

Biblioteka MySensors a Domoticz

System bezprzewodowych czujników z tanich modułów Arduino i nRF24L01+

Do napisania tego artykułu skłoniły mnie próby uruchomienia radiowego systemu komunikacji pomiędzy dwoma zestawami Arduino z modułami nRF24L01+. Zadanie okazało się niełatwe. Dość długie i niestety nie do końca udane prace nad własnym programem wspierałem wiedzą zaczerpniętą z Internetu. Poszukując nowych informacji na ten temat, trafiłem w końcu na stronę opisującą bibliotekę MySensors. Próbowałem wcześniej realizować swoje pomysły, stosując kilka innych bibliotek obsługujących nRF24L01, ale tylko biblioteka MySensors wyróżniła się tym, że od razu i bez problemów umożliwiła radiową wymianę danych pomiędzy kontrolerami. Ponadto drugim ważnym atutem biblioteki MySensors jest jej bardzo dobra współpraca z oprogramowaniem Domoticz.

Informacje tu zawarte będą przydatne w szczególności dla osób zaczynających swoje eksperymenty z uruchomieniem własnej, domowej sieci czujników radiowych i zdalnie sterowanych urządzeń. Układy takie stale cieszą się dużym zainteresowaniem elektroników hobbyistów i nie tylko, dlatego zapraszam do przetestowania proponowanych rozwiązań. Cały testowy systemik można zmontować, oprogramować i uruchomić w kilkadziesiąt minut.

Do niedawna poważnym utrudnieniem w realizacji takich projektów były wysokie ceny podzespołów oraz ich skomplikowane aplikacje. Ich uruchomienie zajmowało sporo czasu, a także wymagało

specjalistycznej wiedzy i umiejętności. Na szczęście sytuacja ta zmieniła się dzięki pracy różnych społeczności i łatwemu dostępowi do takich platform, takich jak Arduino i nRF24L01+. Do tej pory niewiele pisano w czasopiśmie wydawanym przez AVT na temat możliwości wykorzystania modułów radiowych nRF24L01, a szkoda, bo przecież są one bardzo atrakcyjne ze względu na swoje właściwości i cenę. W Internecie można znaleźć wiele informacji na temat współpracy Arduino z nRF24L01, ale nie wszystkie przykłady jest łatwo dostosować do swoich potrzeb, a część z nich w ogóle nie działa.

Powstało co najmniej kilka dobrych bibliotek wspierających współpracę Arduino z transceiverem nRF24L01, ale użyta w prezentowanych programach biblioteka MySensors wydaje się idealnie przygotowana do pracy w systemach automatyki domowej. Pozwala ona na szybkie, wręcz ad hoc wdrażanie pomysłów w dziedzinach zdalnego pomiaru i sterowania z możliwością tworzenia sieci czujników. Zawarte w niej mechanizmy pozwalają na zachowanie energooszczędności w układach pracujących pasywnie, które dzięki temu mogą być zasilane z baterii przez długi czas. Zaprezentowane tu programy wykorzystują mały ułamek możliwości biblioteki, mającej wiele innych zalet. Na przykład zabezpieczenie transmisji danych czy zdalne radiowe programowanie mikrokontrolera – FOTA. Biblioteka współpracuje również z innymi kontrolerami np. ESP8266 oraz z różnymi typami modułów radiowych. Stosując bezprzewodowe czujniki w połączeniu z popularnym oprogramowaniem Domoticz, otrzymamy system sterowania różnymi obiektami z wizualizacją wskazań. Właściwie zamiast dalszego opisu wystarczyłoby podać link do strony projektu <https://www.mysensors.org>, na której wszystko jest dobrze opisane. Dla zachęcenia czytelników, którzy jeszcze się tym tematem nie zajmowali, dalej przedstawię, jak w kilku krokach

uruchomić przykładowy, mały system z wizualizacją zbieranych danych. Zakładam przy tym, że osoby chcące przetestować proponowane rozwiązania mają chociaż minimalną wiedzę na temat pracy w środowisku Arduino.

Podstawowe informacje o systemie

Urządzenia z oprogramowaniem opartym na bibliotece MySensors tworzą sieć węzłów przekazujących dane do bramy. W celu realizacji projektu musimy wykonać jeden układ bramy „Gateway” i co najmniej jeden układ węzła „Node”. Węzeł przekazuje przez radio dane z podłączonych do niego fizycznych czujników i urządzeń wykonawczych określanych jako „Child”. Każdy węzeł może obsługiwać do 256 czujników. W sieci teoretycznie może być 256 węzłów, ale niektóre adresy są zarezerwowane przez system. Jak wspomniałem, drugim niezbędnym elementem potrzebnym do pracy sieci jest brama, która obsługuje wszystkie węzły. Brama pośredniczy w wymianie danych przez radio, przekazując je przez łącze szeregowe do komputera. Komputer z programem kontrolującym to ostatni, trzeci element systemu. Program kontrolujący użyty w tym przykładzie to Domoticz, który przetwarza otrzymane dane i umożliwia ich wizualizację oraz inteligentne sterowanie urządzeniami wykonawczymi. Komputer PC lub inny, wraz z oprogramowaniem Domoticz, pracuje w sieci jako serwer, dzięki czemu otrzymane dane możemy obserwować na urządzeniach w niej zalogowanych. Taki system może być sterownikiem własnego inteligentnego domu i jest bardzo ciekawy ze względu na swoje OGROMNE możliwości.

Krok pierwszy – potrzebny sprzęt

Domyślałam się, że prawie każdy elektronik ma w domu zestaw Arduino. W moich zapasach od wielu miesięcy leżały również moduły nRF24L01+, które wcześniej kupiłem ze względu na ich niską cenę. Do projektu wymagane są transceivery oparte na układzie nRF24L01+. Ten znaczek plus oznacza nowszą wersję zastosowanego układu o nieco szerszych możliwościach. Wszystkie transceivery, które dotychczas kupiłem, były z układem scalonym nRF24L01+, ale nie zawsze sprzedawcy określają ich typ i należy na to zwrócić uwagę.

Do budowy najmniejszej „sieci” potrzebujemy dwóch sztuk Arduino z procesorem Atmega328, dysponujących wyprowadzonym zasilaniem 3,3 V (zasilanie wymagane dla radia), dwóch modułów nRF24L01+ i przewodów połączeniowych. Stosując moduły nRF24L01, koniecznie musimy im zapewnić zasilanie napięciem nieprzekraczającym 3,6 V. Jeżeli moduł Arduino nie ma wyprowadzonego napięcia 3,3 V o wydajności prądowej co najmniej 30 mA, to trzeba zastosować zewnętrzny stabilizator. Moduły o większej mocy, typu nRF24L01+PA+LNA, pobierają prąd nawet do 120 mA.

Po połączeniu Arduino z radiem do układu dołączamy czujniki. Praktycznie mogą to być dowolne sensory obsługiwane przez Arduino. Do testowego układu zastosowano czujnik DS18B20 i potencjometr. W przygotowanych do tego projektu przykładowych węzłach pierwsze obsługiwane przez węzeł urządzenie (child 0) to dioda LED, której jasność będzie można regulować, potencjometrem z bramy i innego węzła. Pozostałe „dzieci” to czujniki. Child 1 będzie przysyłał dane odczytane z czujnika temperatury DS18B20. Child 2 prześle informacje o stanie wejścia przetwornika analogowo-cyfrowego A1, który mierzy napięcie zadane z potencjometru i podaje wynik w procentach. Odczyt ten steruje jasnością diody LED w innym węzle. Czujnik child 3 generuje przyrastającą wartość pseudotemperatury o przebiegu piłokształtnym w zakresie od 20 do 25 stopni. Ostatni, czwarty czujnik będzie przekazywał aktualną wartość napięcia zasilającego procesor. Informacja taka jest szczególnie przydatna przy zasilaniu węzła z baterii. Precyzja pomiaru napięcia VCC zależy od właściwego podania w programie napięcia referencyjnego Napięcie_AREF, które trzeba zmierzyć w użytym procesorze (domyślnie jest 1,1 V). Drugi programowy parametr korygujący wynik pomiaru zasilania procesora to VccKorektor, który powinien mieć wartość

Tabela 1. Połączenia modułu radiowego z Arduino. W prezentowanych przykładach przerwanie IRQ nie są używane

Uno Nano Pro-Mini	Mega	NRF24L01	Kolor
GND	GND	GND	czarny lub brązowy
3,3 V lub 5 V > stab > 3,3 V	3,3 V	VCC	czerwony
9	49	CE	pomarańczowy
10	53	CSN	żółty
13	52	SCK	zielony
11	51	MOSI	niebieski
12	50	MISO	fioletowy
*NC	*NC	IRQ	szary

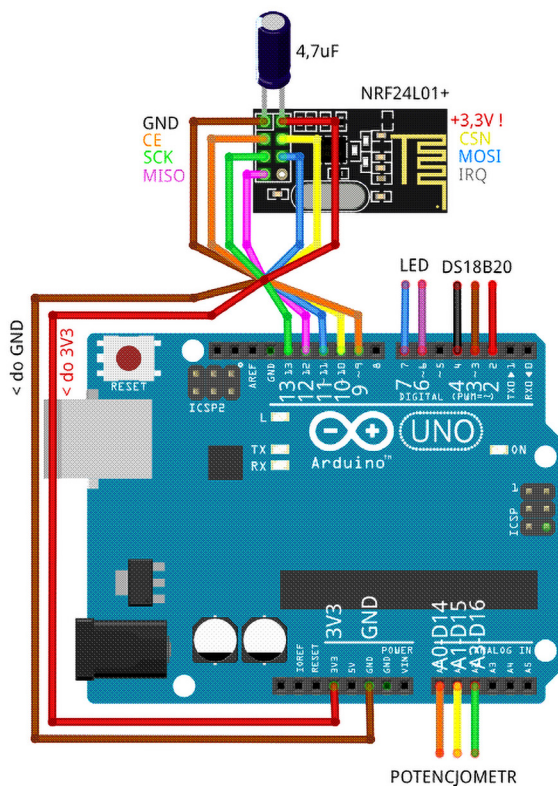
napięcia faktycznego VCC podzielonego przez wartość napięcia wyświetlonego przez czujnik.

Jeżeli nie mamy pod ręką potrzebnych elementów na czujniki, to w ostateczności do pierwszych testów nie musimy ich podłączać. Węzeł prześle wtedy tylko wartość napięcia zasilania VCC i pseudo-temperaturę a pozostałe dane będą przypadkowe.

Krok drugi – łączenie modułów

Połączenie modułów radiowych z płytkami Arduino i czujnikami wykonujemy, wzorując się na załączonych rysunkach. Wskazane jest, by przewody były jak najkrótsze. Do połączeń możemy użyć przewodów z kolorowych taśm, co bardzo ułatwia montaż. W tabeli 1 zestawiono połączenia modułu radiowego z Arduino w zależności od jego typu.

Radio i czujniki są tak samo dołączone zarówno dla układu węzła, jak i dla bramy. Dlatego każdy z widocznych na rysunkach 1...4 układów może pracować jako węzeł lub brama, a jego funkcja zależy



Rysunek 1. Połączenia pomiędzy modułem radiowym a Arduino Uno

tylko od programu wgranego do procesora. Pamiętajmy o dołączeniu kondensatora pomiędzy przewody zasilające radio. Jest to zalecane przez twórców MySensors i dość istotne przy stosowaniu połączeń długimi przewodami. Wyprowadzenia sterujące modułu radia mogą bezpośrednio współpracować z układami zasilanymi napięciem 5 V.

Do modułu Arduino dołączamy czujniki, które w tym przykładzie, dla uproszczenia układu połączeń, są zasilane z wyjść procesora, ale można zasilac je wprost z linii zasilających płytki. Oczywiście

musimy pamiętać o rezystorach przy diodzie LED i czujniku DS18B20. W związku z łączeniem układów mam jedną praktyczną uwagę. Proponuję w układach montowanych do dłuższej pracy stosować połączenia lutowane. Jednym z częściej występujących błędów w testowanych czujnikach była utrata komunikacji pomiędzy modułem Arduino i radiem. Po poprawieniu połączeń na „goldpinach” i restarcie wszystko wracało do normalnej pracy. Niestety te goldpiny, dołączane teraz do modułów, nie są już takie gold, jak były kiedyś ;).

Krok trzeci – oprogramowanie Arduino

Pora na wgranie programów do Arduino. Do kompilacji oprogramowania testowego zastosowano najnowszą wersję biblioteki MySensors 2.3.0. Dodatkowo do obsługi czujnika DS18B20 jest potrzebna biblioteka OneWire.

W celu umożliwienia szybkiego uruchomienia proponowanych układów przygotowane zostały cztery szkice, które można pobrać wraz z niezbędnymi bibliotekami z naszego serwera FTP. Układ bramy z czujnikami programujemy kodem ze szkicu Brama_z_Czujn_Stejuje_LED_w_14.ino. W tym małym systemie testowym brama z czujnikami podłączonymi bezpośrednio do niej pracuje dobrze. Należy jednak pamiętać, że dla płynności działania układu bramy powinno się unikać obsługiwanie przez nią zbyt wielu czujników. Może to powodować gubienie pakietów odbieranych przez radio. Dlatego jest wskazane, aby procesor bramy wykonywał pomiary bardzo rzadko. Zbyt duży ruch w sieci, czyli częste przesyłanie informacji o stanie czujników lub obsługa zbyt wielu czujników, może zakłócać jej pracę.

Brama współpracująca z wieloma zdalnymi węzłami powinna przekazywać tylko dane radiowe. Program dla bramy bez czujników, obsługującej tylko połączenia radiowe, to szkic GatewaySerial_z_LED.ino, który jest niemal identyczny, jak szkic bramy zawarty w przykładach biblioteki MySensors. W proponowanych programach do obsługi bramy dioda LED sygnalizuje krótkimi błyskami odbiór transmisji z węzłów.

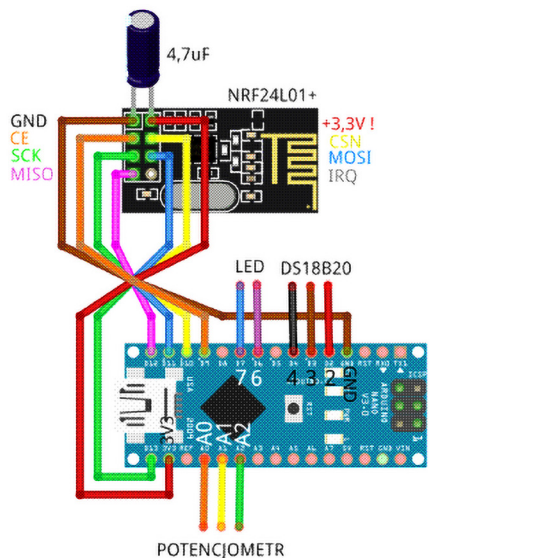
Do układu pierwszego węzła wgrywamy szkic Node14_z_LED_i_Czuj.ino. Zgłosi się on w systemie jako numer 14 (0x0E). Do ewentualnego kolejnego węzła proponuję program ze szkicu Node30_z_LED_i_Czuj.ino. Powyższe dwa szkice współpracują ze sobą w zakresie wzajemnej regulacji jasności świecenia diod LED przez potencjometry. Dodając kolejne nowe węzły, należy w ich programie w linii #define MY_NODE_ID xx zmienić numer identyfikacyjny węzła z xx na inny np. kolejny z zakresu od 1 do 244. Zastrzeżone są numery 0 dla bramy i 255 rozgłoszeniowy. Można też wgrać i testować inne programy z przykładów biblioteki MySensors. Na przykład szkic PassiveNode.ino, który co 2 sekundy generuje i wysyła losowe wartości temperatury. Jeżeli programy wgrane do zestawów miały zdefiniowaną opcję #define MY_DEBUG, to na monitorze przyłączonym do portu szeregowego węzła możemy obserwować zakodowane komunikaty o ich stanie i trybie pracy.

Dużo ciekawych informacji na temat pracy bramy i czujników uzyskamy, podłączając port szeregowy bramy do komputera z uruchomionym programem MYSCONTROLLER – więcej na stronie projektu. Programy kompilujemy, wgrywamy i gotowe.

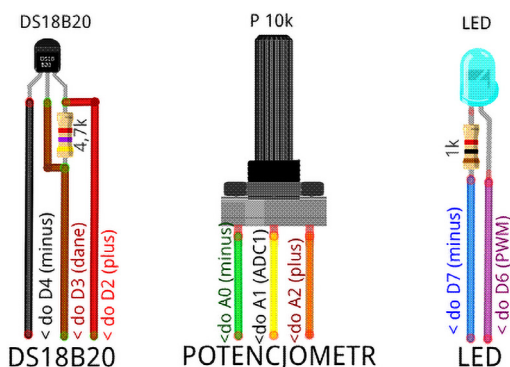
Krok czwarty – rejestracja i wizualizacja danych za pomocą Domoticza

Program Domoticz jest jednym z popularniejszych programów rejestrujących i przetwarzających różne dane z rozmaitych systemów automatyki domowej. Może on pracować między innymi na komputerze PC lub Raspberry Pi. Na PC z Windowsem XP nie udało mi się uruchomić tego oprogramowania, natomiast pod kontrolą Windows 7 i 10 wszystko ruszyło bez problemu. Instalacja i podstawowa konfiguracja przebiegają szybko i sprawnie.

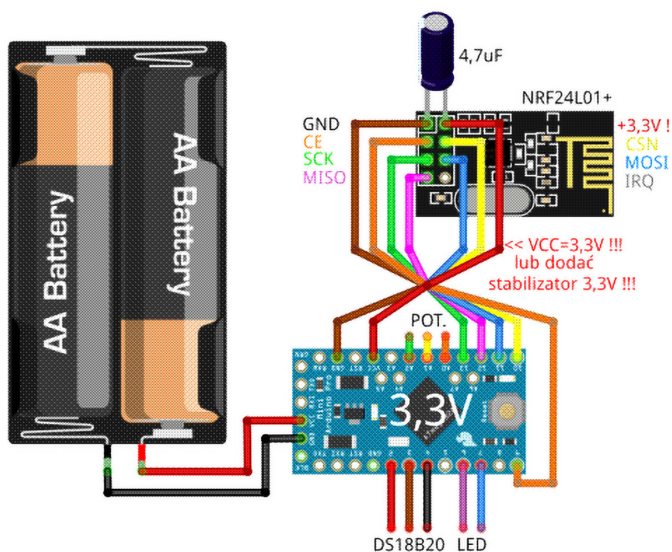
Z strony www.domoticz.com/downloads/ ściągamy stabilną wersję programu Domoticz dla Windows. Rozpakowujemy plik i przeprowadzamy standardową instalację. Po instalacji program uruchamia się jako strona internetowa i jest po angielsku. Wybieramy u góry



Rysunek 2. Połączenia pomiędzy modułem radiowym a Arduino Nano



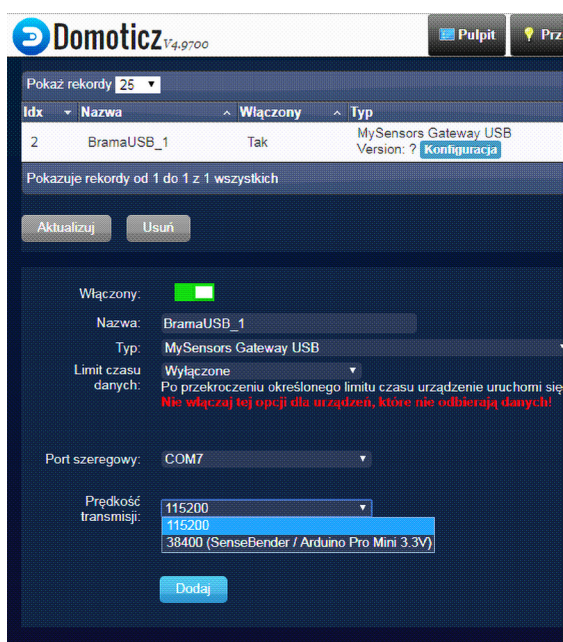
Rysunek 3. Połączenia czujnika, potencjometru i LED



Rysunek 4. Połączenia pomiędzy modułem radiowym a Arduino Pro Mini

zakładkę Setup → Settings, gdzie ustawiamy w Language opcję Polish, a w Location określamy swoją lokalizację. Zatwierdzamy nowe ustawienia, klikając na czerwone pole Apply Settings. Teraz dołączamy do komputera bramę i wybieramy u góry Konfiguracja → Sprzęt. Na stronie, która zostanie wyświetlona (**rysunek 5**), w polu Nazwa nadajemy nazwę naszej bramie. Wybieramy typ – MySensors Gateway USB i określamy port COM, do którego jest dołączone Arduino obsługujące bramę. Na koniec wybieramy prędkość transmisji – domyślnie jest to 115200 dla Arduino taktowanego kwarcem 16 MHz. Jeżeli Arduino sterujące bramą taktowane jest z częstotliwością 8 MHz, to ustawiamy prędkość na 38400 bps. Przyciskiem Dodaj na dole strony dodajemy nową bramę do listy sprzętu.

Następnie włączamy zasilanie węzła. Jeżeli znajduje się on w zasięgu bramy, to w logach Domoticza (Konfiguracja → Logi) pojawią się informacje o transmisjach z czujników. Czujniki trzeba teraz



Rysunek 5. Dodanie bramy do sprzętu Domoticza



Rysunek 6. Widok pulpitu z wybranymi czujnikami



Rysunek 7. Lista urządzeń połączonych z Domoticzem

Ważniejsze parametry modułów radiowych:

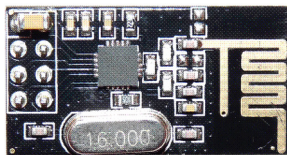
- Napięcie zasilające od 1,9 do 3,6 V.
- Wyprowadzenia sterujące mogą współpracować z układami zasilanymi napięciem 5 V.
- Cztery poziomy mocy nadawania.
- Trzy prędkości transmisji: 250 kbps, 1 Mbps i 2 Mbps.
- 125 kanałów radiowych: od 2,4 GHz do 2,525 GHz z krokiem co 1 MHz.
- Sygnał przerwania po odebraniu lub wysłaniu pakietu danych.
- Funkcje automatycznej wymiany danych adresowanymi połączeniami z kontrolą długości pakietu, CRC i potwierdzeniem ACK.
- Pobór prądu przez nRF24L01+: 12,3 mA w trybie odbioru RX przy transmisji 2 Mbps, 11,3 mA w trybie nadawania TX przy maksymalnej mocy nadawania 0 dBm, 900 nA w trybie Power Down.
- Pobór prądu przez nRF24L01+ PA+LNA: 45 mA w trybie odbioru RX przy transmisji 2 Mbps, 115 mA w trybie nadawania TX przy maksymalnej mocy nadawania 0 dBm, 4,2 µA w trybie Power Down.

dodać do listy urządzeń w Domoticzu, więc wybieramy Konfiguracja → Urządzenia. Na wyświetlonej stronie powinna się pokazać lista urządzeń, jakie połączyły się przez bramę z Domoticzem (**rysunek 7**). Jeżeli wykonaliśmy układ bramy z czujnikami, to również one będą na liście. Pojawienie się nowych czujników na liście urządzeń może potrwać do kilku minut, zależnie od programu węzła. Z prawej strony wierszy opisujących czujniki są zielone kółeczka. Klikając na kółko, otworzymy okno, w którym możemy zmienić nazwę czujnika i dodać go do obsługiwanych urządzeń. Po dodaniu czujników w zakładkach Przełączniki, Temperatura i Użytkowe pojawią się nasze nowe czujniki. Zaznaczając w ich polach gwiazdki, dodamy je do pulpitu Domoticza (**rysunek 6**). Edytując właściwości przełącznika LED i potencjometru, wybierzmy typ Dimmer. Zmianę wprowadzamy w oknie przełącznika, klikając Edit, następnie w polu Typ przełącznika wybieramy Dimmer i potwierdzamy przez Zapisz. Po tej zmianie w oknie przełącznika powinien pojawić się suwak, którym można regulować z Domoticza jasność świecenia diody w węźle.

Teraz o bardzo użytecznej właściwości systemu. Jeżeli wszystko poszło dobrze, a nasz komputer jest zalogowany w domowej sieci, to możemy obserwować wykresy zarejestrowanych pomiarów oraz sterować naszymi nowymi urządzeniami z każdego innego komputera, tabletu czy smartfona mającego dostęp do tej sieci. Po wpisaniu w przeglądarce na smartfonie adresu IP komputera, na którym pracuje nasz serwer Domoticz z dopisanym po dwukropku numerem

portu 8080, np. 192.168.1.100:8080, zostanie pokazana strona Domoticza wyświetlająca wszystkie dane.

Przedstawione tu podstawowe informacje przekazują małeńki zasób wiadomości na temat możliwości Domoticza i biblioteki MySensors. W Internecie można znaleźć wiele przykładów i tutoriali o stosowaniu tych programów a ten artykuł ma na celu zachęcenie osób, które jeszcze się tym tematem nie zajmowały, ale myślą o praktycznych rozwiązaniach takich układów. W moim domu taki minisystem pomiarowy, oparty na Raspberry Pi, działa z powodzeniem już od kilku miesięcy i uważam, że idealnie nadaje się do małej automatyki domowej.



Fotografia 8. Moduł „mały” o oznaczeniu nRF24L01

Kilka słów o modułach nRF24L01

Spośród kilkunastu wersji modułów nRF24L01 dostępnych w sprzedaży można wyróżnić dwa modele. Pierwszy to moduł „mały”, oferowany jako nRF24L01, widoczny na **fotografii 8**, który w otwartym terenie ma deklarowany zasięg do 100 m. Drugi, „większy” moduł to nRF24L01+PA+LNA, widoczny na **fotografii 9**, w którym nadajnik ma większą moc i zasięg do 1000 m. Potrzebuje on zasilania prądem o natężeniu do 120 mA. Nie wszystkie płytki Arduino na wyjściu zasilania 3,3 V mają taką wydajność prądową, więc przy stosowaniu tego modułu może być potrzebny dodatkowy stabilizator. Jedną z przyczyn złej pracy modułów są problemy z zasilaniem, dlatego często jest zalecane podłączenie



Fotografia 9. Moduł „większy” o oznaczeniu nRF24L01+PA+LNA

modułu do zasilania porządnymi przewodami i dodanie przy nim kondensatora o pojemności od kilku do kilkudziesięciu mikrofaradów (fot. 9). Bardzo dobrze spisują się kondensatory SMD 0805 wlotowane pomiędzy wyprowadzenia (piny) zasilające moduł radiowy. Obydwa prezentowane moduły są takie same pod względem interfejsu elektrycznego i oprogramowania.

Wiesław Pytlewski
elewp@wp.pl

REKLAMA

m.technik
Ciekawi świata są zawsze młodzi
w prezencie na każdą okazję

GEORGE DE MESTRAL
Pomysł przyczepiony do psiego ogona
nr 2, luty 2019
e-suplement www.mt.com.pl

m.technik
Ciekawi świata są zawsze młodzi

CYBERWOJNA
wygra, kto przetrwa
RAPORT: Zagadki Wszechświata
Im bardziej go znamy, tym mniej jest poznany

<http://bit.ly/2DKgsBJ>
przejrzyj i kupisz na
www.ulubionykiosk.pl