
**AVT
5271**

VAGLogger

Przyrząd diagnostyczny dla samochodów z grupy VW-Audi



Diagnostyka współczesnych systemów sterowania silnikami spalinowymi wymaga nie tylko odpowiedniej wiedzy, ale i odpowiednich narzędzi diagnostycznych. Większość czynności diagnostycznych można wykonać tylko przy użyciu dedykowanego dla danego modelu samochodu testera.

Rekomendacje: przyrząd może przysłać się każdemu, kto lubi troszkę „pogrzebać” w silniku samochodu.

Dla aut grupy Volkswagen-Audi (producenta takich marek jak Volkswagen, Skoda, Seat oraz Audi) powstało kilka programów umożliwiających diagnostykę z użyciem prostego interfejsu podłączonego do portu szeregowego lub USB. Jednym z bardziej znanych jest VagCom. Umożliwia on odczyt i kasowanie błędów nie tylko ze sterownika silnika, ale także z innych podzespołów samochodu (ABS, poduszki powietrzne itp.). Jest możliwy odczyt grup pomiarowych, a także adaptacje i kodowanie sterowników. Jedną z bardziej przydatnych funkcji jest zapis do pliku, bloków pomiarowych zbieranych podczas testowego przejazdu samochodu. Późniejsza analiza takiego zbioru danych pomaga zdiagnozować niesprawność danego podzespołu silnika szczególnie, jeżeli problem występuje sporadycznie lub na przykład tylko przy dużym obciążeniu silnika. Powstały plik tekstowy zawiera grupy danych rozdzielone średnikami – format odczytywany przez większość popularnych arkuszy kalkulacyjnych, co można wykorzy-

stać do utworzenia wykresu danej wartości w funkcji czasu.

Z racji bezpieczeństwa rejestracja danych podczas jazdy wymaga obsługi komputera przenośnego przez drugą osobę. W wielu przypadkach jest to kłopotliwe, dlatego postanowiłem opracować samodzielne urządzenie, które zapisywałoby odczytane ze sterownika silnika dane na karcie SD. Położyłem nacisk na intuicyjną, zredukowaną do absolutnego minimum, obsługę, która nie rozprasza uwagi kierowcy pojazdu. Wybór konfiguracji logowania, rozpoczęcie oraz zatrzymanie pracy jest kontrolowane przy użyciu jednego przycisku. Stan urządzenia jest sygnalizowany dwukolorową diodą świecącą oraz akustycznie.

Opracowany tester komunikuje się ze sterownikiem silnika (ECU) przy użyciu protokołu KW1281. Z tego też powodu diagnostyka jest ograniczona do aut grupy Volkswagen-Audi, które miały swoją premierę rynkową do roku 2003. Po tym czasie producent wycofał się z diagnostyki silnika

AVT-5271 w ofercie AVT:
AVT-5271A – płytka drukowana

Podstawowe informacje:

- Przyrząd diagnostyczny dla samochodów z grupy VW
- Przetestowano działanie z następującymi modelami samochodów:
 - Audi A4 1.9TDI z 1996 roku
 - Skoda Octavia 1.8T z 2005 roku
 - Skoda Octavia 1.9TDI z 2000 roku
 - Audi A3 1.6 z 2002 roku
 - VW Polo 1.2 z 2003 roku
- Płytkę dwustronna o o wymiarach 60×41 mm

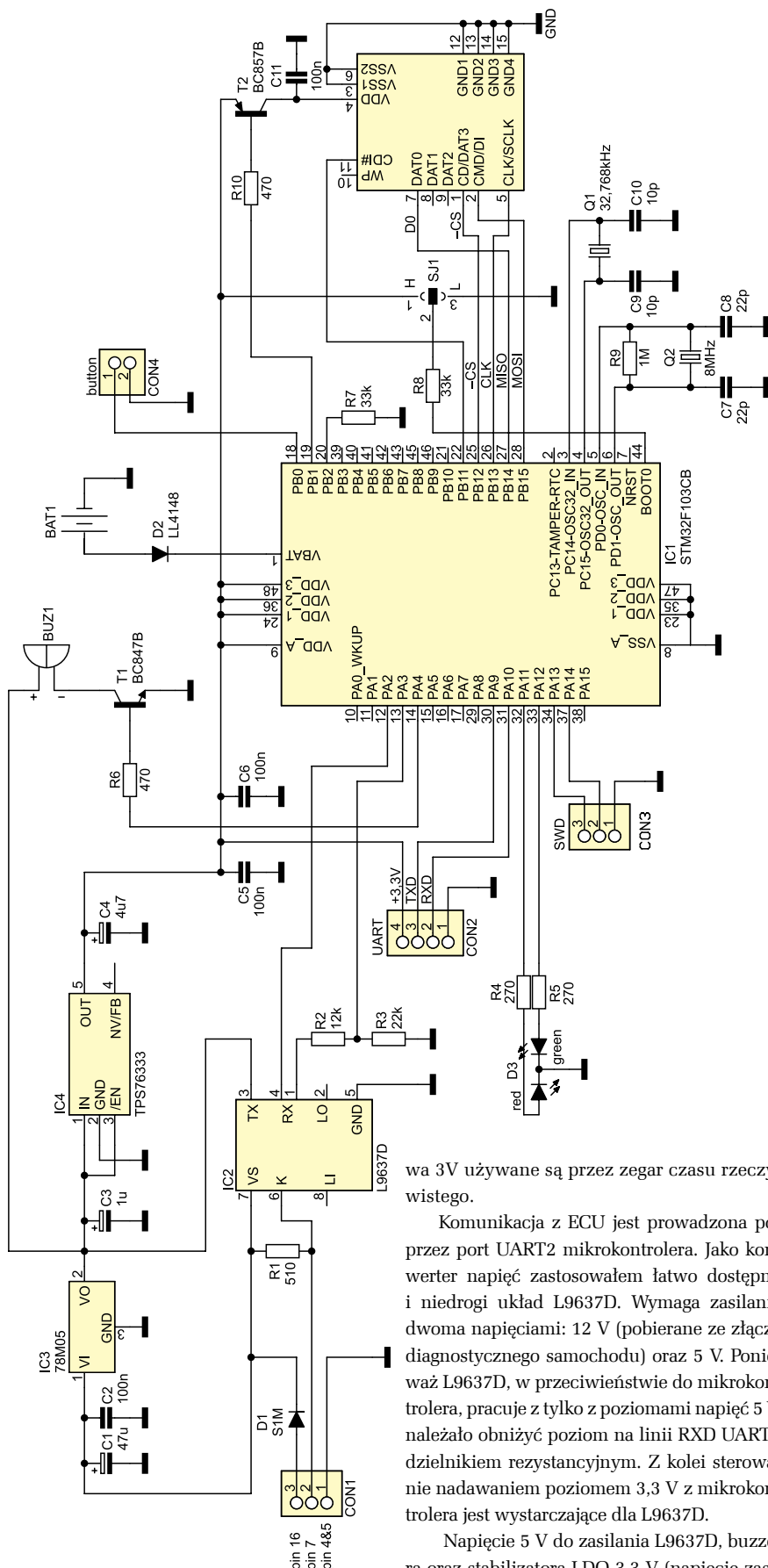
Dodatkowe materiały na CD i FTP:

- <ftp://ep.com.pl>, user: 10142, pass: 5x7bu87r
- wzory płytek PCB
 - karty katalogowe i noty aplikacyjne elementów oznaczonych w wykazie elementów kolorem czerwonym

(a także innych podzespołów) po asynchronicznej linii szeregowej K, a jej miejsce zajęła komunikacja z użyciem magistrali CAN. Należy także wspomnieć, iż można się spotkać z samochodami grupy Volkswagen-Audi, które mimo diagnostyki za pomocą linii K, wykorzystują protokół inny niż KW1281 lub też odpowiedzi są niezgodne z oczekiwanymi (przykładem jest Skoda Felicia 1.3MPI z 1997 roku).

Budowa

Sercem urządzenia jest mikrokontroler z rodziny Cortex-M3 STM32F103CB, który jest dostępny w obudowie PQFP48. Mikrokontroler ma 128 kB pamięci programu oraz



Rysunek 1. Schemat ideowy przyrządu

20 kB pamięci RAM. Taktowany jest rezonatorem kwarcowym 8 MHz, którego częstotliwość jest wewnętrznie powielana z użyciem PLL. Rezonator 32,768 kHz oraz bateria lito-

wa 3V używane są przez zegar czasu rzeczywistego.

Komunikacja z ECU jest prowadzona poprzez port UART2 mikrokontrolera. Jako konwerter napięć zastosowałem łatwo dostępny i niedrogi układ L9637D. Wymaga zasilania dwoma napięciami: 12 V (pobierane ze złącza diagnostycznego samochodu) oraz 5 V. Ponieważ L9637D, w przeciwieństwie do mikrokontrolera, pracuje z tylko z poziomami napięć 5 V, należało obniżyć poziom na linii RXD UART2 dzielnikiem rezystancyjnym. Z kolei sterowanie nadawaniem poziomem 3,3 V z mikrokontrolera jest wystarczające dla L9637D.

Napięcie 5 V do zasilania L9637D, buzzeria oraz stabilizatora LDO 3,3 V (napięcie zasilania mikrokontrolera oraz karty SD) jest wytwarzane typowym stabilizatorem liniowym 78M05. Niewielki pobór prądu nie powoduje dużych strat na tym stabilizatorze i nie zaobserwowałem nadmiernego nagrzewania się układu nawet podczas dłuższej pracy. Należy

Wykaz elementów

Rezystory:

R1: 510 Ω
R2: 12 kΩ
R3: 22 kΩ
R4, R5: 270 Ω
R6: 470 Ω
R7: 33 kΩ
R8: 33 kΩ
R9: 1 MΩ
R10: 470 Ω

Kondensatory:

C1: 47 μF/25 V
C2: 100 nF (1206)
C3: 1 μF/10 V
C4: 4,7 μF/10 V
C5: 100 nF
C6: 100 nF
C7: 22 pF
C8: 22 pF
C9: 10 pF
C10: 10 pF
C11: 100 nF

Półprzewodniki:

D1: S1M lub inna prostownicza SMD
D2: LL4148 (MINIMELF)
D3: dioda LED dwukolorowa, 5 mm, wspólna katoda
T1: BC847B
T2: BC857B
IC1: STM32F103CB
IC2: L9637D
IC3: 78M05
IC4: TPS76333

Inne:

Q1: rezonator kwarcowy 8MHz
Q2: rezonator kwarcowy 32.768kHz
BUZ1: buzzer z generatorem na napięcie 5 V
BAT1: bateria CR2032 z wyprowadzeniami do montażu poziomego
Przycisk zwiny
Wtyk OBD2 duży
Gniazdo karty SD bez wyrzutnika

pamiętać, że wyłączenie stacyjki w samochodzie nie odcina zasilania na pinie 16 złącza diagnostycznego i urządzenie należy każdorazowo odłączyć po skończonej pracy.

Oprogramowanie

Oprogramowanie zostało napisane w języku C i skompilowane kompilatorem GCC. Wykorzystałem bezpłatne biblioteki do obsługi peryferiów dostarczone przez ST-Microelectronics w wersji 2.0.3 [2]. System plików FatFS jest autorstwa osoby podpisującej się pseudonim ChaN i również jest dostępny bezpłatnie [1].

Komunikacja ze sterownikiem samochodu

Komunikacja ze sterownikiem silnika jest dokonywana przy użyciu dwukierunkowej jedнопроводowej magistrali K. Dane są przesyłane asynchronicznie (tak jak w przypadku RS232) z jednym bitem startu, ośmioma bitami danych i jednym bitem stopu. Prędkość transmisji wynosi, w przypadku aut, w których wyższą warstwą jest protokół KW1281, 9600 lub 10400 bodów. Wyjątkiem jest sekwencja inicjalizacji połączenia.



Komunikacja testera ze sterownikiem polega na wymianie komunikatów o długości od 4 do 256 bajtów.

Pojedynczy komunikat składa się z następujących pól:

- długość komunikatu, nie uwzględnia znacznika końca komunikatu (1 bajt)
- kolejny numer komunikatu, inkrementowany każdorazowo o 1 (1 bajt)
- typ komunikatu (1 bajt)
- blok danych, może być pusty (maksymalna długość tego pola to 252 bajty)
- znacznik końca komunikatu, ma zawsze wartość 0x03 (1 bajt)

Kolejne bajty komunikatu (z wyjątkiem znacznika końca) muszą być każdorazowo potwierdzane przez drugą stronę poprzez odesłanie zanegowanej wartości odebranego bajta.

Zaimplementowałem obsługę następujących typów komunikatów:

- potwierdzający (ACK, 0x09)
- żądanie odczytu grupy pomiarowej (GROUP_REQUEST, 0x29)
- odpowiedź z zawartością grupy pomiarowej (GROUP_RESPONSE, 0xE7)
- koniec sesji (END_OF_SESSION, 0x06)
- dane ASCII (ASCII_BLOCK, 0xF6)

Inicjalizacja połączenia następuje poprzez wysłanie przez tester adresu sterownika, z którym ma nastąpić komunikacja. W przypadku silnika jest to adres 01h. Wysłanie sekwencji inicjalizacyjnej następuje z prędkością 5 bodów (czas trwania pojedynczego bitu to 200 ms).

Zaadresowany sterownik, z właściwą sobie prędkością transmisji, odpowiada wysłaniem trzech bajtów: 55h (bajt synchronizacji) oraz dwubajtowego słowa kodowego oznaczającego numer protokołu (w przypadku KW1281 są to kolejno bajty o wartościach 01h oraz 8Ah). Mierząc czas trwania słowa synchronizującego (naprzemienna sekwencja zer i jedynek) można określić zastosowaną przez sterownik prędkość transmisji i odpowiednio zaprogramować UART mikrokontrolera w celu poprawnego odebrania kolejnych dwóch bajtów.

Jako potwierdzenie tester odsyła zanegowaną wartość ostatniego odebranego bajta (w przypadku protokołu KW1281 będzie to 75h).

Po inicjalizacji sterownik (w tym przypadku sterownik silnika) przesyła pierwszy komunikat. Zwykle pierwsze kilka komunikatów zawiera dane tekstowe ASCII (numer części, model sterownika itp.). Tester potwierdza kolejne komunikaty odebrane od sterownika wysyłając ramki typu ACK o długości 4 bajtów (z pustym polem danych).

Kiedy ECU wyśle wszystkie bloki ASCII nastąpi wzajemna wymiana komunikatów ACK (służy to podtrzymaniu połączenia). Po odebraniu kolejnych 8 komunikatów ACK tester wysyła pierwsze zapytanie o grupę pomiarową (nie jest to wymóg protokołu a tylko moje założenie). Komunikat tego typu ma długość 5

bajtów a jego pole danych zawiera numer żądanej grupy (od 0 do 255). Odpowiedzią ze strony sterownika jest komunikat typu *GROUP_RESPONSE* z polem danych o długości 12 bajtów (w przypadku grupy zerowej jest to 8 bajtów).

Każda grupa pomiarowa zawiera cztery wartości (po trzy bajty na każdą). Pierwszy bajt takiej podgrupy zawiera rodzaj (typ) wartości i mówi o tym, jakiej procedury należy użyć, aby przeliczyć dwa kolejne bajty na wartość rzeczywistą. Wzory do przeliczenia odebranych wartości są dostępne na stronie [3]. Powyższe nie dotyczy grupy o numerze 0, grupa ta zawiera osiem wartości, po jednym bajcie na każdą wartość.

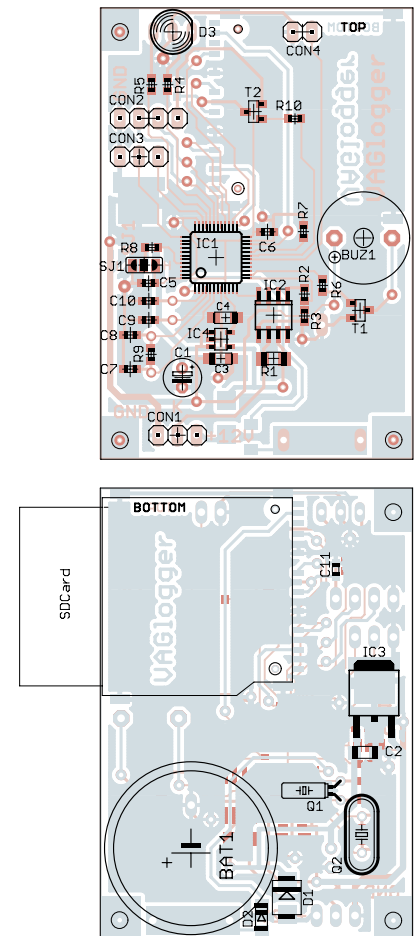
Jednocześnie można wysłać zapytanie tylko o jedną grupę pomiarową. VAGlogger pozwala na logowanie maksymalnie trzech grup a zapytania o kolejne numery są dokonywane sekwencyjnie.

Zakończenia komunikacji ze sterownikiem, tester może dokonać wysyłając ramkę typu *END_OF_SESSION*. W programie VAGlogger przyjąłem, że nieotrzymanie odpowiedzi od sterownika w ciągu 500 milisekund jest traktowane jako zerwanie połączenia.

Montaż i programowanie mikrokontrolera

VAGlogger wykonano na dwustronnej płytce o wymiarach 60×41 mm. Przelotki zostały umieszczone poza obrysem elementów, dzięki czemu można ją wykonać w warunkach domowych (bez metalizacji otworów). Płytkę została wykonana metodą fotochemiczna na gotowym laminacie firmy Bungard o grubości 0,8 mm i naświetlanym lampą UV do opalania twarży. Laminat jest wywoływany identycznie jak laminat pokryty lakierem światłoczułym Positiv 20, czyli w roztworze wodorotlenku sodu. Płytkę została wytrawiona w roztworze B327. Po wytrawieniu ścieżek, płytkę należy zabezpieczyć przed utlenianiem – osobiście używam sprayu Flux firmy Contact Chemie.

Przed zamontowaniem elementów warto odznaczyć w obudowie wtyku otwory pod śruby mocujące. Jeśli płytkę nie ma metalizacji otworów, to montaż należy rozpocząć od przelotek. Można je wykonać np. ze srebrzonego drutu o średnicy 0,2...0,3 mm. Wygięty w kształt litery U drut przekładamy przez dwa sąsiadujące ze sobą otwory. Sprężystość drutu zapewnia pewne osadzenie przelotek przed lutowaniem i nie ma ryzyka ich wypadnięcia nawet po obróceniu płytki. Część przelotek znajdują się blisko innych punktów lutowniczych, dlatego warto je skrócić dopiero po przylutowaniu reszty elementów. Mikrokontroler ma raster wyprowadzeń 0,5 mm a jego montaż wykonałem przy użyciu lutownicy kolbowej z cienkim grotem. Jako ostatnie należy zamontować elementy przewlekane (buzzer, bateria litowa). Dioda świecąca jest montowana na skróconych wyprowadzeniach bezpośrednio do PCB, przycisk na krótkich przewodach.



Rysunek 2. Schemat montażowy przyrządu

Jako złącza CON1 (podłączenie złącza diagnostycznego samochodu) oraz CON2 (UART1 mikrokontrolera, używane tylko do jego zaprogramowania) należy zastosować kątowe listwy kołkowe. Proszę zwrócić uwagę, że niektóre wyprowadzenia muszą zostać przylutowane po obu stronach płytki (najpierw od spodu PCB a później po delikatnym podważeniu plastikowej części listwy u góry). Złącza CON3 nie trzeba montować – jest to złącze debugera SWD. Po montażu należy optycznie sprawdzić płytkę pod kątem wystąpienia zwarc.

Kolejnym etapem jest zaprogramowanie mikrokontrolera. Przed podłączeniem zasilania należy wykonać zwór między rezystorem R8 (wyprowadzenie BOOT0 mikrokontrolera) a +3,3 V – na płytce jest to zwora oznaczona SJ1-H, w celu wymuszenia na mikrokontrolerze startu w trybie bootloadera. Układ należy zasilic podłączając napięcie 12V między piny 1 i 3 złącza CON1. Poprzez konwerter 3.3V/RS232 podłączyć płytkę VAGlogger (złącze CON2) do portu szeregowego komputera PC. Jako konwerter można użyć typowej aplikacji układu MAX3232 z kondensatorami 100 nF.

Plik hex z kodem programu należy załadować do mikrokontrolera używając programu Flash Download Demonstrator dostępnego na stronie ST-Microelectronics [2]. Domyślne ustawienia portu szeregowego sugerowane przez aplikację są właściwe.

Jeżeli programowanie powiodło się, odłączyć zasilanie od układu, usunąć zworę SJ1 z pozycji H i zamontować ją w pozycji L (wyprowadzenie BOOT0 mikrokontrolera musi zostać podłączone poprzez rezystor R8 do masy). W tym ustawieniu mikrokontroler po resecie będzie wykonywać kod z pamięci FLASH.

Płytką drukowaną jest zamontowana w obudowie wtyku OBD2 przy pomocy trzech śrub M2.5 z łbem stożkowym i o długości 15mm. Każda ze śrub jest przymocowana do obudowy przykręconą z drugiej strony nakrętką. Kolejne dwie nakrętki mocują płytkę na odpowiedniej wysokości (równej wysokości połówki obudowy wtyku).

Przed ostatecznym zamontowaniem płytki w obudowie należy odznaczyć miejsca, w których pilnikiem zostaną wykonane otwory na kartę SD, diodę świecącą oraz przycisk.

Obsługa urządzenia

Do prawidłowej pracy VAGlogger potrzebuje pliku konfiguracyjnego. Należy przygotować kartę SD z systemem plików FAT16 lub FAT32. W katalogu głównym utworzyć plik konfiguracyjny o nazwie *config.txt*. Kolejne linie pliku to ustawienia dla kolejnych konfiguracji. Pojedyncza konfiguracja może zawierać od jednej do trzech numerów grup (rozdzielone średnikami).

Przykładowa zawartość pliku *config.txt*:

1;2;5

115;118

32;33

VAGlogger ma zegar czasu rzeczywistego podtrzymywany bateria litową. Czas oraz data są ustawiane na podstawie zawartości pliku *settime.txt* umieszczonego podobnie jak *config.txt* w katalogu głównym karty. Prawidłowo odczytany plik jest kasowany następnie z systemu plików.

Format pliku *settime.txt* jest następujący:

gg.mm.ss

dd.mm.rr

Plik z poniższą zawartością spowoduje ustawienie godziny 19:59:00 oraz daty na 14 września 2010 roku:

19:59:00

14.09.10

Kartę należy włożyć do urządzenia przed włączeniem zasilania, ponieważ jej inicjalizacja oraz odczyt plików konfiguracyjnych następuje tylko jednorazowo po resecie urządzenia.

Dla bezpieczeństwa elektroniki w samochodzie, układ należy podłączać przy wyłączonym zapłonie. Po uruchomieniu silnika należy wybrać długimi naciśnięciami przycisku aktywną konfigurację. Domyślnie urządzenie startuje w konfiguracji pierwszej (sygnalizowane ciągłym świeceniem się zielonej diody, w tym przypadku logowane są grupy podane w pierwszej linii pliku *config.txt*). Kolejne numery konfiguracji są sygnalizowane odpowiednio zapaleniem się diody na kolor czerwony lub pomarańczowy. Jednocześnie numer konfiguracji jest sygnalizowany odpowiednią liczbą krótkich sygnałów.

Krótkie naciśnięcie przycisku rozpoczyna proces łączenia się ze sterownikiem silnika. Jest on sygnalizowany szybkim miganiem zielonej diody LED oraz pojedynczym długim sygnałem dźwiękowym. Po prawidłowym nawiązaniu połączenia kolor diody zmienia się na kolor wybranej konfiguracji, tempo pulsowania diody zmniejsza się a użytkownik jest informowany akustycznie o numerze wybranej konfiguracji (odpowiednią liczbą krótkich sygnałów dźwiękowych). Rozpoczyna się proces odpytywania sterownika silnika o kolejne grupy pomiarowe i zapisywanie ich do pliku. Pojedyncze naciśnięcie przycisku powoduje rozłączenie ze sterownikiem i zamknięcie bieżącego pliku.

Brak odpowiedzi od sterownika, niewłaściwy numer protokołu lub zerwanie komunikacji w trakcie logowania jest sygnalizowane szybkim pulsowaniem diody czerwonej diody LED oraz dwoma długimi sygnałami dźwiękowymi. Po dwóch sekundach układ wraca do trybu wyboru konfiguracji.

Dowolny błąd związany z systemem plików zarówno podczas inicjalizacji urządzenia jak i podczas procesu logowania, taki jak: brak karty, brak pliku konfiguracyjnego lub jego

błędny format, błąd zapisu pliku np. z powodu przepełnienia karty, błąd komunikacji z kartą, jest sygnalizowany trzema długimi sygnałami dźwiękowymi oraz pulsowaniem diody na kolor pomarańczowy. Długie przytrzymanie przycisku powoduje reset urządzenia.

W konstrukcji loggera użyłem zwykłego gniazda SD bez wyrzutnika. Wstrząsy obecne w aucie mogą spowodować poluzowanie się karty w gnieździe i konsekwencji tego błąd opisany powyżej. Prosty rozwiązanie jest założenie gumki recepturki. Lepszym rozwiązaniem byłoby użycie gniazda SD typu push-pull. Niestety ich wadą są dużo większe wymiary.

Pliki tworzone przez VAGlogger mają trzycyfrową nazwę oraz rozszerzenie CSV (poczynając od *001.CSV*). Po uruchomieniu urządzenia katalog główny karty jest każdorazowo odczytywany w celu stwierdzenia numeru ostatniego zapisanego pliku – kolejny zapisywany plik będzie miał numer o jeden większy od największego znalezionej.

Podsumowanie

VAGlogger powstał z myślą o samochodach zbudowanych na bazie Golfa 4 (Seat Leon, Audi A3, Skoda Octavia). Protokół KW1281 występował także w innych pojazdach z tych lat, wobec czego jego funkcjonalność nie jest ograniczona tylko do tej rodziny.

Działanie urządzenia zostało przetestowane na następujących pojazdach:

- Audi A4 1.9TDI z 1996 roku
- Skoda Octavia 1.8T z 2005 roku
- Skoda Octavia 1.9TDI z 2000 roku
- Audi A3 1.6 z 2002 roku
- VW Polo 1.2 z 2003 roku

Jacek Greniger
jacek.greniger@gmail.com

Linki:

[1] http://elm-chan.org/fsw/fff/00index_e.html

[2] <http://www.st.com/mcu/devicedocs-STM32F103CB-110.html>

[3] http://www.blafusel.de/obd/obd2_kw1281.html

R E K L A M A

MIERZ Z MONUMENTALNĄ PRECYZJĄ

Sprawdź bogatą ofertę urządzeń pomiarowych na:
www.micros.com.pl

 **MICROS** Kraków, ul. E. Godlewskiego 38
tel. 12 636 95 66 fax 12 636 93 99
biuro@micros.com.pl

