( $105,7 \cdot 10 - 870 = 187$ ).

W pętli głównej odbierane są pakiety i parsowane tak, jak zostało to wyjaśnione wyżej. Przed wgraniem programu

należy pamiętać, by ustawić ESP8266 w tryb „flash”, ściągając pin GPIO0 do GND. Po restarcie na monitorze powinny się pojawić informacje o obecnym stanie urządzenia oraz o nowych pakietach.

Oprogramowanie zostało napisane w taki sposób, by łatwo było dodać nową komendę. Potrzeba jedynie dodać nowy wpis do struktury ENUM, następnie ustawić warunki w funkcji *get\_command* oraz w instrukcji switch dodać kod, który ma się wykonać po wykryciu danej komendy, dzięki czemu jest to bardzo dobra baza do rozwijania tego oprogramowania jako sterownika z wykorzystaniem aplikacji na smartfonie, komputerze czy też mikrokomputerze takim jak Raspberry PI lub BeagleBone. Jak widać, da się niskim kosztem zrobić stereofoniczny odbiornik radiowy, obsługujący RDS, sterowany telefonem klasy Smartfon. Wystarczy wykorzystać do tego ESP8266, który ma o wiele większe możliwości niż zaprezentowane w tym kursie. Moc obliczeniowa układu SoC w module jest na tyle duża, że poradzi sobie spokojnie z o wiele bardziej wymagającymi aplikacjami, takimi jak monitorowanie wielu czujników, obsługa wyświetlacza oraz sterowanie urządzeniami.

JAKUB KISIEL



Rysunek 2. Ekran główny aplikacji dla smartfona

REKLAMA

Wydanie specjalne magazynu *Gitarzysta* wyciąga rękę do wszystkich tych, którzy chcą nagrywać muzykę w domu – a badania pokazują, że jest to obecnie niezwykle szeroka i dynamicznie rosnąca grupa odbiorców. Zaczynamy od opisanie kompletnych podstaw, sposobów podłączania instrumentu, dostępnych programów do nagrywania muzyki, sprzętu niezbędnego do tego. Polecamy wybrane produkty z działów gitara elektryczna, gitara akustyczna, gitara basowa, perkusja. Omawiamy zasady nagrywania tych instrumentów, dzielimy się poradami zawodowców, podpowiadamy jak wykonać miks i mastering. Co więcej omawiamy także strukturę piosenki i tego jak powinien zostać napisany przebieg. Na koniec kilka wskazówek dotyczących biznesu muzycznego i PR. *Przewodnik Domowe Studio* to pozycja obowiązkowa dla każdego!



PRZEJRZYSZ I KUPISZ NA WWW.ULUBIONYKIOSK.PL