

Zaskakująco proste uniwersalne łącze bezprzewodowe

Do czego to służy?

Uniwersalne łącze bezprzewodowe tworzą zaskakująco proste moduły, mające 8 wejść i 8 wyjść. Dwa moduły tworzą najprostszą łącze dwukierunkowe. Wtedy nie jest potrzebna żadna konfiguracja, a działanie jest proste i oczywiste: na wyjścia każdego modułu przekazywany jest bezprzewodowo stan wejść modułu współpracującego.

Większa liczba modułów pozwala stworzyć sieć o dowolnej konfiguracji. Możliwa jest jednoczesna praca wielu sieci, zawierających wiele modułów.

A to wszystko dzięki zastosowaniu niedrogich transceiverów RFM12 oraz popularnego mikrokontrolera ATmega8.

Przedstawiane moduły rozwiążą mnóstwo problemów, związanych ze sterowaniem na odległość i monitoringiem. Liczba zastosowań takich modułów jest ogromna za sprawą

ich ogromnej elastyczności. Moduły możemy wykorzystać np. w sterowaniu modeli RC, a także w robotach. Niemalą zaletą jest możliwość śledzenia stanu wejść po drugiej stronie, umożliwi to np. nadzorowanie stanu czujników. Innym zastosowaniem może być sterowanie oświetleniem i innymi pasywnymi urządzeniami. Przykładowo, mając kilka pomieszczeń, możemy sterować oświetleniem w każdym z każdego. Ważną cechą jest możliwość pracy na dziesiątkach różnych kanałów, co umożliwia zrealizowanie dużej liczby sieci działających jednocześnie obok siebie.

Cechy sprzętowe:

- 8 wejść podciągniętych do „plusa”,
- 8 wyjść, maksymalny prąd 5mA/na wyjście,
- wejście konfigurujące tryb master/slave (przydatne jedynie bez tablicy routingu),
- zasilanie 3–5V (25mA przy 5V),

- moduł RF 433MHz,
- zasięg w otwartym terenie >150m.

Podstawowe wykorzystanie:

praca w parach,

Dodatkowe możliwości:

- praca w sieci typu master i slave
- możliwość dowolnej konfiguracji kierunków przekazywania informacji,
- maksymalna liczba modułów w jednej sieci: 9,
- 153 kanały fizyczne (różna częstotliwość),
- 16 kanałów wirtualnych (oddalone sieci),
- czas opóźnienia przełączania wyjść: dla dwóch modułów: 25ms – w jednym kierunku, 50ms – w obu kierunkach, dla 9 modułów min. 0,5s – w obu kierunkach, max. 1,2s.

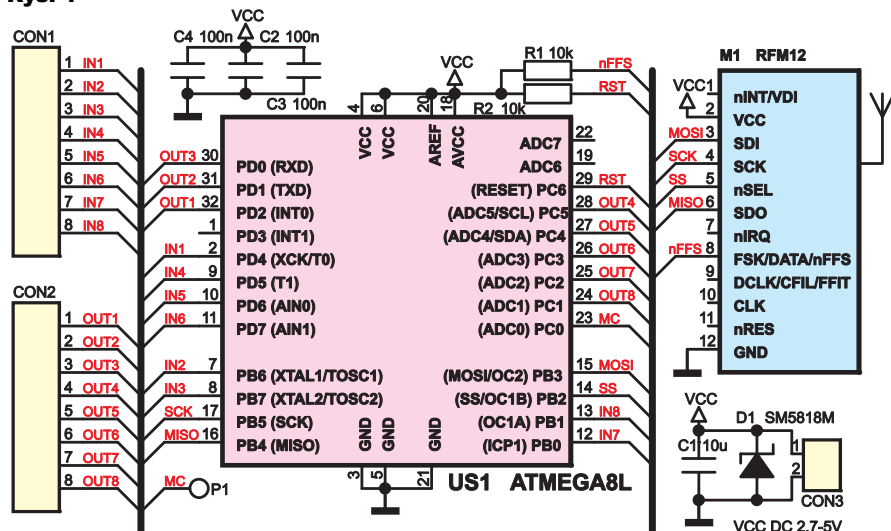
Jak to działa?

Schemat ideowy jednego modułu został przedstawiony na **rysunku 1**. Właściwie jedynymi elementami zastosowanymi w modułach są mikrokontroler US1 ATmega8L oraz ostatnio coraz bardziej popularny niedrogi transceiver M1 RFM12, firmy HOPE RF.

Transceiver został zbudowany na bazie układu RF12, na jego płytce znajduje się zaledwie kilka elementów zewnętrznych. Układ ten jest wyposażony w interfejs SPI, za pomocą którego można go nie tylko skonfigurować, ale również odbierać i nadawać dane. Ja zdecydowałem się na wykorzystanie tylko SPI (Serial Peripheral Interface), czyli 4 wyprowadzeń, ponieważ chciałem dokładnie sprawdzać stan flag statusu. Wadą tego rozwiązania jest mniejsza szybkość transmisji danych.

Konfiguracja rozpoczyna jest komendą *Configuration Setting Command* (0x8000),

Rys. 1



wysyłane są w niej ustawienia dotyczące częstotliwości pracy nadajnika i odbiornika. W tym przypadku jest to 433MHz, gdyż na taką częstotliwość dobrane są elementy modułu RFM12. Włączana jest kolejka FIFO (First Input First Output) oraz wewnętrzny rejestr danych, a dodatkowo należy też wybrać pojemność obciążającą zastosowany kwarc. Wstępnie wyłączany jest zarówno nadajnik, jak i odbiornik, komendą *Power Management Command* (0x8200). Dodatkowo wyłączam nieużywane wyjście zegara, którego przeznaczeniem jest taktowanie np. procesora. Komenda *Frequency Setting Command* (0xA000) jest bardzo istotna – za jej pomocą określana jest dokładna częstotliwość pracy odbiornika i nadajnika. Cały zakres częstotliwości wynoszący 9,525MHz został podzielony ze skokiem 62,5kHz, co pozwoliło na uzyskanie 153 kanałów. Ustawienie dwóch sąsiednich kanałów nie powoduje przenikania sygnału między nimi, nawet jeśli moduły leżą obok siebie. Komenda *Data Rate Command* (0xC600) ustawia prędkość transmisji i została ona ustawiona na 4800bps, dyktuje to, ile czasu ma program, by przesłać lub odebrać kolejny bajt z kolejki. Oczywiście, czym większa prędkość transmisji, tym mniejszy zasięg i większa podatność na błędy. Transceiver posiada małą 16-bitową kolejkę FIFO, która może być i jest używana zarówno przy transmisji, jak i odbiorze danych. Zastosowanie tak małej kolejki jest trochę uciążliwe dla sterownika zajmującego się obsługą modułu, gdyż musi on naprzemiennie badać stan flag oraz odczytywać lub zapisywać kolejkę, co powoduje spadek wydajności. Kolejna komenda, *Receiver Control Command*

związanych z torem odbiorczym. Istotnym parametrem jest szerokość pasma filtru pasmowoprzepustowego – została ona ustawiona na 67kHz, oraz wzmocnienie przedwzmacniacza, które ustawiono na maksimum, tj. 0dB. Określana jest też siła sygnału, przy której jest ustawiana wewnętrzna flaga DRSSI. Składa się ona wraz z innymi flagami CR_LOOK (Clock Recovery Look) oraz DQD (Data Quality Detector) na sygnał VDI (Valid Data Indicator), który jest używany m.in. do rozpoczęcia napełniania kolejki. Odbiornik konfiguruję tak, aby koniunkcja wszystkich trzech flag składała się na VDI. *Data Filter Command* (0xC200) określa sposób odzyskania sygnału zegara (CR_LOOK), polega to na zliczaniu kolejnych przejść 0/1. Tak więc, zanim jakiegokolwiek użyteczne dane zostaną nadane, wysyłany jest ciąg 0/1. Dodatkowo ustawiany jest próg DQD oraz filtr cyfrowy. Komenda niezbędna przy odbiorze to *FIFO and Reset Mode Command* (0xCA00), ustawia ona m.in. wartość progową liczby odebranych bitów, która spowoduje ustawienie flagi FIIT. Wartość ta zostaje ustawiona na 8. Ponadto ustawiany jest wymóg odbioru słowa/sygnatury składającej się z dwóch bajtów (0x2DD4), a powodującej rozpoczęcie napełniania kolejki, gdy DVI i bit ff są ustawione. Współpracujące transceivery mają możliwość automatycznego odstrojenie się od częstotliwości głównej. Umożliwia to lepsze zgranie nadajnika z odbiornikiem, zwłaszcza gdy rezonator kwarcowy nie jest zbyt dokładny oraz przy pracy z wieloma różnymi modułami. Włączenie tej możliwości dokonywane jest za pomocą *AFC Command* (0xC400). Konfiguracja nadajnika dokonywana jest za pomocą *TX Configuration Control Command* (0xB800) i sprowadza się wyłącznie do ust

wyżsiowej (0dB) oraz głębokości modulacji FSK (30kHz).

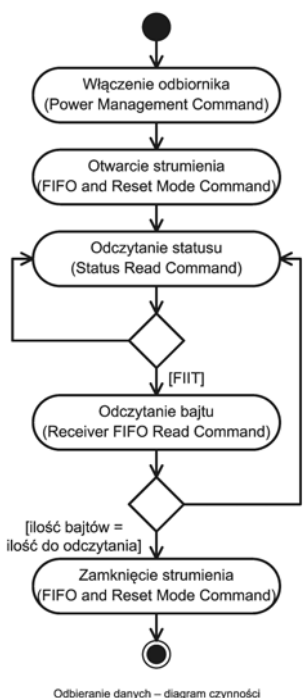
Procedura odbioru składa się z szeregu operacji, przedstawionych na diagramie czynności – **rysunek 2**. Początkowo uruchamiany jest tor odbiorczy poprzez zapis komendy *Power Management Command*, z ustawionym bitem *er* (Enable Receiver). Następnie, zapisywana jest komenda *FIFO and Reset Mode Command*, z ustawionym bitem *ff* (FIFO fill). Po czym następuje cykliczne sprawdzanie, czy odebrano bajt. Jest to realizowane komendą *Status Read Command* (0x0000) i testowania bitu FIIT. Jeśli bit jest ustawiony, następuje odczyt odebranych danych za pomocą *Receiver FIFO Read Command* (0xB000), bit FIIT zostanie wtedy wyzerowany. Czynność ta powtarzana jest aż do odebrania żądanej liczby bajtów, składających się na pakiet. Jeśli się to stanie, bit *ff* (FIFO fill) jest czyszczony. Brak wyzerowania tego bitu mogłoby skutkować ciągłym napływem danych, nawet przy braku nadawania.

Podobna jest procedura nadawania, przedstawiona na diagramie czynności – **rysunek 3**. Początkowo uruchamiany jest tor nadajnika. Od razu następuje cykliczny odczyt statusu (Status Read Command). Jeśli bit RGIT jest ustawiony, przesyłany jest bajt do kolejki nadajnika komendą *Transmitter Register Write Command* (0xB800). Czynność ta powtarzana jest aż do wysłania wszystkich bajtów pakietu, w tym bajtów poprzedzających (0xAA), preambuły (0x2DD4), właściwej zawartości pakietu, bajtu zaślepki.

Oczywiście, wysyłanie zawartości pakietu w oryginalnej, niezmienionej postaci nie byłoby możliwe, bo ciągi zer i jedynek spowodowałyby rozszynchronizowanie odbiornika i nie mógłby on poprawnie odebrać danych. Dlatego

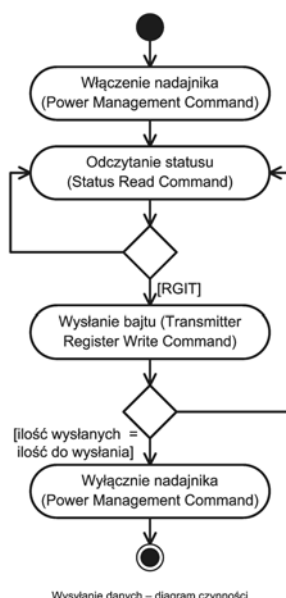
(0x9000),
ustawia szereg opcji

Rys. 2



Odbieranie danych – diagram czynności

Rys. 3



Wysyłanie danych – diagram czynności

Rys. 4

| | | Wejścia modułu o ID: | | | | | | | | | | | | | | | | |
|----------------------|----|----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | | | | | |
| Wyjścia modułu o ID: | 0 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | |
| | 1 | 10 | 00 | 06 | 00 | 48 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| | 2 | 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 3 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 4 | 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 5 | 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 6 | 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 7 | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 8 | 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | FF | FF | FF |
| | AO | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

Opis przeznaczenia poszczególnych bajtów pamięci EEPROM.

- Bajt zabezpieczający. Musi być równy 0, jeśli zawartość pamięci ma być sprawdzana przy konfiguracji.
- Bajt określający wirtualny numer kanału.
- Wartość kanału może być z przedziału 0 – 15.
- Identyfikator modułu. Każdy moduł musi mieć unikalny identyfikator z przedziału 0 – 8. Numer 0, przewidziany jest dla modułu władcy, moduł ten musi mieć zdefiniowaną tablicę routing'u. Identyfikator modułu ściśle wiąże dany moduł z kolumną i wierszem w tablicy routing'u.
- Bajt określający numer kanału fizycznego. Wartość tego bajtu powinna być z przedziału 0 – 152. Domyślnie jest to wartość 72. Pozostawianie komórki pamięci niezaprogramowanej, powoduje przyjęcie ustawienia domyślnego.
- Tablica routing'u. Tablica ma wymiary 9/9 komórek pamięci. Kolumnom tablicy odpowiadają wejścia modułów o numerze kolumny. Wierszom tablicy odpowiadają wyjścia modułów o numerze wiersza.

też wysyłany pakiet zamieniany jest zgodnie z regułą 1 do 2 bitów. Zero jest kodowane jako 01, a jeden jako 10. Pakiet wyposażony jest też w sumy kontrolne, zapobiegając błędnej interpretacji danych.

Zasada działania sieci opiera się na założeniu, że jeden moduł musi władać pozostałymi. Stąd też specjalny pin konfiguracyjny, określający, który moduł jest władcą (master).

| | | Wejścia modułu o ID: | | | | | | | | |
|----------------------|---|----------------------|----|----|----|----|----|----|----|----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| Wyjścia modułu o ID: | 0 | 20 | 00 | FF | 00 | 00 | 00 | 00 | 00 | 00 |
| | 1 | 30 | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 2 | 40 | 00 | 00 | 00 | 00 | A0 | 00 | 5F | 00 |
| | 3 | 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 4 | 60 | 00 | 00 | 0F | 55 | 00 | 00 | 00 | 00 |
| | 5 | 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 6 | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 7 | 90 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 |
| | 8 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

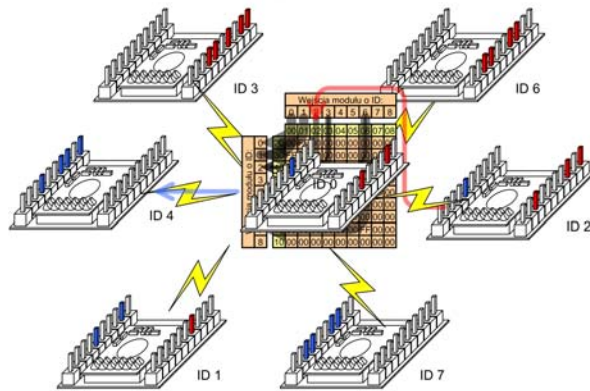
Przykład tablicy routingu.

Rys. 5

To władca wysła pierwszy pakiet, a jego sługa (slave), odbierając go, odpowiada. Proces ten trwa bez przerwy, a pasmo sieci jest całkiem zajęte, ponieważ gdy tylko moduł odbierze dane, od razu przechodzi do wysyłania i znowu do odbioru. **Niezaprogramowanie pamięci EEPROM umożliwia prostą pracę dwóch modułów.** W konfiguracji tej wejścia jednego modułu dołączone są do wyjść drugiego i na odwrót.

Moduły mogą też tworzyć większą sieć, przypomina ona wtedy topologię gwiazdy. Obrazowo mówiąc, pośrodku znajduje się władca, a dookoła otaczają go słudzy. Aby było możliwe takie działanie sieci, wymagane jest zdefiniowanie statycznej tablicy routingu i umieszczenie jej w pamięci EEPROM procesora władcy. Władcę można utożsamiać z routerem. W tablicy routingu znajdują się wytyczne dotyczące tego, który pin jakiego modułu ma zostać przekazany, na które wyjście jakiego modułu. W tej konfiguracji moduły muszą być kolejno ponumerowane, numeracji należy dokonać samodzielnie, wpisując odpowiednie wartości do pamięci EEPROM każdego modułu. Identyfikator modułu służy musi być z przedziału wartości 1–8, a władcy – musi być równy 0. Oczywiście w sieci na tym samym kanele nie może być dwóch modułów o tym samym identyfikatorze. W sieci musi być jeden władca, czyli moduł o identyfikatorze 0. Brak władcy uniemożliwia działanie sieci – słudzy nie mogą komunikować się między sobą bezpośrednio.

Tablica routingu ma kształt tabeli o takiej samej liczbie wierszy, co kolumn. Kolumn i wierszy jest dziewięć, czyli tyle, ile może być maksymalnie modułów w jednej sieci. Identyfikatory kolumn i wierszy odpowiadają identyfikatorom modułów. Kolumna o danym identyfikatorze odpowiada wejściom modułu o tym identyfikatorze, a wiersz wyjściom modułu o identyfikatorze wiersza. Komórka pamięci występująca na przecięciach wierszy i kolumn określa, które piny z wejścia mają być przekazane na wyjście. Wpisanie jedynki na danej pozycji powoduje przekazanie wej-



Przykład stanu wyjść w sieci opartej o tablicę routingu pokazaną poprzednio. Kolorem czerwonym oznaczono stan niski na wejściach, niebieskim stan wysoki na wyjściach.

Rys. 6

ścia o tym numerze na wyjście o tym samym numerze. Jeśli w danym wierszu jest więcej jedynek na danej pozycji, stan wyjścia jest traktowany jako alternatywa z nich. Wejścia jednego modułu mogą być przekazane na wyjścia różnych modułów lub żadnego. Możliwa jest też dziwna konstrukcja przekazania wejść na wyjścia tego samego modułu, np. w celu kontroli zestawienia połączenia. Jeśli moduł nie uczestniczy w wymianie, tj. jego wejścia nie są przekazywane na żadne wyjścia i do jego wyjść nie są przekazywane żadne wejścia, to jest on wyłączony z sieci.

Rysunek 4 przedstawia opis przeznaczenia poszczególnych komórek pamięci. Na **rysunku 5** przedstawiono przykład tablicy routingu. Analizując rysunek, możemy doszukać się następujących tras: – Wszystkie wejścia modułu 0 przypisane są do wszystkich wyjść modułu 1. – Wszystkie wejścia modułu 1 przypisane są do wszystkich wyjść modułu 0. – Wejścia 6 i 8 modułu 3 przypisane są do wyjść modułu 2 o tych samych numerach. Dodatkowo, wejścia modułu 3 o numerach 1, 3, 5, 7 przypisane są do wyjść o tych samych numerach modułu 4. – Wejścia 1–4 modułu 2 przekazywane są na wyjścia o tych samych numerach, modułu 4. – Na stan wyjść modułu 4 składa się alternatywa odpowiednich wyjść, modułów 2 i 3. – Wejścia 1–5, 7 modułu 6 przypisane są do wyjść o tych samych numerach modułu 2. – Dodatkowo, wszystkie wejścia modułu 6 przekazywane są na wszystkie wyjścia modułu 7. – Moduł o identyfikatorze 5 i 8 nie uczestniczy w wymianie, więc nie musi być go w sieci. Obrazowo przedstawia to **rysunek 6**.

Władca, czyli moduł o identyfikatorze 0 i wpisanej zawartości tablicy routingu, odpytuje kolejno wszystkie moduły, znajdujące się w sieci, o stan ich wejść. Stan wejść tych modułów jest buforowany w celu późniejszego przekazania na wyjścia innego modułu. Ponieważ sieć tworzona poprzez moduły odpo-

wiada trzem pierwszym warstwom modelu OSI, pewność transmisji nie jest spełniona. Oznacza to, że w przypadku błędu dane są odrzucane. Jeśli władca nie otrzymał żądanego stanu wejść któregoś ze sług, to przyjmuje za niego stan, w którym żadne wejście nie jest aktywne (połączone z masą), wiadomość ta rozsyłana jest do odpowiednich modułów w kolejnym cyklu. Zapobiega to utrzymywaniu się stanów wysokich na wyjściach modułów po zerwaniu

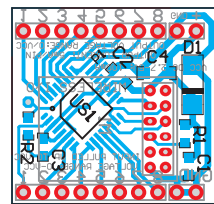
połączenia z modułem ustalającym ten stan. Inny przypadek – groźny jest wtedy, gdy moduł władcy z jakichś przyczyn przestanie działać, wtedy cała sieć obumiera, a wyjścia sług po chwili automatycznie przejdą w stan zerowy. **Ponieważ wszystkie moduły są jednakowe, tylko zawartość pamięci EEPROM lub stan pinu konfiguracyjnego decyduje o tym, który moduł jest władcą, a które sługami.**

Montaż i uruchomienie

Układ można zmontować na płytce drukowanej pokazanej na **rysunku 7**. Montaż rozpoczynamy od wlutowania mikrokontrolerów (uwaga na właściwe wlutowanie, wyżłobione kółko na obudowie mikrokontrolera musi zgadzać się z tym na płytce drukowanej), a kończymy wlutowaniem goldpinów i podstawki pod moduł RFM. Do modułów RFM lutujemy dostarczone razem z nimi anteny. Po upewnieniu się, że wszystko zostało zlutowane poprawnie, wgrujemy oprogramowanie (można je ściągnąć z Elportalu) do mikrokontrolera – jeśli jeszcze go tam nie ma. Napięcie zasilające moduł powinno być stabilizowane, w zakresie 2,7–5V – w czasie programowania 5V. **Rysunek 8** pokazuje połączenie układu. Szczególną uwagę

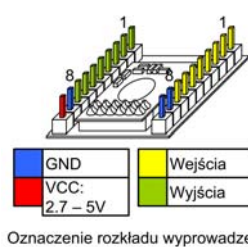
zwracamy na poprawność zaprogramowania FuseBits, bez zaprogramowania których program na pewno nie będzie działał poprawnie. **W wariantcie z dwoma modułami nie jest konieczne programowanie pamięci EEPROM, w tym przypadku w jednym z modułów wystarczy zwrócić pin 23 (ozna-**

czenie P1 na PCB) procesora z masą, nadając mu w ten sposób funkcję master. Jeśli zdecydujemy się na wgranie ustawień do pamięci EEPROM, uzyskamy możliwość zdefiniowania statycznej tablicy routingu, fizycznego i wirtualnego kanału pracy oraz identyfikatora urządzenia. Szczegóły dotyczące zawartości pamięci EEPROM były



Rys. 7

Rys. 8



Oznaczenie rozkładu wyprowadzeń

podawane wcześniej. Zwarcie wcześniej omawianego pinu z masą ma większy priorytet niż zaprogramowane ID modułu w pamięci EEPROM. Tak więc zalecane jest wypełnienie obszaru tablicy routingu zerami, nawet dla modułów typu slave. Razem z plikami wsadowymi przygotowałem też obraz pamięci EEPROM, ułatwiający stworzenie wymaganych ustawień. Tablica routingu w module typu slave (ID: 1–8) nie będzie normalnie używana, można ją jednak aktywować bez zmian w pamięci EEPROM, zwierając wcześniej omawiany pin do masy. Wtedy moduł stanie się modulem master i zacznie używać tej tablicy. Możliwość taka może być potrzebna w przypadku awarii wcześniejszego modułu master. Po zmianie ustawień pinu moduł należy zrestartować. Przypominam, że w sieci nie mogą równocześnie pracować dwa moduły typu master (na tych samych kanałach).

Sieci wirtualne to tylko dodatkowe zabezpieczenie przed przeniknięciem sygnału z innej sieci w przypadku, gdy dwie sieci na tym samym kanale fizycznym znajdują się z jakichś przyczyn obok siebie. Wtedy żadna sieć nie będzie działać. Jeśli spodziewamy się natłoku sieci z tymi modułami, powinniśmy zmienić domyślne ustawienia kanałów.

I tu bardzo ważna uwaga przy wkładaniu transceiverów. Przynajmniej moje podstawki

pod transceiver były słabej jakości, co skutkowało spadkiem napięcia uniemożliwiającym poprawną pracę transceivera. Poszukiwania ewentualnej przyczyny niedziałania należy rozpocząć od sprawdzenia napięcia na stykach zasilania transceivera (trzeba dokładnie **zmierzyć napięcie na płytce transceivera**, w miejscu wlutowania goldpinów, pin pierwszy i ostatni od strony układu scalonego), powinno ono być równe napięciu zasilania układu. Zazwyczaj dociśnięcie pomaga. Oczywiście transceiver wkładamy tak, aby pokrywał się z płytką drukowaną, a nie wystawał poza nią. UWAGA! Wlutowanie transceivera bez podstawki uniemożliwi późniejsze zmiany pamięci EEPROM.

Maksymalny prąd, jaki może być pobierany z wyjścia modułu nie może przekroczyć 5mA. Na wyjściach nie powinny pojawiać się „obce” napięcia wymuszające przepływ prądu. W przypadku sterowania np. oświetleniem należy we własnym zakresie dodać matrycę tranzystorów (ULN2803A) wzmacniającą prąd oraz przełączniki lub tyristory (nie są to skomplikowane obwody). Do wejść modułu nie doprowadzać „obcych” napięć. Wejścia są podciągnięte do plusa zasilania, w takim przypadku ich stan jest nieaktywny i odpowiada stanowi niskiemu na wyjściu. Do wejść można bezpośrednio dołączyć przyciski, pod warunkiem że są umieszczone nie dalej jak 20cm.

Wykaz elementów

| | |
|--------------------------------------|------------------------|
| R1,R2 | 10k Ω /SMD/0805 |
| C1 | 10 μ F/SMD/1206 |
| C2-C4 | 100nF/SMD/0805 |
| D1 | SM5818M |
| US1 | ATmega8L |
| M1 | RFM12-433-D |
| CON1,CON2 | goldpin 1x8 |
| CON3,CON4 | goldpin 1x2 |
| Dwurzędowe gniazdo pod małe goldpiny | |

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2956.

W przypadku większych odległości zalecane jest stosowanie transoptorów, a w przypadku prądu przemiennego zalecane jest stosowanie transoptorów z dwoma diodami (np. PC814), dodatkowo na wejściu modułu wykorzystać filtr tłumiący tętnienia. **Jeszcze raz przypomnam (dla wystraszonych), że moduły mogą pracować bez jakiegokolwiek ingerencji w pamięć EEPROM procesora.**

Ustawiając moduły, należy pamiętać, że największy zasięg uzyskamy, ustawiając anteny w tej samej polaryzacji, czyli równolegle do siebie w obu płaszczyznach.

Szymon Janek
sx13@o2.pl

R E K L A M A