Zadania realizowane przez Wydział Elektryczny PW

Programowanie i wykorzystanie modułu ADAM4500

Wstęp

W ćwiczeniu wykorzystujesz kontroler ADAM4500 do sterowania pracą modułu wykonawczego ADAM4018 przez port RS-485 (COM2) i obsługi portu RS-232 (COM1).

W pierwszej części ćwiczenia korzystasz z TC 2.0 i zajmujesz się wyłącznie RS-485, w który wpięty jest moduł ADAM 4018 pod adresem 01. Niniejsze wprowadzenie pozwoli Ci oswoić się z pracą 4018. Realizuj postawione zadania, o ile uznasz je za interesujące i ważne dla zrozumienia działania modułu, który będziesz wykorzystywał w dalszej części laboratorium.
W skrajnym przypadku możesz pominąć tę część ćwiczenia, bazując na wiedzy z podręcznika ADAM4018.
Bezpośrednie korzystanie z ADAMA4018 umożliwia sprzętowa przekładka interfejsu RS-232 standardowo dostępnego w starszych PC na RS-485 (moduł ADAM 4520), oraz aplikacja ADAM40005000.exe (dostępna z pulpitu). Masz wstępnie zestawione stosowne stanowisko. Zauważ, że moduł wykonawczy (ADAM 4018) wpięty jest w system czterema przewodami (2 zasilania i 2 RS-485), co pozwoli Ci go łatwo przenieść do zasobów ADAM4500, który jest używany w zasadniczej części ćwiczenia. Sterującym portem PC jest COM2, na COM1 jest mysz.

Podłącz kabelek RS PC-ADAM4520. Włącz zasilanie zestawu ADAM4520/ADAM4018 (10-30V, sprawdź polaryzację!). Uruchom ADAM4k5k z pulpitu.
Wybierz COM2 i korzystając z „lornetki” dokonaj inspekcji obiektów zainstalowanych na magistrali. Zgłosi się tylko jeden, więc przerwij szukanie po jego wykryciu. Kliknij na znaleziony moduł uruchamiając dedykowaną dla niego aplikację komunikacyjną. Otwórz wszystkie kanały Channel setup… (daj update, ale nie główny lecz w Channel setup).
Daj Refresh i zauważ że nie działa w pelni, nie pokazując wyników w kanałach. Nie przejmuj się, wkrótce poznasz przyczynę. Przejdź na poziom interfejsu (COM2) i wywołaj terminal (błyskawica na prawo od lornetki). „Porozmawiaj” z modułem sprawdzając działanie komend (tu AA to 01, bo adres modułu jest ustawiony na 01):

$AA2, $AAM, $AAF, #AA, #AA0, #AA1,…, #AA7, #AA8, $AA6, $AA5…

Projekt jest współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

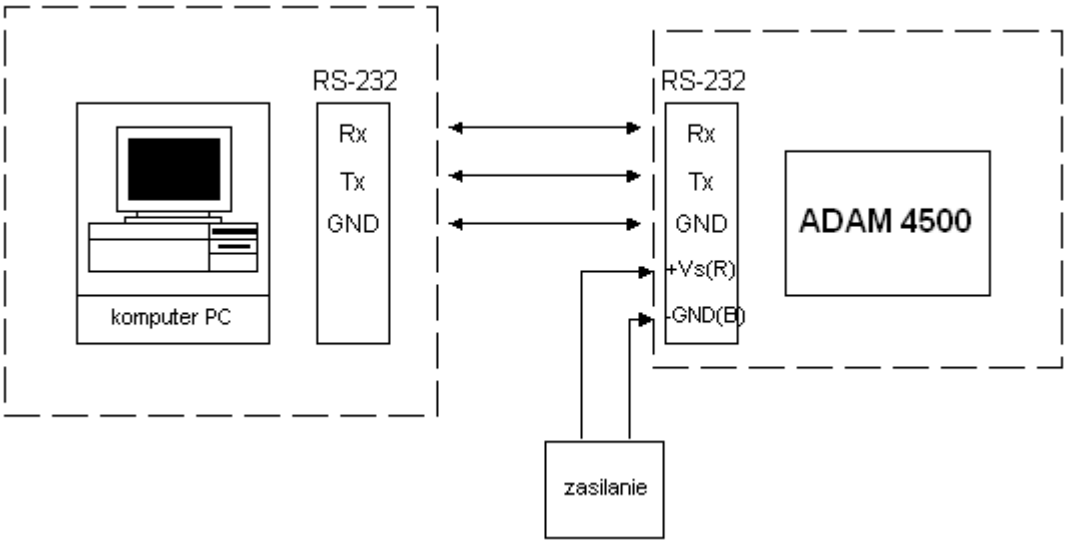
Skonfrontuj odpowiedzi z opisem z dokumentacji (podręcznik ADAM-4000ed7.pdf), rozdział 4.4. Zwróć szczególną uwagę na obsługę komend $AAM i #AAn dla n=0 do 7 – będziesz je wykorzystywał w dalszej części ćwiczenia.
Zamknij uruchomione programy narzędziowe na PC, wyłącz zasilanie stanowiska. Odłącz cztery przewody od modułu 4520 (RS485: DATA+/DATA- i zasilanie i przyłącz je do modułu kontrolera ADAM4500 (RS485/COM2 i zasilanie), nie pomyl +/-.

Przejdź do pierwszej części ćwiczenia

.

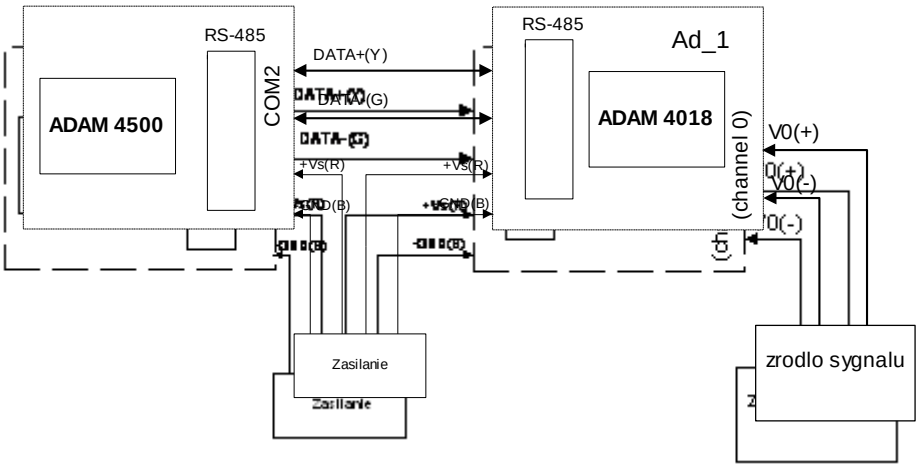
1. Moduł ADAM-4500, instrukcja wykonawcza – Turbo C 2.0.

Moduł ADAM-4500 posiada dwa porty komunikacyjne (COM1: konfigurowalny RS-485/RS-232 i COM2: RS-485) przeznaczone do komunikacji ze współpracującymi modułami kontrolno-pomiarowymi, oraz bezpośrednio niedostępne dla tworzonego oprogramowania łącze RS-232 do współpracy z PC (rys. 1). W ćwiczeniu do zarządzania modułami pomiarowymi wykorzystywany jest COM2 (rys.2), COM1 ustawiony jest do pracy w trybie RS-232 i jako full-duplex używany w testach transferów nadawanie-odbiór (2 część ćwiczenia).



Rys 1. Schemat połączenia PC z modułem ADAM4500

Uruchomione na PC oprogramowanie zarządzające kontrolerem ADAM-4500 pracuje w okienku DOS tworząc na ekranie PC odległą konsolę kontrolera. Konsola ta umożliwia obsługę dwóch dysków modułu 4500 (C i D), odbiera komunikaty z klawiatury które mogą sterować pracą programu na etapie jego uruchamiania, oraz przechwytuje komunikaty generowane przez funkcje printf używane w uruchamianym programie. Stanowi to istotną zaletę „systemu" uruchomieniowego, gdyż pozwala w klasyczny sposób korzystać z monitora alfanumerycznego na etapie przygotowywania samodzielnej aplikacji pracującej w module pozbawionym realnej konsoli. Dodatkowo poprzez funkcję copy systemu DOS użytkownik może przejrzeć zawartość plików tekstowych kontrolera wysyłając je na konsolę (con:).



Rys 2. Schemat połączenia modułu ADAM4500 z modułem ADAM4018 (Schemat połączeń PC z modułem ADAM 4500 jak na rys. 1).

1.1 Tworzenie oprogramowania

W ćwiczeniu oprogramowanie kontrolera tworzone jest w środowisku DOS Turbo C v.2.0 - wyłącznie dla niego producent udostępnia biblioteki (część 1), oraz nowocześniejszym (o ile można tak mówić o „historycznych" produktach) środowisku Borland C v. 3.1 (część 2).

Na dysku lokalnym PC znajduje się katalog z kompilatorami (e:\tc, e:\bc…), oraz katalog z programem służącym do komunikacji z modułem ADAM-4500 (e:\adam4500). Ponieważ na pulpicie masz stosowne skróty do w/w otwórz właściwości każdego z nich (ppm – prawy przycisk myszy) i w „Program" sprawdź ustawienia w wierszu poleceń i katalogu roboczym.

Przy programowaniu modułu 4500 w podkatalogu tc\library powinny znaleźć się niezbędne biblioteki 4500x.lib, a w podkatalogu tc\include zbiór adam4500.h (header) w którym zdefiniowane są funkcje biblioteczne. Przejrzyj pobieżnie zawartość headera i odnotuj, że zawiera on funkcje obsługi portów COM, watchdoga i diody LED, opisane w dokumentacji z którą zapoznałeś się w domu, oraz dwie dodatkowe funkcje związane z obsługą portów szeregowych przy wykorzystaniu przerwań. W katalogu adam4500 znajduje się program do komunikacji („zdalny terminal") z modułem ADAM4500 (plik adam4500.exe). Obsługa programu adam4500.exe wyjaśniona jest w rozdziale 3 instrukcji użytkowania kontrolera ADAM4500 (ADAM-4500 User's Manual - jest w folderze dostępnym z pulpitu przez ikonę Adam4500 zasoby: adam4500.pdf. Uruchamianie modułu opisane jest w rozdziale 2 tej instrukcji.

Programy powinny być skompilowane do postaci kodu akceptowanego przez mikroprocesor 80186 lub 80188. Należy to ustawić w opcjach kompilatora. W przypadku programu, korzystającego z biblioteki serii 4500*.lib należy otworzyć projekt (menu **Project** w kompilatorze TC 2.0). Przykładowy plik projektu (wykorzystywanego w dalszej części niniejszej instrukcji) wygląda następująco (plik DIODA.PRJ):

DIODA.C

4500S.LIB

Jak widać składa się on z nazwy pliku źródłowego i nazwy wykorzystywanej biblioteki.

Po napisaniu i pomyślnym skompilowaniu należy program przesłać do i uruchomić na module ADAM4500. W tym celu uruchomić należy program adam4500.exe (ikona na pulpicie). W opcji „COMport" należy wybrać port szeregowy PC, do którego podłączony jest moduł (aktualnie COM2 PC). Parametry transmisji nie podlegają konfiguracji i są zadane na stałe. ADAM4500 posiada dwa rodzaje pamięci, FlashROM (dysk C:) oraz SRAM (dysk D:). Dysk C pełni rolę twardego dysku kontrolera i jest standardowo dostępny tylko do odczytu. W ćwiczeniu do realizacji oprogramowania wykorzystywany jest wyłącznie dysk D o dostępie read/write. Modyfikacje dysku C są niebezpieczne z uwagi na możliwość utracenia łączności PC-kontroler i w przypadkach koniecznych mogą być realizowane wyłącznie pod kontrolą prowadzącego zajęcia.

Niniejsza instrukcja dotyczy więc pracy w trybie przenoszenia programów na dysk D: ADAMa 4500. Opcja „Terminal" programu komunikacyjnego umożliwia obsługę modułu tworząc jego zdalny terminal. Poruszanie się po nim jest analogiczne do obsługi środowiska DOS komputerów klasy PC- działa większość komend systemowych DOS. Wybierając kombinację <Alt-T> można dokonać transferu dowolnego pliku z PC do pamięci RAM (dysk D) modułu. Kolejno pojawią się okna: „źródłowe" i „docelowe", w które wpisuje się nazwę pliku wraz ze ścieżką odpowiednio: źródłową i docelową. Program uruchamia się tak jak w DOS'ie wpisując jego nazwę. Skrótem <Alt-X> można opuścić terminal modułu.

1.2 Przykładowy program:

DIODA.C (DIODA.EXE)

Schemat połączeń tak jak na rys. 1. Program w pętli 10 razy zapala i gasi diodę (funkcja led()), z uwzględnieniem czasu opóźnienia (funkcja delayy()). W programie **dioda.exe** wykorzystana jest biblioteka 4500S.LIB dlatego też źródło zawiera stosowny nagłówek.

```
#include "adam4500.h"
#include <dos.h>

void delayy(int time)
 {
   union REGS wejscie;
   union REGS wyjscie;
   int go=0;
   int tim=0;

   wejscie.r_ax=0;
   int86(0x1a,&wejscie,&wyjscie);
   go=wyjscie.dx;
   do
```

```
    {
        wejscie.r_ax=0;
        int86(0x1a,&wejscie,&wyjscie);
        tim=(wyjscie.dx-go);
    } while (tim<(time*20));

 }

void main(void)
        {
        int i;
        led_init();
        for(i=0; i<10; i++)
        {
                led(0);
                delayy(1);
                led(1);
                delayy(1);
        }

}
```

1.3 Realizacja ćwiczenia

1. Skonfiguruj, połącz z PC kabelkiem RS i włącz zasilanie stanowiska sprzętowego ADAM500/ADAM4018.
2. Uruchom aplikację ADAM4500.exe (**ikona na pulpicie**).
3. Wybierz port COM2 (wciśnij **C**, a następnie ustaw jako aktywny port **COM2**).
4. Zapoznaj się z zawartością dysków C i D kontrolera. W tym celu należy wejść w tryb terminala (np. wcisnąć **T)**, co umożliwi pracę bezpośrednio na module ADAM 4500. Moduł powinien zgłosić znak zachęty C:\>, jeśli tak nie jest wyłącz i włącz zasilanie Adama by go zresetować. Sprawdź działanie dir i przechodzenie c:/d:. System nie ma helpa, ale odpowiada „Bad command…" gdy nie rozpoznaje polecenia i „invalid…" gdy zna, ale nie może wykonać polecenia (np. type). To pozwala eksperymentować z wywołaniami. Wykorzystaj komendy systemu operacyjnego DOS: „copy c:\autoexec.bat con:" oraz „type c:autoexec.bat" do transferów pliku tekstowego autoexec.bat pomiędzy dyskiem ADAM-4500 i konsolą. Sprawdź skuteczność transferów i skomentuj zawartość pliku. Wejdź na dysk D: i sprawdź co się stanie gdy wykonasz:

Copy con: qq

1 2 3 4 5 CTRL-z enter

Druga linijka wprowadza dane z klawiatury bezpośrednio do nasłuchującego copy na konsoli (con:). CTL-z „zamyka" nadawany plik i pozwoli zamknąć realizację polecenia copy z pierwszej linii. Wpisuj wiec po kolei 1,2…

Obejrzyj zawartość wygenerowanego pliku wysyłając go na konsolę.

5. Dokonaj transferu pliku tekstowego z PC (np. e:\rob\tcrob\dioda.c) na dysk D kontrolera 4500 (*użyj ALT-T*). Odczytaj zawartość pliku poprzez jego transfer na konsolę (*copy d:\dioda.c con:*).

Uwaga: w/w scieżka e:\rob\tcrob jest scieżką roboczą środowiska Turbo C, na niej umieszczasz i kompilujesz programy źródłowe, z niej odbierasz kody wynikowe (*.exe), które ładujesz do ADAM4500.

6. Zapoznaj się z zawartością pliku „dioda.c". Przeanalizuj program sterowania diodą w module ADAM 4500. Należy zwrócić uwagę, że funkcja „delayy (int time)" nie wykorzystuje żadnych funkcji bibliotecznych i odwołuje się bezpośrednio do BIOS („Podręcznik użytkownika ADAM 4500", s. 11). Program główny main odwołuje się do bibliotek (led_init, led (0), led (1) „Podręcznik użytkownika ADAM 4500", s. 19).
7. Dokonaj transferu programu „dioda.exe" na dysk D modułu 4500. Użyj ALT-T**.**
8. Uruchom w module program dioda.exe. Sprawdź czy działa zgodnie z oczekiwaniem.

Poprawnie wykonywany program świadczy o prawidłowym działaniu środowiska i kontrolera. Stanowi dobry punkt wyjścia do tworzenia własnej aplikacji.

9. Tym razem uruchomisz własną, nieznacznie zmodyfikowaną wersję programu dioda. Uruchom Turbo C (ikona TC na pulpicie). Zostanie otworzone okno programu TC. Otwórz do edycji program źródłowy dioda.c (wyjście z edytora – F10). Istotną jego zaletę edytora stanowi możliwość uzyskania pomocy ogólnej (F1) i pomocy kontekstowej odnośnie dostępnych struktur języka C (sprawdź to: naprowadź kursor na funkcję i wciśnij CTRL-F1).
10. Kompilację programu realizujemy nie przez run (co zwykle ma miejsce), lecz przez compile gdyż program uruchamiany jest na zewnątrz (najlepiej użyć *complie -> build*

all). Dostaniesz ostrzeżenia i błędy. Jak z nich wynika, głównym problemem jest linker, który „nie widzi" funkcji bibliotecznych. Musisz załadować projekt – jest już przygotowany, obejrzyj go (w TC: File>load>*.prj…, w Windows: send to>notatnik). Załaduj: Project i dalej…

11. Skompiluj (nie powinno być błędów), przerób rob.c tak by wypisywał powitanie (np. Dzialam...), zróżnicuj czasy gaszenia o zapalania diody.

12. Uruchom tak zmieniony program. Sprawdź skuteczność swych modyfikacji.

Obecnie panujesz nad środowiskiem tworzenia i uruchamiania programu oraz przygotowaniem własnego projektu.

13. Postępując podobnie jw. uruchom program testplik.c - najpierw w środowisku PC, potem, po niezbędnych modyfikacjach w kontrolerze. Zweryfikuj poprawność tworzonych plików w jednym i drugim środowisku.

14. Zapoznaj się z programem prezent.c, uruchomić jego wersję wykonawczą (prezent.exe) i skonfrontuj działanie z programem źródłowym, przeanalizuj jego strukturę.

15. Przeanalizuj działanie poszczególnych podfunkcji programu, np. read_voltage, write_status. Zwrócić szczególną uwagę na te, które wykorzystują funkcje biblioteczne.

16. Zmodyfikuj nieznacznie i uruchom swą wersję programu prezent. Dokonaj bardziej śmiałych modyfikacji zmierzających do stworzenia programu zbierania danych do tekstowego pliku dyskowego.

17. Zmodyfikuj program tak by zapisywał dane do pliku, sprawdź poprawność jego działania.

18. Przeorganizuj program tak, by pracował autonomicznie i samodzielnie kończył pracę po zebraniu określonej (niewielkiej) porcji danych. Wyposaż go w stosowne fragmenty oczekiwania na rozłączenie łącza i sygnalizacji końca pracy. Skutkiem działania tego programu powinien być plik z danymi pomiarowymi.

19. Pozamykaj programy narzędziowe, wyłącz zasilanie modułów ADAM4500/ADAM4018.

Na tym etapie ćwiczenia panujesz nad modułem ADAM4500 i umiesz uruchamiać w nim własną aplikację, korzystającą z interfejsu szeregowego i obsługi dysku przez s.o. DOS. Korzystasz z starego kompilatora TC, ale nie wykorzystujesz jego zalet związanych z dostępem do funkcji bibliotecznych obsługi portów COM. Jak się zapewne już zorientowałeś funkcje te naprawdę są mało użyteczne – np. brak w nich funkcji wysyłania stringu. Możesz więc bez żalu porzucić TC i przejść do programowania w nieco nowszym BorlandC 3.1 (część 2 ćwiczenia).

2. Moduł ADAM-4500, instrukcja wykonawcza – Borland C 3.1.

W części pierwszej ćwiczenia nauczyłeś się korzystać z modułu ADAM4500 do obsługi własnej aplikacji. Oprogramowanie w minimalnym stopniu korzysta z funkcji bibliotecznych (program dioda), które mogą być zastąpione wywołaniami źródłowymi zaczerpniętymi z przykładowego programu example4.c. Nie będziesz obecnie tego realizował, ale wskaż zmiany, które musisz wprowadzić w kodzie źródłowym by wynieść program dioda.c poza środowisko TC2.0 (zaczerpnij wiedzę z example4.c).

Jeśli jesteś zainteresowany i czujesz się na siłach to możesz poświęcić chwilkę na „podniesienie" dioda.c w środowisku BC – zajmie Ci to zapewne mniej niż kwadrans…

Praca w nowym środowisku nie jest nastawiona na obsługę aplikacji, jak w części pierwszej ćwiczenia, lecz podniesienie funkcjonalności obsługi transferów danych przez wykorzystanie funkcji bibliotecznych środowiska (bioscom) i przerwań. Punktem wyjścia będą dwa programy obsługi portu COM, które uruchomisz na PC a potem, po niezbędnych przeróbkach, na module ADAM4500: termpoll.c i buff1024.c. Programy te są opisane w autorskim podręczniku "Interfacing the Serial / RS232 Port" C. Peacocka, dostępnym w sieci i udostępnionym w zasobach dyskowych związanych z ćwiczeniem

Termpoll PC

Termpoll jest prostym programem wysyłającym i odbierającym dane z portu COM. W wersji na COM2 skompilujesz i uruchomisz go na PC. Zauważ, że struktura programu jest identyczna ja programów przykładowych które uzywałeś w pierwszej części ćwiczenia. Do obserwacji sygnałów wyjściowych użyj oscyloskopu (oczywiście pracujesz teraz na porcie COM2 komputera PC – masz go na kabelku RS, który był/jest podłączony do ADAM4500). Najpierw sprawdź wyjście (nóżka 2 lub 3) – wejście ma poziom bliski zeru, wyjście - napięcie ujemne (-3 do -15V zgodnie ze standardem). Podłącz sondę oscyloskopu do wyjścia, ustaw czułość na 5V/działkę i podstawę czasu na 1 ms/działkę.
Skompiluj i uruchom program. Otwórz tym razem BC – ikona BC 3.1 na pulpicie, z uwagi na znaczne podobieństwo TC i BC nie jesteś już prowadzony „za rękę". Pracę w BC możesz prowadzić bez projektu – nie korzystasz z bibliotek, wystarczy więc, że załadujesz program do okna edycyjnego (File>open…). Daj Compile>buid all i uruchom exec'a. Jeśli widzisz reakcję na ekranie oscyloskopu to przełącz trigger mode na normal i sprawdź zmiany w sygnałach wystawianych na wyjściu przy nadawaniu cyfr 0,1,2…
Zgodnie z oczekiwaniami esc kończy działanie programu. Na ekranie terminala nic ciekawego się nie dzieje. Zastanów się co zrobić, by udrożnić terminal – może połączyć wej z wyj na łączu RS?
Zrób tak i ciesz się odbiornikiem – zauważ, że znak z klawiatury „przechodzi" teraz fizycznie przez COM. Sprawdziłeś działanie portu w trybie zapisu i odczytu. Przerób program tak, by używając funkcji bioscom wyeliminować bezpośrednie odwołania do portów układu obsługi łącza RS.

Teraz masz samodzielne zadanie uruchomienia swojej wersji programu w ADAM4500. Powinieneś wiedzieć co robić… Pamiętaj że port RS-232 jest tylko na COM1/ADAM4500.
Zgłoś prowadzącemu działanie programu w ADAMIE, lub problem z jego uruchomieniem o ile się zacinasz.

Buff1024

Program jest bardziej skomplikowany - korzysta z przerwań, uruchom jego (nieco przerobioną) finalną wersję (buff_PC.exe buffADAM.exe) w PC i Adamie. Nie dostaniesz do niej źródeł, a będziesz dążył do uzyskania podobnej funkcjonalności (aby zobaczyć wyniki musisz dać esc).
Punktem wyjścia jest oryginalny program źródłowy buff1024.c.
Przed rozpoczęciem pracy powinieneś zwrócić uwagę na przechwywanie i ustawianie wektora przerwań (getvect/setvect), strukturę podprogramu obsługi przerwania i jego interakcje z hardwarem (outport (0x20,0x20)), oraz przemyśleć sposób testowania pracy programu, który tym razem napędzany jest przerwaniami. Nie jest to łatwe i wymaga świadomego i przemyślanego działania, ale doprowadza do stworzenia narzędzi umożliwiających prowadzenie nasłuchu na łączu komunikacyjnym w trakcie normalnej pracy modułu, który nie obciąża procesora.