



# multiTID – wielofunkcyjny, samochodowy komputer pokładowy (1)

Kilkakrotnie podejmowałem wyzwanie zaprojektowania komputera pokładowego i to zarówno takiego, który korzysta z oryginalnego, wbudowanego w deskę rozdzielczą wyświetlacza pokładowego, jak i własnego LCD, OLED czy TFT. Korzystając z doświadczenia, które zdobyłem projektując wspomniane urządzenia, postanowiłem zbudować komputer pokładowy będący syntezą wcześniej zastosowanych rozwiązań, lecz wyróżniający się największą i niespotykaną w innych rozwiązaniach funkcjonalnością. Miał to być komputer, który „tchnie” nowe życie w starsze pojazdy. MultiTID, czyli urządzenie, o którym mowa, ma możliwość współpracy z oryginalnymi radiodbiornikami montowanymi w pojazdach spod znaku Opla, a więc daje możliwość zastąpienia wbudowanego (dość archaicznego), trójfunkcyjnego wyświetlacza pokładowego nazywanego skrótem TID (Triple Info Display) przez niezwykle efektowny, duży i szerokoekranowy kolorowy wyświetlacz TFT o przekątnej 4.2”. Co więcej, ma możliwość pomiaru spalania i rejestracji przeróżnych statystyk w pojazdach wyposażonych w instalację zasilania gazem LPG i to niezależnie dla obu rodzajów paliwa.

**Rekomendacje:** prezentowany komputer pokładowy może zostać zamontowany w pojeździe dowolnej marki. W pojeździe innej marki po prostu nie będzie korzystał z funkcjonalności wyświetlania danych za pomocą wyświetlacza radiodbiornika.

Oto lista funkcji, w które jest wyposażony komputer pokładowy multiTID:

- Pełna funkcjonalność typowego, samochodowego komputera pokładowego.
- Możliwość emulacji wyświetlacza TID montowanego w pojazdach marki Opel (8- i 10-znakowego), a więc możliwość obsługi komunikatów wysyłanych przez oryginalne radiodbiorniki.
- „Obsługa” dwóch rodzajów paliwa (Pb/LPG),
- Czytelny, intuicyjny, graficzny interfejs użytkownika wzorowany na najnowszych rozwiązaniach z segmentu Premium.
- Możliwość regulacji jasności podświetlenia wyświetlacza graficznego zgodnie z ustawieniem jasności podświetlenia zegarów pojazdu.
- Łatwość instalacji uzyskana poprzez zastosowanie oryginalnego złącza połączeniowego wyświetlacza TID (oraz złącz dodatkowych dla funkcji spalania).

Ponadto, postawiłem dość wysokie wymagania dotyczące funkcjonalności komputera pokładowego, którego zdecydowałem się wyposażyć w następujące funkcje:

- Pokazywanie temperatury na zewnątrz pojazdu (z zastosowaniem czujnika montowanego w pojazdach marki Opel) oraz ostrzeżenia o śliskiej nawierzchni (dla temperatury zewnętrznej niższej, niż 5°C).
- Pokazywanie chwilowej prędkości pojazdu (w km/godz.).
- Pokazywanie średniej prędkości pojazdu na przejechanym odcinku drogi (w km/godz.),
- Pokazywanie maksymalnej prędkości pojazdu na przejechanym odcinku drogi (w km/godz.),
- Pokazywanie chwilowego zużycia paliwa (w l/godz. dla prędkości ≤5 km/godz. oraz l/100 km dla pozostałych prędkości) dla obu rodzajów paliwa.
- Pokazywanie średniego zużycia paliwa (w l/100 km) dla obu rodzajów paliwa.
- Pokazywanie paliwa pozostającego w baku pojazdu (w l, jak i graficznie – bargraf) dla obu rodzajów paliwa.
- Pokazywanie przewidywanego zasięgu pojazdu na paliwie pozostającym w baku pojazdu (w km) dla obu rodzajów paliwa.
- Pokazywanie przejechanego dystansu od ostatniego kasowania (w km).
- Pokazywanie aktualnego czasu i daty (z zastosowaniem mechanizmu podtrzymania zasilania).
- Pokazywanie statystyk spalania na każde przejechane 10 km/1 km (dla ostatnio przejechanych 150 i 15 km) dla obu rodzajów paliwa.
- Automatyczna detekcja bieżącego rodzaju paliwa.

Jak widać, cel jest dość ambitny, jednak doświadczenia zdobyte przy konstruowaniu poprzednich urządzeń upraszczają znacząco proces jego implementacji. Zanim jednak przejdziemy do opisu samego urządzenia warto przybliżyć nieco temat pokładowego wyświetlacza TID, gdyż możliwość jego

emulacji jest jedną z głównych zalet i zarazem unikalnych cech urządzenia multiTID, bo umożliwia współpracę sterownika z oryginalnymi radioodbiornikami dedykowanymi do pojazdów spod znaku Opla. Co więcej, sterownik nasz obsługuje wszystkie rodzaje radioodbiorników montowanych przez tego producenta, które to wyposażono w magistralę sterującą z sygnałami SDA/SCL/MRQ i może emulować wyświetlacze TID z 8-znakową i 10-znakową organizacją pola tekstowego, a więc wyświetlacze pokładowe montowane w takich modelach jak Astra F, Corsa B, Combo, Astra G, Corsa C, Meriva, Movano, Omega B, Vectra B, Tigra. Wyjątkiem są nowsze pojazdy wyposażone w wyświetlacz pokładowy obsługujący magistralę CAN. Ta cecha naszego sterownika czyni go wyjątkowo atrakcyjnym, z punktu widzenia użytkownika pojazdu tej marki i co warto podkreślić, nie znajduje odpowiednika ani w konstrukcjach amatorskich, ani też na rynku komercyjnym.

Zanim przejdziemy do szczegółów implementacyjnych warto przyjrzeć się rozwiązaniu zastosowanemu przez Opla, gdyż jest to przykład dobrze wykonanej pracy inżynierskiej i dogłębnie przemyślanej konstrukcji. Zgodnie z tym, o czym wspomniano wcześniej, firma Opel od wielu lat stosuje w swoich pojazdach zewnętrzne, zintegrowane z deską rozdzielczą, podświetlane wyświetlacze LCD, których funkcje zależne są od modelu auta, jego wyposażenia, roku produkcji jak i modelu samego wyświetlacza. Ogólnie rzecz ujmując można wyróżnić kilka typów wyświetlaczy, których poglądowe porównanie przedstawiono w **tabeli 1**.

Poza wymienionymi, podstawowymi różnicami, w zależności od roku produkcji i wyposażenia pojazdu, stosowano wyświetlacze o różnej organizacji i rozdzielczości. Starsze miały organizację 1×8 znaków (tylko wielkie litery

**Tabela 1. Porównanie funkcjonalności wyświetlaczy pokładowych stosowanych przez firmę Opel**

Nazwa	Funkcje	Typ magistrali sterującej
DID	Data, godzina, obsługa radioodbiornika	Zmodyfikowana I <sup>2</sup> C
TID	Funkcje DID + temperatura zewnętrzna	
MID	Funkcje TID + obsługa komputera pokładowego	
NAVI	Funkcje MID + obsługa nawigacji satelitarnej	CAN

**DODATKOWE MATERIAŁY NA FTP:**

<ftp://ep.com.pl>

**USER: 66532, PASS: 8nnjjeaa**

**W ofercie AVT\***

**AVT-5562**

**Podstawowe informacje:**

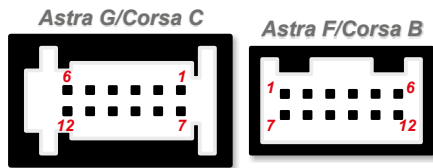
- Napięcie zasilania: 8...15 V DC.
- Maksymalny prąd obciążenia (z napięcia +12 V): 10 mA.
- Prąd podtrzymania zegara RTC (z napięcia BATT): 1 mA.
- Maksymalny prąd podświetlenia (z napięcia ILL+): 75 mA.
- Dokładność pomiaru temperatury: 1°C.
- Zakres pomiarowy temperatury zewnętrznej: -30...35°C.
- Zakres pomiarowy prędkości pojazdu: 0...255 km/godz.
- Zakres pomiarowy chwilowego zużycia paliwa: 0...99,9 l/100 km.
- Zakres pomiarowy średniego zużycia paliwa: 0...25,5 l/100 km.
- Zakres pomiarowy paliwa dostępnego w baku: 0...99,9 l.
- Zakres pomiarowy przejechanej odległości: 0...9999 km.
- Zakres pomiarowy dystansu do przejechania na dostępnym paliwie: 0...999 km.
- Zakresy regulacji parametrów konfiguracyjnych:
  - Stała wtryskiwaczy: 1...999 ml/min.
  - Stała przetwornika drogi: 1...99 impulsów/obrót.
  - Obwód opony: 50...255 cm.
  - Liczba cylindrów: 2...8.
  - Pojemności baków: 25...99 l.
- Przesunięcie belek informacyjnych: 0:9 pikseli.

**Projekty pokrewne na FTP:**

(wymienione artykuły są w całości dostępne na FTP)

AVT-5545	Komputer samochodowy Mee MK II (2.0) (EP 7-8-9/2016)
AVT-5495	Uniwersalny komputer samochodowy Mee (EP 3/2015)
AVT-3095	Komputer samochodowy (EdW 4-5/2014)
AVT-5405	TripCo – komputer samochodowy (EP 7/2013)
AVT-5395	TIDex – komputer dla samochodów z silnikiem Diesla (EP 5/2013)
AVT-5397	Komputer pokładowy z funkcją tempomatu (EP 5/2013)
AVT-1664	Transceiver CAN (EP 2/2012)
AVT-5280	Urządzenie diagnostyczne do sieci CAN (EP 3/2011)
AVT-5271	VAGlogger – Przyrząd diagnostyczny dla samochodów z grupy VW – Audi (EP 1/2011)
AVT-5160	Climatic – sterownik klimatyzacji samochodowej (EP 12/2008)
AVT-286	„Komputerek” pokładowy do samochodu (EP 5-6/1996)

\* Uwaga:  
Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx AT płytka drukowana i zaprogramowany układ (czyli połączenie wersji AVT wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf.  
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlotowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf.  
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu).  
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>



**Rysunek 1. Rozmieszczenie wyprowadzeń dwóch złącz wyświetlacza TID stosowanych w najbardziej popularnych modelach Opla**

i cyfry, wyświetlacz alfanumeryczny), które były dostępne dla radioodbiornika plus dodatkowe piktogramy obrazujące tryb pracy radia. Nowsze pracują w organizacji 1×10 znaków (plus piktogramy, pełne ASCII, wyświetlacz mozaikowy 5×7 pikseli) w dolnym wierszu, zaś górny wiersz zarezerwowano dla wbudowanego systemu mikroprocesorowego (zegar, termometr). Dodatkowo, przewidziano możliwość synchronizacji wbudowanego weń zegara czasu rzeczywistego sygnałem RDS radioodbiornika. Niestety, od roku 2005/2006 firma Opel zdecydowała się (wzorem innych producentów w branży motoryzacyjnej) na zastosowanie magistrali CAN we wszystkich nowych pojazdach osobowych, co pociągnęło potrzebę implementacji tego interfejsu także w wyświetlaczu LCD. Niemniej jednak, w zdecydowanej większości popularnych modeli pojazdów tej marki znajdziemy wyświetlacze, które z powodzeniem mogą być zastąpione przez opisywane urządzenie. Oczywiście, w zależności od typu pojazdu i rodzaju zastosowanego wyświetlacza, stosowano różne złącza. Na **rysunku 1** zamieszczono rozkład wyprowadzeń dwóch złącz wyświetlacza TID stosowanych w najbardziej popularnych modelach Opla, tj. Astra F/G i Corsa B/C. W **tabeli 2** umieszczono opis wyprowadzeń tych złącz. Paleta stosowanych złącz jest znacznie szersza, w związku z czym zainteresowanych czytelników odsyłam na bardzo ciekawą stronę poświęconą tematyce car-audio, znajdującą się pod adresem <https://goo.gl/NaAfwS>.

Wyświetlacz jest systemem mikroprocesorowym wyposażonym w funkcje dodatkowe (oprócz możliwości wyświetlania danych z magistrali), takie jak: wbudowany kalendarz, zegar czy też termometr. Kolejnym ciekawym zagadnieniem jest rodzaj magistrali sterującej.

**Tabela 2. Opis wyprowadzeń dwóch złącz wyświetlacza TID stosowanych w najbardziej popularnych modelach Opla**

Astra G/Corsa C	
Nr pinu	Funkcja/Opis
1	Zasilanie: 12V po przekręceniu kluczyka stacyjki
2	AA – 12V po włączeniu radioodbiornika
3	Zasilanie: 12V z akumulatora
4	Regulacja jasności podświetlenia wyświetlacza (sygnał PWM)
5	NTC – wyprowadzenie do podłączenia czujnika temperatury zewnętrznej
6	Masa
7	NTC – wyprowadzenie do podłączenia czujnika temperatury zewnętrznej
8	Wyprowadzenie testowe – do sprzętu diagnostycznego
9	WEG – sygnał prędkości pojazdu (korekta wskazań termometru)
10	SCL - sygnał sterujący magistrali – Serial Clock
11	SDA - sygnał sterujący magistrali – Serial Data
12	MRQ - sygnał sterujący magistrali – Master Request
Astra F/Corsa B/Tigra	
Nr pinu	Funkcja/Opis
1	Zasilanie: 12V z akumulatora
2	NTC – wyprowadzenie do podłączenia czujnika temperatury zewnętrznej
3	Masa
4	NTC – wyprowadzenie do podłączenia czujnika temperatury zewnętrznej
5	Zasilanie: 12V po przekręceniu kluczyka stacyjki
6	Oświetlenie
7	Regulacja jasności podświetlenia wyświetlacza (sygnał PWM)
8	AA – 12V po włączeniu radioodbiornika
9	SCL - sygnał sterujący magistrali – Serial Clock
10	MRQ - sygnał sterujący magistrali – Master Request
11	SDA - sygnał sterujący magistrali – Serial Data
12	WEG – sygnał prędkości pojazdu (korekta wskazań termometru)

Jest ona bardzo zbliżona funkcjonalnością do dobrze znanej magistrali I<sup>2</sup>C, lecz poszerzona o dodatkowy sygnał sterujący – MRQ (Master Request). Na tę magistralę składają się 4 linie oznaczonych: AA, MRQ, SDA i SCL, przy czym wyłącznie trzy ostatnie realizują rzeczywistą transmisję danych, zaś linia AA sygnalizuje zdarzenie załączenia/wyłączenia radioodbiornika przygotowując tym samym wbudowany wyświetlacz do odbierania danych. Po wyłączeniu radioodbiornika napięcie na tej linii wynosi 0 V, zaś po jego załączeniu 12 V. Linia MRQ jest używana przez układ master (radioodbiornik) do sprawdzenia obecności układu slave (wyświetlacza TID) na magistrali danych oraz do rozróżnienia rodzaju przesyłanych danych, o czym później. Zanim jednak radioodbiornik rozpocznie jakąkolwiek transmisję danych sprawdza, tuż po włączeniu jego zasilania, stan wszystkich linii transmisyjnych wykonując tzw. *Power On Test*, dla którego przebiegi sygnałów

sterujących pokazano na **rysunku 2**, a którego celem jest sprawdzenie ciągłości wszystkich linii danych jak i wykłuczenie potencjalnych zwarc.

Tuż po włączeniu zasilania radioodbiornika, któremu to towarzyszy zmiana poziomu napięcia występującego na linii AA z 0 V na 12 V, następuje po czasie T1 (100...500 ms) ściągnięcie wszystkich linii danych (MRQ, SDA i SCL) do masy na czas T2 (500...1000 μs), następnie zwolnienie ich (także na czas T2) oraz naprzemienne ściągnięcie do masy kolejnych linii SDA, SCL i na końcu MRQ. Taka sekwencja zdarzeń pozwala, co napisano wcześniej, na zbadanie stanu wszystkich linii sterujących i zakończenie transmisji w przypadku wykrycia potencjalnych problemów. Jeśli powyższa procedura zakończy się powodzeniem (brak zwarc i nieciągłości linii danych), master przechodzi do właściwej transmisji danych, dla której to przebiegi sygnałów sterujących pokazano na **rysunku 3**.

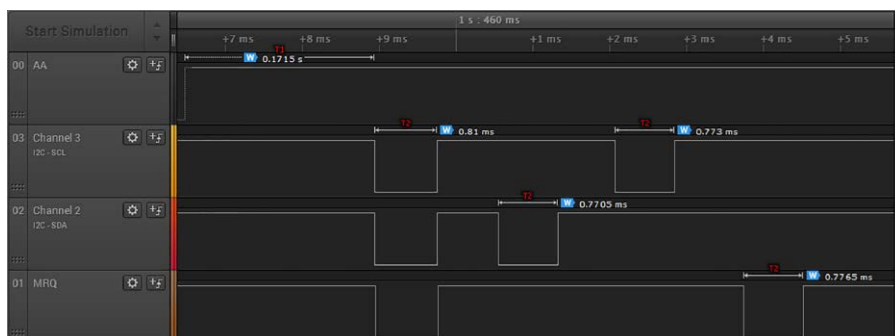


Po czasie T3 (1...2 ms) od ostatniego zwolnienia linii MRQ będącego jednocześnie końcowym elementem procedury *Power On Test*, układ Master ponownie zeruje linię MRQ, tym razem jednak z zupełnie innego powodu. Po pierwsze, sygnalizuje w ten sposób zamiar transmisji danych, a po drugie, sprawdza obecność slave na magistrali danych oraz jego gotowość na odbieranie pakietu danych, gdyż działanie takie podejmowane jest każdorazowo na początku każdej transmisji z radioodbiornika do wyświetlacza TID. Po wymuszeniu poziomu niskiego na linii MRQ, slave musi w czasie T4 (100  $\mu$ s...15 ms) zewrzeć linię SDA do masy sygnalizując w ten sposób swoją obecność, jak i gotowość do odbierania danych. Jeśli

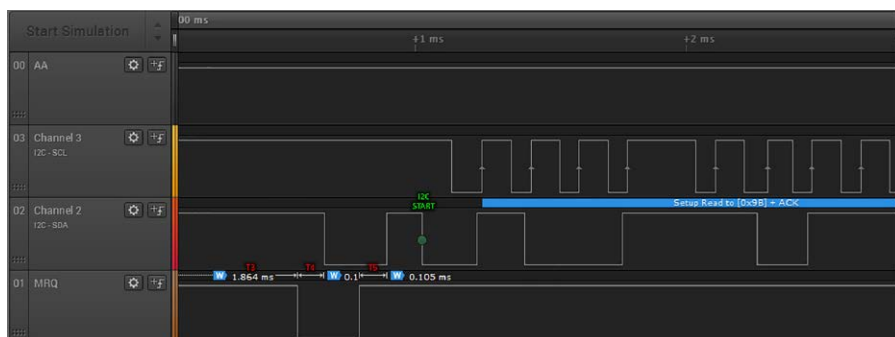
tego nie zrobi, master powinien zakończyć w tym miejscu bieżącą transmisję danych. Dalej, wyzerowaniu linii SDA przez slave, master zwalnia linię MRQ (zostaje ona ustawiona), co powinno skutkować zwolnieniem linii danych SDA przez slave po czasie T5 (100...200  $\mu$ s). Na tym etapie rozpoczyna się właściwa transmisja danych zgodna ze specyfikacją standardu I<sup>2</sup>C, z jednym, drobnym, acz istotnym wyjątkiem, o którym mowa będzie w dalszej części artykułu.

Wygląd kompletnej ramki danych pokazano na **rysunku 4**. Transmisja rozpoczyna się od wygenerowania sekwencji *Start* przez master'a, po czym jako pierwszy bajt transmitowany jest adres slave o wartości 0x4D (0x9B w notacji

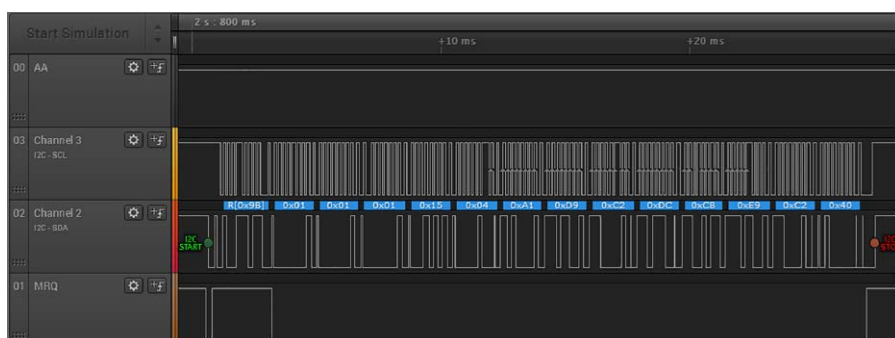
8-bitowej z ustawionym bitem R/W). Właśnie w tym miejscu jest widoczna subtelna różnica pomiędzy standardowym interfejsem zgodnym ze standardem I<sup>2</sup>C, a rozwiązaniem zastosowanym przez inżynierów Opla. Skoro transmitowany bajt adresu ma ustawiony najmniej znaczący bit będący jednocześnie bitem R/W (odczyt/zapis) w standardzie I<sup>2</sup>C powinno to oznaczać, że master (radioodbiornik) zgłasza chęć odczytania danych ze slave (wyświetlacza TID) i kolejne dane, które zostaną przesłane będą danymi przesyłanymi ze slave do master, a jest zgoła inaczej, ponieważ master kontynuuje w tym miejscu transmisję danych do slave, tak jakby najstarszy bit przesłanego wcześniej adresu był wyzerowany. Z czego wynika ta drobna, acz znacząca różnica? Odpowiedź jest prosta. Z chęci maksymalnego zabezpieczenia się przed możliwością wystąpienia zaburzeń transmisji i będących tego następstwem, niepoprawnych danych, o co, jak łatwo się domyślić, nietrudno w tak niesprzyjającym środowisku, jak instalacja samochodowa. Otóż, zgodnie z pomysłem inżynierów Opla, każdy bajt danych składa się z 7 bitów właściwych danych oraz bitu kontroli parzystości umieszczonego na pozycji najmniej znaczącej. I niestety, dotyczy to wszystkich bajtów danych, w tym adresu urządzenia, co powoduje, że dla adresu 0x4D (w notacji 7-bitowej) otrzymujemy bit kontroli parzystości o wartości „1”, czyli ustawiony bit R/W. Różnica jest drobna, lecz niesie za sobą poważne konsekwencje implementacyjne. Otóż, w przypadku takiej konstrukcji ramki danych, do obsługi transmisji danych nie możemy użyć wygodnego, sprzętowego interfejsu TWI (odpowiednik I<sup>2</sup>C)



**Rysunek 2.** Przebiegi sygnałów sterujących magistrali danych w trakcie sekwencji *Power On Test*



**Rysunek 3.** Przebiegi sygnałów sterujących magistrali danych po wykonaniu sekwencji *Power On Test*



**Rysunek 4.** Kompletna ramka danych

REKLAMA

Projekty na...Texas

STM32

www.stm32.eu

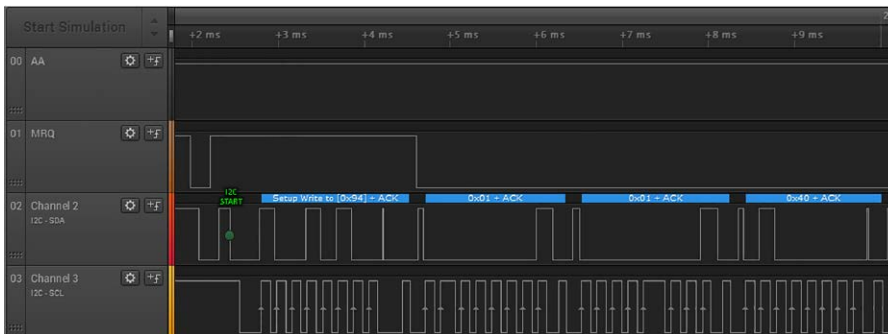
ST life.augmented

KAMAMI

mikrokontrolera ATmega644PA, którego zastosowanie znacznie uprościłoby mechanizmy emulujące wyświetlacz TID. Zwyczajnie, i co oczywiste, sprzęg TWI nie przewiduje sytuacji, w której to master wysyła żądanie odczytu do slave (wysyła jego adres z ustawionym bitem R/W), po czym najzwyczajniej w świecie kontynuuje transmisję danych w kierunku do slave. Takiego scenariusza transmisji w module TWI (i stosownych statusów jego stanu w rejestrze TWSR mikrokontrolera) po prostu nie przewidziano, gdyż jest on zgodny ze standardem I<sup>2</sup>C. Czy to stanowi jakiś problem? W zasadzie nie, lecz z pewnością skomplikuje mechanizm obsługi interfejsu Opla, gdyż do obsługi transmisji I<sup>2</sup>C będziemy musieli zaprzęgnąć inne zasoby sprzętowe mikrokontrolera, w tym przypadku przerwania zewnętrzne oraz zaimplementować niezbędne procedury programowe, co na pewno przyczyni się do większego obciążenia rdzenia.

W tym miejscu należy wspomnieć o jeszcze jednej funkcji linii MRQ, którą można zaobserwować analizując przebiegi kompletnej ramki danych z rys. 4. Otóż, linia danych MRQ jest wyzerowana w trakcie przesyłania przez radioodbiornik wszystkich bajtów danych za wyjątkiem bajta adresu, co w prosty sposób odróżnia ten bajt od pozostałych bajtów danych. Takie rozwiązanie stanowi zapewne kolejny element poprawy integralności przesyłanej ramki danych przyczyniający się do zwiększenia marginesu bezpieczeństwa, bo identyfikacja bajta adresu mogłaby być wykonana przecież w prostszy sposób, choćby przez fakt, że jest to zawsze pierwszy, przesyłany bajt danych. Niemniej jednak, rozwiązanie Opla z pewnością należy uznać za bardzo przemyślane. Transmisję właściwych danych, jak to zwykle ma miejsce w standardzie I<sup>2</sup>C, kończy sygnał *stop* wygenerowany przez master.

Jak do tej pory nie powiedziałem nic na temat samych danych, które to radioodbiornik wysyła do wyświetlacza pokładowego TID. Dane te to 13 bajtów, z których pierwsze trzy bajty odpowiadają za wyświetlanie piktogramów pokazywanych na ekranie wyświetlacza TID, zaś kolejne 10 bajtów to kody znaków ASCII przeznaczonych do wyświetlenia w polu tekstowym, przy czym



Rysunek 5. Wygląd ramki danych wyświetlacza TID w wypadku nieobsługiwanego adresu I<sup>2</sup>C

Tabela 3. Znaczenie poszczególnych bitów trzech pierwszych bajtów danych wyświetlacza TID

	Bajt 1 (status radia)	Bajt 2 (status magnetofonu)	Bajt 3 (status CD)
Bit 7	przecinek	„CD-In”	0
Bit 6	„RDS”	„Dolby C”	Symbol „Track”
Bit 5	„TP”	„Dolby B”	„RDM”
Bit 4	Symbol „Stereo”	„Cr”	„PGM”
Bit 3	0	„CPS”	„DISC”
Bit 2	„AS”	0	0
Bit 1	Nawias kwadratowy dla „TP”	0	0
Bit 0	Bit parzystości	Bit parzystości	Bit parzystości

obsługiwanych jest kilka znaków specjalnych o kodach poniżej wartości 32 (spacji). W tabeli 3 opisano znaczenie poszczególnych bitów trzech pierwszych bajtów danych odpowiedzialnych za wyświetlanie specjalnych piktogramów na ekranie wyświetlacza TID. W tabeli 4 umieszczono kody ASCII i opis znaków specjalnych wyświetlanych przez oryginalny wyświetlacz TID a obsługiwane przez sterownik multiTID. Wcześniej opis dotyczy wyświetlaczy o organizacji 10-znakowej. Dla starszych wyświetlaczy, o organizacji 8-znakowej, jest przesyłanych wyłącznie 10 bajtów danych, z czego 2 pierwsze bajty odpowiadają za wyświetlanie piktogramów pokazywanych na ekranie wyświetlacza TID (tylko status radia i magnetofonu), zaś kolejne 8 bajtów to kody znaków ASCII przeznaczonych do wyświetlenia w polu tekstowym wyświetlacza (tylko wielkie litery i cyfry, bez znaków specjalnych). Inny jest też adres samego wyświetlacza, który przyjmuje wtedy wartość 0x4A. Już tylko dla porządku dodam, że każdy bajt danych przesłany przez master do slave jest potwierdzany przez układ podrzędny (TID) poprzez wygenerowanie sygnału ACK (wyzerowanie

Tabela 4. Kody ASCII znaków specjalnych wyświetlanych przez oryginalny, 10-znakowy wyświetlacz TID

Kod ASCII	Znak
1	Pomniejszona cyfra 1
2	Pomniejszona cyfra 2
3	Pomniejszona cyfra 3
4	Pomniejszona cyfra 4
5	Pomniejszona cyfra 5
6	Pomniejszona cyfra 6
8	Pomniejszona, wielka litera A
9	Pomniejszona, wielka litera F
10	Pomniejszona, wielka litera U
11	Pomniejszona, wielka litera M
12	Pomniejszona, wielka litera L
13	Pomniejszona, wielka litera P
14	Symbol „nutki”

linii SDA) w dziewiątym takcie zegara magistrali (SCL) i to niezależnie czy przesyłane dane dotyczą naszego urządzenia (przesłano wcześniej zgodny adres I<sup>2</sup>C), czy też nie, co również wynika się standardom wyznaczonym przez specyfikację interfejsu I<sup>2</sup>C. Sytuację taką przedstawiono na rysunku 5, który to prezentuje przebiegi na magistrali danych zarejestrowane podczas współpracy radioodbiornika CAR300

Listing 1. Treść pliku nagłówkowego emulatora wyświetlacza TID

```

//Zmienne globalne modułu TIDemulator
extern volatile uint8_t TIDdataReady; //Flaga gotowości danych
extern volatile uint8_t TIDdata[14]; //Dane (element 0 to adres I2C)

//Prototypy funkcji
void initTIDemulator(uint8_t TID_type);
//Definicje portów TID'a
#define I2C_SDA_DDR DDRC
#define I2C_SDA_PIN PINC
#define I2C_SDA_NR PC1 //ISR: PCINT17
#define I2C_SCL_PIN PINB
#define I2C_SCL_NR PB2 //ISR: INT2
#define MRQ_PIN PIND
#define MRQ_NR PD3 //ISR: INT1
#define AA_PIN PIND
#define AA_NR PD2 //ISR: INT0
//Adres wyświetlacza TID
#define TID_TYPE_8DIGITS 0x4A
#define TID_TYPE_10DIGITS 0x4D
#define TID_TIMESTAMP 0x47
//Stany pracy interfejsu TID
#define RADIO_IS_OFF 0x00
#define WAITING_FOR_FIRST_MRQ 0x01
#define FIRST_MRQ_STARTED 0x02
#define WAITING_FOR_SECOND_MRQ 0x03
#define SECOND_MRQ_STARTED 0x04
#define WAITING_FOR_MRQ_BEFORE_TRANSMISSION 0x05
#define START_DETECTED 0x06
//Definicje czasów dla Preskalera Timeral = 256 i fosc = 12.288MHz [w taktach timera]
#define TIME_TOLERANCE 14 //300us
#define T2_MIN (24 - TIME_TOLERANCE) //500us - TIME_TOLERANCE
#define T2_MAX (48 + TIME_TOLERANCE) //1000us + TIME_TOLERANCE
//Średnie wartości pozostałych czasów [us]
#define T4_MID 120 //120us
#define T5_MID 120 //120us
//Makra dla portów
#define SDA_SET I2C_SDA_DDR &= ~(1<<I2C_SDA_NR) //Ustawiamy, jako wejście podciągnięte przez zewnętrzny rezystor
#define SDA_RESET I2C_SDA_DDR |= (1<<I2C_SDA_NR) //Ustawiamy, jako wyjście z domyślnym stanem „0”
#define SDA_READ (I2C_SDA_PIN & (1<<I2C_SDA_NR))>>I2C_SDA_NR //Wartość: 0x00 lub 0x01
#define SDA_IS_RESET (!(I2C_SDA_PIN & (1<<I2C_SDA_NR)))
#define SDA_IS_SET (I2C_SDA_PIN & (1<<I2C_SDA_NR))
#define SCL_IS_RESET (!(I2C_SCL_PIN & (1<<I2C_SCL_NR)))
#define SCL_IS_SET (I2C_SCL_PIN & (1<<I2C_SCL_NR))
#define MRQ_IS_RESET (!(MRQ_PIN & (1<<MRQ_NR)))
#define MRQ_IS_SET (MRQ_PIN & (1<<MRQ_NR))
#define AA_IS_SET (AA_PIN & (1<<AA_NR))
//Makra dla przerwań
#define SDA_INTR_INIT_ON_BOTH_EDGES PCMSK2 = (1<<PCINT17) //Zmiana stanu na PCINT17 (sygnał SDA) generuje przerwanie
#define SDA_INTR_ENABLE FCICR |= (1<<PCIE2) //Zezwolenie na przerwanie PCINT2 (PCINT23..16)
#define SDA_INTR_DISABLE PCICR &= ~(1<<PCIE2) //Zablokowanie przerwania PCINT2 (PCINT23..16)
#define SDA_INTR_CLEAR_FLAG PCIFR |= (1<<PCIF2) //Skasowanie flagi przerwania PCINT2 (PCINT23..16)
#define SDA_INTR_NAME PCINT2_vect //Nazwa wektora przerwania PCINT2 (SDA)
#define SCL_INTR_INIT_ON_RISING_EDGE EICRA |= (1<<ISC21)|(1<<ISC20) //Rosnące zbocze sygnału na INT2 (sygnał SCL) generuje przerwanie
#define SCL_INTR_ENABLE EIMSK |= (1<<INT2) //Zezwolenie na przerwanie INT2
#define SCL_INTR_DISABLE EIMSK &= ~(1<<INT2) //Zablokowanie przerwania INT2
#define SCL_INTR_CLEAR_FLAG EIFR |= (1<<INTF2) //Skasowanie flagi przerwania INT2
#define SCL_INTR_NAME INT2_vect //Nazwa wektora przerwania INT2 (SCL)
#define MRQ_INTR_INIT_ON_BOTH_EDGES EICRA |= (1<<ISC10) //Dowolne zbocze sygnału na INT1 (sygnał MRQ) generuje przerwanie
#define MRQ_INTR_ENABLE EIMSK |= (1<<INT1) //Zezwolenie na przerwanie INT1
#define MRQ_INTR_DISABLE EIMSK &= ~(1<<INT1) //Zablokowanie przerwania INT1
#define MRQ_INTR_CLEAR_FLAG EIFR |= (1<<INTF1) //Skasowanie flagi przerwania INT1
#define MRQ_INTR_NAME INT1_vect //Nazwa wektora przerwania INT1 (MRQ)
#define AA_INTR_INIT_ON_BOTH_EDGES EICRA |= (1<<ISC00) //Dowolne zbocze sygnału na INT0 (sygnał AA) generuje przerwanie
#define AA_INTR_ENABLE EIMSK |= (1<<INT0) //Zezwolenie na przerwanie INT0
#define AA_INTR_DISABLE EIMSK &= ~(1<<INT0) //Zablokowanie przerwania INT0
#define AA_INTR_CLEAR_FLAG EIFR |= (1<<INTF0) //Skasowanie flagi przerwania INT0
#define AA_INTR_NAME INT0_vect //Nazwa wektora przerwania INT0 (AA)

```

z nowym, 10-znakowym wyświetlaczem pokładowym TID.

Uważny Czytelnik zastanowi się z pewnością, w jakim celu pokładowy wyświetlacz TID miałby potwierdzać każdy bajt danych, które nie są do niego adresowane, to znaczy nie zostały „okraszone” stosownym adresem układu podrzędnego (w naszym przypadku 0x4D). To kolejny przykład przemyślanej konstrukcji Opla. Otóż wiele fabrycznych radiodbiorników może pracować tak w starszych, jak i nowszych modelach pojazdów tego producenta, które to mogą być wyposażone w różne rodzaje wyświetlaczy TID. Aby umożliwić współpracę radiodbiornika z nowymi i starszymi typami wyświetlaczy, które to przecież charakteryzują się inną organizacją ekranu i odrębnym adresem w przestrzeni I<sup>2</sup>C, wprowadzono zasadę,

iż radiodbiornik wysyła pakiety danych dla starego i nowego typu wyświetlaczy, co oczywiście, stosownie je modyfikując. W związku z tym, każdy przesyłany komunikat transmitowany jest w dwóch „wersjach”, dla starego i nowego typu wyświetlacza, stąd różne adresy i różna liczba towarzyszących im danych na jednej i tej samej magistrali. Nie tłumaczy to oczywiście faktu potwierdzania (sygnałem ACK) przez wyświetlacz TID nie swojego adresu I<sup>2</sup>C, jak i danych nie dla niego przeznaczonych, lecz myślę, że wynika to z potrzeby ciągłego „informowania” (przy użyciu sygnału ACK) radiodbiornika o nasłuchu magistrali I<sup>2</sup>C przez układ podrzędny. Jak widać, zastosowana przez Opla magistrala, na pozór bardzo podobna do rozwiązania Philips’a (I<sup>2</sup>C), ma wiele drobnych, acz znaczących różnic, przez

które trudno byłoby ją „obsłużyć” korzystając ze sprzętowego sprzęgu TWI.

Na koniec prawdziwa „wisienka na torcie”. Jak wiadomo, oryginalny wyświetlacz TID wyposażono w zegar czasu rzeczywistego, którego wskazania są wyświetlane w pierwszej linii ekranu (obok

REKLAMA



Projekty na...  
**STM32**

[www.stm32.eu](http://www.stm32.eu)

life.augmented

**KAMAMI**



temperatury zewnętrznej). W związku z tym producent modułu wyświetlacza przewidział możliwość synchronizacji wskazań tego zegara sygnałem czasu nadawanym przez stacje radiowe, a zawartym w treści wybranych komunikatów RDS. W jaki sposób wyświetlacz TID synchronizuje swój zegar RTC sygnałem czasu radiodbiornika? Wprowadzono dodatkowy adres I<sup>2</sup>C wyświetlacza TID, dla którego przesyłane dane są interpretowane przez ten moduł jako dane bieżącego czasu sygnału RDS. Adres ten to 0x47 (0x8F przy ustawionym bicie R/W), natomiast odpowiednia ramka danych składa się z 7 bajtów o następującej organizacji: X-M-G-D-X-X-X, gdzie: M to minuty, G to godziny, D to dzień miesiąca. Bajty X nie zostały przez mnie jednoznacznie zidentyfikowane, choć z całą pewnością ich obecność nie jest przypadkowa. Jeśli bajty te przechowują bieżący miesiąc i rok, jak się zapewne domyślicie, czas oczekiwania na potwierdzenie tej tezy byłby dość długi. Wygląd ramki danych znacznika

czasu wyświetlacza TID zarejestrowanej 17/08/2016 o godzinie 17:49 przedstawiono na **rysunku 6**. Jak poprzednio, bit najmniej znaczący jest bitem kontroli parzystości, w związku z czym cały bajt należy przesunąć w prawo o jedno miejsce).

To tyle, jeśli chodzi o opis modułu wyświetlacza pokładowego. Przejdźmy zatem do szczegółów implementacyjnych. Jako, że nasze urządzenie multiTID realizuje szereg dość skomplikowanych funkcjonalności, które wymagają ścisłych

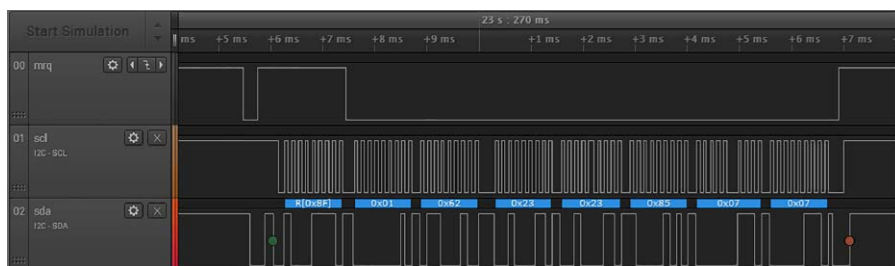
zależności czasowych i zaangażowania wielu peryferiów mikrokontrolera, obsługa emulacji wyświetlacza TID (czyli obsługa radiodbiornika) została zrealizowana z wykorzystaniem 4 zewnętrznych przerwań systemowych. Taka mnogość wykorzystywanych przerwań wynika z faktu obsługi aż 4 sygnałów sterujących (AA, MRQ, SCL, SDA) przy braku możliwości realizacji sprzętowej obsługi magistrali I<sup>2</sup>C. W **tabeli 5** pokazano zestawienie wykorzystywanych, zewnętrznych

```
Listing 2. Funkcja ISR odpowiedzialna za obsługę sygnału AA interfejsu danych wyświetlacza TID
//Przerwanie odpowiedzialne za obsługę sygnału AA - wyzwalane po każdym zboczach sygnału na AA
ISR(AA_INTR_NAME)
{
    if(AA_IS_SET) //Włączenie radiodbiornika
    {
        Status = WAITING_FOR_FIRST_MRQ;
        MRQ_INTR_INIT_ON_BOTH_EDGES; //Konfiguracja i włączenie przerwania dla MRQ
        MRQ_INTR_CLEAR_FLAG;
        MRQ_INTR_ENABLE;
    }
    else //Wyłączenie radiodbiornika
    {
        Status = RADIO_IS_OFF; //Startowy stan algorytmu
        SDA_SET; //Zwolnienie SDA, gdyby było ściągane przez Slave'a
        MRQ_INTR_DISABLE;
        SDA_INTR_DISABLE;
        SCL_INTR_DISABLE;
    }
}
```

```
Listing 3. Funkcja ISR odpowiedzialna za obsługę sygnału MRQ interfejsu danych wyświetlacza TID
//Przerwanie odpowiedzialne za obsługę sygnału MRQ - wyzwalane po każdym zboczach sygnału na MRQ
ISR(MRQ_INTR_NAME)
{
    register uint16_t pulseLength;
    static uint16_t lastTimer1;
    switch(Status)
    {
        case WAITING_FOR_FIRST_MRQ:
            if(SDA_IS_RESET && SCL_IS_RESET && MRQ_IS_RESET) Status = FIRST_MRQ_STARTED; lastTimer1 = TCNT1;
            else Status = WAITING_FOR_FIRST_MRQ;
            break;
        case FIRST_MRQ_STARTED:
            pulseLength = TCNT1 - lastTimer1;
            if(SDA_IS_SET && SCL_IS_SET && MRQ_IS_SET && pulseLength>T2_MIN && pulseLength<T2_MAX) Status = WAITING_FOR_SECOND_MRQ;
            else Status = WAITING_FOR_FIRST_MRQ;
            break;
        case WAITING_FOR_SECOND_MRQ:
            if(SDA_IS_SET && SCL_IS_SET && MRQ_IS_RESET) Status = SECOND_MRQ_STARTED; lastTimer1 = TCNT1;
            else Status = WAITING_FOR_FIRST_MRQ;
            break;
        case SECOND_MRQ_STARTED:
            pulseLength = TCNT1 - lastTimer1;
            if(SDA_IS_SET && SCL_IS_SET && MRQ_IS_SET && pulseLength>T2_MIN && pulseLength<T2_MAX) Status = WAITING_FOR_MRQ_BEFORE_TRANSMISSION;
            else Status = WAITING_FOR_FIRST_MRQ;
            break;
        case WAITING_FOR_MRQ_BEFORE_TRANSMISSION:
            if(SDA_IS_SET && SCL_IS_SET && MRQ_IS_RESET)
            {
                //Sekwencja Slave'a: ściąganie SDA do „0” po czasie T4
                delay_us(T4_MID); SDA_RESET;
                //Czekamy, aż Master zwolni MRQ (MRQ = 1)
                while(MRQ_IS_RESET);
                //Sekwencja Slave'a: zwolnienie SDA po czasie T5
                delay_us(T5_MID); SDA_SET;
                //Czekamy na sekwencję START
                while(SDA_IS_SET);
                //Czekamy, aż Master ściagnie SCL do „0”,
                //czyli przygotowuje się do wysłania bitu
                while(SCL_IS_SET);
                Status = START_DETECTED;
                //Konfiguracja i włączenie przerwania SCL dla
                //odbioru danych (poszczególnych bitów)
                SCL_INTR_INIT_ON_RISING_EDGE;
                SCL_INTR_CLEAR_FLAG;
                SCL_INTR_ENABLE;
                //Konfiguracja i włączenie przerwania SDA dla
                //wykrycia późniejszej sekwencji STOP
                SDA_INTR_INIT_ON_RISING_EDGE;
                SDA_INTR_CLEAR_FLAG;
                SDA_INTR_ENABLE;
                //Wyczyszczenie flagi MRQ
                MRQ_INTR_CLEAR_FLAG;
            }
            else Status = WAITING_FOR_FIRST_MRQ;
            break;
    }
}
```

**Tabela 5. Lista zewnętrznych przerw systemowych wraz z opisem ich konfiguracji i funkcjonalności niezbędnych dla obsługi magistrali radioodbiornika Opla**

Nazwa przerwania	Sposób konfiguracji	Funkcjonalność
INT0	Wyzwalane przy każdym zboczu sygnału na wejściu AA	Wykorzystywane do obsługi sygnału AA w zakresie detekcji włączenia/wyłączenia radioodbiornika
INT1	Wyzwalane przy każdym zboczu sygnału na wejściu MRQ	Wykorzystywane do obsługi sygnału MRQ w zakresie kontroli procesu odbioru ramki danych
INT2	Wyzwalane przy narastającym zboczu sygnału na wejściu SCL	Wykorzystywane do obsługi sygnału SCL w zakresie odczytu bitów danych
PCINT2	Wyzwalane zmianą stanu na wejściu SDA	Wykorzystywane do obsługi sygnału SDA w zakresie wykrywania sekwencji Stop

**Rysunek 6. Wygląd ramki danych znacznika czasu wyświetlacza TID****Listing 4. Funkcja ISR odpowiedzialna za obsługę sygnału SDA interfejsu danych wyświetlacza TID**

```
//Przerwanie odpowiedzialne za obsługę sygnału SDA - wyzwalane po każdej zmianie stanu na SDA
ISR(SDA_INTR_NAME)
{
    if(SCL_IS_SET)
    {
        if(SDA_IS_SET) //Odebrano sekwencję STOP
        {
            //Wyłączamy przerwanie SDA i oczekujemy na nową sekwencję MRQ
            // (przed pakietem danych)
            Status = WAITING_FOR_MRQ_BEFORE_TRANSMISSION;
            SDA_INTR_DISABLE;
            //Zerujemy zmienną, bo sygnał STOP mógłby wystąpić przed końcem ramki
            byteIndex = 0;
        }
    }
}
```

**Listing 5. Funkcja ISR odpowiedzialna za obsługę sygnału SCL interfejsu danych wyświetlacza TID**

```
//Przerwanie odpowiedzialne za obsługę sygnału SCL - wyzwalane przy rosnącym zboczu sygnału na SCL
ISR(SCL_INTR_NAME)
{
    register uint8_t Bytes = (TIDtype == TID_TYPE_8DIGITS)? 11:14;
    static uint8_t bitIndex, readValue, TID[14];
    readValue = (readValue<<1)|SDA_READ;
    if(++bitIndex == 8) //Odczytano 8 bitów bieżącego bajta danych
    {
        TIDisr[byteIndex++] = readValue; //Element nr 0 zawiera adres I2C TID'a
        bitIndex = readValue = 0;
        //Generujemy ACK po każdych, odebranych 8 bitach danych
        while(SCL_IS_SET); //Czekamy, aż Master ustawi SCL = 0
        SDA_RESET; //sygnał ACK generowany przez Slave'a
        while(SCL_IS_RESET); //Czekamy, aż Master ustawi SCL, tj. odbierze ACK
        while(SCL_IS_SET); //Czekamy, aż Master wyzeruje SCL, czyli przejdzie do następnego bitu
        SDA_SET; //Zwolnienie SDA przez Slave'a
        SCL_INTR_CLEAR_FLAG;
        SDA_INTR_CLEAR_FLAG;
        //Sprawdzamy, czy mamy komplet bajtów by oczekiwać na sekwencję
        //STOP (niezależnie od rodzaju TID'a)
        if((TID[0]>>1 == TID_TYPE_8DIGITS && byteIndex == 11)
            ||(TID[0]>>1 == TID_TYPE_10DIGITS && byteIndex == 14)
            ||(TID[0]>>1 == TID_TIMESTAMP && byteIndex == 8))
        {
            if((TID[0]>>1) == TIDtype) //Sprawdzamy, czy to obsługiwany TID
            {
                for(uint8_t i=0; i<Bytes; ++i) TIDdata[i] = TID[i];
                TIDdataReady = 1;
            }
            byteIndex = 0;
            //Wyłączamy przerwanie SCL i oczekujemy na sekwencję STOP
            SCL_INTR_DISABLE;
        }
    }
}
```

przerwań systemowych, ich konfigurację oraz realizowaną funkcjonalność.

W tym momencie posiadamy już niezbędną wiedzę w zakresie sposobu komunikacji radioodbiornika z oryginalnym wyświetlaczem pokładowym TID, w związku z czym pora na szczególne implementacyjne mechanizmów pozwalających na emulację tego rodzaju wyświetlacza w zakresie interfejsu naszego komputera pokładowego. Zanim jednak przedstawię ciała funkcji zewnętrznych przerw systemowych realizujących obsługę poszczególnych sygnałów magistrali danych muszę przedstawić plik nagłówkowy, który ułatwia napisanie wspomnianych funkcji a zarazem czyni je bardziej czytelnymi. Treść pliku nagłówkowego, o którym mowa pokazano na **listingu 1**. Napisano go w sposób dość „uniwersalny”, aby łatwo można było zmieniać wiele z predefiniowanych nazw i stałych, bez potrzeby edycji funkcji, które to korzystają z jego zawartości. Pora na przedstawienie funkcji obsługi zewnętrznych przerw systemowych, które łącznie, odpowiedzialne są za obsługę interfejsu danych firmy Opel. Te funkcje pokazano na **listingach** od 2 do 5. Do kompletu brakuje funkcji inicjalizacyjnej, która ustawia parametry emulowanego wyświetlacza TID oraz aktywuje niezbędne, startowe funkcje ISR – pokazano ją na **listingu 6**.

Przejdźmy zatem do krótkiej analizy przedstawionych mechanizmów programowych. Można zauważyć, że funkcja obsługująca sygnał MRQ sprawdza zależności czasowe podczas sekwencji *Power On Test* i dopiero po ich pozytywnej weryfikacji pozwala na dalszą analizę sygnałów sterujących wykonywaną przez funkcje ISR, odpowiednio dla SDA i SCL. Oczywiście, sam start procesu obsługi magistrali wyświetlacza TID jest możliwy wyłącznie po załączeniu radioodbiornika, czemu towarzyszy wywołanie przerwania dla sygnału AA.

**ROBERT WOŁGAJEW, EP****Listing 6. Funkcja inicjalizacji modułu emulatora wyświetlacza TID**

```
inline void initTIDemulator(uint8_t TID_type)
{
    AA_INTR_INIT_ON_BOTH_EDGES;
    AA_INTR_CLEAR_FLAG;
    AA_INTR_ENABLE;
    TIDtype = TID_type;
}
```





# multiTID

## – wielofunkcyjny, samochodowy komputer pokładowy (2)

*Po lekturze pierwszej części artykułu wiemy już wszystko w zakresie obsługi wyświetlacza pokładowego oraz emulowania jego funkcjonalności, więc pora na zaprezentowanie szczegółów budowy komputera samochodowego multiTID.*

Schemat ideowy komputera samochodowego multiTID pokazano na **rysunku 7**. Został on zbudowany z zastosowaniem mikrokontrolera Atmega644-PA, zegara czasu rzeczywistego z podtrzymaniem zasilania MCP7940N-I/SN oraz wyświetlacza TFT pełniącego rolę graficznego interfejsu użytkownika.

Mikrokontroler, jak to zwykle bywa, stanowi „serce”. Jest taktowany za pomocą rezonatora kwarcowego 12,288 MHz dla zapewnienia dużej dokładności pomiaru czasu oraz prędkości ładowania obrazków stanowiących elementy graficznego interfejsu użytkownika. Oczywiście, lepiej byłoby zastosować maksymalną, dostępną częstotliwość taktowania mikrokontrolera tj. 20 MHz, jednak przy napięciu zasilania 3,3 V jest to wartość progowa, dla której deklarowana jest poprawna praca mikrokontrolera.

W celu realizacji założonej funkcjonalności urządzenia, w programie obsługi niniejszego sterownika, wykorzystano dwa sprzętowe układy czasowo-licznikowe znajdujące się „na pokładzie” mikrokontrolera oraz jedno przerwanie zewnętrzne. Sposób konfiguracji wspomnianych peryferiów oraz realizowaną przez nie funkcjonalność pokazano w **tabeli 6**. W aplikacji użyto także ostatniego dostępnego układu

czasowo-licznikowego Timer2. Skonfigurowano go do pracy w trybie CTC, w którym generuje cykliczne przerwania (co 10 ms) wywołując obsługę klawiatury (przyciski MENU/<, OK/>). Dzięki zastosowaniu tego typu rozwiązania, program obsługi aplikacji urządzenia nie używa opóźnień, co zapewnia jego bezproblemową pracę oraz możliwość detekcji czasu naciśnięcia przycisku (krótki/długi/przytrzymanie itd.), dzięki czemu udało się poprawić ergonomię.

Jak opisano w tab. 6, multiTID dokonuje w czasie każdej sekundy pomiaru sumarycznej liczby impulsów doprowadzanych na wejście T0 mikrokontrolera z przetwornika drogi pojazdu oraz pomiaru sumarycznego czasu wtrysków, których to sygnał jest doprowadzony na wejście ICP1 licznika Timer1. W celu realizacji drugiej z funkcjonalności zaprojektowano dwa kompletne, bezpieczne układy wejściowe formujące sygnały wtryskiwaczy paliwa (oddzielnie dla Pb i LPG) dla potrzeb wejściowych obwodów mikrokontrolera. Wybór wtryskiwacza paliwa, z którego impulsy podawane są na wejście ICP mikrokontrolera, jest realizowany poprzez scalony przełącznik NC7SB3157 (U5), dla którego sygnałem sterującym jest poziom logiczny na wyjściu PD7 mikrokontrolera. Jest on zmieniany w procedurze obsługi przerwania PCINT1, które

jest wywoływane zmianą poziomu na wprowadzeniu PB6 na skutek załączenia zaworu na reduktorze instalacji LPG (czyli zmiany paliwa zasilającego silnik na LPG, w tym „start” wtryskiwaczy gazu). To rozwiązanie pozwala na efektywną realizację funkcji automatycznej detekcji rodzaju paliwa i reakcję po stronie programu obsługi.

Na **listingu 8** pokazano wzory zaczerpnięte bezpośrednio z programu obsługi aplikacji służące do obliczenia wszystkich parametrów rzeczywistych komputera pokładowego.

Kilka słów komentarza wymaga mechanizm akumulowania przejechanej drogi i zużytego paliwa. Komputer multiTID oblicza i wyświetla średnie wartości zużycia paliwa i prędkości, objętość paliwa pozostającego w baku pojazdu oraz pokonany dystans. Jak łatwo się domyślić, w celu wyznaczenia wspomnianych wartości jest niezbędna znajomość całkowitego zużycia paliwa i dystansu od momentu wyzerowania liczników, a co za tym idzie – niezbędny staje się mechanizm akumulowania mierzonych wartości, niezależnie dla każdego rodzaju paliwa.

### Ustawienia Fuse-bitów:

```
CKSEL3...0: 1111
SUT1...0: 11
CKDIV8: 1
CKOUT: 1
JTAGEN: 1
EESAVE: 1
BODLEVEL2:0: 110
```

**Wykaz elementów:****Rezystory:** (SMD 0805)

R1, R3: 10 kΩ  
 R2: 6,34 kΩ/1%  
 R4, R5, R7, R9, R11, R12, R14...R16, R18: 3 kΩ  
 R6, R10, R13, R19: 1 kΩ  
 R8, R17: 8,2 kΩ  
 R20, R21: 10 Ω

**Kondensatory:** (SMD 0805)

C1, C2, C8, C9: 22 pF  
 C4: 470 nF  
 C3, C5, C10, C13, C14: 100 nF  
 C6, C7, C11: 10 μF/16 V (SMD „A”)  
 C12: 1 μF

**Półprzewodniki:**

U1, U3 – MCP1703-3302E/DB (obudowa SOT223)  
 U2 – MCP7940N-I/SN (obudowa SO08)  
 U4 – ATMEGA644PA (obudowa TQFP44)  
 U5 – NC7SB3157 (obudowa SC-70-6)  
 U6, U7 – LTV817S-SMD (obudowa SO04)  
 T1, T2 – BC817 (obudowa SOT23)  
 D1, D2 – 1N4148 (obudowa MINIMELF)

**Inne:**

TFT – wyświetlacz graficzny Multi-Inno Technology MI0420CT-5 (kontroler FT1509, taśma ZIF 45 pin)  
 L1 – dławik 10μH (obudowa SMD 0805)  
 Q1 – rezonator kwarcowy 12,288 MHz niski  
 Q2 – rezonator kwarcowy, zegarkowy 32768 Hz  
 MENU/<, OK/> – microswitch THT wysokość 6 mm  
 LPG – gniazdo IDC męskie typu P1303 (PIN3, raster 2,54mm)  
 PB – gniazdo IDC męskie typu P1302 (PIN2, raster 2,54mm)  
 TID\_CON – gniazdo wyświetlacza Opel TID lub gniazdo IDC do druku (PIN 2×6, raster 2,54mm)

Mikrokontroler Atmega644-PA ma nielotną pamięć EEPROM, jednak zapewnia ona ograniczoną liczbę gwarantowanych cykli zapisu. Dlatego posłużono się pamięcią RAM dostępną w układzie zegara czasu rzeczywistego MCP7940N, której zawartość jest podtrzymywana nawet po wyłączeniu zasilania za pomocą baterii BAT. Dla wygody wszystkie dane obliczeniowe uformowano w strukturę zaopatrzoną w sumę kontrolną CRC8.

Kolejne słowo komentarza należy się układowi odpowiedzialnemu za pomiar temperatury zewnętrznej. Samochody marki Opel mają rezystancyjny czujnik temperatury zewnętrznej montowany w okolicach przedniego zderzaka. Jest to termistor o ujemnym współczynniku temperaturowym (NTC), którego zaciski dostępne są w złączu wyświetlacza TID. Skoro tak, to sposób pomiaru temperatury nasuwa się sam. Dysponując przetwornikiem A/C wystarczy zastosować dzielnik rezystancyjny, którego jednym z elementów jest termistor w pojeździe, a drugim dokładny rezystor o odpowiedniej rezystancji. Dzielnik rezystancyjny, o którym mowa zbudowano z termistora dostępnego na zaciskach NTC/NTC wyświetlacza TID oraz rezystora R2.

Wszystkie wejścia PORTA mikrokontrolera, do których w wewnętrznej strukturze układu są dołączone wejścia przetwornika A/C użyto do sterowania pracą wyświetlacza TFT. Jak w takiej sytuacji zmierzyć napięcie występujące na termistorze? Zastosowano tu pewną „sztuczkę”. Napięciem wejściowym (mierzonym) przetwornika A/C jest specjalne, wysokostabilne, wewnętrzne

źródło napięcia odniesienia VBG (bandgap), a napięcie z dzielnika NTC/R2 jest napięciem referencyjnym przetwornika A/C. W ten sposób odwrócimy „normalny” sposób pomiaru napięcia przez przetwornik A/C. Nie mając de facto do dyspozycji żadnego z wejściowych portów mikrokontrolera, do których dołączono wejścia przetwornika A/C, możemy dokonywać pomiaru napięcia korzystając głównie z faktu, iż przetwornik może mierzyć napięcie VBG.

Napięcie wyjściowe dzielnika NTC/R2 będące zarazem napięciem referencyjnym (odniesienia) przetwornika A/C wyraża się wzorem  $U_{WY} = U_{REF} = R_{NTC} / (R_{NTC} + R_2) \times 3,3 \text{ V}$ . Z kolei, wartość zmierzona przez przetwornik A/C wyraża się wzorem  $ADC = (U_{IN} \times 1024) / U_{REF}$ , gdzie  $U_{IN} = VBG = 1,1 \text{ V}$ . Za pomocą obu wzorów otrzymujemy wyrażenie pozwalające wyznaczyć wartość mierzoną przez przetwornik A/C –  $ADC = (1,1 \times 1024 \times (R_{NTC} + R_2)) / (3,3 \times R_{NTC})$ . Tego sposobu możemy użyć prawie zawsze, gdy brakuje wejść, aby skoryzować z wbudowanego przetwornika A/C. By obliczyć wartość temperatury zewnętrznej, program obsługi aplikacji korzysta z predefiniowanej i zapisanej w pamięci Flash tablicy stałych odwzorowujących charakterystykę zastosowanego termistora NTC. Już zupełnie na koniec dodam, iż zdecydowaną większość pamięci programu aplikacji zajęły wzorce obrazków wyświetlanych w ramach graficznego interfejsu użytkownika (niektóre z nich, mimo kompresji, zajmują po 5 kB) oraz wzorce czcionek.

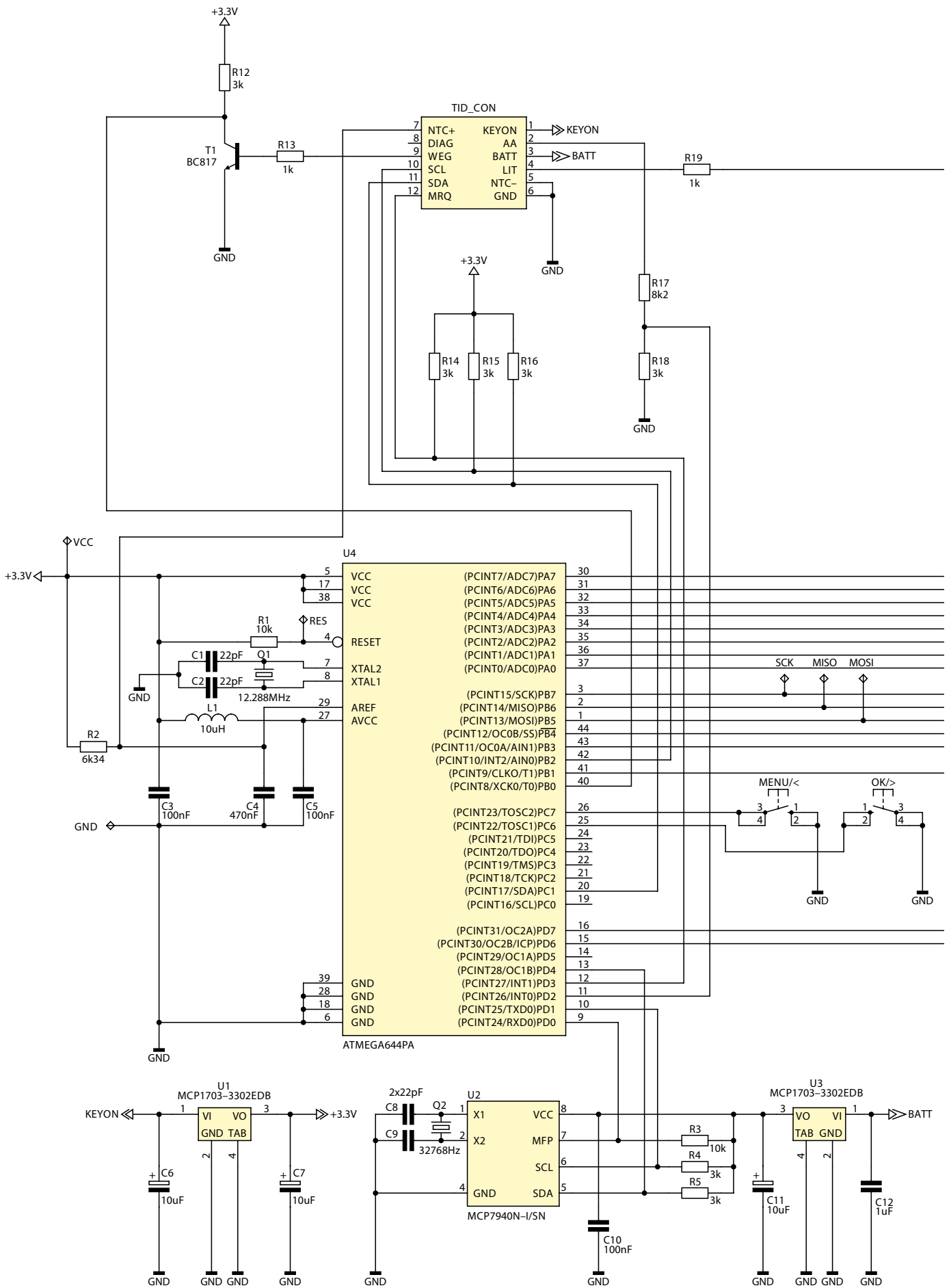
Robert Wołgajew, EP

Listing 8. Wzory używane do obliczania wyświetlanych parametrów na ekranie

```
//
//Obliczamy spalone w czasie ostatniej sekundy paliwo [ul] - jeśli tylko nie jesteśmy w trybie kalibracji
if(Config.Calibration[Fuel] == 0) spentFuelPerls = ((10UL*Config.Cylinders*injectionTime*Config.CcPerMin[Fuel])/2880UL); else spentFuelPerls = 0;
//Akumulujemy wartości zużytego paliwa [ul] oraz przejechanego dystansu [m] niezbędne do obliczenia wartości średnich
Accu.spentFuel[Fuel] += spentFuelPerls;
Accu.Distance[Fuel] += ((10UL*WEGpulses*Config.Wheel) / (1000UL*Config.PulsPerRot));
//Obliczamy również ilość paliwa pozostającego w baku pojazdu [ul] pod warunkiem, że jeszcze jakieś pozostało
if(Accu.remainingFuel[Fuel] >= spentFuelPerls) Accu.remainingFuel[Fuel] -= spentFuelPerls;
//Dla potrzeb procedury kalibracji akumulujemy sumaryczny czas wtrysków, jeśli tylko ta procedura jest aktywna
if(Config.Calibration[Fuel]) Accu.injectionTime += injectionTime;
//Obliczamy wartości chwilowe dostępne z pomiarów wykonywanych co 1 sekundę
Speed = ((360UL*WEGpulses*Config.Wheel) / (1000UL*Config.PulsPerRot)); //Prędkosc chwilowa [km/h]
if(Speed > Accu.Vmax) Accu.Vmax = Speed; //Zapamiętujemy wartość maksymalnej prędkości
//Zużycie chwilowe dla wszystkich wtrysków: w [l/h *10] gdy Speed<=5 i w [l/100km *10] gdy Speed>5
if(Speed<=5) Consum = ((5UL*Config.Cylinders*injectionTime*Config.CcPerMin[Fuel]) / 400000UL);
else Consum = ((5UL*Config.Cylinders*injectionTime*Config.CcPerMin[Fuel]*Config.PulsPerRot) / (144UL*WEGpulses*Config.Wheel));
if(Consum > 999) Consum = 999; //Zabezpieczenie przed przekroczeniem zakresu
//Na podstawie akumulat. wartości zużytego paliwa i przejech. dystansu obliczamy wartości średnie prędkości oraz zużycia paliwa
SpeedAvg = ((360UL*Accu.Distance[Fuel]) / (100UL*Accu.Measurements[Fuel])); // [km/h]
if(Accu.Distance[Fuel]>999) ConsumAvg = (Accu.spentFuel[Fuel]/Accu.Distance[Fuel]); else ConsumAvg = 0; //l/100km *10
//Obliczamy dostępny zasięg pojazdu na pozostałym w baku paliwie [km] = pozostałe paliwo/średnie zużycie paliwa
if(Accu.spentFuel[Fuel]) availableDistance = (Accu.remainingFuel[Fuel]*Accu.Distance[Fuel]) / (1000UL*Accu.spentFuel[Fuel]); else availableDistance = 0;
if(availableDistance > 999) availableDistance = 999; //Zabezpieczenie przed przekroczeniem zakresu

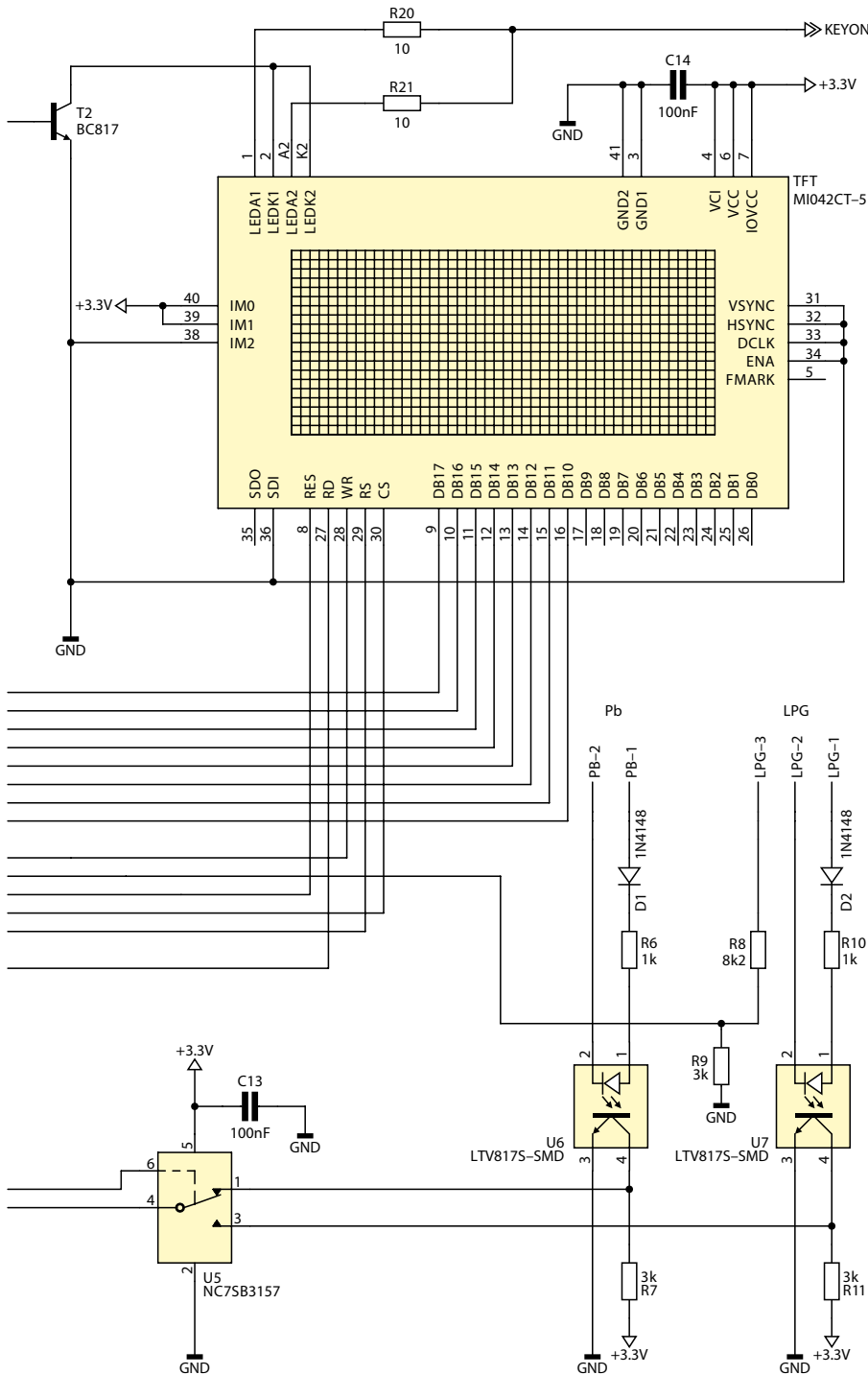
/*
- InjectionTime - sumaryczny czas wtrysku zliczony w czasie 1s [ms*48]
- WEGpulses - liczba impulsów z przetwornika drogi zliczona w czasie 1 sekundy
- Fuel - bieżący rodzaj paliwa - zmienna ustawiana w procedurze przerwania PCINT1
- spentFuelPerls - bieżące paliwo spalane w czasie ostatniej sekundy [ul]
- Accu.Measurements - akumulator liczby interwałów pomiarowych [s] - oddzielnie dla każdego paliwa
- Accu.spentFuel - akumulator ilości spalonego paliwa [ul] - oddzielnie dla każdego paliwa
- Accu.remainingFuel - akumulator ilości paliwa pozostającego w baku [ul] - oddzielnie dla każdego paliwa
- Accu.Distance - akumulator przejechanego dystansu [m] - oddzielnie dla każdego paliwa
- Consum - chwilowe zużycie paliwa [l*10/h], dla prędkości<5 km/h lub [l*10/100km], dla prędkości>5 km/h
- ConsumAvg - średnie zużycie paliwa [l*10/100km]
- Speed - prędkość chwilowa [km/h]
- SpeedAvg - prędkość średnia [km/h]
- availableDistance - orientacyjny, dostępny dystans na paliwie pozostającym w baku [km] - oddzielnie dla każdego paliwa
- Config.CcPerMin - stała wtryskiwacza [ml/min] - oddzielnie dla każdego paliwa
- Config.PulsPerRot - stała przetwornika drogi [imp/obr]
- Config.Cylinders - liczba wtryskiwaczy paliwa
- Config.Wheel - obwód opony [cm]
*/

```



Rysunek 7. Schemat ideowy komputera samochodowego multiTID





**DODATKOWE MATERIAŁY NA FTP:**

<ftp://ep.com.pl>

**USER: 33948, PASS: 5gcckdmq**

**W ofercie AVT\***

**AVT-5562**

**Podstawowe informacje:**

- Napięcie zasilania: 8...15 V DC.
- Maksymalny prąd obciążenia (z napięciem +12 V): 10 mA.
- Prąd podtrzymania zegara RTC (z napięciem BATT): 1 mA.
- Maksymalny prąd podświetlenia (z napięciem ILL+): 75 mA.
- Dokładność pomiaru temperatury: 1°C.
- Zakres pomiarowy temperatury zewnętrznej: -30...35°C.
- Zakres pomiarowy prędkości pojazdu: 0...255 km/godz.
- Zakres pomiarowy chwilowego zużycia paliwa: 0...99,9 l/100 km.
- Zakres pomiarowy średniego zużycia paliwa: 0...25,5 l/100 km.
- Zakres pomiarowy paliwa dostępnego w baku: 0...99,9 l.
- Zakres pomiarowy przejechanej odległości: 0...9999 km.
- Zakres pomiarowy dystansu do przejechania na dostępnym paliwie: 0...999 km.
- Zakresy regulacji parametrów konfiguracyjnych:
  - Stała wtryskiwaczy: 1...999 ml/min.
  - Stała przetwornika drogi: 1...99 impulsów/obrót.
  - Obwód opony: 50...255 cm.
  - Liczba cylindrów: 2...8.
  - Pojemności baków: 25...99 l.
- Przesunięcie belek informacyjnych: 0:9 pikseli.

**Projekty pokrewne na FTP:**

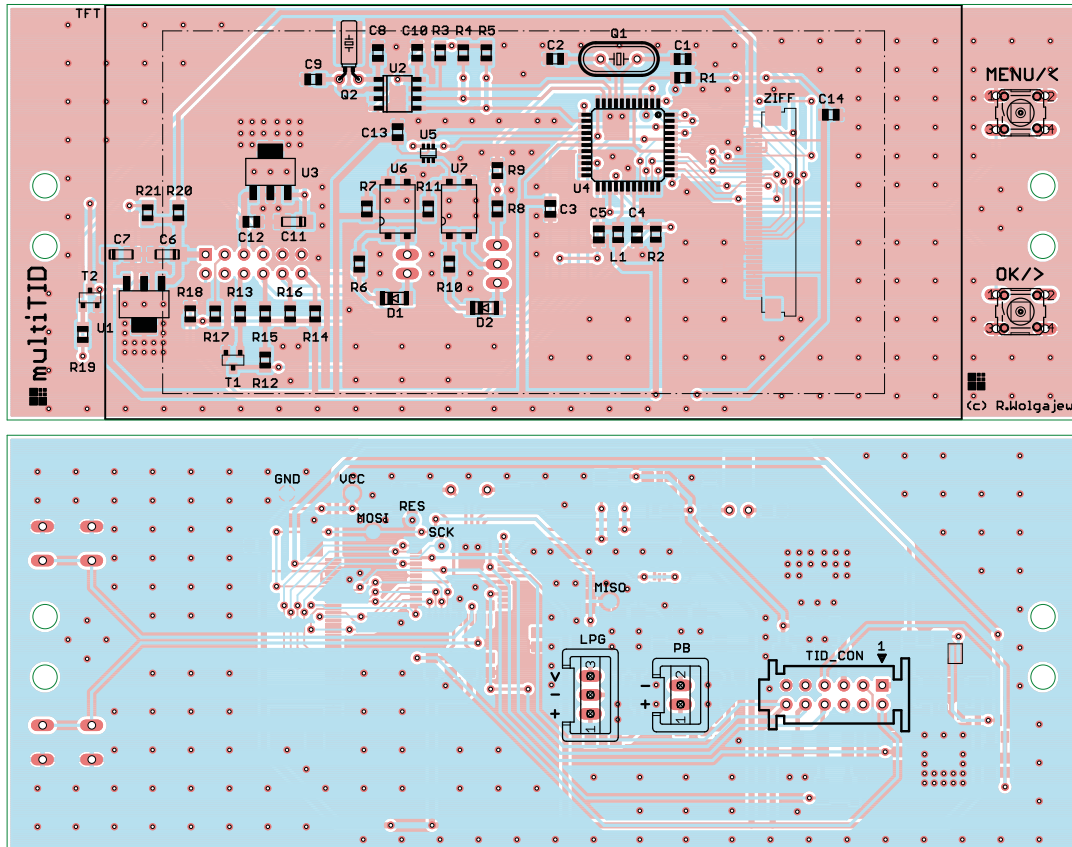
(wymienione artykuły są w całości dostępne na FTP)

AVT-5545	Komputer samochodowy Mee MK II (2.0) (EP 7-8-9/2016)
AVT-5495	Uniwersalny komputer samochodowy Mee (EP 3/2015)
AVT-3095	Komputer samochodowy (EdW 4-5/2014)
AVT-5405	TripCo – komputer samochodowy (EP 7/2013)
AVT-5395	TIDex – komputer dla samochodów z silnikiem Diesla (EP 5/2013)

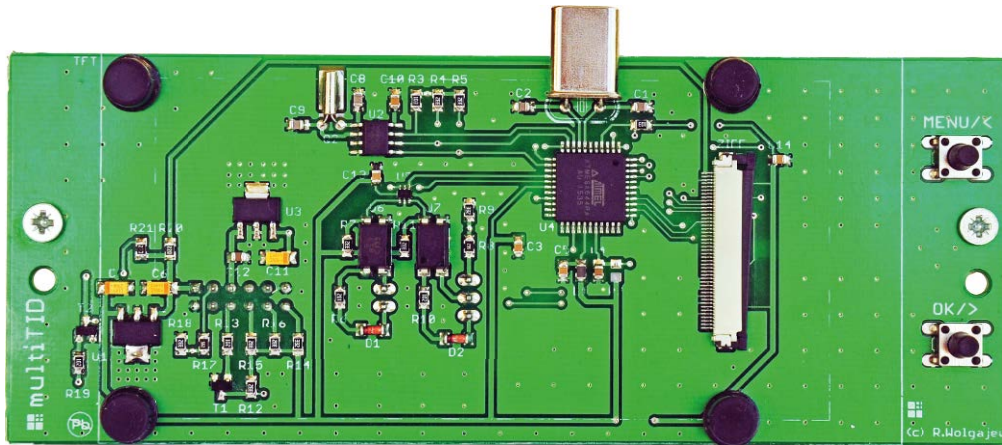
**\* Uwaga:**  
Zestawy AVT mogą występować w następujących wersjach:  
**AVT xxxx UK** to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
**AVT xxxx A** płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
**AVT xxxx A1** płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
**AVT xxxx B** płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku.pdf  
**AVT xxxx C** to nic innego jak zmontowany zestaw B, czyli elementy wylutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku.pdf  
**AVT xxxx CD** oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu).  
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

**Tabela 6. Konfiguracja peryferiów Atmega644-PA dla potrzeb pomiarów**

Nazwa	Sposób konfiguracji	Funkcjonalność
Timer0	Licznik impulsów zewnętrznych podawanych na wejście T0 – reakcja na zbocze opadające.	Liczy impulsy przetwornika drogi WEG zapewniając pomiar przejechanego dystansu.
Timer1	Licznik taktowany wewnętrznym przebiegiem o częstotliwości 48 kHz (preskaler=256, 48 taktów na 1 ms). Wejście Input/Capture dotychczas do układu formującego przebieg z wtryskiwacza paliwa.	Mierzy sumaryczny czas wtrysku w czasie 1 s. zapewniając pomiar zużytego paliwa.
Przerwanie INT0	Wyzwalane zboczem opadającym na wejściu INT0, do którego doprowadzono wyjście częstotliwości wzorcowej zegara RTC.	Służy do odmierzania czas pomiaru równego 1 s.



Rysunek 8 Schemat montażowy komputera samochodowego multiTID



Fotografia 9 Wygląd obwodu drukowanego zmontowanego sterownika multiTID od strony elementów (bez wyświetlacza TFT)

REKLAMA

# Elektronika Praktyczna na facebook



# multiTID

## – wielofunkcyjny, samochodowy komputer pokładowy (3)

*Omówiliśmy już sposób działania oraz budowę komputera multiTID. Teraz przyszedł czas na zamontowanie go w samochodzie, skonfigurowanie, a wreszcie użytkowanie!*

Schemat montażowy komputera samochodowego multiTID pokazano na rysunku 8, opublikowanym w poprzedniej części artykułu w EP 1/2017. Zaprojektowano zwarty, dwustronny obwód drukowany, złożony w głównej mierze z elementów przeznaczonych do montażu powierzchniowego. Co bardzo ważne w wypadku pracy w środowisku o potencjalnie dużym poziomie zaburzeń, jakim bez wątpienia jest instalacja samochodowa, zadbano o odpowiednie prowadzenie masy i sygnałów krytycznych.

Z uwagi na fakt, że moduł wyświetlacza TFT jest dołączony do płytki z użyciem gniazda ZIF o gęstym rastrze (45 wyprowadzeń co 0,5 mm), montaż rozpoczynamy od przylutowania tego gniazda. Należy przy tym posłużyć się cyną z odpowiednią ilością topnika oraz cienką płecionką, która umożliwi usunięcie nadmiaru cyny. Jakość tak wykonanego połączenia sprawdzamy pod lupą oraz kontrolujemy zwarcia multimetrem. Wspomniana kontrola będzie znacznie łatwiejsza, jeśli zmontowana płytkę sterownika przemyjemy alkoholem izopropylowym w celu wypłukania nadmiaru kalafonii.

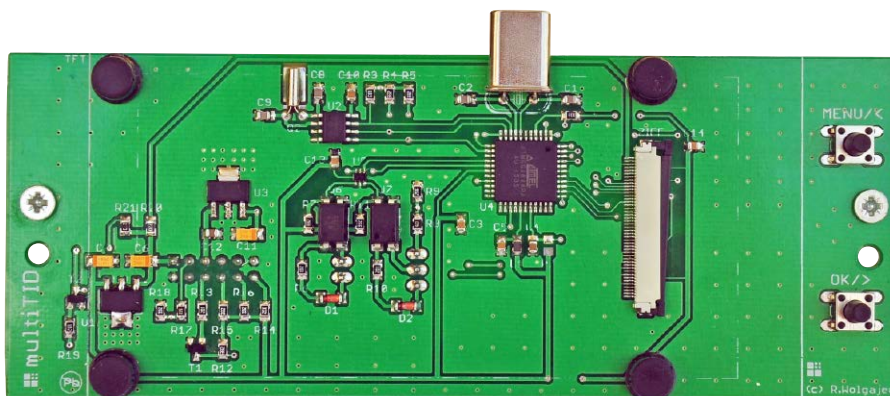
Następnie lutujemy mikrokontroler, pozostałe elementy półprzewodnikowe SMD, elementy bierne, a na samym końcu wszystkie elementy przeznaczone do montażu przewlekane.

Ostatnimi podzespołami, które należy wlotować po stronie wyprowadzeń (BOTTOM), są gniazda podłączeniowe: TID\_CON (najlepiej oryginalne, wylutowane z wyświetlacza TID), Pb, LPG. Na samym końcu przyłączamy taśmę wyświetlacza TFT do złącza ZIF, dbając o odpowiednie zablokowanie zatrzasków złącza, natomiast sam panel TFT przyklejamy do płytki sterownika, korzystając z 4 dystansów o wysokości równej wysokości najwyższego elementu montowanego po stronie TOP oraz dwustronnej taśmy klejącej. Na **fotografii 9** pokazano wygląd zmontowanego komputera multiTID bez wyświetlacza TFT.

### Połączenia

Konstruując multiTID, kierowałem się założeniem, że jego włączenie w miejsce oryginalnego wyświetlacza pokładowego TID powinno sprowadzać się do zwykłego wpięcia przewodów. Idąc tym tokiem myślenia, urządzenie multiTID wyposażylem w 12-pinowe złącze o rozmieszczeniu wyprowadzeń zgodnym z oryginałem, co umożliwi wlotowanie w to miejsce elementu wymontowanego z oryginalnego, 10-znakowego wyświetlacza pokładowego (z Astry G, Corsy C itp.). Oczywiście, w tym miejscu możemy również zastosować listwę goldpin 2x6 pinów.

Rozmieszczenie wyprowadzeń, o którym wspomniałem wcześniej, dotyczy wersji wyświetlacza TID o organizacji 10-znakowej, więc jeśli urządzenie multiTID ma być zainstalowane w pojeździe wyposażonym



**Fotografia 9. Wygląd obwodu drukowanego zmontowanego sterownika multiTID od strony elementów (bez wyświetlacza TFT)**



**DODATKOWE MATERIAŁY NA FTP:**

ftp://ep.com.pl

USER: 97325, PASS: 6yfwxr8q

**W ofercie AVT\*****AVT-5562****Podstawowe informacje:**

- Napięcie zasilania: 8...15 V DC.
- Maksymalny prąd obciążenia (z napięcia +12 V): 10 mA.
- Prąd podtrzymania zegara RTC (z napięcia BATT): 1 mA.
- Maksymalny prąd podświetlenia (z napięcia ILL+): 75 mA.
- Dokładność pomiaru temperatury: 1°C.
- Zakres pomiarowy temperatury zewnętrznej: -30...35°C.
- Zakres pomiarowy prędkości pojazdu: 0...255 km/godz.
- Zakres pomiarowy chwilowego zużycia paliwa: 0...99,9 l/100 km.
- Zakres pomiarowy średniego zużycia paliwa: 0...25,5 l/100 km.
- Zakres pomiarowy paliwa dostępnego w baku: 0...99,9 l.
- Zakres pomiarowy przejechanej odległości: 0...9999 km.
- Zakres pomiarowy dystansu do przejechania na dostępnym paliwie: 0...999 km.
- Zakresy regulacji parametrów konfiguracyjnych:
  - Stałe wtryskiwaczy: 1...999 ml/min.
  - Stała przetwornika drogi: 1...99 impulsów/obróć.
  - Obwód opony: 50...255 cm.
  - Liczba cylindrów: 2...8.
  - Pojemności baków: 25...99 l.
- Przesunięcie belek informacyjnych: 0÷9 pikseli.

**Projekty pokrewne na FTP:**

(wymienione artykuły są w całości dostępne na FTP)

AVT-5545	Komputer samochodowy Mee MK II (2.0) (EP 7-8-9/2016)
AVT-5495	Uniwersalny komputer samochodowy Mee (EP 3/2015)
AVT-3095	Komputer samochodowy (EdW 4-5/2014)
AVT-5405	TripCo – komputer samochodowy (EP 7/2013)

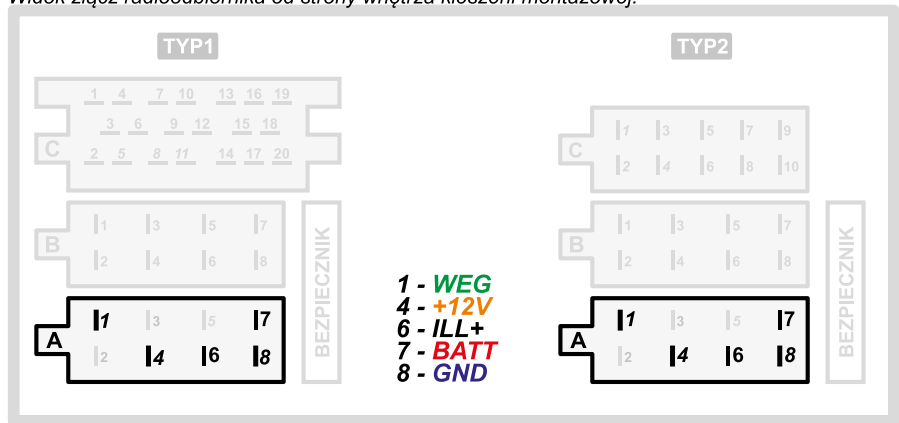
**\* Uwaga:**

Zestawy AVT mogą występować w następujących wersjach:  
**AVT xxxx UK** to oprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
**AVT xxxx A** płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wydrukowane są oznaczenia), bez elementów dodatkowych.  
**AVT xxxx A+** płytka drukowana i oprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
**AVT xxxx B** płytka drukowana (lub płytki) oraz komplet elementów wymiennych z zestawem pdf.  
**AVT xxxx G** to nic innego jak zmontowany zestaw B, czyli elementy wylutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf.  
**AVT xxxx CD** oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu).  
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf. Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

w starszą wersję wyświetlacza (np. Astra F, Corsa B) lub pojeździe o innym rodzaju złącza (np. Omega, Vectra B), konieczne jest odpowiednie dopasowanie sygnałów sterujących oryginalnego złącza wyświetlacza pojazdu do sygnałów sterujących złącza sterownika multiTID. W aucie innym niż Opel, stosowane sygnały, które zgrupowane są standardowo w złączu wyświetlacza TID, „pobieramy” tym razem ze złącza ISO radioodbiornika.

Wygląd typowych złączy ISO radioodbiornika wraz z oznaczeniem interesujących nas wyprowadzeń pokazano na rysunku 10.

Widok złącza radioodbiornika od strony wnętrza kieszeni montażowej.

**Rysunek 10. Rodzaje złączy radioodbiornika z opisem wyprowadzeń**

Co ciekawe, w zależności od marki pojazdu, piny o numerach 4 i 7 (oznaczone jako +12 V i BATT) mogą być zamienione miejscami, dlatego ich rzeczywiste funkcje należy ustalić eksperymentalnie. Napięcie na pinie +12 V powinno występować wyłącznie po włączeniu zapłonu (przekręceniu kluczyka), a na pinie BATT na stałe. Napięcie na pinie ILL+ to zwyczajowo przebieg prostokątny o amplitudzie ok. +12 V i wypełnieniu zależnym od ustawienia pokrętki regulującego jasność podświetlenia zegarów pojazdu. To napięcie jest używane do regulacji jasności podświetlenia wyświetlacza urządzenia multiTID. Jeśli nie zależy nam na takiej funkcjonalności, pin ILL+ należy zewrzeć z pinem +12 V.

Wszystkie połączenia, o których mowa powyżej, zapewniają jednak realizację części funkcjonalności sterownika tj. wszystkich funkcji niezwiązanych z pomiarem spalania. Aby umożliwić pomiar spalania, należy do urządzenia multiTID dostarczyć sygnał z wtryskiwacza/wtryskiwaczy paliwa (dla przypadku zasilania auta wyłącznie benzyną) oraz sygnał z zaworu reduktora instalacji LPG (dla przypadku zasilania auta zarówno benzyną, jak i gazem LPG).

**Połączenie multiTID z modułem wtryskiwacza paliwa**

Połączenia układu w tym zakresie należy wykonać nader starannie, zachowując dużą ostrożność, by nie doprowadzić do zwarcia przewodów zasilających wtryskiwacz, co mogłoby skutkować uszkodzeniem wyjściowych obwodów sterujących elektronicznego układu sterującego pracą silnika ECU. Każdy wtryskiwacz ma 2 wyprowadzenia. Pierwsze z nich to stałe napięcie +12 V, które zostaje podane po przekręceniu kluczyka stacyjki i które należy doprowadzić do złącza „+” stosownego gniazda urządzenia multiTID (Pb i LPG, przy zasilaniu gazem LPG). Drugie to sygnał sterujący z modułu ECU (komutowana masa), który z kolei należy doprowadzić do wejścia „-”. Przewody połączeniowe należy starannie

zabezpieczyć przed możliwością ewentualnego przetarcia izolacji i powstania zwarcia – dotyczy to zwłaszcza wszelkiego rodzaju otworów przelotowych, przez które zostaną one przeprowadzone. Aby zabezpieczyć wejściowy układ pomiarowy czasu wtrysku przed zaburzeniami (np. od będącej zwykle w pobliżu listwy zapłonowej), najlepiej zastosować dwużyłowy przewód ekranowany o odpowiednim przekroju, a ekran tego przewodu po obu stronach połączyć z masą pojazdu. Możliwe jest także połączenie układu formującego impulsy wtryskiwaczy paliwa bezpośrednio z wyjściem odpowiedniego sterownika (ECU silnika i komputera instalacji LPG). Wtedy należy odpowiednio zmniejszyć wartości rezystorów R6/R10 ustalających prądy diod LED transoptorów LTV817.

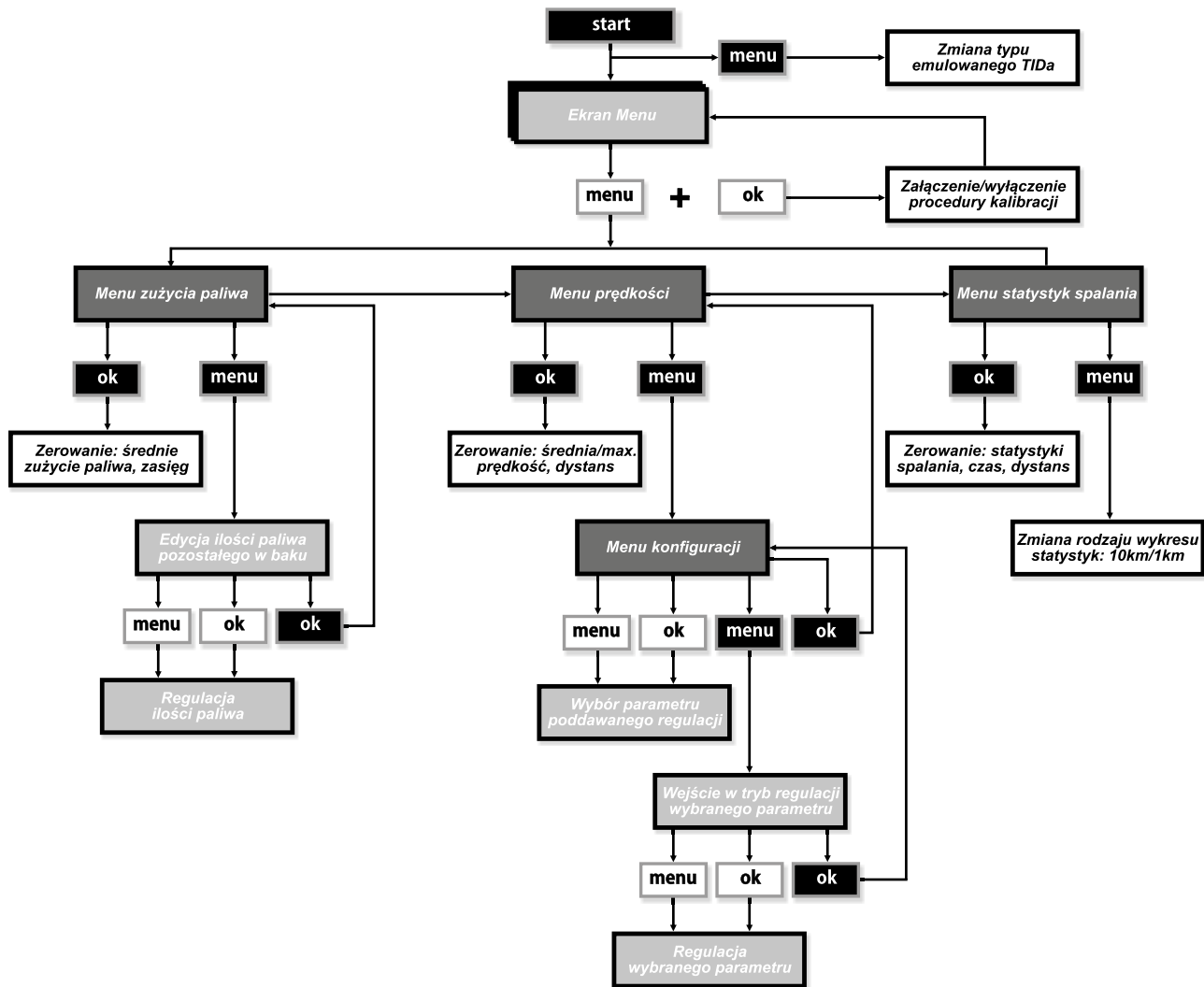
**Obsługa**

Sterownik multiTID jest urządzeniem, które zwykle będzie obsługiwane podczas jazdy samochodem, więc podstawowym kryterium przyjętym podczas tworzenia menu użytkownika były ergonomia, łatwość obsługi i czytelność interfejsu użytkownika. Zgodnie z tymi założeniami, na płytce sterownika przewidziano jedynie 2 elementy sterujące umownie oznaczone „MENU/<” i „OK/>”.

Przycisk „MENU/<” służy do zmiany aktualnie wyświetlanego ekranu menu, a przycisk „OK/>” do wykasowania liczników

REKLAMA

Projekty na...  
**STM32**  
[www.stm32.eu](http://www.stm32.eu)  
 life.augmented  
**KAMAMI**



Rysunek 11. Diagram obrazujący system Menu i sposób obsługi sterownika multiTID

dystansu, średnich wartości obliczeniowych lub wyzerowania statystyk spalania. Zastosowanie tylko dwóch przycisków pozwala również na dołączenie do urządzenia multiTID oryginalnej „manetki” komputera pokładowego, która poprawia ergonomię użytkownika urządzenia. Nie są to jednak wszystkie funkcje wspomnianych elementów, gdyż ich funkcjonalność zależy od aktualnego trybu pracy układu multiTID.

Na **rysunku 11** pokazano diagram obrazujący system menu oraz sposób obsługi urządzenia. Na tym rysunku symbole przycisków wypełnione kolorem czarnym oznaczają długie naciśnięcie wybranego przycisku. Warto zauważyć, że funkcja wyświetlania ilości dostępnego paliwa w zbiorniku nie korzysta z sygnału informującego o rzeczywistym poziomie paliwa, gdyż jej implementacja, skalowanie i samo włączenie sterownika do instalacji pojazdu byłoby dość kłopotliwe. Mechanizm jej działania jest w tym wypadku bardzo prosty i zakłada każdorazowe uzupełnianie bieżącego odczytu o ilość zatankowanego paliwa, co jest możliwe poprzez wejście w tryb edycji paliwa dostępnego w zbiorniku. A więc po każdym tankowaniu pojazdu należy wejść w edycję dostępnego

paliwa i zwiększyć jego ilość o zatankowaną wartość (zmiana o pełne litry). Mechanizm pomiarowy sterownika multiTID będzie następnie odejmował od wartości dostępnego paliwa zużywanego paliwa, co umożliwi realizację wspomnianej funkcjonalności.

Słowo komentarza należy się także samemu interfejsowi graficznemu. Konstruując sterownik multiTID, chciałem, aby formą prezentacji danych oraz możliwościami dorównywał rozwiązaniom znanym ze współczesnej motoryzacji, zwłaszcza w tzw. segmencie Premium. Podstawowym zadaniem było w takim razie zaprojektowanie czytelnego i ładnego interfejsu użytkownika z wykorzystaniem efektywnych, kolorowych elementów graficznych. Nie było to zadanie łatwe i opracowanie interfejsu graficznego zajęło mi najwięcej czasu, włączając w to wykonanie wielu drobnych ikon. Zdecydowałem się na kolorystykę zbliżoną do sepia, gdyż taka wydała mi się najbardziej stonowana i elegancka, a przy okazji pasująca do kolorów podświetlenia stosowanych przez Opla.

Na **rysunku 12** pokazano wygląd „Menu zużycia paliwa” wraz z opisem wszystkich wyświetlanych danych (w tym danych znajdujących się na górnej i dolnej

belce informacyjnej, które wyświetlane są zawsze, niezależnie od aktywnego ekranu menu). Znaczenie poszczególnych danych jest następujące:

1. Bieżąca godzina.
2. Bieżąca data.
3. Chwilowe zużycie paliwa (w l/godz. dla prędkości poniżej 5 km/godz., dla pozostałych prędkości w l/100 km) wraz z oznaczeniem aktualnie używanego paliwa.
4. Dostępna ilość aktualnie używanego paliwa (w tym, graficzna prezentacja w formie bargrafu, który zmienia kolor na czerwony w przypadku, jeśli w zbiorniku jest poniżej 10% paliwa).
5. Prognozowany dystans na dostępnym, bieżącym rodzaju paliwa (liczony w oparciu o średnie zużycie paliwa).
6. Temperatura na zewnątrz pojazdu.
7. Ikonka „śnieżynki” informująca o śliskiej nawierzchni (dla temperatury poniżej 4°C).
8. Ikonka informująca o aktywnym trybie kalibracji stałej wtryskiwacza.
9. Informacje tekstowe przesyłane przez radiodiodniok Opla plus ikonki reprezentujące status radia/magnetofonu/CD.

10. Średnie zużycie paliwa w l/100 km.

Na **rysunku 13** pokazano wygląd „Menu prędkości” z opisem wyświetlanych danych. Znaczenie poszczególnych danych jest następujące:

1. Chwilowa prędkość pojazdu w km/godz.
2. Maksymalna, zarejestrowana prędkość pojazdu w km/godz.
3. Przejechany dystans w km (od ostatniego kasowania tego licznika) dla aktualnie używanego paliwa.
4. Średnia prędkość pojazdu w km/godz.

Na **rysunku 14** pokazano wygląd „Menu statystyk” wraz z opisem wszystkich wyświetlanych danych.

Znaczenie poszczególnych danych jest następujące:

1. Słupki reprezentujące średnie zużycie paliwa, dla bieżącego rodzaju paliwa, na każde przejechane 10 km lub 1 km (w zależności od wybranej opcji). Możliwość wyświetlenia kolejnego „słupka” zależy, co oczywiste, od aktualnie przejechanego dystansu oraz wybranej skali wykresu. Wykres pokazuje zawsze statystyki dla ostatnio przejechanych 150 km lub 15 km i jest aktualizowany co jednostkę skali dystansu (przesuwa się w lewo po przejechaniu 150 km lub 15 km).
2. Linia reprezentująca średnie zużycie paliwa dla bieżącego rodzaju paliwa w l/100 km (wraz z wartością).
3. Skala dla wykresu średniego zużycia paliwa dla bieżącego rodzaju paliwa (zmieniana z poziomu menu).
4. Średnia prędkość pojazdu w km/godz. (od ostatniego kasowania tego licznika).
5. Przejechany dystans w km dla bieżącego rodzaju paliwa (od ostatniego kasowania tego licznika).
6. Czas jazdy (od ostatniego kasowania tego licznika).

Na **rysunku 15** pokazano wygląd „Menu konfiguracyjnego”. Znaczenie poszczególnych opcji Menu konfiguracyjnego jest następujące:

- Tire perimeter: obwód koła w [cm].
- Pulses/rotation: stała przetwornika drogi.
- Pb injector flow: stała wtryskiwacza benzyny [cm<sup>3</sup>].
- LPG injector flow: stała wtryskiwacza gazu LPG [cm<sup>3</sup>].
- Cylinders: liczba cylindrów.
- Pb tank capacity: pojemność zbiornika paliwa [l].
- LPG tank capacity: pojemność zbiornika gazu LPG [l].
- Time: aktualny czas (format 24-godzinny).
- Date: aktualna data.
- Info bar offset: przesunięcie górnej i dolnej „belki informacyjnej” w kierunku środka ekranu [piksel].
- Interface type: kolorystyka interfejsu użytkownika [Sepia/Blue].

Opcja „Info bar offset” służy do przesunięcia górnej i dolnej „belki informacyjnej”

w kierunku środka ekranu o zadaną liczbę pikseli obrazu (0...9). Funkcja ta może być użyteczna w przypadku, gdy pionowy wymiar okna przeznaczonego na montaż oryginalnego wyświetlacza TID jest mniejszy niż wysokość ekranu, co powodowałoby przesłanianie „belek” informacyjnych i zawartych tam informacji. Z kolei opcja „Interface type” pozwala na zmianę kolorystyki

całego interfejsu użytkownika z kolorów sepia na niebieski i odwrotnie. Wygląd przykładowego Menu zużycia paliwa w kolorystyce niebieskiej pokazano na **fotografii 16**.

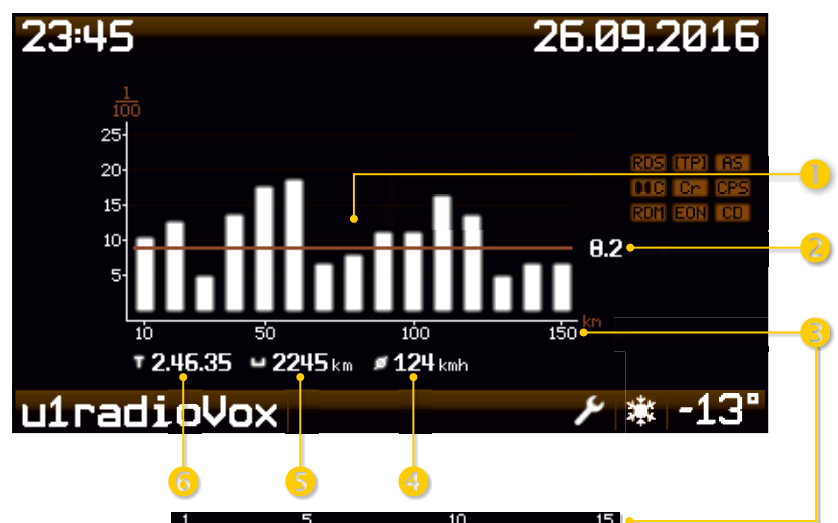
Wybrana opcja Menu jest podświetlana w kolorze białym, a wejściu w jej edycję towarzyszy negacja tła wybranej wartości (tło staje się białe, zaś wybrane wartości czarne). Wyjściu z trybu konfiguracyjnego



Rysunek 12. Wygląd „Menu zużycia paliwa”



Rysunek 13. Wygląd „Menu prędkości”



Rysunek 14. Wygląd „Menu statystyk”



towarzyszy zapisanie edytowanych wartości w nieulotnej pamięci EEPROM mikrokontrolera i/lub aktualizacja czasu i daty wbudowanego zegara RTC.

### Setup i tryb kalibracji stałej wtryskiwacza

Urządzenie multiTID wyposażono w specjalny, konfiguracyjny tryb pracy, dzięki któremu możemy określić pewne, niezbędne parametry regulacyjne nieodzowne z punktu widzenia funkcjonalności komputera pokładowego. Tryb ten uruchamiamy poprzez jednoczesne naciśnięcie przycisków „MENU/<” i „OK/>”. Za jego pomocą możemy określić wielkość następujących stałych niezbędnych w procesie obliczania zużycia paliwa, prędkości jazdy oraz drogi:

- Stała wtryskiwacza dla każdego z rodzajów paliwa (w [ml/min]): jest to wielkość charakterystyczna dla każdego wtryskiwacza elektronicznego wtrysku paliwa informująca nas o ilości paliwa, jakie może on wprowadzić do komory spalania w jednostce czasu (przy założeniu 100% czasu otwarcia zaworu i stałym, charakterystycznym dla każdego wtryskiwacza ciśnieniu zasilającym).
- Stała przetwornika drogi (impulsatora, w [imp/obr]): jest to wielkość charakterystyczna dla każdego impulsatora informująca nas o liczbie impulsów przypadających na 1 obrót koła (osi).
- Obwód opony (w [cm]).
- Liczba cylindrów (a więc i liczba wtryskiwaczy zamontowanych w silniku pojazdu).
- Pojemność zbiornika paliwa dla każdego z rodzajów (w [l]).

Ponadto tryb konfiguracyjny pozwala nam na ustawienie bieżącej daty i godziny wbudowanego zegara RTC. Wspomniane powyżej stałe można znaleźć w Internecie na forach o tematyce motoryzacyjnej lub elektronicznej, lecz wydaje się, że lepszym sposobem jest ich wyznaczenie empiryczne. Dla przykładu, obwód opony, a w zasadzie drogę, jaką pokona pojazd w czasie jednego, pełnego obrotu koła, możemy wyznaczyć dość łatwo. W tym celu zaznaczamy (np. kredą) najniższe położone miejsce styku opony z powierzchnią drogi, następnie standardowo obciążony pojazd przetaczamy, aby koło wykonało jeden, pełny obrót, po czym mierzymy pokonany odcinek drogi. Co oczywiste, wszystkie wprowadzone wartości zostaną zachowane w nieulotnej pamięci EEPROM urządzenia.

Niestety, jak pokazała praktyka, pewnych trudności może czasami nastęrczać znalezienie parametrów stosowanych w naszym pojeździe wtryskiwaczy, ponieważ są one często wykonywane na zamówienie producenta pojazdu i na próżno szukać ich oznaczeń na stronach producentów podzespołów. Na szczęście przewidziano pewien



Rysunek 15. Wygląd „Menu konfiguracyjnego”



Fotografia 16. Wygląd przykładowego „Menu zużycia paliwa” w kolorystyce niebieskiej urządzenia multiTID

mechanizm, za którego pomocą sterownik multiTID jest w stanie samodzielnie wyznaczyć poszukiwaną stałą na podstawie informacji o zużytym paliwie i pomiarze sumarycznego czasu wtrysków. Do tego celu przewidziano specjalny tryb kalibracyjny, który może być uruchomiony poprzez jednoczesne naciśnięcie przycisków „MENU/<” i „OK/>”, co zostanie zasygnalizowane wyświetleniem ikony na dolnej belce informacyjnej graficznego interfejsu użytkownika. Ponowne wykonanie wspomnianych czynności powoduje obliczenie żądanej stałej wtryskiwacza a następnie opuszczenie procesu kalibracji. Co oczywiste, do czasu zakończenia procesu kalibracji nie są dostępne następujące wartości obliczeniowe: chwilowe i średnie zużycie paliwa, ilość paliwa dostępnego w baku pojazdu (jest to sygnalizowane wyświetleniem symboli „-” w odpowiednich polach wspomnianych wartości) oraz menu statystyk spalania. Aby przeprowadzenie procesu kalibracji miało w ogóle sens, należy zastosować następujący algorytm postępowania:

- Zużyć całe, dostępne paliwo, aż do zaświecenia się lampki sygnalizującej tzw. rezerwy paliwa.
- Zatankować 20 l paliwa.
- Uruchomić procedurę kalibracji.
- Zużyć całe, dostępne paliwo (zatankowane wcześniej 20 l), aż do ponownego

zaświecenia się lampki sygnalizującej tzw. rezerwy paliwa.

- Zakończyć procedurę kalibracji.

Po wykonaniu tych czynności układ multiTID obliczy i zapisze, w nieulotnej pamięci EEPROM mikrokontrolera, wartość stałej wtryskiwacza, po czym przejdzie do normalnego trybu pracy. Oczywiście stałą taką wyznaczamy dla każdego z rodzajów paliwa oddzielnie, co oznacza, że w czasie aktywnego procesu kalibracji nie wolno zmieniać rodzaju paliwa. Gdyby obliczona przez sterownik wartość stałej wtryskiwacza powodowała zaniżanie lub zawyżanie rzeczywistego spalania paliwa, w każdej chwili możemy dokonać odpowiedniej korekty poprzez wejście w menu konfiguracji urządzenia i zwiększenie (przy zaniżaniu spalania) lub zmniejszenie (przy zawyżaniu spalania) wspomnianej wartości. Należy zaznaczyć, że tak jak w przypadku oryginalnych rozwiązań typu „komputer pokładowy”, obliczane wartości zużycia paliwa są obarczone pewnym błędem wynikającym choćby z założenia stałego ciśnienia zasilającego wtryskiwacz czy też z zaokrąglenia obliczeniowych. Testy praktyczne pokazały, że maksymalny błąd pomiarowy jest na poziomie 0,5 l na całą pojemność baku pojazdu, czyli ok. 1%.

Robert Wołgajew, EP