

# Tracker GPS (2)

## Zabezpieczenie auta przed kradzieżą

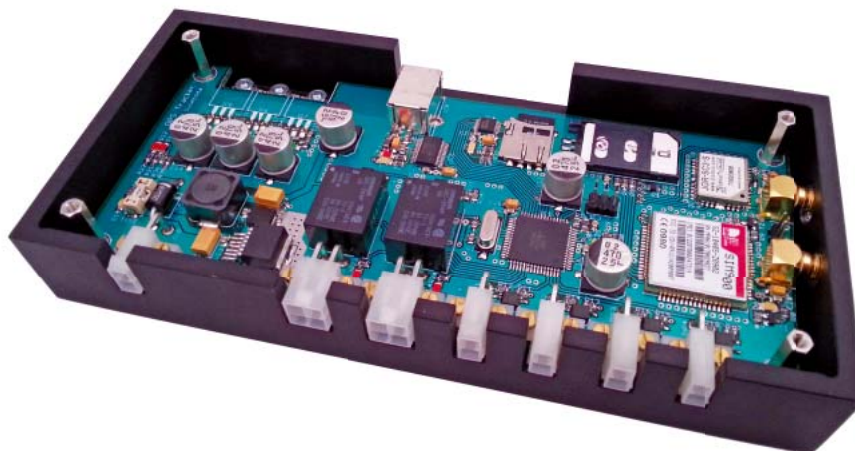


*W Elektronice Praktycznej nr 12/2013 zaprezentowaliśmy projekt trackera przeznaczanego do zabezpieczenia samochodu przed kradzieżą. Wiadomo, że istnieją zaawansowane, skuteczne i zarazem drogie systemy alarmowe. Jednak istotnym jest również, by koszt takiego systemu nie był porównywalny z wartością samochodu, abstrahując od rozważań – poniżej jakiej wartości samochodu warto w ogóle martwić się o zabezpieczenia. Oczywiście jest też, że specjaliście mogą powstrzymać jedynie nieszablone rozwiązania.*

### Program mikrokontrolera

Program mikrokontrolera zajmuje w aktualnej wersji (v1.28) 27,8 kB, co stanowi jedynie około 21% pamięci Flash dostępnej w ATmega128L. Istnieje więc potencjał dla rozbudowy funkcjonalności urządzenia.

Po włączeniu trackera rozpoczyna się procedura uruchamiania poszczególnych podsystemów. Po wysłaniu przez port USB komunikatu powitalnego jest wykonywany test diod LED. Następnie sprawdzane są wartości napięć na wszystkich szynach zasilających układ. Jeżeli wartości te nie mieszczą się w przedziałach zdefiniowanych jako normalne, proces uruchamiania zostaje zawieszony (sygnalizowane jest to dwoma błyskami LED D11 i 12). Wówczas moduły GPS, GSM i karta SD pozostają wyłączone, a mikrokontroler jedynie sprawdza cyklicznie wartości napięć. Jeżeli napięcia mieszczą się w normie, to procedura uruchamiania jest kontynuowana. Komunikat „\$PSON (Power Supply ON)” wysłany przez interfejs USB potwierdza ten status. Następnie mikrokontroler sprawdza czy karta microSD znajduje się w slocie. Jeżeli tak, wykonywana jest procedura inicjalizacji, jeżeli nie jest realizowany następny krok procedury uruchamiania trackera. Poszczególne etapy inicjalizacji karty pamięci potwierdzane są odpowiednimi komunikatami na interfejsie USB, a bezbłędne jej zakończenie sygnalizuje zaświecenie diody D9. Program prawidłowo rozpoznaje i inicjalizuje karty o pojemności do 2 GB. Karty HC (High Capacity) mogą nie być prawidłowo roz-



poznawane i wywoływać błędy w działaniu procedur mikrokontrolera.

W następnych krokach uruchamiane są: odbiornik GPS, a później modem GSM (sygnały podane na wejścia PEN i PWR). Obydwa kroki potwierdzone są odpowiednimi komunikatami. Przy czym uruchomienie modemu GSM jest nieco bardziej skomplikowane i zostało zrealizowane w kilku etapach.

W następnym kroku uruchamiane są: odbiornik GPS, a później modem GSM (sygnały podane na wejścia PEN i PWR). Obydwa kroki potwierdzone są odpowiednimi komunikatami. Przy czym uruchomienie modemu GSM jest nieco bardziej skomplikowane i zostało zrealizowane w kilku etapach. Oprócz podania impulsu na wejście PWR konieczne jest wysłanie kilku komend AT. Najpierw zostaje wyłączona funkcja echo, następnie, jeżeli wymaga tego karta SIM, jest wprowadzany numer PIN (przechowywany w pamięci trackera – konfigurację trackera omówiono dalej). W tym kroku możliwe jest również wykrycie braku lub błędu karty SIM. W obu wypadkach tracker przełącza się w tryb pracy z ograniczoną funkcjonalnością – bez możliwości komunikowania się z użytkownikiem przy użyciu wiadomości SMS (modem zostaje wyłączony). Jeżeli numer PIN zostanie przyjęty lub nie jest wymagany (zależnie od ustawień karty SIM), kolejnym krokiem konfiguracji modemu jest przełączenie go z trybu PDU na tekstowy format wiadomości SMS. Dalej jest ustawiany najbardziej rozbudowany format komunikatu o statusie logowania do sieci (z kodami LAC i CID). W ostatnim kroku kasowane są wszystkie wiadomości SMS zapisane w pamięci karty SIM. Ten krok kończy także procedurę uruchamiania trackera. Zakończenie sekwencji uruchamiania podsystemów potwierdzone jest komunikatem „\$PSSC (Power Supply Sequence Completed)”. Następnie rozpoczyna się wykonywanie rozkazów pętli głównej programu.

Jednym z podstawowych, a w zasadzie najważniejszym, zadaniem realizowanym w pętli głównej jest sprawdzanie buforów interfejsów USART0 i USART1. Odebrane ciągi

**Dodatkowe materiały na CD/FTP:**  
ftp://ep.com.pl, user: 63241, pass: 741obq51  
• artykuł z EP12/2012

znaków są na bieżąco interpretowane i na tej podstawie wyznaczane są kolejne zadania, które mikrokontroler realizuje w sekwencji wykonywanej co minutę.

W pierwszej fazie mikrokontroler identyfikuje wielkości zawarte w komunikacie RMC wysłanym przez odbiornik GPS. Jeżeli komunikat ma potwierdzony status ważności (znacznik ma wartość „A”), odczytane dane o czasie kopiowane są do rejestrów programowego zegara czasu rzeczywistego. Po pierwszym zsynchronizowaniu czasu, na karcie microSD utworzony zostaje plik, w którym następnie zapisywany będzie ślad przebytej trasy. Utworzenie pliku potwierdza komunikat „\$SDWE (Data Write Enabled)”. Wcześniejsze zsynchronizowanie czasu jest o tyle ważne, że nazwy plików tworzone są według schematu: MMDDhhmm.TXT (gdzie MM – miesiąc, DD – dzień, hh – godzina, mm – minuta), co pozwala zachować ich częściową unikalność, a także ułatwia ich późniejszą identyfikację. Po utworzeniu pliku zapisywane są w nim kompletne komunikaty RMC (oznaczenia jak powyżej w opisie odbiornika) w formacie „\$GPRMC,hhmmss.sss,a,ddmm.mmmm,L,dddmm.mmm-m,L,s,ss,a,a,DDMMYY”.

Wpisy na kartę microSD są dokonywane podczas postoju w interwałach określonych przez użytkownika i mogą wynosić od 1 do 99 minut, a w czasie jazdy co minutę. Przy czym przejście w „tryb parkingowy” odbywa się, jeżeli przez pięć minut prędkość jest równa 0 km/h, a „wybudzenie” z trybu parkingowego następuje w momencie, gdy prędkość pojazdu przekroczy limit prędkości SL0. Tymczasem do portu USB trafiają wszystkie komunikaty z odbiornika GPS, zatem dane wysyłane

Tabela 9. Wartości progowe napięć na poszczególnych szynach zasilających

Szyna zasilająca	Napięcie				
	minimalne		nominalne	maksymalne	
	dopuszczalne	normalne		normalne	dopuszczalne
[V]					
VCC	3,2	3,3	3,6	3,9	4,1
VCCAT	3,3	3,4	3,6	3,9	4,1
VCCSD	2,9	3,1	3,3	3,5	3,7
VREL	5,0	5,4	6,0	6,6	7,0
VACCU	9,0	11,0	12,0	15	17

są co 10 sekund. Użytkownik może uzyskać informację o aktualnym położeniu i prędkości samochodu komunikatem SMS, wysyłając na numer karty SIM wiadomość o treści „POSITION”. Odpowiedź zawiera informację o aktualnej prędkości i położeniu: „GPS, SPD=S.Skm/h, LAT=DDd MM.MMMMm L, LON=DDd MM.MMMMm L eom”. Jeżeli zapytanie nadejdzie, gdy dane z modułu GPS nie mają potwierdzonego statusu ważności, tracker odsyła komunikat zwrotny o treści: „No Valid GPS Data eom”.

W tym miejscu trzeba wyjaśnić, że na kartę pamięci i do USB trafiają ramki danych w postaci wysyłanej przez odbiornik GPS. Tam prędkość jest wyrażona w milach na godzinę. Natomiast do wszelkich operacji wykonywanych przez algorytm mikrokontrolera (w tym w odpowiedzi SMS), aktualna prędkość jest przeliczana na kilometry na godzinę (współczynnik 1,852).

W buforze interfejsu USART0 mikrokontroler gromadzi znaki wysyłane przez modem GSM. Najczęściej są to odpowiedzi modemu na zapytania wydane przez nadrzędne procedury i są one „oczekiwane” w tym sensie, że procedura która wysłała zapytanie może zakończyć się dopiero po odebraniu odpowiedzi (ewentualnie komunikatu o błędzie). Jako „nieoczekiwane” należy określić komunikaty +CMTI informujące o nadejściu nowej wiadomości SMS. Jednak odebranie tego komunikatu inicjuje tą samą, co komunikaty oczekiwane, procedurę odczytującą i interpretującą.

Zapytaniem, które cyklicznie kierowane są do modemu, w ramach sekwencji programu głównego są: +CREG i +CSQ. Na podstawie odpowiedzi, procedura odczytująca dane wysłane

z modemu, formułuje komunikat \$ATNR, który może wystąpić w sześciu wariantach:

\$ATNR,0,SQ (Not Registered)

\$ATNR,1,xLAC,xCID,SQ (Registered – Home Network)

\$ATNR,2,SQ (Network Search)

\$ATNR,3,SQ (Registration Denied)

\$ATNR,4,SQ (Unknown)

\$ATNR,5,SQ (Registered – Roaming)

Komunikat ten, raz na minutę wysyłany jest przez port USB. Te same dane wykorzystywane są do dokonania wpisu na kartę pamięci gdy dane GPS nie mają statusu ważności. Wówczas zamiast komunikatu RMC zapisywane są heksadecymalne kody LAC, CID i dziesiętnie siła sygnału przekaźnika BTS „\$GPRMC,hhmmss.sss,v,xCID,xLAC,SQ”. Użytkownik wysyłając wiadomość SMS o treści „BTS”, może na bieżąco uzyskać informację o BTS-ie do którego modem jest aktualnie zalogowany.

Raz na minutę sprawdzane są wartości napięć wszystkich szyn zasilających trackera. Bieżące wartości wysyłane są na port USB w komunikacie „\$VADC,x.v.vv.vv.v.vv.v.vv.vv.vv.vv.vv.vv (VCCUC,VCCAT,VCCSD,VREL,VACCU)”, w którym x przyjmuje wartość 0, gdy wszystkie wartości mieszczą się w normalnych dla siebie przedziałach lub wartość 1, gdy chociaż jedna z nich jest przekroczone. Bieżąca informacja o napięciach jest dostępna dla użytkownika po wysłaniu SMS-a z komunikatem o treści „Voltage”. W odpowiedzi dostaje on wiadomość o treści „VCCUC=v.vvV, VCCAT=v.vvV, VCCSD=v.vvV, VREL=v.vvV, VACCU=v.vvV eom”, jeśli wszystkie wartości mieszczą się w przedziałach normalnych to z prefiksem „volt\_OK”. Wartości progowe napięć

na poszczególnych szynach zasilających umieszczono w tabeli 9.

Jeżeli wystąpi jakaś anomalia, to w zależności od tego, jak dalece wartości zmierzone wykraczają poza przedziały napięć będące w normie, algorytm decyduje o podjęciu odpowiednich działań. W przypadku, gdy na jednej, dwu, trzech lub czterech szynach napięcia przekraczają wartości normalne, ale mieszczą się w dopuszczalnych granicach, funkcjonalność urządzenia nie zostaje zakłócona, a do użytkownika wysyłana jest wiadomość SMS z komunikatem, że wartości normalne napięć zostały przekroczone (prefiks „volt\_OVR”) wraz z podaniem ich liczbowych wartości. Natomiast przekroczenie, na choćby jednej z szyn, wartości dopuszczalnej lub na wszystkich szynach zasilających przekroczenie przedziałów wartości normalnych, skutkuje wyłączeniem większości bloków funkcjonalnych trackera (modemu GSM, odbiornika GPS oraz karty microSD). Innymi słowy tracker przechodzi w tryb pracy z ograniczonym poborem mocy, ten sam, co na początku procedury uruchamiania (charakterystykę poboru mocy przedstawiono w rozdziale opisującym doświadczenia z eksploatacji). Fakt ten sygnalizowany jest komunikatem „\$PSLD (Power Supply Limit)” wysłanym przez port USB oraz wiadomością SMS z komunikatem zawierającym aktualne wartości napięć oraz adnotację o wyłączeniu urządzenia (prefiks „volt\_OVR Power Off”). Jeżeli wiadomość SMS nie może być wysłana, np. gdy modem nie jest zalogowany do sieci domowej, procedura wyłączenia jest wstrzymywana do momentu, gdy uda się wysłać powiadomienie do użytkownika. W trybie ograniczonego poboru mocy mikrokontroler co pewien

Ponad 3 500 dostawców



element14

Polecenie	Odpowiedź/Potwierdzenie	Komunikat USB	Opis
BTS	BTS: CID=xxxx, LAC=yyyy, SQ=zz eom		Status logowania do sieci GSM
CLR	OC CLEARED eom	\$OC3C (OC3 Input Sequence cleared)	Kasowanie alarmu
DISable	DISABLE eom	\$R1ON (REL1 On - Disable)	Załączenie przekaźnika 1
ENable	ENABLE eom	\$R1OF (REL1 Off - Enable)	Wyłączenie przekaźnika 1
STOp	STOP eom	\$R2ON (REL2 On - Disable)	Załączenie przekaźnika 2
DRive	DRIVE eom	\$R2OF (REL2 Off - Enable)	Wyłączenie przekaźnika 2
POSition	GPS, SPD=S.Skm/h, LAT=DDd MM.MMMMm L, LON=DDDd MM.MMMMm L eom		Aktualna pozycja i prędkość
VOLTage	volt_OK VCCUC=v.vvV, VCCAT=v.vvV, VCCSD=v.vvV, VREL=v.vvV, VACCU=v.vvV eom volt_OVR ... eom volt_OVR Power Off ... eom		Napięcia szyn zasilających

czas sprawdza napięcia, a jego powtórne uruchomienie jest możliwe dopiero wówczas, gdy wszystkie zmierzone wartości mieszczą się w przedziałach normalnych.

W cyklu minutowym sprawdzane są także stany wejść OC1–4, przy czym w aktualnej wersji algorytmu wejścia OC1–3 współpracują ze sobą, a OC4 funkcjonuje samodzielnie. Jeżeli użytkownik uaktywni kontrolę sekwencji wejść, odpowiednia procedura zaczyna kontrolować kolejność pojawiania się na nich stanów wysokich. Prawidłowa kolejność w sekwencji oczywiście powinna być taka: OC1, OC2 i OC3. Wykrycie prawidłowej sekwencji potwierdzone jest komunikatem wysłanym przez USB „\$OC3E (OC3 Init Sequence OK)”. Sekwencja może trwać dowolnie długo i może zostać przerwana w dowolnym momencie. Warunkiem koniecznym nieaktywowania alarmu jest aby stany wysokie, które już się pojawiły na wejściach o niższym numerze, nie zniknęły przed pojawieniem się kolejnych. Niedopełnienie któregośkolwiek warunku skutkuje aktywowaniem alarmu, tj. wysłaniem wiadomości SMS o treści „OC3 Init Sequence Activated eom”, a na port USB trafia komunikat „\$OC3A (OC3 Init Sequence Activated)”.

Wejście OC4 monitorowane jest niezależnie od pozostałych. Pojawienie się na nim stanu wysokiego od razu aktywuje alarm. Nie można więc mówić, jak w przypadku wejść OC1...3, o prawidłowej sekwencji. Nie ma też komunikatu informującego o prawidłowym, tzn niewywołującym alarmu, stanie tego wejścia. Aktywowanie alarmu – czyli pojawienie się stanu wysokiego – powoduje, że do użytkownika zostaje wysłana wiadomość „OC4 Input Activated eom” oraz „\$OC4A (OC4 Input Activated)” przez port USB.

Kolejne sekwencje stanów pojawiające się na wejściach OC, zarówno prawidłowe jak i nieprawidłowe oraz zmiana stanu wysokiego na niski na OC4, nie powodują ani ponownego wyzwolenia alarmu ani skasowania go. Stan alarmowy pozostaje aktywny aż do momentu skasowania go przez użytkownika wiadomością SMS o treści „CLR”. Wykasowanie alarmu potwierdzone jest wiadomością SMS o treści „OC CLEARED eom” i komunikatem „\$OC3C

Tabela 11. Komunikaty wysyłane przez interfejs USB

Zarządzanie energią	
\$PSON	(Power Supply ON)
\$PSLD	(Power Supply Limit)
\$PGPO	(GPS Power ON)
\$PATO	(GSM Power ON)
\$PATD	(GSM Power OFF)
\$PATE	(GSM Power OFF Error)
\$PSSC	(Power Supply Sequence Completed)
Komunikacja z modemem	
\$ATAC	(AT OK)
\$ATAE	(AT Error)
\$ATEC	(AT Echo OFF)
\$ATEE	(AT Echo Error)
\$ATPC	(PIN Correct)
\$ATPE	(PIN Check Error)
\$ATPI	(PIN Input Error)
\$ATCR	(AT Call Ready)
\$ATFC	(AT Text Mode ON)
\$ATFE	(AT Text Mode Error)
\$ATMC	(Network Reg Mode Correct)
\$ATME	(Network Reg Mode Error)
\$ATDC	(SMS Messages Deleted)
\$ATDE	(SMS Delete Error)
\$ATNR,0,ZZ	(Not Registered)
1,XXXX,YYYY,ZZ	(Registered - Home Network)
2,ZZ	(Network Search)
3,ZZ	(Registration Denied)
4,ZZ	(Unknown)
5,ZZ	(Registered - Roaming)
6,33	(SIM Error - Offline Mode)
\$SMS0	(SMS Inf @ NumX Sented)
\$SMSE	(SMS Inf @ NumX Error)
\$SMSC	(SMS Inf @ NumX Canceled)
\$SMSD	(SMS Message Received & Deleted)
\$SMSI	(SMS Ignored)
\$SMSR	(SMS Receive Error)
\$SMSU	(SMS Unauthorised Sender)
Komunikaty procedury obsługi wejść OC	
\$OC3E	(OC3 Init Sequence OK)
\$OC3A	(OC3 Init Sequence Activated)
\$OC3C	(OC3 Input Sequence Cleared)
\$OC4A	(OC4 Input Activated)
Komunikaty procedur obsługi wyjść przekaźnikowych	
\$RXON	(RELX On - Disable)
\$RXOF	(RELX Off - Enable)
Obsługa karty microSD	
\$SDCI	(Card Inserted)
\$SDIC	(Card Initialization Correct)
\$SDIE	(Card Initialization Error)
\$SDDC	(File System Correct)
\$SDDE	(File System Error)
\$SDWE	(Data Write Enabled)
\$SDCR	(Card Removed)
\$SDST,0	(No Card)
2	(Card In Slot)
4	(Waiting for Time Sync)
5	(Card Write Enable)
6	(Card Write Disable)
7	(Card Communication Error)
8	(Card Filesystem Error)

Tabela 1. c.d.

Monitoring napięć szyn zasilających	
\$VADC,X,V.VV,V.VV,V.VV,V.VV,V.VV	(VCCUC,VCCAT,VCCSD,VREL,VACCU)
Zezwolenie na odbiór danych z portu USB	
\$PCCE	
Konfiguracja trackera (zapis/odczyt danych z pamięci)	
\$MSLN,Y,XXX	(Speed Limit X Saved)
\$MSLV,Y,XXX	(Speed Limit Mem X [km/h])
\$MPIN,Y,XXX	(Speed Limit Period X Saved [min])
\$MPIV,Y,XXX	(Speed Limit Period Mem X [min])
\$MPIN,2,XXX	(Voltage Limit Period Saved [min])
\$MPIV,2,XXX	(Voltage Limit Period [min])
\$MPIN,3,XXX	(OC Input Period Saved [min])
\$MPIV,3,XXX	(OC Input Period [min])
\$MPIN,9,XXX	(SD Write Period Saved [min])
\$MPIV,9,XXX	(SD Write Period [min])
\$MSPN,XXXX	(SIM Card PIN Saved)
\$MSPV,XXXX	(SIM Card PIN)
\$MCNN,Y,XXXXXXXXXXXX	(Phone Number Y Saved)
\$MCNV,Y,XXXXXXXXXXXX	(Phone Number Mem Y)
\$MMON,0	(OC Inactive)
1	(OC1-3)
2	(OC4)
3	(OC1-3 + OC4)
4	(SLO @OC1-3)
6	(SLO @OC1-3 + OC4)
8	(SLO @OC4)
9	(SLO @OC4 + OC1-3)
\$MMOV,X	(OC Input Mode)
\$MSMN,0	(No Memory Selected)
1	(Mem: 0)
2	(Mem: 1)
3	(Mem: 0,1)
\$MSMV,X	(Selected Memory)
\$MSUC	(Settings Updated)

(OC3 Input Sequence Cleared)” wysłanym na port USB.

Użytkownik może uaktywnić kontrolę sekwencji wejść OC1...3 i OC4, by pracowały pojedynczo lub równolegle, niezależnie od siebie. Przewidziano także tryb pracy, w którym tracker – oprócz powiadomienia o wystąpieniu stanu alarmowego – wysyła dodatkową wiadomość z informacją o aktualnym położeniu, jeśli prędkość samochodu (według wskazań GPS) przekroczy limit SLO (prefiks SLO\_OVR).

Limity prędkości to kolejne wartości, które sprawdzane są przez algorytm programu w cyklu minutowym. W trackerze można ustawić dwa limity prędkości: wspomniany już SLO oraz SL1. Limit SLO służy do określenia wartości prędkości, powyżej której algorytm rozpoznaje, że pojazd przemieszcza się. Konieczność wprowadzenia limitu, o wartości różnej od zera i definiowanej przez użytkownika, wynika stąd, że dość często wyznaczone kolejno pozycje GPS różnią się nieznacznie od siebie, a taka zmiana interpretowana jest przez algorytm modułu jako przemieszczanie się z niezerową

prędkością. Pojedyncze odczyty o niezerowej prędkości zakłócałyby pracę trackera zmieniając w trybie parkingowym częstotliwość zapisów śladu na kartę microSD oraz ingerując w procedurę wykrywania stanów alarmowych wejść OC. SL1 to limit prędkości maksymalnej. Jeżeli wartość prędkości SL1 zostanie przekroczona, tracker wysyła wiadomość SMS z informacją o położeniu i prędkości z prefiksem „SL1\_OVR”. Przekroczenie limitu prędkości maksymalnej ma znaczenie drugorzędne i w algorytmie nie przewidziano innych działań z tym związanych.

Aby zminimalizować ryzyko, że wiadomość SMS z ważnym komunikatem zostanie przeoczona przez użytkownika lub że nie zostanie dostarczona przez operatora sieci komórkowej, niektóre z wiadomości są powtarzane co pewien czas. Dotyczy to SMS-ów z informacjami o przekroczeniu normalnych zakresów napięć, aktywowaniu wejść oraz przekroczeniu limitu prędkości. Pierwsze powiadomienie SMS wysyłane jest w ciągu minuty od przekroczenia limitów napięć

i prędkości oraz wykrycia stanu alarmowego na wejściach OC. Kolejne komunikaty wysyłane są w okresach określonych przez użytkownika i zapisanych w pamięci urządzenia. Okres powiadomień może wynosić od 1 do 999 minut. Zapis tych wartości do pamięci trackera możliwy jest jedynie przez interfejs USB, najlepiej z poziomu dołączonej aplikacji na komputer PC. Zagadnienie to szerzej omówiono przy okazji opisu uruchamiania urządzenia.

Jak już wielokrotnie wspomniano, użytkownik może komunikować się z trackerem za pomocą poleceń i komunikatów wysyłanych w postaci wiadomości tekstowych SMS. W pamięci trackera utworzono dwa rekordy do przechowywania numerów telefonów. Rekordy mają długość po dwanaście znaków, a zapisane w nich numery telefonów powinny mieć format „+yy-xxxxxxx” (gdzie: yy – numer kierunkowy kraju, dla Polski to oczywiście 48). Zależnie od tego, którym rekordom użytkownik nada status aktywności, tracker wysyła wszystkie komunikaty na obydwa lub tylko jeden

# Dostawa w ciągu 24h



element14

z zapamiętanych numerów. Użytkownik może ponadto zablokować funkcję wysyłania wszystkich wiadomości SMS, jeżeli żaden z rekordów nie będzie ustawiony jako aktywny. Sposoby wprowadzania numerów do pamięci i warianty wyboru aktywnego rekordu zostały opisane w dalszej części artykułu.

Przechowywane w pamięci numery telefonów służą także do weryfikowania nadawcy poleceń. Wiadomości wysłane z telefonów o innych numerach są traktowane jako nieautoryzowane i ignorowane. Zignorowana wiadomość SMS jest jedynie sygnalizowana wysłaniem na port USB komunikatu: „\$SMSU (SMS Unauthorised Sender)”. Akceptowane są natomiast polecenia wydawane z numerów z obu rekordów, niezależnie od tego czy mają ustawiony status aktywności czy nie. Jednak odpowiedzi na zapytania i potwierdzenia odsyłane są jedynie na numery z aktywnych rekordów. W tabeli 10 zestawiono obsługiwane polecenia SMS.

Użytkownik wysyłając polecenie do trackera nie musi wpisywać całej komendy w postaci takiej jak w cytowanej tabeli. Wystarczy wpisanie jedynie trzech liter wyróżnionych wielkimi literami. De facto, procedura interpretująca treść polecenia SMS analizuje cały tekst wiadomości w poszukiwaniu trzyznaczkowych słów kluczowych. Pozostałe znaki będące treścią wiadomości, niezależnie od tego czy znajdują się przed czy za słowem kluczowym, są ignorowane. Nieistotne jest także czy słowo kluczowe w całości lub fragmencie wpisane jest małą czy wielką literą. Na przykład, wiadomości o treści: „Dri”, „PeNdRiVe 8GB”, „He loved to drive his Jaguar” zostaną zinterpretowane tak samo.

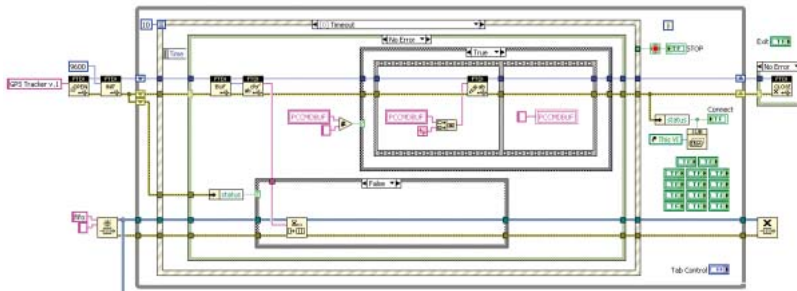
Stosownie do odebranego polecenia jest wywoływana odpowiednia procedura, a odebrana wiadomość od razu jest usuwana z pamięci karty SIM. Taki tok postępowania gwarantuje, że pamięć karty SIM nie zostanie zapełniona i modem będzie mógł odbierać wszystkie przychodzące wiadomości tekstowe. Tracker potwierdza wykonanie tych czynności wysyłając przez USB komunikat „\$SMSD (SMS Message Received & Deleted)”, natomiast użytkownik otrzymuje zwrótną wiadomością SMS – potwierdzenie wykonania polecenia. Wiadomości zwrótnie zestawiono w drugiej kolumnie tabeli 10.

Wysłanie każdej wiadomości SMS potwierdzane jest wysłaniem przez port USB komunikatu „\$SMSX (SMS Inf @ NumX Sented)”, w którym X oznacza rekord pamięci numerów telefonów użytkownika.

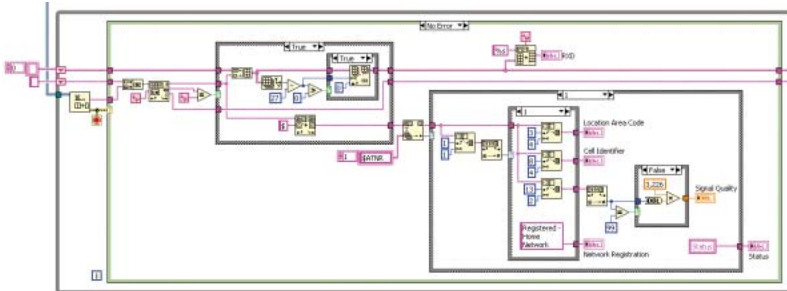
Do tej pory nie została jeszcze opisana, jeśli wziąć pod uwagę założenia projektu, jedna z bardziej istotnych funkcji trackera – mianowicie sterowanie wyjściami przekaźnikowymi. Użytkownik może to zrobić jedynie za pomocą poleceń SMS o treści: ENAbLe – DISAbLe (dla przekaźnika REL1) i DRiVe – STOp (dla przekaźnika REL2). Użytkownik

Tabela 12. Polecenia konfiguracyjne trackera GPS

Funkcja	Polecenie	Argumenty
Numer PIN karty SIM	\$P,xxxx	x = 0..9 (czterocyfrowy PIN)
Numery telefonów	\$N,y,xxxxxxxxxxxx	y = 0, 1 (rekord 0 lub 1)
Aktywne numery telefonów	\$S,x	x = 0..3 0 – obydwa nieaktywne 1 – aktywny numer z rekordu 0 2 – aktywny numer z rekordu 1 3 – aktywne obydwa numery
Tryb pracy wejść OC	\$O,x	x = 0..4,6,8..9 0 – wejścia OC nieaktywne 1 – aktywne wejścia OC1–3 2 – aktywne wejścia OC4 3 – aktywne wejścia OC1–3 i OC4 4 – aktywne OC1–3 wraz SLO 6 – aktywne OC1–3 wraz SLO oraz OC4 8 – aktywne OC4 wraz SLO 9 – aktywne OC4 wraz SLO oraz OC1–3
Limity prędkości	\$L,y,xxx	y = 0, 1 (limit SLO lub SL1) x = 0..999 (prędkość wyrażona w km/h)
Okres wysyłania powiadomień	\$I,y,xxx	y = 0..3 0, 1 – limity prędkości SLO lub SL1 2 – limity napięć 3 – alarmy wejść OC x = 0..999 (okres powiadomień wyrażony w min)
Odczyt wszystkich rekordów pamięci trackera	\$M	



Rysunek 7. Pętla obsługująca komunikację przez interfejs USB (LabVIEW)



Rysunek 8. Procedura odczytująca kody LAC, CID i SQ z komunikatu \$ATNR (LabVIEW)

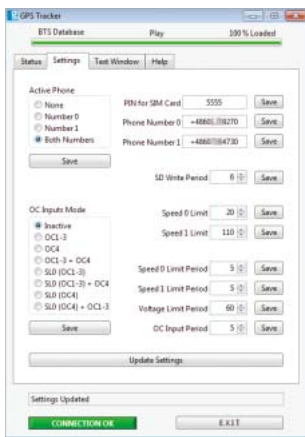
dotrzymuje potwierdzenie wykonania każdego z tych poleceń w postaci wiadomości SMS z pełną nazwą wykonanego polecenia (tabela 10). Odpowiedni komunikat trafia także na port USB. Jedyną sytuacją, w której algorytm trackera samodzielnie może zmienić stan wyjść przekaźnikowych jest przełączenie w tryb ograniczonego poboru mocy. Wówczas, niezależnie od aktualnego stanu na tych wyjściach, napięcie zasilające obydwa przekaźniki jest odłączane, co oznacza, że

przełącznik REL1 zostaje przełączony w stan ENABLE, a REL2 w stan DRIVE.

Ostatnią ważną kwestią wymagającą omówienia jest komunikacja z komputerem. Ponieważ do transmisji danych pomiędzy modemem GSM a modulem GPS wykorzystano oba dostępne w ATmega128 sprzętowe interfejsy USART, komunikację przez interfejs USB zrealizowano programowo. Wymiana danych odbywa się z szybkością 9600 bps (8N1). Tworząc protokół komunikacji przyję-



Rysunek 9. Program do konfiguracji trackera – okno tekstowe



Rysunek 10. Program do konfiguracji trackera – zakładka Settings

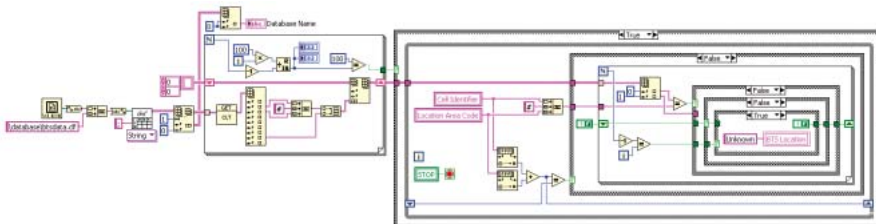
to konwencję, zgodnie z którą wszystkie wysyłane i odbierane komunikaty rozpoczynają się znakiem „\$”, następny jest jedno- lub czteroznakowy kod komunikatu, dalej opcjonalne argumenty oraz komentarz ujęty w nawiasy okrągłe. Komunikat zamykają znaki powrotu karetki <CR> i końca linii <LF>. Odbierane przez tracker komunikaty konfiguracyjne mają jednoznakowy kod, a wysyłane przez urządzenie mają kod czteroznakowy. Komunikaty konfiguracyjne zestawiono w tabeli 12, a komunikaty wysyłane przez port USB w tabeli 11.

Aby oszczędzać czas mikrokontrolera, procedura od-

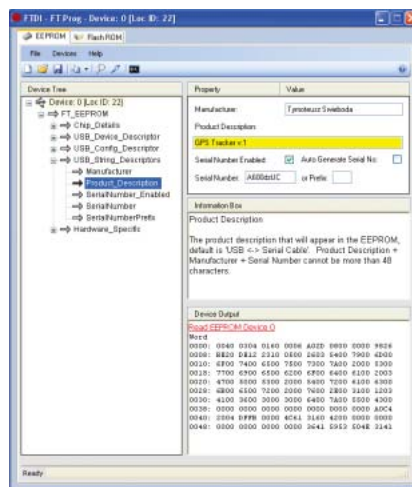
„\$PCCE”.

### Program dla PC

Konfiguracja trackera jest możliwa przy użyciu zaledwie siedmiu komend (tab. 12), jednak aby ograniczyć uciążliwości związane z ręcznym wprowadzaniem poleceń konfiguracyjnych, na przykład za pomocą jakiegoś terminala, zdecydowano o utworzeniu programu przeznaczonego do tego celu. Ponadto, taki program mógłby mieć zaimplementowane kilka funkcji o charakterze diagnostycznym. Taka właśnie jest geneza powstania aplikacji utworzonej w środowisku LabVIEW.



Rysunek 11. Procedura wyszukująca aktualne kody LAC i CID w bazie danych BTS (LabVIEW)



Rysunek 12. Okno aplikacji FT\_PROG – edycja pola Product Description

bierająca znaki na linii TXD układu FTDI jest wywoływana co 10 sekund. W ten sposób powstają kilkumilisekundowe okna czasowe, w których komputer PC może nadawać polecenia. Początek okna komunikacji jest sygnalizowany nadaniem przez tracker komunikatu

Program zawiera dwie pętle: jedna jest odpowiedzialna za komunikację z układem trackera i wykrywanie zdarzeń, a druga przetwarza i interpretuje dane. Pętle współpracują ze sobą według modelu producent – konsument, a przepływ danych odbywa się za pośrednictwem kolejki FIFO.

Rysunek 7 przedstawia fragment kodu pętli realizującej funkcję komunikacji przez interfejs USB. Po uruchomieniu programu, pętla nawiązuje połączenie z układem FTDI z opisem „GPS Tracker v.1”. Jednocześnie jest tworzona kolejka FIFO. Później, w każdym przebiegu pętli, dane z bufora konwertowane są na ciąg znaków ASCII i kolejno zapisywane do kolejki. Równolegle jest monitorowany stan panelu operatorskiego i jeżeli użytkownik wykona jakąś akcję, pętla ta rejestruje takie zdarzenie (struktura Event). Na tej podstawie jest generowana stosowna komenda i umieszczana w buforze PCCMDBUF. W kolejnym przebiegu pętli łańcuch znaków z tego bufora jest automatycznie wysyłany do układu FTDI, a bufor czyszczony.

Druga pętla pobiera znaki z kolejki FIFO i odbudowuje z nich poszczególne komunikaty nadawane przez tracker. Jako marker wykorzystywany jest symbol „\$” znajdujący się na początku każdego komunikatu. Następnie wyodrębnione komunikaty są identyfikowane na podstawie czteroznakowego kodu. Po tej operacji wywołana jest odpowiednia procedura. Dla przykładu na **rysunku 8** przedstawiono fragment kodu drugiej pętli z procedurą odczytującą kody LAC, CID i SQ. W jej kodzie wykorzystano fakt, że dla każdej z tych wielkości zawartych w komunikacie \$ATNR przewidziano stałą liczbę znaków – określone fragmenty łańcucha znaków są wycinane i zapisywane do odpowiednich zmiennych.

Mając nakreśloną zasadę działania programu, można przejść do omówienia panelu operatorskiego. W dolnej części okna programu znajdują się: pole tekstowe, w którym wyświetlane są potwierdzenia ostatnio wykonanych operacji, kontrolka informująca o statusie połączenia i przycisk EXIT, który zatrzymuje działanie programu. Centralną część panelu to kontrolka z czterema zakładkami. Każda z zakładek grupuje kontrolki z jednej z głównych kategorii zadań: aktualny status trackera, ustawienia i okno tekstowe. Czwarta zakładka zawiera zestawienie informacji, które mogą być pomocne podczas obsługi programu.

Okno tekstowe, to nic innego jak uproszczony terminal wyświetlający komunikaty wysyłane z trackera przez port USB (**rysunek 9**). Śledzenie i analiza zawartości tego okna jest najlepszą metodą diagnostyczną opisywanego urządzenia. Użytkownik ma też możliwość manualnego wysyłania poleceń konfiguracyjnych do trackera.

Zakładka „Settings” zawiera kontrolki służące do zapisywania w pamięci trackera wszystkich ustawień (**rysunek 10**). Każdy z przycisków „Save” wysyła tylko jedno polecenie konfiguracyjne. W ten sposób zmniejszono ryzyko powstania błędów, ale przede wszystkim ograniczono liczbę cykli zapisu do niemodyfikowanych rekordów pamięci. W zakładce umieszczono także przycisk „Update settings”, którym użytkownik może odświeżać wszystkie pola zakładki.

W zakładce „Status” zebrano kontrolki informujące o aktualnym stanie pracy podstawowych bloków funkcjonalnych trackera: wartości napięć szyn zasilających, status logowania modemu do sieci GSM, dane z modułu GPS i znacznik ich ważności oraz status karty microSD. Nieco miejsca trzeba poświęcić na opis pola tekstowego o nazwie „BTS Location”.

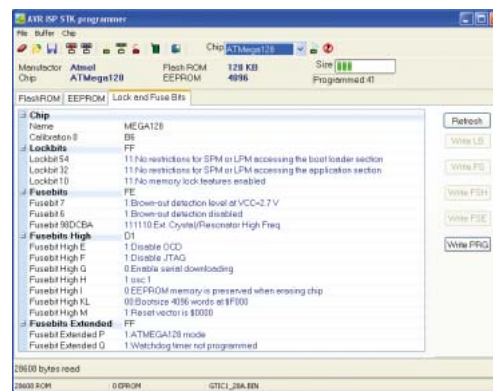
Kody CID i LAC pozwalają jednoznacznie zidentyfikować stację przekaźnikową sieci komórkowej, do której zalogowany jest modem. Korzystając z tego, że na stronie

internetowej [btsearch.pl](http://btsearch.pl) dostępne są bazy danych zawierające informacje o stacjach BTS wszystkich krajowych operatorów, możliwe stało się poszerzenie funkcjonalności programu o identyfikację stacji, do której w danym momencie zalogowany jest tracker. Informacja ta prezentowana jest właśnie w polu tekstowym „BTS Location”. Dołączony do projektu program instalacyjny oprogramowania na PC zawiera bazę danych obejmującą stacje BTS sieci Play w całym kraju.

Instalator aplikacji do obsługi trackera został wygenerowany z założeniem, że będzie uruchamiany w systemach operacyjnych Windows XP i Windows 7. Działanie programu na innych wersjach Windows nie było testowane, co oczywiście nie oznacza, że nie jest możliwe. Instalacja programu przebiega typowo. Wraz z programem instalowane są także sterowniki do układu FTDI, który po podłączeniu trackera przewodem USB powinien być widoczny w menedżerze urządzeń jako „USB Serial Port (COMx)”.

Użytkownik może podmienić instalowaną domyślnie bazę danych, stosownie do wybranego operatora sieci komórkowej. Na stronie [btsearch.pl/nobbi-monitor.php](http://btsearch.pl/nobbi-monitor.php) znajduje się narzędzie umożliwiające eksport danych do formatu CLF w wersji 3.0. Oczywiście wcześniej należy wybrać operatora oraz zaznaczyć, z których województw nadajniki mają zostać włączone do bazy. Pobrany plik CLF wymaga pewnych modyfikacji. W pierwszej kolejności, korzystając na przykład z systemowego notatnika, trzeba skopiować jego zawartość (oprócz pierwszego wiersza) do nowego pliku TXT. W pierwszym wierszu nowego pliku należy wpisać informację o bazie, np. nazwę, a w drugim liczbę jej rekordów (tj. pozostałych wierszy). Następnie trzeba zmienić nazwę pliku na „btsdata.clf” i podmienić plik oryginalny, który znajduje się w folderze o nazwie „database”.

**Rysunek 12** przedstawia fragment kodu programu z procedurą obsługującą bazę danych. Po uruchomieniu programu i odtworzeniu ścieżki dostępu, plik btsdata.clf jest otwierany, a zawarte w nim dane kopiowane do tablicy. Ponieważ przy większej ilości danych (baza, jeżeli zawiera dane o BTS-ach wszystkich operatorów, zawiera ponad 54 tys. rekordów) proces ten może trwać dłuższą chwilę, w górnej części okna programu umieszczono pasek informujący o postępie tego procesu. W trakcie normalnej pracy programu, jeżeli wartości CID i LAC ulegną zmianie, to przekazywane są do pętli, w której są porównywane z odpowiednimi polami w kolejnych rekordach tablicy. Po stwierdzeniu zgodności zawartości rekordu prezentowana jest w polu „BTS Location”.



**Rysunek 14. Konfiguracja fuse bits mikrokontrolera**

Z doświadczeń przeprowadzonych z instalacją i użytkowaniem programu na różnych komputerach wynika, że zwykle przy pierwszej próbie podłączenia trackera po uruchomieniu programu na komputerze PC, połączenie zostaje nawiązane, a mimo to dane nie są przesyłane. W takiej sytuacji połączenie należy zrestartować odłączając jedną z wtyczek przewodu USB i od razu ponownie podłączyć do tego samego gniazda.

## Uruchomienie

W uruchomianiu układu można wyróżnić kilka etapów, które nie powinny nastręczać większych trudności. Pierwszy, to po prostu podłączenie układu do źródła napięcia około 12 V i sprawdzenie wartości napięcia poszczególnych szyn zasilających. Powinny one wynosić: VREL = 6,0 V, VCC = 3,6 V, VCCAT = 3,6 V i VCCSD = 3,3 V.

W drugim etapie trzeba skonfigurować interfejs USB. Ze strony internetowej producenta trzeba pobrać narzędzie FT\_PROG. Jeżeli na komputerze zostanie zainstalowany program do konfigurowania trackera, nie trzeba pobierać sterowników do układu (dostępnych na tej samej witrynie) – w trakcie jego instalacji instalują się również odpowiednie sterowniki. Po zainstalowaniu sterowników i uruchomieniu aplikacji (FT\_PROG jest plikiem EXE i nie wymaga instalowania) można podłączyć tracker do portu USB. Podłączanie zasilania do płytki nie jest konieczne, ponieważ w zastosowanej aplikacji układ IC5 pobiera je z portu USB. Po wybraniu opcji wyszukiwania, podłączony układ FT232RL powinien zostać wykryty i zidentyfikowany. Edycji wymaga pole „USB\_String\_Descriptors/Product Description”, którego zawartość należy zastąpić tekstem „GPS Tracker v.1” (**rysunek 13**). Dzięki temu tracker będzie poprawnie wykrywany przez aplikację na PC.

Konfiguracja fuse bits mikrokontrolera to trzeci etap. Trzeba wyłączyć JTAG oraz wybrać źródło taktowania mikrokontrolera – zewnętrzny rezonator kwarcowy. Należy pamiętać, że pomyłka przy wyborze źródła tak-

towania, zwłaszcza w wersji SMD, może być trudna do skorygowania. Docelową konfigurację fuse bits przedstawiono na **rysunku 14**.

Czwarty etap składa się z dwóch kroków i polega na zainstalowaniu oprogramowania mikrokontrolera, a jednocześnie na skonfigurowaniu odbiornika GPS – układu IC3. W rozdziale opisującym układ JGR-SC3-S wspomniano, że przy ustawieniach fabrycznych, wysła on co sekundę cztery wiadomości NMEA: GGA, GSA, GSV i RMC. Algorytm mikrokontrolera wykorzystuje tylko informacje zawarte w wiadomości RMC i z uwagi na maksymalne uproszczenie procedur, tylko ten rodzaj wiadomości potrafi zinterpretować. Zatem pozostałe wiadomości należy wyłączyć. Ponadto, należy też zwiększyć długość interwałów wysyłania komunikatów do 10 s. W tym celu do odbiornika GPS należy wysłać polecenie „\$PSRF109,NMEA9600,NULL38400,GGA0,LL0,GSA0,GSV0,RMC10,VTG0,USER0\*37”.

Ponieważ czynność tę wystarczy wykonać tylko jeden raz, nie było sensu implementowania specjalnych procedur w użytkowym oprogramowaniu mikrokontrolera. Dlatego do projektu dołączono drugi zestaw plików gTIC1\_CONFIG.BIN i gTIC1\_CONFIG.HEX zawierający uproszczoną wersję kodu, który w założeniu miał służyć jedynie do skonfigurowania odbiornika GPS. Ponieważ w tej wersji kodu zaimplementowano także komunikację z komputerem PC, więc warto przed skopiowaniem plików na mikrokontroler podłączyć tracker do PC-ta i w oknie tekstowym obserwować pojawiające się komunikaty. Jeżeli komunikat rozpoczynający się znakami „\$GPRMC” jest poprawnie interpretowany i pojawia się co około 10 sekund oznacza to, że konfiguracja odbiornika GPS przebiegła poprawnie. Zatem mikrokontroler trzeba zaprogramować dwukrotnie – za pierwszym razem instalując program konfiguracyjny, a później docelową aplikację.

Ostatni, piąty etap polega na zapisaniu w pamięci trackera numeru PIN, numerów telefonów, limitów prędkości, interwałów wysyłania powiadomień oraz wyborze trybu pracy wejść OC. Zapisanie tych ustawień w pamięci jest możliwe za pomocą opisanej w poprzednim rozdziale aplikacji na komputer PC, ewentualnie „na piechotę” za pomocą dowolnego terminala, np. Docklight.

Jeżeli wszystkie opisane etapy zostaną zrealizowane poprawnie, układ trackera można uznać za w pełni uruchomiony. Dopiero teraz do trackera można włożyć kartę SIM – wcześniej, gdy w pamięci numeru PIN znajdowały się przypadkowe znaki, uruchomienie trackera wiązałoby się z poważnym ryzykiem zablokowania karty (tracker nie umożliwia wprowadzania przez użytkownika numeru PUK).

Oczywiście modem i odbiornik należy wyposażać w anteny. Przy wyborze anten

istnieje niemal pełna dowolność. Ograniczeniem jest jedynie rodzaj złącza. Powinno to być męskie złącze SMA. Najprościej jest wykorzystać popularne (dostępne w wielu sklepach internetowych) zintegrowane anteny GPS/GSM do zastosowań motoryzacyjnych, które sprzedawane są z kilkumetrowym przewodem z wtyczką. W egzemplarzu modelowym zastosowano, kupioną na rynku wtórnym, antenę GPS/GSM z Volkswagena Touarega.

Płytkę należy umieścić w obudowie. Prototyp miał obudowę wykonaną specjalnie dla niego, ale oczywiście można dopasować gotową, dostępną w ofercie różnych producentów.

Po zmontowaniu i uruchomieniu trackera można rozpocząć pierwsze testy praktyczne. Jeżeli wypadną pomyślnie, można przejść do ostatniego etapu – zainstalowania trackera w samochodzie. Decydując się na ten krok, należy zdawać sobie sprawę, że jest on dość poważną ingerencją w instalację elektryczną pojazdu. Wszelkie modyfikacje należy przeprowadzić w taki sposób, by mieć pewność, że nie wpłyną one na funkcjonowanie samochodu, a tym bardziej na bezpieczeństwo użytkownika. Autor projektu nie ponosi odpowiedzialności za ewentualne szkody, które mogą zaistnieć w wyniku zainstalowania i użytkowania trackera.

Podłączając sygnały do wejść trackera można skorzystać z tego, że masa układu ma ten sam potencjał, co masa samochodu, więc do połączeń można użyć przewodu jednożyłowego. Z doświadczeń zdobytych podczas eksploatacji prototypu wynika, że nie warto oszczędzać przy zakupie złączy Mini-Fit. Lepiej kupić elementy o dobrej jakości, które zagwarantują pewne połączenie elektryczne gniazda i wtyczki.

### Doświadczenia z eksploatacji

Tracker należy traktować jak układ zasilany bateryjnym. W czasie normalnej pracy, zależnie od wartości napięcia w instalacji elektrycznej samochodu, tracker zużywa 1,1...1,3 W. Uaktywnienie każdego z wyjść przekaźnikowych zwiększa pobór mocy o około 0,36 W. Po przełączeniu w tryb ograniczonego poboru energii zapotrzebowanie na moc spada do 0,5...0,6 W. Gdy okresy postoju nie są dłuższe niż kilka dni, ta „prądożerność” trackera nie jest problemem. Należy jednak mieć na uwadze ten aspekt eksploatacji.

W codziennym użytkowaniu tracker działa przewidywalnie i zgodnie z założeniami. Sporadycznie przydarzały się problemy z wysyłaniem (a może odbieraniem) wiadomości SMS, ale wydaje się, że to raczej efekt związany ze specyfiką wybranego operatora.

**Ytmoteusz Świeboda**  
gleitmo@vp.pl

## UKŁADY INTERNETOWE



Karta przekazników sterowana przez Internet  
**AVT5250**



Karta I/O sterowana przez Internet  
**AVT966**



Karta wejść z interfejsem Ethernet  
**AVT953**



Moduł I/O sterowany przez Internet  
**AVTMOD05**



[www.sklep.avt.pl](http://www.sklep.avt.pl)

AVT-Korporacja Sp. z o.o., 03-197 Warszawa, ul. Leszczyńska 11,  
tel.: 22 257 84 50, fax: 22 257 84 55, e-mail: handlowy@avt.pl