

GPS-owy rejestrator trasy, część 2

AVT-388

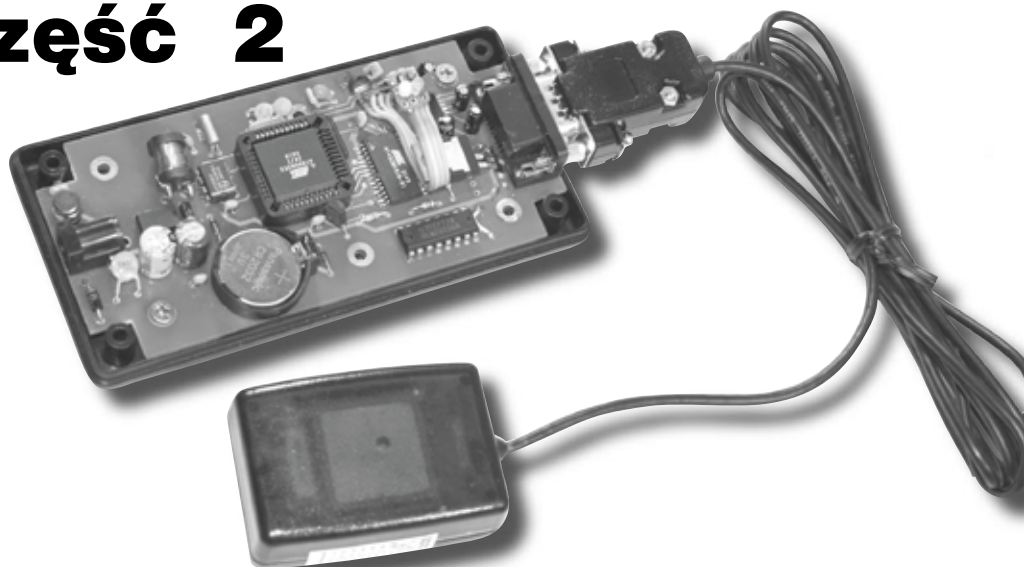
Określenie położenia (pozycjonowanie) i nawigacja są coraz bardziej istotne w wielu dziedzinach życia. Ich zastosowania obejmują przede wszystkim transport morski, lotniczy i kołowy, ale także turystykę i sport. Najwcześniejszy znany system nawigacyjny, rydwan wskazujący kierunek południowy, pochodzi zgodnie z legendą z około 2600 r. p.n.e. z Chin. Od tego czasu powstało wiele znacznie doskonalszych systemów pozycjonujących i nawigacyjnych, chociaż prawdziwie rewolucyjną zmianę w tej dziedzinie zaszły stosunkowo niedawno, z chwilą pojawienia się globalnego systemu nawigacji satelitarnej NAVSTAR GPS.

Rekomendacje: urządzenie o nieocenionych walorach dla użytkowników pojazdów, którym zależy na precyzyjnym monitorowaniu ich trasy.



PODSTAWOWE PARAMETRY

Płytką o wymiarach 124 x 59 mm
Zasilanie +12... +28 V
Konieczny zewnętrzny (dodatkowy) odbiornik GPS
interfejs do odbiornika GPS: port szeregowy
Gniazdo przyłączeniowe GPS: DB9 (żeńskie)
Format danych z GPS: NMEA-0183



Program zawarty w mikrokontrolerze AT89S8252 rejestratora trasy GPS został napisany w języku C i skompilowany z wykorzystaniem wersji demonstracyjnej zintegrowanego środowiska uruchomieniowego RIDE firmy Reasonance. W wersji demonstracyjnej objętość kodu wynikowego jest ograniczona do 4 kB, co przy pewnej dyscyplinie programistycznej, pozwala na tworzenie stosunkowo rozbudowanych aplikacji. Rozmiar programu rejestratora trasy zbliża się do granicy 4 kB i w obecnej wersji wynosi 3814 bajtów, ale wciąż pozostaje jeszcze wiele możliwości optymalizacji programu pod kątem zmniejszenia rozmiaru kodu. Zastąpienie funkcji printf() z biblioteki standardowej uproszczoną autorską funkcją jest jedną z takich możliwości. Zmniejszenie rozmiaru programu umożliwiłoby dodawanie do niego nowych elementów, podnoszących funkcjonalność rejestratora.

Program wykonywany przez mikrokontroler składa się z następujących modułów zawartych w pięciu plikach źródłowych:

- programu głównego,
- funkcji komunikacji rejestratora z komputerem PC,
- funkcji rejestracji danych z odbiornika GPS,
- funkcji służącej do formatowania danych odebranych z odbiornika GPS,

- funkcji służących do komunikacji z pamięcią Data Flash AT45DB081B poprzez SPI,
- funkcji służących do komunikacji z układem zegara-kalendarza PCF8583 poprzez I²C,
- funkcji pomocniczych stosowanych np. do konfigurowania bloków wewnętrznych mikrokontrolera, realizacji opóźnienia czasowego, itp.

Plik źródłowy zawierający program główny mikrokontrolera AT89S8252 przedstawiono na liście **list. 1**.

W programie głównym jest włączony watchdog, który stanowi zabezpieczenie na wypadek zawieszenia się programu rejestratora. Następnie funkcja *Init()* sprawdza czy rejestrator jest połączony z GPS, czy z komputerem, poprzez sprawdzenie poziomu linii In-Car. W zależności od wyniku tego sprawdzenia jest ustawiana odpowiednia prędkość transmisji portu szeregowego UART mikrokontrolera. Kolejny etap pracy programu to sprawdzenie czy urządzenie jest uruchamiane po raz pierwszy. Jeśli tak, ustawiany jest domyślny okres rejestracji wynoszący 20 sekund, a liczba zapisanych w pamięci rekordów jest zerowana. Następnie, w zależności od tego czy urządzenie jest połączone z komputerem, czy z odbiornikiem GPS, program przechodzi do funkcji *Registration()*, od-

Wszystkie listingi do tego artykułu są umieszczone na płycie CD EP5/2005 oraz na stronie <http://www.ep.com.pl>.

Tab. 2. Komendy do komunikacji rejestratora z komputerem

Komenda	Opis/Uwagi	Odpowiedź rejestratora	Opis/Uwagi
?P<CR>	pytanie o ustawiony okres rejestracji trasy pojazdu	<Period><CR> <LF>	okres rejestracji jest równy Period*10 s, po zaokrągleniu do dziesiątek sekund dla 1<=Period<=6 i do pełnych minut dla Period>6
?M<CR>	pytanie o stan zajętości pamięci Serial DataFlash rejestratora (liczba zapisanych rekordów)	<Liczba>,[spacja] <Pojemność> <CR><LF>	np. 12005, 65535 oznacza, że przy wielkości pamięci rejestratora wynoszącej 65535 rekordów, zapisano 12005 rekordów
#P<Period><CR>	ustawienie nowego okresu rejestracji wynoszącego Period*10 sekund, po zaokrągleniu do dziesiątek sekund dla 1<=Period<=6 i do pełnych minut dla Period>6 (np. #P4<CR> ustawia okres 40 sekund, #P8<CR> ustawia okres 1 minuta)	<CR><LF>OK <CR><LF>	
#M<CR>	odczyt całej zawartości pamięci rejestratora	<... dane ...> <CR><LF>OK <CR><LF>	przesyłanie danych z pamięci rejestratora odbywa się pakietami z kontrolą prawidłowości transmisji
#D<CR>	kasowanie pamięci rejestratora	<CR><LF>OK <CR><LF>	dane nie są fizycznie kasowane, a jedynie zerowana jest zmienna przechowująca liczbę zapisanych rekordów

powiedzialnej za rejestrację danych lub do funkcji *Communication()*, odpowiedzialnej za komunikację urządzenia z komputerem PC.

Funkcja *Communication()* służy do odbierania, interpretacji i wykonywania komend przesyłanych z komputera PC przez port szeregowy do rejestratora. W tym trybie pracy rejestratora prędkość transmisji portu szeregowego UART mikrokontrolera wynosi 57600 b/s.

Plik źródłowy zawierający funkcję realizującą komunikację rejestratora z komputerem PC przedstawiono na **list. 2**.

Komunikacja komputera PC z rejestratorem odbywa się za pomocą zbioru komend zestawionych w **tab. 2**. Wszystkie komendy wysyłane z komputera PC i odpowiedzi rejestratora, za wyjątkiem transmisji zarejestrowanych danych nawigacyjnych z pamięci Serial DataFlash rejestratora, są przekazywane w trybie znakowym (tekstowym). Komendy muszą być pisane wielkimi literami. Znak <CR> (odpowiadający naciśnięciu klawisza ENTER) jest znakiem powrotu karetki (o kodzie ASCII 13), a znak <LF> jest znakiem przejścia do nowego wiersza (o kodzie ASCII 10).

Transmisja danych nawigacyjnych z rejestratora odbywa się pakietami z potwierdzeniem prawidłowości odebranych danych, w sposób przedstawiony na **rys. 3**. Długość pojedynczego pakietu wynosi

258 bajtów – 256 bajtów danych (16 rekordów po 16 bajtów) oraz 2 bajty sumy kontrolnej (LSB, MSB). Format danych zawartych w pojedynczym rekordzie zapisanym w pamięci Serial DataFlash zostanie omówiony w dalszej części artykułu, podczas omawiania rejestracji danych z odbiornika GPS.

Suma kontrolna jest wyliczana przez zsumowanie wartości wszystkich bajtów danych zawartych w pakiecie i przekazywana w postaci liczby 16-bitowej (2 bajty po 8 bitów). Po odebraniu prawidłowej sumy kontrolnej komputer odsyła do rejestratora bajt potwierdzenia ACK (o kodzie ASCII 6). W przypadku niezgodności sumy kontrolnej odsyłany jest znak NAK (o kodzie ASCII 21), po czym rejestrator przesyła ponownie błędnie odebrany pakiet.

Do zapisywania danych nawigacyjnych z odbiornika GPS w pamięci szeregowej Serial DataFlash służy funkcja *Registration()*, której plik źródłowy przedstawiono na **list. 3**. W trybie rejestracji prędkość transmisji portu szeregowego UART mikrokontrolera wynosi 4800 b/s i taka powinna być ustawiona prędkość transmisji portu szeregowego odbiornika GPS.

Oprócz rejestracji danych funkcja *Registration()* sprawdza zajętość pamięci. Zapisanie ponad 80% dostępnej pamięci jest sygnalizowane krótkimi mignięciami żółtej diody

LED co 1 sekundę. Dzięki temu operator jest wcześniej informowany, że przy najbliższej okazji powinien odczytać zgromadzone dane i zwolnić pamięć rejestratora. Zapelnienie całej dostępnej pamięci jest sygnalizowane ciągłym świeceniem żółtej diody świecącej LED. Program mikrokontrolera wchodzi wówczas do pętli nieskończonej *while(1)* i rejestrator przerywa zapisywanie danych.

Odbiór wiadomości RMC (*Recommended Minimum Specific GNSS Data*) z odbiornika GPS, sprawdzenie jej poprawności, właściwe sformatowanie i zapisanie danych nawigacyjnych w tablicy *Record[]* jest realizowane przez funkcję *GetGPRMC()*. Funkcja ma jako argument wejściowy adres tablicy *Record[]*, natomiast zwraca wartość bitową *Valid*. Zmienna *Valid* przyjmuje wartość 1, jeśli dane nawigacyjne w wiadomości RMC są prawidłowe, zaś 0 w przeciwnym przypadku.

Odebrane dane GPS, nawet jeśli są prawidłowe mogą być zapisane w pamięci Data Flash tylko wówczas, gdy od poprzedniego zapisu upłynął zadany okres rejestracji. Odliczanie zadanych odcinków czasu jest realizowane przez timer w zewnętrznym zegarze-kalendarzu RTC. Po upływie czasu odpowiadającego okresowi rejestracji, RTC ustawia poziom niski na linii In-RTC. Stwierdzenie niskiego poziomu na tej linii przez program mi-

Pakiet 1																Potwierdzenie od komputera		Pakiet 2	Potwierdzenie od komputera
Rekord																Suma kontrolna			ACK lub NAK
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	LSB	MSB	ACK lub NAK	

Rys. 3. Struktura ramki transmisji danych

krokontrolera powoduje ustawienie zmiennej bitowej *WritePermission*, co oznacza zezwolenie na zapis danych do pamięci DataFlash.

Odliczanie czasu przez zewnętrzny zegar RTC nie jest konieczne i mogłoby być zrealizowane przez wewnętrzny timer mikrokontrolera. W rejestratorze zastosowano jednak zegar RTC, ponieważ ma on wewnętrzną pamięć RAM, którą wykorzystano do przechowywania informacji o liczbie rekordów zapisanych w pamięci DataFlash. Dzięki podtrzymaniu baterijnemu, informacja ta nie jest tracona po wyłączeniu zasilania rejestratora.

Przechowywanie zmiennej zawierającej liczbę zapisanych rekordów w pamięci DataFlash lub EEPROM mikrokontrolera nie jest możliwe ze względu na wymaganą bardzo dużą liczbę aktualizacji tej zmiennej. Jednokrotne zapisanie całej pamięci DataFlash to zapisanie w niej 65535 rekordów i tyle razy zmienia się wartość zmiennej przechowującej liczbę zapisanych rekordów. Zgodnie z danymi producenta, liczba cykli kasowań/zapisu do pamięci EEPROM wynosi 100000, zaś do pamięci Serial DataFlash 50000 na stronę. Stąd wniosek, że w przypadku umieszczenia zmiennej w pamięci Serial DataFlash lub EEPROM mikrokontrolera, już po jedno- lub dwukrotnym zapisaniu całej pamięci danymi GPS, gwarantowana przez producenta liczba cykli kasowań/zapisu zostałaaby przekroczona. Z tego względu zastosowano podtrzymywaną bateryjnie szeregową pamięć RAM zegara – kalendarza RTC. Ponadto zastosowanie RTC zwiększa uniwersalność rejestratora. Możliwe jest na przykład, co sprawdzili autorzy, przyspieszenie pierwszego ustalenia położenia przez tani odbiornik GPS, pozbawiony wewnętrznego podtrzymania baterijnego zegara RTC. Można wówczas wysłać do odbiornika komunikat inicjalizujący, zawierający aktualny czas i datę pobrane z zegara RTC rejestratora. Z kolei podczas normalnej pracy odbiornika GPS jego dane mogą posłużyć do synchronizacji zegara RTC w rejestratorze.

Doświadczenia autorów z rejestratorem i różnymi typami tanich odbiorników GPS wykazały, że odbiorniki takie nie zawsze pracują zgodnie z oczekiwaniami. Niekiedy, zwłaszcza po uruchomieniu w

warunkach słabej widoczności satelitów GPS i/lub w pojeździe poruszającym się z dużą prędkością zdarza się, że odbiornik przez długi czas nie może ustalić swojego położenia. Dzieje się tak nawet, gdy warunki obserwacji satelitów ulegną po pewnym czasie poprawie. W celu przyspieszenia rozpoczęcia rejestracji danych, w programie mikrokontrolera wprowadzono proste zabezpieczenie polegające na zerowaniu odbiornika GPS przez chwilowe wyłączenie i włączenie jego zasilania. W funkcji *Registration()* liczba wiadomości bez prawidłowo określonego położenia jest zliczana i przechowywana w zmiennej *InvalidMsgCount*. Przekroczenie ustalonej wartości progowej *MAX_INVALID_MSGS* powoduje wyłączenie zasilania odbiornika GPS. Brak danych przychodzących z odbiornika przez około 2 sekundy powoduje zerowanie mikrokontrolera przez watchdog'a i ponowne uruchomienie programu oraz włączenie zasilania odbiornika GPS. Wartość progową *MAX_INVALID_MSGS* ustalono na 300, co przy 1-sekundowym okresie wiadomości RMC oznacza wyłączenie zasilania GPS po około 5 minutach braku prawidłowych danych nawigacyjnych.

W funkcji *Registration()* wykonuje się ponadto sprawdzanie prędkości i wykrywanie braku ruchu pojazdu, w celu zwiększenia okresu rejestracji, jeśli pojazd nie zmienia swego położenia. Umożliwia to znaczne wydłużenie czasu rejestracji danych, poprzez zmniejszenie liczby zapisów do pamięci podczas postojów, krótkotrwałych zatrzymań na skrzyżowaniach czy podczas jazdy w korkach. Informację o ruchu lub braku ruchu pojazdu przechowuje zmienna bitowa *Parking*. Każdorazowe stwierdzenie odebrania prawidłowej wiadomości RMC (zmienna *Valid* równa 1) o prędkości ≤ 2 mph (*miles per hour* – mil morskich na godzinę, przy czym 1 mph=1,852 km/h) powoduje zwiększenie o 1 wartości zmiennej pomocniczej *SpeedFlag*. Jeśli natomiast prędkość jest większa od 2 mph, zmienna *SpeedFlag* jest zmniejszana o 1. Zwiększenie wartości zmiennej *SpeedFlag* i osiągnięcie przez nią wartości progowej *SPEED_FLAG_LIMIT* powoduje wykrycie braku ruchu i ustawienie zmiennej bitowej *Parking*. Zmniejszenie wartości zmiennej *SpeedFlag*

do zera powoduje wykrycie ruchu pojazdu i wyzerowanie zmiennej bitowej *Parking*. Wartości parametrów, tj. prędkości granicznej 2 mph i stałej *SPEED_FLAG_LIMIT=10* zostały wybrane doświadczalnie wskutek kompromisu pomiędzy szybkością wykrycia ruchu i braku ruchu pojazdu oraz liczbą fałszywych wykryć tych stanów.

Zapis danych GPS do pamięci jest realizowany wówczas, gdy odebrana wiadomość RMC jest prawidłowa (ustawiona zmienna bitowa *Valid*) i jeśli jest zezwolenie na zapis (ustawiona zmienna bitowa *WritePermission*). Wówczas program sprawdza czy od ostatniego zapisu rejestrator był wyłączony. Informację taką można uzyskać przez sprawdzenie flagi POF (*Power Off Flag*) mikrokontrolera. Jeśli rejestrator był wyłączony, jest to uwzględniane w zapisywanym rekordzie, przez ustawienie niewykorzystywanego do innych celów najbardziej znaczącego bitu w bajcie nr 15 tablicy *Record[]*. Stosowany sposób formatowania danych GPS w tablicy *Record[]* sprawia, że w niektórych bajtach tej tablicy nie wszystkie bity są wykorzystywane. Daje to możliwość modyfikacji programu i zapisywania w tych bitach dodatkowych danych, np. informacji o włączeniu/wyłączeniu jakiegoś urządzenia, itp.

Po sprawdzeniu POF, uzyskany z odbiornika GPS rekord danych jest zapisywany w pamięci DataFlash, zmienna przechowująca liczbę zapisanych rekordów jest zwiększana o 1, a timer w zegarze RTC jest ustawiany tak, aby odliczał czas do następnego zapisu danych. Przy ustawianiu alarmu timera wykorzystywana jest ustalona uprzednio wartość zmiennej bitowej *Parking*. Jeśli zmienna *Parking* jest równa 1 (stwierdzono brak ruchu pojazdu) okres rejestracji jest wydłużany do wartości odpowiadającej stałej *SOMETIMES* (chyba, że ustawiony przez operatora okres rejestracji jest dłuższy niż ten, który odpowiada stałej *SOMETIMES*).

W każdym obiegu funkcji *Registration()* wywoływana jest funkcja *GetGPRMC()*, przedstawiona na listingu list. 4, odbierająca wiadomość z odbiornika GPS i formatująca ją w zadany sposób. Funkcja oczekuje na wiadomości przychodzące przez port szeregowy i poszukuje wiadomości rozpoczynającej się znakami

Tab. 3. Format danych w wiadomości RMC

Numer pola	Nazwa	Przykład	Format/Opis
1	ID wiadomości	\$GPRMC	nagłówek wiadomości RMC
2	Czas UTC	031312.876	hhmmss.sss – godziny, minuty, sekundy, ułamkowe części sekundy
3	Status	A	A – dane poprawne, V – dane niepoprawne
4	Szerokość geograficzna	2446.5270	ddmm.mmmm – stopnie, minuty, ułamkowe części minuty
5	Półkula N/S	N	N – północna, S – południowa
6	Długość geograficzna	12100.1485	dddmm.mmmm – stopnie, minuty, ułamkowe części minuty
7	Półkula E/W	E	E – wschodnia, W – zachodnia
8	Prędkość podróżna	000.0	prędkość względem Ziemi w milach na godzinę (mph)
9	Kurs rzeczywisty	000.0	kurs względem Ziemi w stopniach
10	Data	210802	ddmmyy – dzień, miesiąc, rok
11	Deklinacja magnetyczna	003.3	lokalna odchyłka kierunku północy magnetycznej od rzeczywistej
12	Kierunek deklinacji	W	E – wschodni, W – zachodni
13	Suma kontrolna	*76	XOR wszystkich bajtów pomiędzy '\$' a '*'

„\$GPRMC”. Następnie poszczególne bajty wiadomości RMC są zapisywane tak, aby zmieściły się w 16 bajtach tablicy *Record[]*.

Wiadomość RMC jest jedną z najczęściej wykorzystywanych wiadomości zdefiniowanych w standardzie NMEA. Wiadomość ta zawiera dane, które są wystarczające w większości zastosowań, tj. informację o położeniu, prędkości i kursie obiektu oraz czasie UTC (*Universal Time Coordinated*) i dacie. Z tego względu ta właśnie wiadomość została wybrana w rejestratorze do wydzielenia z niej danych nawigacyjnych. Format wiadomości RMC przedstawiono w **tab. 3**.

Wszystkie wiadomości NMEA rozpoczynają się znakiem '\$', a kończą parą znaków formatujących <CR><LF>. Pole sumy kontrolnej jest opcjonalne i w wielu odbiornikach GPS można samodzielnie określić czy będzie wysyłane, czy nie. Pola wiadomości są oddzielone przecinkami, natomiast pole sumy kontrolnej jest oddzielone od reszty wiadomości znakiem '*'.

Rejestrator trasy odbiera z odpowiednio skonfigurowanego od-

biornika GPS wiadomości RMC, wydziela z nich informację o położeniu, prędkości, czasie i dacie, a następnie zapisuje odpowiednio przygotowany rekord *Record[]* do pamięci Serial DataFlash. Zasadę formatowania danych otrzymanych z odbiornika GPS najłatwiej wyjaśnić na przykładzie.

Załóżmy, że z odbiornika GPS otrzymano następującą wiadomość RMC:

```
$GPRMC,031312.876,A,2446.5270,N,12100.1485,E,000.0,000.0,210802,003.3,W*76<CR><LF>
```

W wyniku działania funkcji *GetGPRMC()*, dane z wybranych pól powyższej wiadomości zostają wydzielone i odpowiednio zapisane w tablicy *Record[]*. Sposób formatowania danych GPS i zapisywania ich do tablicy *Record[]* wyjaśniono w **tab. 4**.

W przypadku, gdy rejestrowane położenie znajduje się na półkuli południowej, informacja o tym jest zapisywana przez ustawienie wartości 1 na najbardziej znaczącym bicie pierwszego bajtu przechowywanej szerokość geograficzną *Record[3]*.

Podobnie, informację o położeniu na półkuli zachodniej zapisuje się przez ustawienie 1 na najbardziej znaczącym bicie pierwszego bajtu przechowywanej długość geograficzną *Record[7]*.

Przykładowo, szerokość geograficzna na półkuli północnej 5234,2345 N zostanie zarejestrowana w bajtach od 3. do 6. tablicy *Record[]*, jako 4 bajty o wartościach: 52,34,23,45, natomiast ta sama szerokość geograficzna, ale dla półkuli południowej 5234,2345 S, zostanie zarejestrowana w postaci 4 bajtów o wartościach: 180,34,23,45. Bajt nr 3 ma w tym drugim przypadku ustawiony najbardziej znaczący bit, co dziesiętnie oznacza dodanie do niego liczby 128 (52+128=180).

Dodatkowo w rekordach zapisywana jest również informacja o momentach włączenia zasilania rejestratora. Dzięki temu można odróżnić przerwy w rejestracji spowodowane brakiem widoczności satelitów przez odbiornik GPS od przerw w pracy samego rejestratora. Daje to możliwość określenia czasu postoju i przerw w pracy pojazdu. Informacja ta jest zakodowana przez ustawienie najbardziej znaczącego bitu ostatniego bajtu rekordu danych.

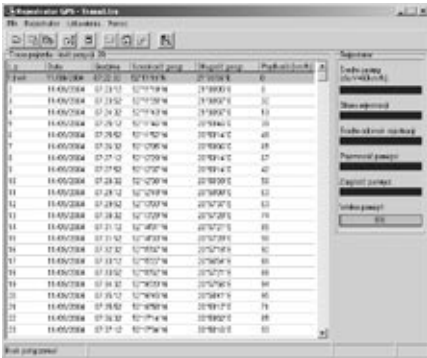
Opisany sposób formatowania danych nie jest najprostszym, ani najbardziej „przejrzystym” z możliwych, ale umożliwia umieszczenie wszystkich istotnych danych z odbiornika GPS w rekordzie o rozmiarze jedynie 16 bajtów, co pozwala na oszczędne wykorzystanie dostępnej pamięci rejestratora.

Program Rejestrator GPS do komunikacji komputera z rejestratorem

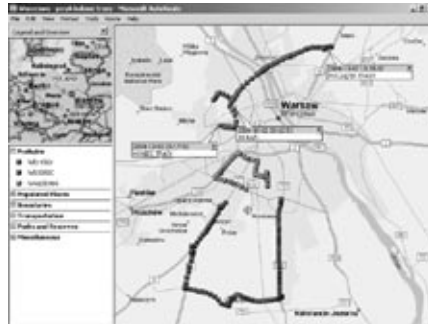
Odczyt danych zarejestrowanych w rejestratorze oraz jego konfigurowanie jest realizowane za pomocą

Tab. 4. Struktura rekordu danych (format danych w tablicy *Record[]*)

Numer bajtu	Przykład	Format	Znaczenie
0	3	godziny UTC	Czas UTC: 3:13:12
1	13	minuty UTC	
2	12	sekundy UTC	
3	24	stopnie	Szerokość geograficzna: 24°46.5270' N (24°46'31.62" N)
4	46	minuty	
5	52	ułamkowe części minuty	
6	70	ułamkowe części minuty	
7	1	stopnie	Długość geograficzna: 121°00.1485' E (121°00'08.91" E)
8	21	stopnie	
9	00	minuty	
10	14	ułamkowe części minuty	
11	85	ułamkowe części minuty	Prędkość: 0 mph
12	0	prędkość podróżna	
13	21	dzień	Data: 21/08/2002
14	08	miesiąc	
15	02	rok	



Rys. 4. Główne okno programu Rejestrator GPS



Rys. 5. Przykładowe trasy pojazdów wyświetlone na mapie programu Microsoft AutoRoute Express

przygotowanego do tego celu programu Rejestrator GPS, pracującego w systemie operacyjnym MS Windows 95/98/2000/XP. Główne okno programu pokazano na rys. 4.

Przed uruchomieniem programu należy podłączyć rejestrator do wolnego portu szeregowego

komputera. Następnie należy ustawić parametry rejestracji w oknie Ustawienia, tzn. wskazać port szeregowy, do którego podłączony jest rejestrator, ustawić okres rejestrowania pozycji pojazdu (np. co 20 sekund) oraz określić przesunięcie względem czasu UTC. Przesunięcie

czasowe względem UTC na terytorium Polski wynosi +1 godz. dla czasu zimowego i +2 godz. dla czasu letniego. Po tych czynnościach należy nawiązać połączenie z rejestratorem, wybierając z menu *Rejestrator>Połącz* i wczytać zarejestrowane dane wybierając z menu *Rejestrator>Wczytaj dane*. Dane te mogą być następnie zapisane na dysku twardym komputera, bądź wyeksportowane do pliku tekstowego, o postaci nadającej się do importowania i wyświetlenia na mapie w programie Microsoft AutoRoute Express. Przykładowe trasy pojazdów zostały pokazane na rys. 5.

Piotr Kaniewski
pkaniewski@wat.edu.pl
Piotr Komur
pkomur@wat.edu.pl

W ofercie handlowej AVT jest dostępna:
 - [AVT-388A] płytka drukowana

ACSELEKTRONIK
 Szydłowiec 26-500 ul.Kolejowa 11 e-mail: acs@acs.ats.pl
 Tel/fax 0486176000, tel 0600332061

OSCYSKOPY CYFROWE
www.acs.ats.pl

PROGRAMATORY PAMIĘCI

Uniwersalne programatory
VI-LAB, ERICA, Ps32

Vi-LAB wirtualne laboratorium

- ✓ programator 1400 układów, ZIF 48Pin 0,3"-0,6"
- ✓ tester TTL, CMOS, PLD
- ✓ emulator czasu rzeczywistego (8MB-16Bit 27xxx, 62xxx, 24Cxxx, 93Cxxx, 25/95xxx)
- ✓ komunikacja port drukarkowy ECP
- ✓ samodzielne dodawanie nowych algorytmów język ISPA

Profesjonalne programatory
XELTEK

SuperPRO 8000, 2000, 680, V, LX, 280, Z

- ✓ obsługa ponad 8000 układów
- ✓ modele z LCD pracujące bez komputera
- ✓ programatory wielokrotne o wydajności 1000 układów/h
- ✓ praca z układami większymi niż 100końcówek

ADS 220 2x60MHz 200MSPS

- ✓ pasmo 2x60MHz
- ✓ rozdzielczość 8bitów/kanał
- ✓ próbkowanie 2x200MSPS, 3.3 x pasmo
- ✓ zakres 5mV-DIV (1:1)
- ✓ zewnętrzny kanał wyzwalania EXT
- ✓ analiza FFT
- ✓ interpolacja przebiegów sin(x)/x
- ✓ autokalibracja 24bitowa
- ✓ wyjście kompensacji sond pomiarowych
- ✓ impedancja wejściowa 1M, pojemność 20pF
- ✓ połączenie z komputerem IEEE1284-ECP
- ✓ pełny resampling przebiegu (możliwość zmiany czasu i wzmocnienia na zatrzymanym przebiegu)
- ✓ automatyczne pomiary: częstotliwość, okres, peak to peak, RMS, wartość średnia
- ✓ symulacja wirtualnej płyty czołowej oscyloskopu
- ✓ oparty na układach AnalogDevices, Burr-Brown, Xilinx
- ✓ w połączeniu z komputerem notebook - idealne stanowisko pomiarowe

MONTAŻ SMT

- na paście
- na kleju

PROGRAMOWANIE KONSTRUOWANIE

- sterowników na bazie mikrokontrolerów 8-bitowych, 16-bitowych, 32-bitowych

PROJEKTOWANIE

- układów elektronicznych
- obwodów drukowanych

MCD electronics

PONADTO OFERUJEMY:

- montaż mieszany; przewlekany, SMT
- lutowanie na fali lutowniczej SOLTEC MIDI z podwójną falą typu SMART WAVE

MCD Electronics
 34-300 Żywiec ul. Lelewela 26
 tel/fax: 33/861 60 35
 e-mail: smt@mcd.com.pl
<http://www.mcd.com.pl>