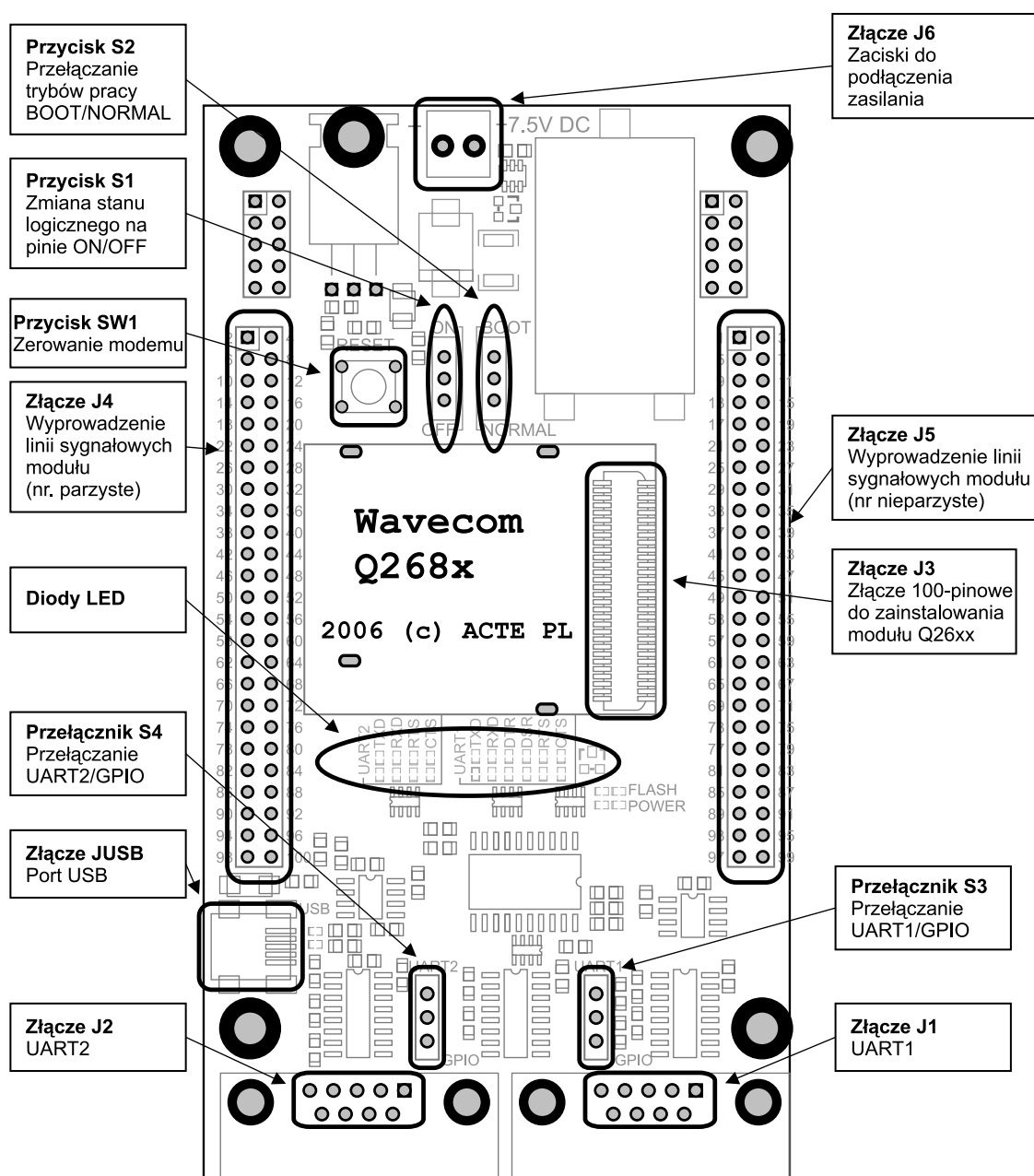


# Moduły GSM w praktyce (2)

## Pierwsze kroki w środowisku OpenAT

Po przerwie wznawiamy kurs na temat modułów GSM nie wymagających mikrokontrolera sterującego, ponieważ program użytkownika jest wykonywany przez sam moduł. W części drugiej przedstawiamy płytę prototypową dla modułów typu WaveCom Q268x oraz sposób instalacji środowiska programistycznego OpenAT, które umożliwi programowanie w języku C. Ponadto opisano metody wgrzywania i uruchamiania oprogramowania dla modułu WaveCom Q2686.



Rys. 7. Widok płyty prototypowej wraz z krótkim opisem elementów

Tab. 2. Nazwy i funkcje sygnałów na złączu UART1 (J1)

Numer pinu	Nazwa sygnału	Kierunek	Opis
1	DCD	Wyjście	Data Carrier Detect
2	RXD	Wyjście	Receive Serial Data
3	TXD	Wejście	Transmit Serial Data
4	DTR	Wejście	Data Terminal Ready
5	GND	–	–
6	DSR	Wyjście	Data Set Ready
7	RTS	Wejście	Request to Send
8	CTS	Wyjście	Clear to Send

### Płytki prototypowa dla modułu WaveCom Q2686

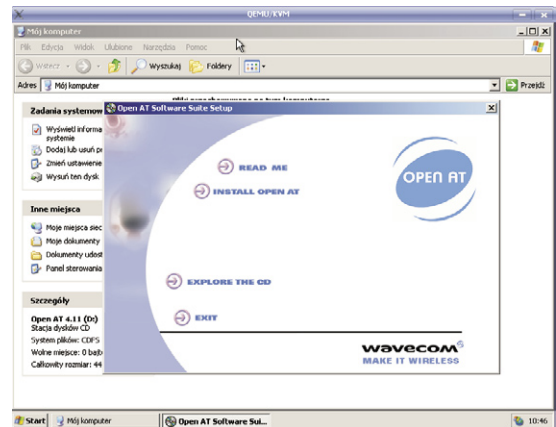
Aby przy pierwszym kontakcie uwolnić użytkownika od zmagania ze sprzętem, w dalszej części kursu będziemy posługiwać się płytką ewaluacyjną dla modułów WaveCom, która pozwala tworzyć aplikacje testowe dla modułów GSM. Płytki posiada 100-pinowe złącze kompatybilne

z modułami Q26xx, układ zasilacza, interfejsy RS232, diody LED oraz złącze USB. Na płytce umieszczono także pola lutownicze w rastrze 2,54 mm, na które wyprowadzono wszystkie linie sygnałowe dostępne na złączu modułu, umożliwiając w ten sposób wygodne dołączanie do płytki prototypowej dowolnych układów zbudowanych przez użytkownika. Poszczególne numery pinów na polach lutowniczych są zgodne z numerami na złączu w module GSM. Na rys. 7 przedstawiono widok płytki prototypowej wraz z krótkim opisem znajdujących się na niej elementów. Na płytce znajduje się seria przełączników umożliwiających dołączenie wejść modułu GSM do wewnętrznych układów peryferyjnych znajdujących się na płytce prototypowej, albo do pól lutowniczych. Dzięki temu rozwiązaniu wszystkie linie dostępne w module mogą być wykorzystane w układzie użytkownika. Przełącznik BOOT/NORMAL służy do włączenia specjalnego trybu wgrzywania oprogramowania za pomocą programów firmowych, podczas normalnej pracy powinien znajdować się w trybie NORMAL. Szczególne znaczenie ma złącze UART1, za pomocą którego możemy skomunikować się z modulem za pomocą komend AT, a także wręcać do niego wcześniej przygotowany program. Nazwy i funkcje sygnałów na tym złączu zestawiono w tab. 2.

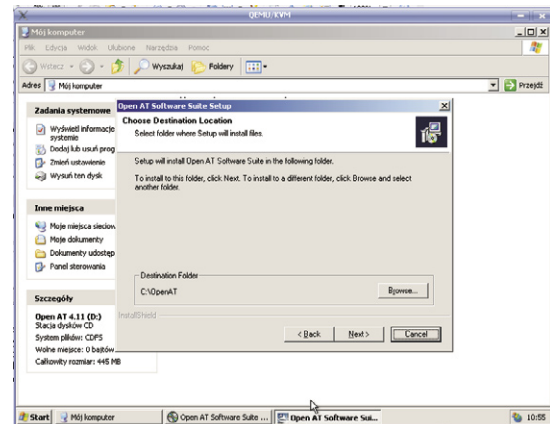
Diody LED na płytce prototypowej sygnalizują stan włączenia oraz stany poszczególnych linii RS232, jednak istnieje możliwość sterowania kilkoma diodami LED bezpośrednio za pomocą modułu, jeżeli niektóre z tych linii są skonfigurowane w trybie IO. Aby uruchomić moduł w normalnym trybie pracy, do złącza J6 należy podłączyć zewnętrzne źródło napięcia o wartości 7...9 V i wydajności prądowej 0,5 A. Złącze portu szeregowego UART1 (J1) należy również podłączyć do złącza szeregowego komputera PC oraz ustawić przełączniki w następującej pozycji: UART1->UART, S1->OFF, S2->NORMAL. Aby skomunikować się z modulem, należy uruchomić program terminalowy z następującymi parametrami portu szeregowego: prędkość – 115200, bity danych – 8, bity stopu – 1, kontrola parzystości – brak, sterowanie przepływem – sprzęt. Po włączeniu modułu powinna zaświecić się dioda POWER, a po wpisaniu w programie terminalowym komendy AT i wciśnięciu klawisza [ENTER] moduł powinien odpowiedzieć „OK”, co świadczy o prawidłowej jego pracy.

### Instalacja środowiska OpenAT, uruchamianie oprogramowania w module OpenAT. Obsługa modułu za pomocą standardowych komend AT

Aby tworzyć własne oprogramowanie dla modułów WaveCom, należy zainstalować środowisko OpenAT. Jako minimalne wymagania programowe producent zaleca używanie systemów Windows 2000/XP oraz komputera minimum Pentium 300 MHz (zalecany 1 GHz) posiadającego minimum 128 MB pamięci (zalecane 512 MB) oraz port szeregowy RS232. Jak więc widzimy, wymagania nie są zbyt wygórowane i spełnia je w zasadzie każdy współczesny komputer. Producent zaleca również, aby w systemie było zainstalowane środowisko Visual C++ w wersji 6.0 lub 2002/2003, jednak na potrzeby niniejszego artykułu nie jest ono potrzebne, ponieważ do pisania programów będziemy wykorzystywać darmowe środowisko Eclipse. Aplikacje dla modułów WaveCom mogą być tworzone w trybie Target Mode, gdzie w łatwy sposób mogą być debugowane, jednak wymaga to konieczności użycia Visual C++. Z tego powodu



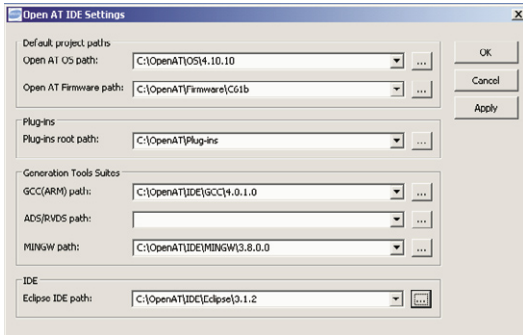
Rys. 8. Okno powitalne instalatora środowiska OpenAT



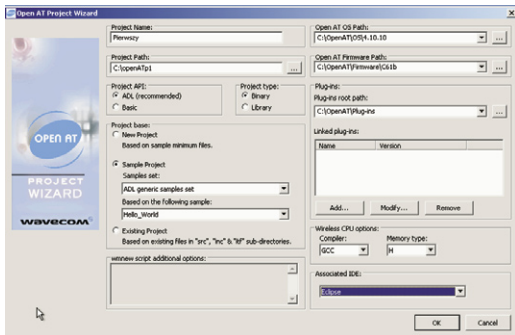
Rys. 9. Okno wprowadzania ścieżki instalacji środowiska OpenAT

w niniejszym cyklu nie będziemy wykorzystywać tego trybu. Drugi dostępny tryb tworzenia aplikacji to Remote Mode, w którym aplikacja jest uruchamiana bezpośrednio w module docelowym i nie wymaga wykorzystywania Visual C++, dlatego w pozostałej części cyklu będziemy wykorzystywać ten tryb uruchamiania programów w module Q2686. Przed instalacją środowiska OpenAT należy upewnić się, że mamy zainstalowany interpreter Javy, a w przypadku jego braku należy zainstalować go ze strony: [http://java.com/en/download/windows\\_manual.jsp](http://java.com/en/download/windows_manual.jsp) Instalację oprogramowania rozpoczynamy od włożenia płytki CD do napędu, co spowoduje wyświetlenie okna powitalnego jak na rys. 8. Aby rozpocząć proces instalacji należy wybrać opcję Install Open AT, wówczas zostanie uruchomiony program instalatora, w którym należy wybrać przycisk Next. Następnie pokaże się kolejne okno z pytaniem o akceptację warunków licencji. Do przeprowadzenia dalszego procesu instalacji należy wybrać opcję „I Agree”. Po tej czynności instalator przechodzi do pytania o ścieżkę instalacji, którą należy ustawić tak, jak na rys. 9, a następnie wciśnięciem przycisku [Next]. Kolejnym etapem instalacji jest pytanie o komponenty, które chcemy zainstalować. Należy zaznaczyć wszystkie dostępne opcje i przejść do dalszego procesu instalacji wybierając klawisz [Next]. Po tej czynności rozpocznie się właściwy proces instalacji środowiska OpenAT, który z uwagi na dużą liczbę plików do skopiowania może trwać nawet kilkanaście minut (w zależności od szybkości komputera). Po zakończeniu instalacji zostało nam jeszcze odpowiednio ustalenie ścieżek środowiska OpenAT. W tym celu z menu start należy uruchomić program „WaveCom->IDE->OpenAT IDE Settings” oraz ustawić ścieżki dostępu do poszczególnych elementów środowiska OpenAT, tak jak na rys. 10. Po zakończeniu instalacji środowiska mamy już komplet programów potrzebnych do pracy z modułami WaveCom. Pokażemy teraz, w jaki sposób skompilować i uruchomić w module przykładowy program wyświetlający co sekundę przez port szeregowy modułu komunikat „Hello from Open AT”.

Do tworzenia aplikacji na podstawie wzorców służy specjalny kreator projektów „Open AT Project Wizard”. Aby go uruchomić, należy z menu Start wybrać: „WaveCom->IDE->OpenAT Project Wizard”. Pokaże się wówczas okno kreatora projektów (rys. 11). Kreator ten służy do tworzenia pustego projektu



Rys. 10. Okno ustawiania ścieżki dostępu do poszczególnych elementów środowiska OpenAT.

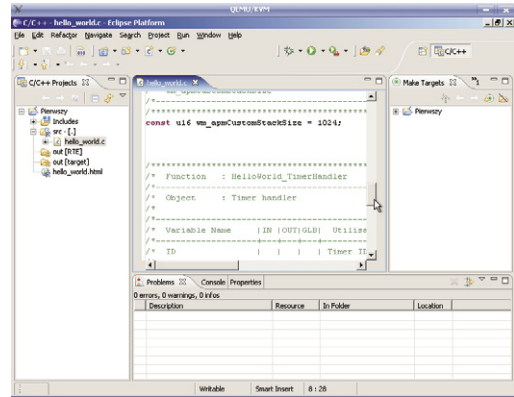


Rys. 11. Okno kreatora projektów

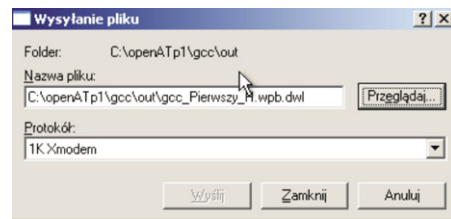
dla środowiska OpenAT lub projektu opartego na jakimś standardowym przykładzie. Aby utworzyć projekt, musimy wypełnić kilka podstawowych pól, a następnie kliknąć przycisk [OK]. Wówczas zostanie utworzony przykładowy lub pusty projekt oraz zostanie automatycznie uruchomione środowisko IDE umożliwiając w ten sposób dalsze pisanie programu. W polu „Project Name” musimy wpisać nazwę tworzonego projektu, natomiast w polu „Project Path” wpisujemy ścieżkę do projektu. W polu „Project API” wybieramy rodzaj interfejsu API środowiska OpenAT. W prezentowanych tutaj przykładach będziemy korzystać zawsze z API ADL. Pole „Project Base” pozwala określić, na jakim przykładowym projekcie będzie się opierał nasz projekt. Pole „Associated IDE” pozwala wybrać rodzaj środowiska, jakie będziemy wykorzystywać. W naszym przypadku powinno to być zawsze darmowe środowisko Eclipse. Po ustawieniu wszystkich opcji, tak jak na rys. 11 oraz wciśnięciu klawisza [OK], zostanie utworzony przykładowy projekt, a następnie zostanie uruchomione środowisko Eclipse (rys. 12).

Środowisko Eclipse zostało szczegółowo omówione przy okazji kursu programowania mikrokontrolerów ARM, dlatego nie będziemy tutaj poświęcać mu zbyt wiele miejsca. Zainteresowanych tą tematyką odsyłam do odpowiednich numerów EP. Pliki źródłowe projektu są umieszczone w polu „C++ projekt” w katalogu src, po rozwinięciu którego pokaże się jeden plik *hello\_world.c* zawierający przykładowy projekt. Aby skompilować ten projekt, należy z menu głównego wybrać „Project→Clean”, a następnie „Project→Build All”. Pierwsza komenda powoduje wyczyszczenie projektu z plików wynikowych i zostanie tylko plików źródłowych, a następna komenda spowoduje ponowne skompilowanie projektu, które przy pierwszym budowaniu może trwać nawet kilkadziesiąt sekund. Po skompilowaniu w podkatalogu projektu *gcc/out/* zostanie utworzony plik *gcc\_Pierwszy\_H\_wpb.dwl*, który jest plikiem wynikowym w formacie XMODEM. Możemy go wgrać do modułu i uruchomić. Samo programowanie oraz obsługa modułu odbywa się z wykorzystaniem zwykłego programu terminalowego i polega na wydawaniu odpowiednich komend AT. Aby zaprogramować modem, łączymy port szeregowy płytki prototypowej modułu WaveCom z komputerem PC, a następnie uruchamiamy program terminalowy (np. *Hyperterminal*, czy linuxowy *minicom*) z parametrami transmisji: 115200,8,n,1, sprzętowa kontrola przepływu.

Po wydaniu w terminalu polecenia AT, moduł powinien odpowiedzieć OK. Jeżeli tak nie jest, należy sprawdzić kabel połączeniowy oraz upewnić się, że ustawiona jest prawidłowa prędkość transmisji. Jeżeli natomiast wszystko



Rys. 12. Okno środowiska Eclipse



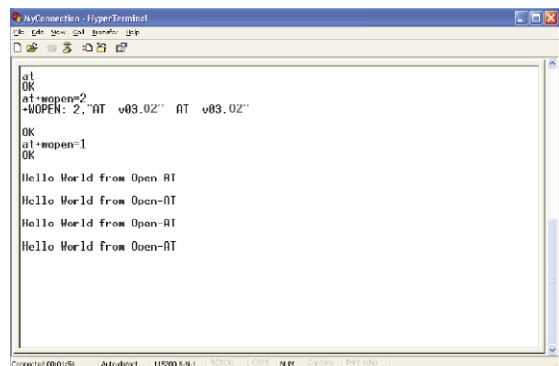
Rys. 13. Okno z parametrami pliku przesyłanego do modułu

przebiega prawidłowo, wówczas możemy przystąpić do zaprogramowania modułu. W tym celu wydajemy komendę AT+WDWL (rozpoczęcie transmisji w trybie XMODEM), po której na ekranie co 1 sekundę powinien się pojawiać znaczek „temidy”. Aby zaprogramować modem, należy z menu „Transfer” wybrać opcję „Send a File”, a następnie wybrać plik do zaprogramowania (*gcc\_Pierwszy\_H\_wpb.dwl*) oraz wybrać protokół transmisji XMODEM (rys. 13).

Po wciśnięciu klawisza Wyślij rozpocznie się proces wysyłania pliku. Po zakończeniu, na terminalu ponownie rozpocznie się cykliczne wyświetlanie znaku „temidy”. Mamy więc już zaprogramowany moduł, następnie należy go wyzerować wydając komendę AT+CFUN=1. Po wyzerowaniu modułu możemy teraz uruchomić zaprogramowaną przez nas aplikację. W tym celu wydajemy komendę AT+WOPEN=1, w wyniku czego na terminalu cyklicznie powinien pojawiać się komunikat jak na rys. 14. Aby wyłączyć uruchomiony program możemy posłużyć się komendą (AT+WOPEN=0).

Jak wiemy sterowanie modułem Q2686 odbywa się poprzez przesyłanie do niego odpowiednich komend AT, przydatne więc będzie poznanie choćby tych najczęściej używanych. Zostały one zebrane w tab. 3.

W codziennej praktyce bardzo użyteczne będą komendy umożliwiające skasowanie pamięci Flash danych użytkownika (AT+WOPEN=3) oraz skasowanie pamięci Flash aplikacji użytkownika (AT+WOPEN=4). Czasami konieczne jest skasowanie pamięci danych przy zmianie aplikacji. W zasadzie nigdy nie musimy kasować pamięci Flash programu, ponieważ następuje to automatycznie podczas wgrzywania nowej wersji programu. Dodatkowego omówienia wymagają komendy AT służące do obsługi SMS, dzięki którym użytkownik może



Rys. 14. Informacje wyświetlane na monitorze po uruchomieniu własnej aplikacji w module

**Tab. 3. Najczęściej używane komendy AT podczas programowania i uruchamiania aplikacji w module Q2686**

Komenda	Opis
AT+CFUN=1	Zerowanie modułu
AT+WOPEN=0	Zatrzymanie aplikacji użytkownika
AT+WOPEN=1	Uruchomienie aplikacji użytkownika
AT+WOPEN=2	Odczytanie wersji bibliotek OpenAT
AT+WOPEN=3	Kasowanie pamięci Flash danych użytkownika
AT+WOPEN=4	Kasowanie pamięci Flash aplikacji użytkownika
AT+WDWL	Przesyłanie aplikacji użytkownika protokołem XMODEM

**Tab. 4. Zestawienie komend służących do wysyłania wiadomości tekstowych**

AT+CMTI: „SM”,1	Komenda wysyłana przez moduł, informująca o odebraniu wiadomości, którą można odczytać spod indeksu 1
AT+CMGR=n	Odczyt wiadomości SMS o indeksie n
AT+CMGD=n	Kasowanie wiadomości SMS o indeksie n
AT+CMGS=“tel”	Wysyłanie wiadomości pod numer telefonu tel

odbierać oraz wysyłać wiadomości SMS bezpośrednio za pomocą terminala. Jest to o tyle użyteczne, gdyż bardzo często operator za pomocą odpowiednich wiadomości SMS przesyła informację np. o konieczności doładowania konta itp. Zestaw użytecznych komend służących do wysyłania wiadomości tekstowych przedstawiono w tab. 4.

Odczyt wiadomości SMS jest bardzo prosty. W momencie, gdy zostanie odebrana wiadomość, moduł informuje użytkownika o jej odebraniu za pomocą komendy AT+CMTI, którą przesyła do terminala wraz z indeksem tej wiadomości. Użytkownik po odebraniu tej wiadomości może odczytać jej treść za pomocą komendy AT+CMGR=i, gdzie i jest indeksem, który został przesłany przez moduł w momencie odebrania wiadomości. Po przeczytaniu wiadomości, jeżeli chcemy, możemy ją skasować za pomocą komendy AT+CMGD=i, gdzie i jest indeksem wiadomości do skasowania. Musimy pamiętać, aby czasami kasować zawartość wiadomości odebranych, ponieważ wiadomości te są przechowywane na karcie SIM modułu. Aby wysłać wiadomość, musimy użyć komendy AT+CMGS=tel. Na przykład jeżeli chcemy wysłać wiadomość pod numer 660428360, musimy wydać komendę AT+CMGS=“+48660428360”, a następnie nacisnąć ENTER. Moduł oczekuje wówczas na tekst, który musimy wpisać w terminalu. Po zakończeniu wpisywania tekstu należy wcisnąć klawisz [CTRL]+[Z], co spowoduje wysłanie wiadomości.

Należy pamiętać, że zaprezentowane tutaj komendy stanowią jedynie minimalny wycinek zestawu komend, tak więc zainteresowani Czytelnicy powinni zapoznać się z dokumentacją dołączoną do modułu.

### Pierwszy program, czyli Porty Wejścia/Wyjścia oraz Timery

Po zdobyciu podstawowej wiedzy na temat modułów Q2686 oraz wstępnym zapoznaniu się ze środowiskiem programistycznym OpenAT napiszemy kilka przykładowych aplikacji, z jakimi możemy się spotkać wykorzystując moduły GSM. Na początku większości kursów dotyczących programowania mikrokontrolerów zapoznajemy się ze sposobem obsługi portów wejścia wyjścia. Podobnie będzie i w naszym cyklu. Najpierw napiszemy prosty program, który będzie błyskał diodami LED znajdującymi się w zestawie uruchomieniowym modułu.

Jak już wspomniano we wcześniejszej części cyklu, programy pisane na moduły WaveCom działają pod kontrolą systemu operacyjnego OS6.60, dlatego pisząc własne programy będziemy musieli się dostosować do pewnych reguł panujących w tym systemie. Po pierwsze, komunikując się z różnymi układami peryferyjnymi modułu nie będziemy nigdy odwoływać się bezpośrednio do sprzętu. Aby wykonać jakąś czynność, na przykład odczytać stan linii portów wejścia-wyjścia, będziemy wykorzystywać zestaw funkcji API udostępnionych przez system OS. Po drugie, aplikacja użytkownika nie jest nigdy wywłaszczana, czyli nie może nigdy używać pętli nieskończonych, a długotrwałe operacje powinny być podzielone na mniejsze fragmenty. Niespełnienie tego warunku spowoduje zadziałanie układu Watchdog, a tym samym wyzerowanie modułu

**List. 1. Minimalna aplikacja dla modułu Q2686**

```
#include "adl_global.h"
//
//Rozmiar stosu dla programu
const ul6 wm_apmCustomStackSize = 1024;
//-----
//Funkcja glowna konfigurujaca
void adl_main ( adl_InitType_e InitType )
{
}
//-----
```

Q2686. Po trzecie, z drugiego warunku wynika, że wszystkie funkcje API systemu nie są blokujące (nigdy nie czekają), a cały system jest oparty na zdarzeniach. Zdarzenie jest to zwykła funkcja, która jest wywoływana w momencie wystąpienia jakiegoś zjawiska, może to być na przykład upływanie określonego czasu. Aby funkcja była wywoływana w momencie wystąpienia zdarzenia, musi być ona wcześniej zarejestrowana przez specjalną funkcję API systemu operacyjnego. Jak więc widzimy, jesteśmy ograniczeni pewnymi regułami, które z góry narzuca nam sam system operacyjny, a programy pisane dla modułu trochę się różnią od programów, do których jesteśmy przyzwyczajeni podczas pracy z mikrokontrolerami. Na list. 1 przedstawiono szkic minimalnej aplikacji dla modułu Q2686

Program pisany dla modułu OpenAT musi spełniać pewne minimalne warunki, czyli musi mieć zdefiniowaną zmienną `wm_apmCustomStackSize` określającą rozmiar stosu aplikacji. Minimalny rozmiar stosu wynosi 1 kB. Stos jest umieszczony w pamięci RAM modułu, a jego organizację przedstawiono na rys. 15.

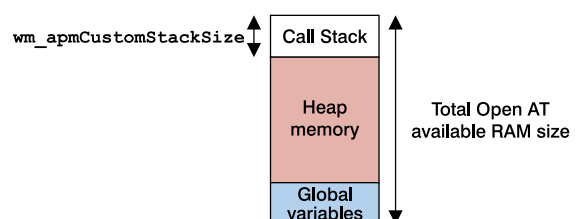
Prawidłowa aplikacja dla środowiska OpenAT musi mieć także zdefiniowaną funkcję `adl_main`, która jest odpowiednikiem zwykłej funkcji `main`. Musimy jednak pamiętać, że nie jest to zwykła funkcja `main` i w przeciwieństwie do niej nie może zawierać pętli nieskończonej. Funkcja ta musi wykonać pewne czynności inicjalizacyjne (np. wywołać funkcje API rejestrujące poszczególne funkcje obsługi zdarzeń), a następnie zakończyć się. Zadania aplikacji będą realizowane przez funkcje obsługi zdarzeń. Argument funkcji `adl_main` określa, z jakiej przyczyny aplikacja została uruchomiona i może być wykorzystywana do wykrycia błędnego działania aplikacji, np. wyzerowania modułu w wyniku zadziałania licznika Watchdog. Umożliwia to podjęcie przez aplikację odpowiednich czynności mających na celu przywrócenie poprzedniego stanu aplikacji i wznowienie jej prawidłowego działania. Argument ten może przyjmować wartości jak w tab. 5.

Lucjan Bryndza SQ7FGB, EP  
lucjan.bryndza@ep.com.pl

Pierwszy odcinek kursu był wyemitowany w EP7/2008.

**Tab. 5. Wartości argumentu funkcji `adl_main`**

Wartość	Opis
ADL_INIT_POWER_ON	Aplikacja została uruchomiona po włączeniu napięcia zasilającego w sposób normalny
ADL_INIT_REBOOT_FROM_EXCEPTION	Aplikacja została wyzerowana, a poprzednio wystąpił wyjątek (sytuacja awaryjna)
ADL_INIT_DOWNLOAD_SUCCESS	Aplikacja została wyzerowana, po zakończeniu procesu instalacji
ADL_INIT_DOWNLOAD_ERROR	Aplikacja została wyzerowana, po wystąpieniu błędu podczas instalacji



**Rys. 15. Organizacja stosu modułu Q2686**