

Prosty modem radiowy
v1.0

Arkadiusz Materek

12 marca 2007

Spis treści

1	Wstęp	2
2	Hardware	2
2.1	Opis układu CC1000	2
2.2	Interfejs CC1000 z mikrokontrolerem	3
2.2.1	Interfejs konfiguracji	4
2.2.2	Interfejs danych	5
2.2.3	Wyjście RSSI/IF i CHP_OUT	6
2.3	Budowa modemu	6
3	Software	8
3.1	Konfiguracja CC1000	8
3.1.1	Wybrane parametry pracy	8
3.1.2	Resetowanie układu	9
3.1.3	Budzenie układu ze stanu uśpienia	9
3.1.4	Przełączanie w stan nadawania	10
3.1.5	Przełączanie w stan odbioru	11
3.1.6	Kalibracja VCO i PLL	11
3.1.7	Inicjalizacja CC1000	12
3.2	Format ramki	12
3.3	Nadawanie ramki	13
3.4	Odbieranie ramki	13
3.5	Transmisja RS232	13
3.6	Algorytm działania modemu	14
4	Podsumowanie	15
	Spis rysunków	15
	Spis tablic	15
	Literatura	16

1 Wstęp

Opisywany modem radiowy został opracowany z myślą o taniej i prostej komunikacji robota z komputerem. Komunikacja z modemem odbywa się przez protokół RS232 z wykorzystaniem linii RX, TX, CTS oraz RTS. Do realizacji transmisji radiowej wykorzystano układ CC1000 firmy Chipcon. Wymaga on sterowania transmisją danych, do tego celu wykorzystano mikrokontroler ATmega8 firmy Atmel.

2 Hardware

2.1 Opis układu CC1000

Układ CC1000 jest nadajnikiem i odbiornikiem (transceiverem) radiowym FM fal UKF. Wymaga jedynie niewielkiej ilości zewnętrznych elementów zależnych od częstotliwości pracy¹. Podstawowe własności układu:

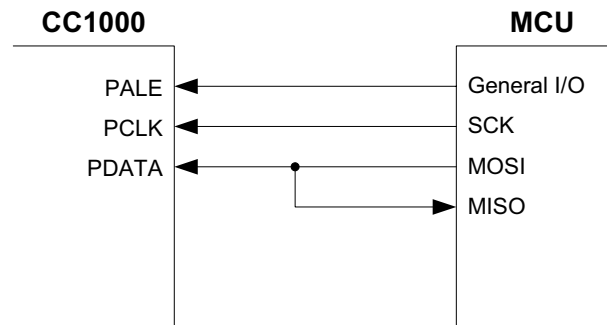
- Zakres częstotliwości 300-1000MHz
- Wysoka czułość (-110dBm przy prędkości transmisji 2.4kBaud)
- Programowalna moc nadajnika (od -20dBm do 10dBm)
- Zasięg do 2000m
- Prędkość transmisji danych do 76.8kBaud
- Wyjście RSSI (pomiar poziomu odbieranego sygnału)
- Kodowanie i synchronizacja bitów (FSK, Manchester, lub brak).
- Bardzo mały pobór prądu
- Częstotliwość nadawania i odbioru programowalne z krokiem 250Hz
- Oprogramowanie generujące konfiguracje

Do budowy modemu wykorzystano gotowy moduł CC1000PP, który zawiera niezbędne elementy do pracy w paśmie 433MHz (dokumentacja modułu znajduje się w [2]). Podstawowe parametry układu można znaleźć w tabeli 1, a szczegółowe informacje na temat układu w dokumentacji technicznej [1]

¹ W Polsce dozwolone są pasma 433MHz i 868MHz

2.2.1 Interfejs konfiguracji

Konfiguracja CC1000 odbywa się przez wpisywanie do rejestrów układu (każdy rejestr można również odczytać). Konfiguracja pamiętana jest w pamięci RAM, co zmusza do każdorazowego konfigurowania całego układu. Podczas konfiguracji układem nadrzędnym (masterem) jest mikrokontroler (generuje sygnał zegara PCLK). Maksymalna częstotliwość zegara (PCLK) wynosi 10MHz, co pozwala na zapisanie wszystkich rejestrów w czasie ok. $46\mu s$. Układ CC1000 czyta bity danych (PDATA) na zboczu opadającym linii PCLK, odczyt bitów przez mikrokontroler powinien być na zboczu opadającym. Dobrym rozwiązaniem w tym przypadku byłoby zastosowanie interfejsu SPI, który jest obsługiwany przez większość dostępnych mikrokontrolerów. Układ CC1000 posiada jedynie jedną linię danych, co wymusza połączenie linii MISO i MOSI², dokładny schemat jest pokazany na rysunku 3. W takim przypadku należy pamiętać, aby przy odbieraniu danych przełączyć pin MOSI w stan wysokiej impedancji. Procedura zapisu rejestru CC1000



Rysunek 3: Schemat połączenia układu CC1000 z mikrokontrolerem (SPI)

przebiega następująco:

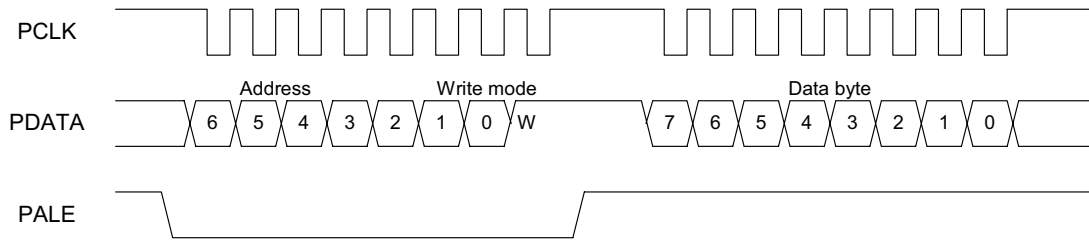
- Ustawienie przez mikrokontroler linii PALE w stan niski
- Wpisanie 7-bitowego adresu i bitu zapisu – 1
- Ustawienie przez mikrokontroler linii PALE w stan wysoki
- Wpisanie wartości rejestru

Procedura odczytu rejestru CC1000 przebiega następująco:

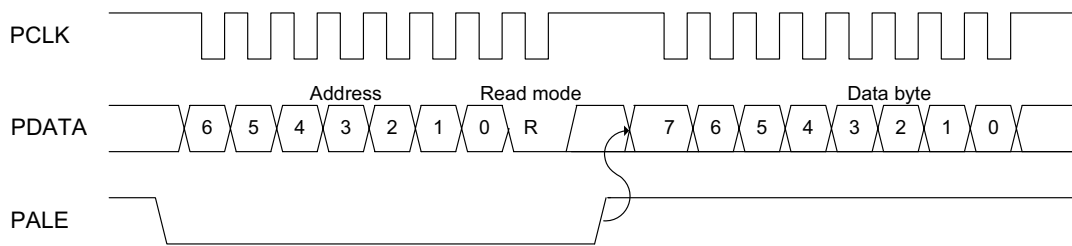
- Ustawienie przez mikrokontroler linii PALE w stan niski
- Wpisanie 7-bitowego adresu i bitu odczytu – 0

² W przypadku mikrokontrolerów AVR uniemożliwia to programowanie przez ISP.

- Ustawienie przez mikrokontroler linii PALE w stan wysoki
- Ustawienie linii PDATA jako wejście i odczytanie zawartości rejestru



Rysunek 4: Operacja zapisu rejestru CC1000



Rysunek 5: Operacja odczytu rejestru CC1000

Szczegóły dotyczące dokładnych timingów można znaleźć w [1] na stronie 13-14.

2.2.2 Interfejs danych

Dane przesyłane są przez linie DCLK i DIO. W zależności od użytego kodowania bitów (NRZ, Manchester i UART).

Gdy używane jest kodowanie NRZ lub Manchester układem nadrzędnym jest CC1000 (generuje zegar transmisji DCLK), linia DIO jest wejściem w trybie nadawania i wyjściem w trybie odbioru.

W trybie nadawania bity na linii DIO są czytane na zboczu narastającym linii DCLK, więc przerwanie mikrokontrolera od linii DCLK powinno być ustawione na zbocze opadające, gdzie mikrokontroler wystawia następny bit do transmisji na linię DIO. W trybie odbioru bity na linii DIO wystawiane są na zboczu opadającym linii DCLK, więc przerwanie mikrokontrolera powinno być ustawione na zbocze narastające linii DCLK, gdzie mikrokontroler powinien odczytać stan linii DIO.

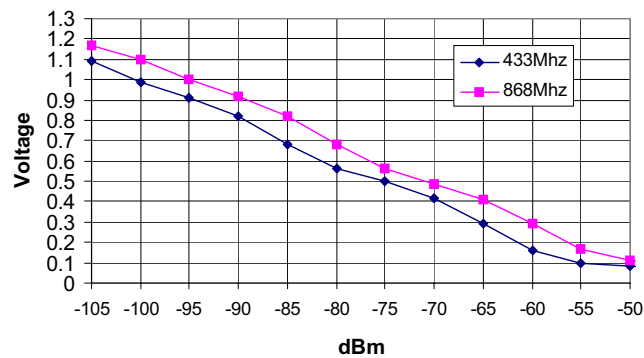
Częstotliwość sygnału zależy od prędkości transmisji radiowej. Dla kodowania NRZ jest równa tej prędkości, a dla kodowania Manchester jest połową mniejsza.

Gdy używany jest tryb UART, wejście i wyjście danych jest na dwóch osobnych liniach - wejście DIO (odpowiednik RXD), wyjście DCLK (odpowiednik TXD).

Użycie tego trybu pracy ułatwia przesyłanie danych np. przez podłączenie bezpośrednio do linii TXD i RXD mikrokontrolera, jednak używanie tego trybu pracy jest nie zalecane.

2.2.3 Wyjście RSSI/IF i CHP_OUT

Wyjście RSSI jest wyjściem analogowym reprezentującym pomiar poziomu mocy odbieranego sygnału radiowego. Zakres napięć wynosi 0-1.2V, gdzie 0 oznacza maksymalną moc a 1.2V minimalną (-110dBm). Na rysunku 6 przedstawiono zależność napięcia na wyjściu RSSI od poziomu odbieranego sygnału, która jest w przybliżeniu liniowa.



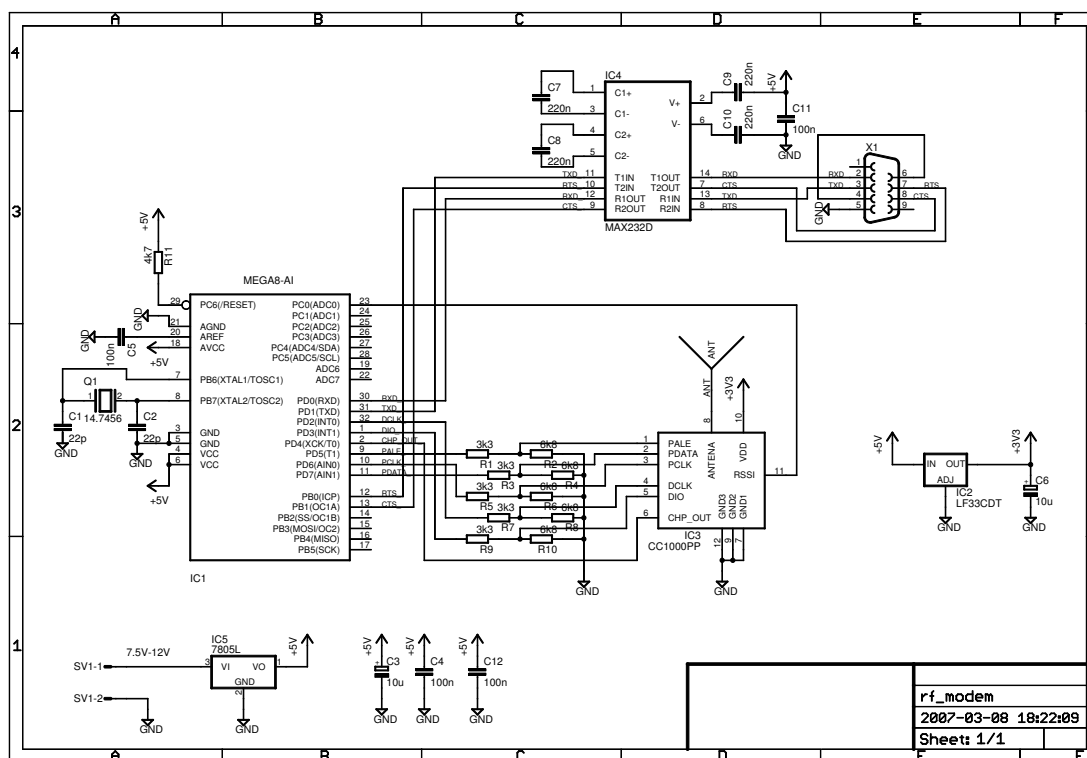
Rysunek 6: Zależność napięcia na RSSI od mocy odbieranego sygnału

Wyjście CHP_OUT jest wygodnym (dla mikrokontrolera - nie wymaga odczytywania rejestru, jedynie sprawdzanie stanu pinu) monitorowania wybranych bitów w rejestrach układu (dokładne informacje w dokumentacji technicznej [1] na stronie 43).

Najczęściej sprawdzanym bitem jest ten odzwierciedlający stan pętli PLL. Gdy częstotliwości nadawania i odbierania wymagają zmiany w rejestrze PLL, niezbędne jest, aby poczekać aż PLL będzie ustalona. W niektórych przypadkach może nastąpić zatrzaśnięcie się generatora, co jest opisane w [4], wtedy należy go zresetować.

2.3 Budowa modemu

Schemat modemu przedstawiony jest na rysunku 7. Mikrokontroler ATMega8 pełni rolę bufora nadawanych i odbieranych danych oraz steruje transmisją radiową. Taktowany jest częstotliwością 14.7456MHz ze względu na możliwość dokładnego ustawienia większości standardowych prędkości transmisji RS232. Układy ATMega8 i MAX232 zasilane są napięciem 5V filtrowanym przez kondensatory C3, C4, C11. Przetwornik analogowo-cyfrowy mikrokontrolera będzie korzystał z



Rysunek 7: Schemat modemu radiowego

wewnętrznego źródła napięcia odniesienia 2.56V, co wymaga kondensatora filtrującego C5 na pinie AREF. Dodatkowo napięcie zasilające przetwornik jest filtrowane przez kondensator C12, który powinien być umieszczony jak najbliżej pinu AVCC. Przetwornik zasilac przez filtr LC (patrz dokumentacja techniczna [5] strona 200).

Układ MAX232 zapewnia transformację napięć standardu TTL na standard RS232 i odwrotnie. W przypadku komunikowania się modemu z urządzeniem działającym w standardzie TTL można zrezygnować z tego układu i linie TXD, RXD, CTS i RTS połączyć bezpośrednio z drugim urządzeniem (należy pamiętać o skrosowaniu TXD-RXD i CTS-RTS).

Moduł CC1000 zasilany jest napięciem 3.3V. Do tego celu można wykorzystać dowolny stabilizator napięcia. Na płytce modułu znajdują się wszystkie elementy niezbędne do pracy układu CC1000. Ze względu na różnice w poziomach napięć pomiędzy mikrokontrolerem a modułem niezbędny jest translator napięć. W najprostszej wersji można wykorzystać zwykły dzielnik napięcia (w praktyce takie rozwiązanie działa bez zarzutu). Innym rozwiązaniem jest zastosowanie gotowego układu przeznaczonego do tego celu np. MAX3372E-MAX3393E³. Wyjście RSSI podłączone jest do wejścia analogowego mikrokontrolera w celu pomiaru poziomu odbieranego sygnału. Antene wykonano z kawałka drutu o długości 16.4cm.

³ dwu i jednokierunkowe translatory napięcia, maksymalna prędkość do 16Mbps. www.maxim-integrated.com

liczba	wartość	oznaczenie na schemacie
rezystory		
5	3k3Ω	R1, R3, R5, R7, R9
1	4k7Ω	R11
5	6k8Ω	R2, R4, R6, R8, R10
kondensatory		
2	22p	C1, C2
4	100n	C4, C5, C11, C12
4	220n	C7, C8, C9, C10
2	10μ	C3, C6
półprzewodniki		
1	ATMega8-16	IC1
1	LF33	IC2
1	CC1000PP	IC3
1	MAX232	IC4
rezonatory		
1	Q1	14.7456MHz
inne		
1	X1	Złącze D-SUB9 (Żeńskie)
1	ANT	16.4cm/50Ω

Tabela 2: Zestawienie elementów medemu radiowego

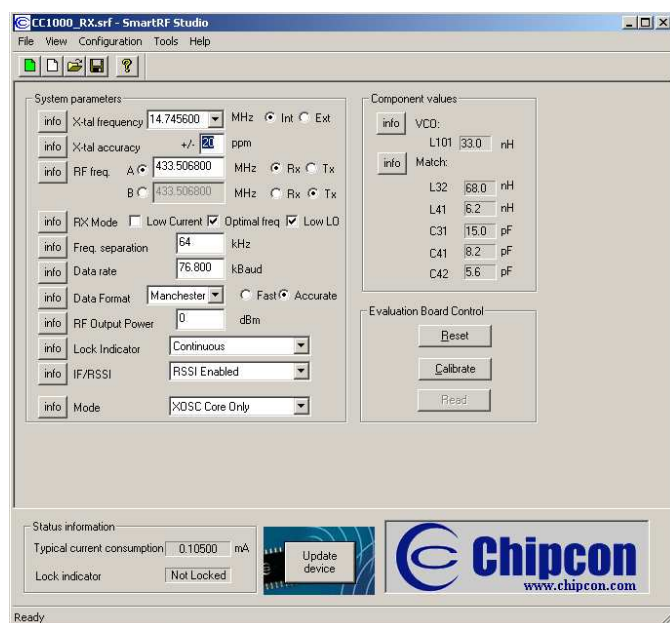
3 Software

W tej części zostaną opisane rozwiązania softwerowe zastosowane w modemie. Program do mikrokontrolera ATMega8 został napisany w języku C, i skompilowany przy użyciu darmowego kompilatora avr-gcc.

3.1 Konfiguracja CC1000

3.1.1 Wybrane parametry pracy

Do wygenerowania konfiguracji użyto program **SmartRF Studio**. Jest to darmowy program firmy Chipcon generujący ustawienia rejestrów oraz wartości elementów towarzyszących układom tej firmy. Przyjęte parametry transmisji radiowej są przedstawione w tabeli 3. Częstotliwości nadawania i odbioru zostały tak dobrane, aby nie wymagały zmiany w rejestrze PLL. Separacja częstotliwości została wybrana maksymalna, kodowanie bitów Manchester, aby maksymalizować czułość układu. Przy tym kodowaniu wybrano maksymalną prędkość transmisji danych 38.4kkBaud. Konfiguracja rejestrów przedstawiona jest w tabeli 4. Pogrubione zaznaczone zostały rejestry, które powinny być modyfikowane przy przełączaniu trybów nadawanie-odbior.



Rysunek 8: Interfejs programu SmartRF Studio

Częstotliwość odbioru	433.5068MHz
Częstotliwość nadawania	433.5068MHz
Separacja częstotliwości	64kHz
Prędkość danych	38.4kBAud
Kodowanie bitów	Manchester
Moc wyjściowa (nadawania)	0dBm
Monitorowanie PLL	ciągle
Wyjście RSSI	włączone

Tabela 3: Przyjęte parametry transmisji radiowej

3.1.2 Resetowanie układu

- Wpisać do rejestru MAIN wartość 0x3A
- Oczekać 2ms
- Wpisać do rejestru MAIN wartość 0x3B
- Oczekać 2ms

Teraz układ jest w stanie uśpienia.

3.1.3 Budzenie układu ze stanu uśpienia

- Wpisać do rejestru MAIN wartość 0x3B.

Nazwa	adres	wartosc RX/TX
MAIN	0x00	0x11/0xE1
FREQ_2A	0x01	0x42
FREQ_1A	0x02	0x00
FREQ_0A	0x03	0x00
FREQ_2B	0x04	0x42
FREQ_1B	0x05	0x04
FREQ_0B	0x06	0x9C
FSEP1	0x07	0x02
FSEP0	0x08	0x80
CURRENT	0x09	0x44/0x81
FRONT_END	0x0A	0x12
PA_POW	0x0B	0x0F
PLL	0x0C	0x48/0x48
LOCK	0x0D	0x10
CAL	0x0E	0x26
MODEM2	0x0F	0xB7
MODEM1	0x10	0x6F
MODEM0	0x11	0x54
MATCH	0x12	0x70
FSCTRL	0x13	0x01
PRESCALER	0x1C	0x00
TEST6	0x40	0x10
TEST5	0x41	0x08
TEST4	0x42	0x3F
TEST3	0x43	0x04
TEST2	0x44	0x00
TEST1	0x45	0x00
TEST0	0x46	0x00

Tabela 4: Konfiguracja rejestrów CC1000

- Odczekać 2ms
- Wpisać do rejestru MAIN wartość 0x39.
- Odczekać 200 μ s

Następnie należy wybrać tryb pracy RX/TX.

3.1.4 Przełączanie w stan nadawania

- Wpisać do rejestru PA_POW wartość 0x00
- Wpisać do rejestru MAIN wartość 0xE1

- Wpisać do rejestru CURRENT wartość 0x81
- Wpisać do rejestru PLL wartość 0x48⁴
- Odczekać 250 μ s
- Wpisać do rejestru PA_POW wartość 0x0F⁵

Teraz układ jest gotowy do nadawania danych.

3.1.5 Przełączanie w stan odbioru

- Wpisać do rejestru MAIN wartość 0x11
- Wpisać do rejestru CURRENT wartość 0x44
- Wpisać do rejestru PLL wartość 0x48⁶
- Odczekać 250 μ s

Teraz układ zaczyna odbierać dane (szumy).

3.1.6 Kalibracja VCO i PLL

- Wpisać do rejestru PA_POW wartość 0x00
- Wpisać do rejestru CAL wartość 0xA6 – zacząć kalibrację
- Zaczekać maksymalnie 34ms, lub dopóki bit CAL_COMPLITE=1
- Wpisać do rejestru CAL POW wartość 0x26 – zakończyć kalibrację

Wynik kalibracji zapisany jest w rejestrach TEST0 i TEST1, następną kalibrację należy wykonać po zmianie temperatury o 40°C, lub gdy napięcie zasilania zmieni się o 0.5V.

⁴ jeśli wartość rejestru PLL jest taka sama dla nadawania i odbioru to można pominąć

⁵ Od tej wartości zależy moc nadawania

⁶ jeśli wartość rejestru PLL jest taka sama dla nadawania i odbioru to można pominąć

3.1.7 Inicjalizacja CC1000

- Zresetować CC1000
- Zaprogramować wszystkie rejestry za wyjątkiem MAIN
- Obudzić układ
- Przełączyć w stan nadawania
- Dokonać kalibracji częstotliwości nadawania
- Przełączyć w stan odbioru
- Dokonać kalibracji częstotliwości odbioru

Po tych czynnościach układ jest gotowy do odbierania danych.

3.2 Format ramki

Dane przesyłane w modemie są pakowane w ramki. Układ CC1000 wymaga przed nadaniem danych nadania preambuły w celu dostrojenia progu komparacji napięć z demodulatora. Wielkość preambuły zależy od sposobu kodowania i od oczekiwanej czułości układu, w [1] na stronie 20 przedstawione są dokładne informacje na ten temat. Po preambule należy nadać unikatowy bajt oznaczający roz-

Preambuła	Start	ACK	CRC	DSIZE	DSIZE	Dane
32 bajty	0x0F	1 bajt	1 bajt	1 bajt	1 bajt	0-64 bajtów

Rysunek 9: Format ramki

poczęcie się ramki, ma on między innymi na celu zsynchronizowanie bitów. Warto też znać ilość danych jaka będzie przesyłana, aby nie była ona przez przypadkowe zakłócenie zmieniona powinna być nadana dwukrotnie. Oczywiście trzeba nadać dane, warto też zaopatrzyć się w sumę kontrolną w celu weryfikacji poprawności odebranych danych. W przypadku poprawnie odebranych danych drugi modem powinien nadać potwierdzenie. Format ramki przesyłanych danych jest następujący:

- Preambuła – 32 bajty 0xAA
- Bajt startu – 0x0F
- Bajt potwierdzenia (ACK)
- Suma kontrolna – suma wszystkich bajtów danych dopełniona do dwóch

- Bajt określający ilość danych w ramce
- Powtórzony bajt ilości danych w ramce
- Dane – maksymalne 64 bajty

3.3 Nadawanie ramki

Nadawanie danych jest realizowane w przerwaniu zewnętrznym mikrokontrolera INT0, wyzwalane zboczem opadającym DCLK. Nadawanie ramki następuje w etapach:

1. Nadanie 32 bajtów preambuły, przed nadaniem ostatniego bajta ustawiana jest flaga tego zdarzenia (**PreambleTXC**).
2. Nadanie pozostałej części ramki opisanej w podpunkcie 3.2, po zakończeniu nadawania ustawiana jest flaga tego zdarzenia (**FrameTXC**).

3.4 Odbieranie ramki

Odczyt odbieranych danych jest realizowany w przerwaniu zewnętrznym mikrokontrolera INT0, wyzwalane zboczem narastającym DCLK. Odbiór ramki następuje w etapach:

1. **Wykrywanie preambuły** – Po przeczytaniu kolejnego bitu z linii DIO sprawdzamy czy bajt utworzony z ostatnich 8 przeczytanych bitów jest równy 0xAA lub 0x55. Jeśli wykryliśmy takich bajtów wystarczająco dużo pod rząd (1/4 długości preambuły) to możemy uznać, że preambuła została wykryta i przejść do następnego etapu odbioru ramki.
2. **Wykrywanie bajta startu ramki** – Czytamy kolejne bity z linii DIO. Jeśli bajt utworzony z ostatnich 8 bitów jest równy, ustalonemu bajtowi (np. 0xF0) to możemy przejść do następnego etapu.
3. **Odbieranie bajtów** – W tym momencie bity w odbieranych bajtach są zsynchronizowane. Po odebraniu bajtu jest on zapamiętywany w buforze i ustawiana jest flaga tego zdarzenia(**ByteRXC**).

3.5 Transmisja RS232

W mikrokontrolerze zaimplementowana została kolejka nadawanych danych o pojemności 256 bajtów i kolejka odbieranych danych także o pojemności 256 bajtów. Prędkość transmisji została ustawiona na 115200kBaud, format ramki to 8 bitów danych 1 bit stopu, brak sprawdzania parzystości.

Zaimplementowano obsługę linii RTS, która jest zerowana w momencie gdy w kolejce odbierania jest mniej niż 1/4 miejsca, i ustawiana w momencie gdy w buforze jest więcej niż 1/2 miejsca. Nadawanie danych jest zrealizowane w przerwaniu RXC (recive complited).

Zaimplementowano obsługę linii CTS, gdy jest w stanie wysokim następuje zaprzestanie nadawania dopóki nie będzie w stanie niskim.

3.6 Algorytm działania modemu

Algorytm działania oparty jest na obsłudze zdarzeń (ustawianych flag). Po obsłudze zdarzenia flagi są zerowane. Wyróżnione zostały następujące zdarzenia:

- **ByteRXC** – został odebrany bajt (flaga ustawiana w przerwaniu INT0)
 - Jeśli zostało odebrane 4 bajty (ACK, CRC, 2 bajty rozmiaru) to sprawdź czy rozmiar ramki jest prawidłowy. Jeśli nie to ustaw flagę błędu ramki **ErrorFrame** i flagę **FrameRXC** i przejdź do etapu wykrywania preambuły.
 - Jeśli odebrano całą ramkę (4+ilość danych bajtów) to ustaw flagę **FrameRXC** i sprawdź sumę kontrolną. Jeśli się nie zgadza to ustaw flagę błędu **ErrorCrc**.
- **FrameRXC** – została odebrana ramka.
 - Jeśli flaga błędu ramki **ErrorFrame** nie jest ustawiona:
 - * Jeśli ramka ma ustawione pole ACK i oczekujemy na potwierdzenie to zaznacz że nie oczekujemy już na potwierdzenie.
 - * Jeśli flaga błędu sumy kontrolnej **ErrorCrc** jest nie ustawiona:
 - Przekopij dane do bufora nadawania UART
 - Jeśli w ramce były dane to wyślij potwierdzenie.
 - * Wyczyść flagi błędów.
- **TimeExceeded** – został przekroczony czas oczekiwania na potwierdzenie.
 - Jeśli liczba prób wysyłania ramki nie jest wyczerpana to włącz nadawanie preambuły. w przeciwnym wypadku zeruj licznik prób wysyłania ramki.
- **PreambleTXC** – przedostatni bajt preambuły został wysłany.
 - Jeśli nie powtarzamy wysyłania ramki to przekopij dane z bufora odbiornika UART.
- **FrameTXC** – transmisja ramki została zakończona.
 - Jeśli wysłano dane to ustaw odmierzenie czasu.

- Włącz wykrywanie preambuły.
- Stan bezczynności modemu (nie nadajemy i nie odbieramy i nie oczekujemy na potwierdzenie).
 - Jeśli bufor odbiornika uart nie jest pusty to włącz nadawanie preambuły.

4 Podsumowanie

Po przetestowaniu modemu byłem zadowolony. Działał poprawnie, jedyną wadą jest mała prędkość transmisji – w jednym kierunku maksymalnie do ok 2kB/s. W przypadku gdy modemy wymieniają się jednym bajtem na przemian ilość przesłanych bajtów przez jeden modem w ciągu sekundy jest równa tylko ok 60.5.

Podczas pracy z modemem zauważono pewien problem: gdy dwa modemy chcą nadawać w tym samym czasie dane, to te dane zostaną utracone. Ze względu na to, że obydwa modemy są w trybie nadawania nie mogą odebrać ramki.

Aby urządzenie uczynić bardziej funkcjonalnym możnaby zaimplementować interfejs konfiguracyjny umożliwiający ustawienia parametrów transmisji radiowej oraz pokazujący statystyki transmisji i poziom odbieranego sygnału. W celu optymalizacji kod można napisać w assemblerze.

Spis rysunków

1	Schemat blokowy układu CC1000	3
2	Schemat połączenia układu CC1000 z mikrokontrolerem	3
3	Schemat połączenia układu CC1000 z mikrokontrolerem (SPI)	4
4	Operacja zapisu rejestru CC1000	5
5	Operacja odczytu rejestru CC1000	5
6	Zależność napięcia na RSSI od mocy odbieranego sygnału	6
7	Schemat modemu radiowego	7
8	Interfejs programu SmartRF Studio	9
9	Format ramki	12

Spis tabel

1	Typowe parametry układu CC1000 przy VCC=3V/25°C	3
2	Zestawienie elementów medemu radiowego	8

3	Przyjęte parametry transmisji radiowej	9
4	Konfiguracja rejestrów CC1000	10

Literatura

- [1] *www.chipcon.com*, „CC1000 Data Sheet”, Chipcon products from Texas Instruments, 2005.
- [2] *www.chipcon.com*, „CC1000PP User Manual Rev. 1.22”, Chipcon products from Texas Instruments, 2003.
- [3] *www.chipcon.com*, „Application note AN009”, CC1000/C1050 Microcontroller interfacing, Chipcon products from Texas Instruments, 2004.
- [4] *www.chipcon.com*, „CC1000 Errata Note 001, rev. 1.0”, Chipcon products from Texas Instruments.
- [5] *www.atmel.com*, „ATMega8 Data Sheet”, ATMEL, 2004.