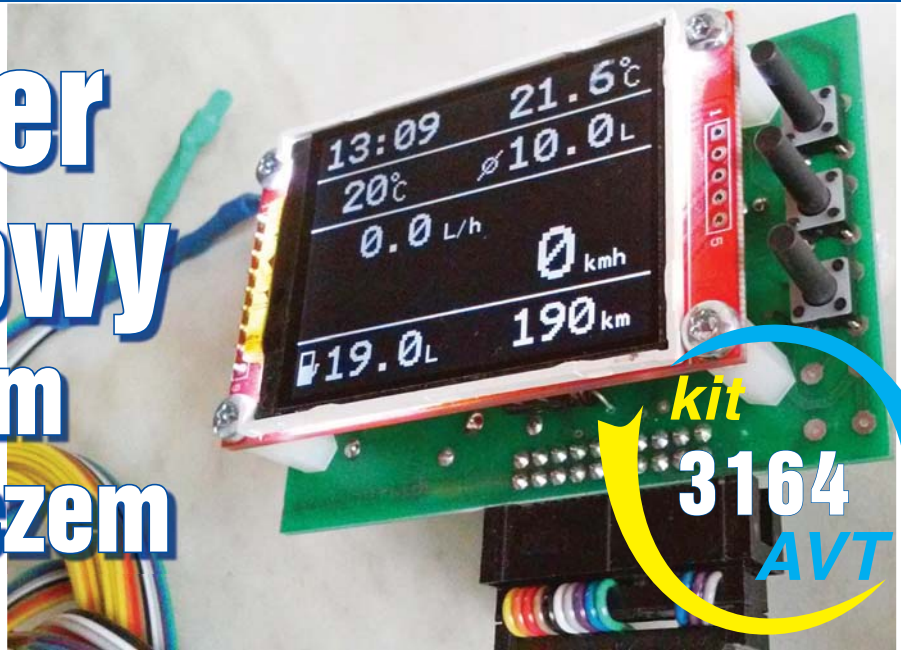


Komputer pokładowy z kolorowym wyświetlaczem



Wszystkie nowe samochody naszpikowane są elektroniką i między innymi mają tzw. komputer pokładowy. Jednak na polskich drogach wciąż jeździ mnóstwo samochodów, które takiego wyposażenia nie mają. Taki komputer można też wykonać samodzielnie. Wcale nie jest to tak trudne, jak mogłoby się wydawać. W EdW przedstawiłem już dwa projekty komputerów samochodowych:

- komputer samochodowy AVT-3095 (<http://serwis.avt.pl/manuals/AVT3095.pdf>)
- graficzny komputer samochodowy - kit AVT-3127 (<http://serwis.avt.pl/manuals/AVT3127.pdf>)

Teraz chciałbym zaprezentować kolejną wersję, wykorzystującą pełnokolorowy wyświetlacz graficzny.

Nieprzypadkowo podaję adresy, gdzie można znaleźć PDF-y z opisem tych dwóch wcześniejszych projektów (AVT3095.pdf, AVT3127.pdf). Teraz, w trzecim projekcie nie powtarzam niektórych bardzo ważnych informacji. Zwłaszcza w pierwszym projekcie zawarłem kluczowe informacje o tym, jak dzia-

ła komputer samochodowy, z jakimi czujnikami współpracuje, jakie sygnały daje poszczególne czujniki i jak te sygnały podać do mikrokontrolera i wykorzystać.

Jeszcze raz powtarzam, że w sumie wcale nie jest to trudne. Nie trzeba się bać – realizacja komputera samochodowego oraz jego dołączenie do instalacji samochodu i kalibracja, wcale nie są trudne. Oczywiście trzeba zachować ostrożność, żeby czegoś nie uszkodzić, ale generalnie elementy i bloki „samochodowe” są odporne. Naprawdę warto wzbogacić swoje auto o komputer pokładowy, zwłaszcza taki z pełnokolorowym wyświetlaczem.

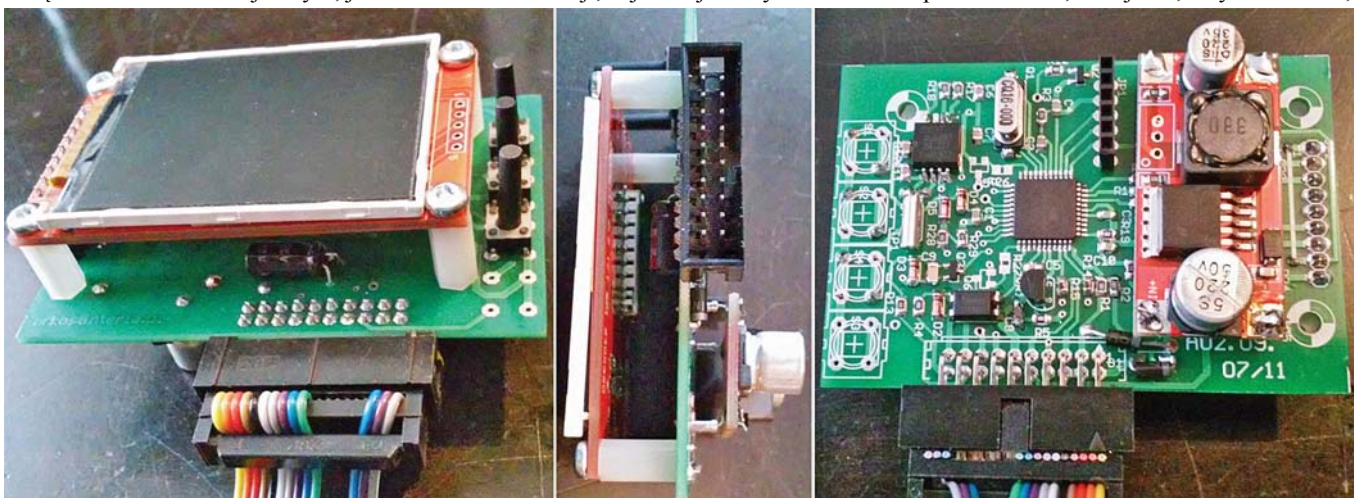
W praktyce podstawowym zastosowaniem komputera jest pomiar spalania paliwa, co powinno pokazać, jak mądrze użytkować auto, by jak najwięcej zaoszczędzić.

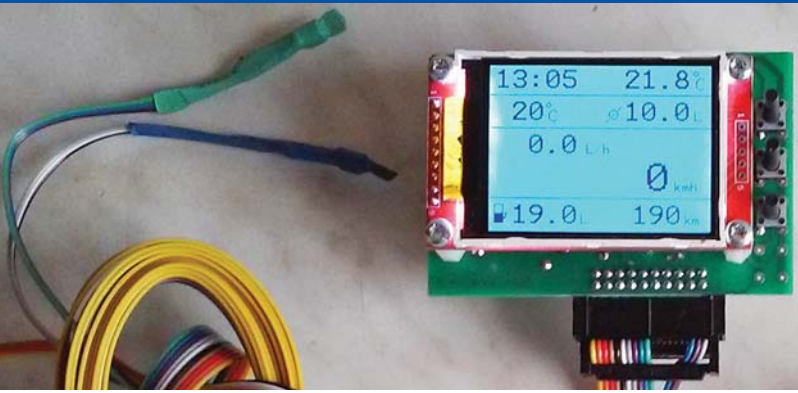
Opisywany komputer ma też o wiele więcej funkcji, takich jak podwójny termometr, zegarek, datownik, pomiar napięcia instalacji, rejestracja trasy i inne. Do

tego wykorzystanie kolorowego wyświetlacza daje wspaniały efekt, który będzie cieszył serce właściciela i co dla niektórych istotne: „bielił oko” zazdrośnym.

Najbardziej atrakcyjnym elementem jest oczywiście kolorowy wyświetlacz. W moim projekcie wykorzystałem kolorowy wyświetlacz TFT ze sterownikiem *ili9341*, który nabyłem nie jako pojedynczy element, tylko jako moduł na płycie przeznaczonej dla Arduino.

Czasy się zmieniły i dziś budując urządzenie elektroniczne, najpierw rozglądamy się po aukcjach, czy nie ma już podobnych, gotowych, kompletnych. Jeśli zdecydujemy się zbudować większe urządzenie zawierające wiele elementów, to zwykle rozglądamy się, czy są jakieś interesujące elementy lub moduły, które można wykorzystać. Po przejrzaniu ofert okazuje się, że jest mnóstwo gotowych przeróżnych półproduktów – modułów elektronicznych, na przykład przetwornice, czujniki, wyświetlacze,





więc przy określeniu zapotrzebowania na urządzenie, zbudowanie go będzie mocno ułatwione. Wystarczy w wyszukiwarce wpisać choćby hasło *arduino*, a wyników jest sporo. Budowa z modułów to jak zabawa klockami Lego, to takie elektroniczne klocki.

Tak jest z wyświetlaczem LCD TFT ze sterownikiem ili9341. Delikatny LCD o przekątnej 2,2 cala jest umieszczony na mocnej płytce PCB, co oczywiście ogromnie ułatwia wykorzystanie. Całość ma małe wymiary i taki moduł można łatwo połączyć z płytką główną goldpinami lub taśmą wielożyłową, co pozwoli na montaż samego wyświetlacza w ciasnym miejscu kokpitu auta. Przesyłanie danych na wyświetlacz odbywa się przez interfejs SPI, co upraszcza połączenie z procesorem, ale też ma wady: niską prędkość przesyłu danych i jak okazało

się w praktyce, nieco słabszą odporność na zakłócenia. W tym projekcie optymalna szybkość obsługi LCD eliminuje wyżej wymienione problemy. Warto wiedzieć, że są też większe LCD 2,4 i 2,8 cala z tym samym sterownikiem, a wszystkie mają gniazdo na karty pamięci SD, a te większe mają także panel dotykowy w postaci nakładki. Na karcie pamięci można zapisać pliki graficzne, które procesor przesyła do LCD, aby je wyświetlić. Oszczędza to wiele cennej pamięci procesora, bowiem pliki graficzne są dużych rozmiarów w porównaniu do rozmiarów pamięci flash procesora. Jednak w tym projekcie nie korzystam z karty pamięci ani panelu dotykowego. Ceny modułów są atrakcyjne, zapewne ze względu na ich dostępność i popularność na rynku. Dla elektronika amatora to raj...

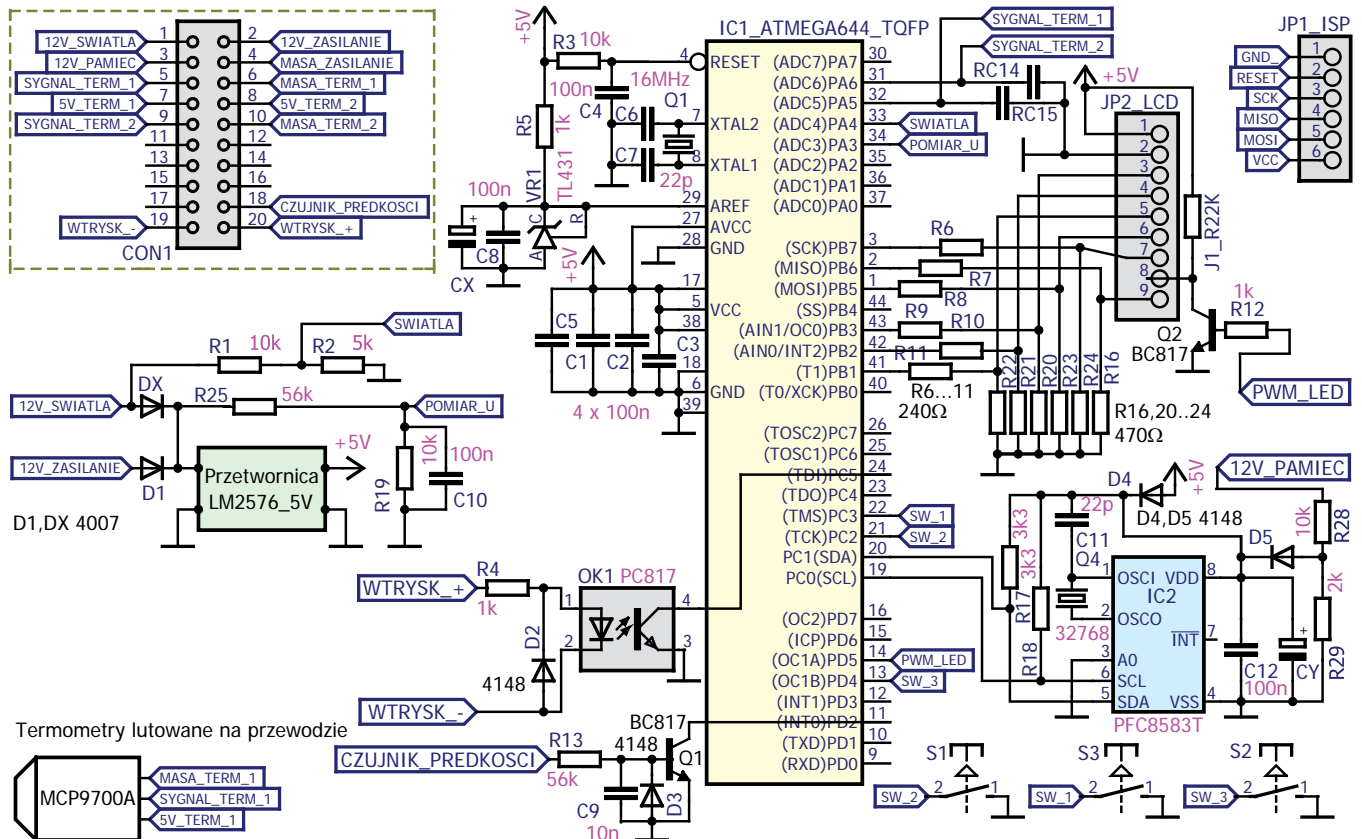
Drugi moduł wykorzystany w opisywanym projekcie to zasilacz – przetwornica DC z układem LM2576. Koszt LCD 2,2' to 30zł, natomiast przetwornicy tylko 4zł! Nieopłacalne jest więc projektowa-

nie zasilacza na płytce. Tu dodam, że w projekcie początkowo, na etapie realizacji prototypu, używałem stabilizatora liniowego LM7805, w budowie DPAK oraz TO-220. Po kilku minutach pracy stabilizatory robiły się ciepłe (DPAK gorący) i wtedy działy się dziwne rzeczy z wyświetlaczem. Nie chciał startować po sieci, ekran „krzaczył”. Wprawdzie pobór prądu całości to ok. 100mA, a pomimo to 1,5-amprowe stabilizatory bez radiatora (w celu minimalizacji płytki) nie dawały sobie rady.

Postanowiłem więc wykorzystać gotowy moduł wyświetlacza i gotowy zasilacz impulsowy. Pozostało jedynie zaprojektować płytę bazową. Dzięki użyciu procesora ATmega644 z 64kb pamięcią na kod programu mamy sporo miejsca do popisu, programując w BASCOM-ie, który, jak wiadomo, daje niezbyt doskonały kod maszynowy.

Procesor jest połączony z wyświetlaczem LCD przez dzielniki rezystorowe, aby dopasować poziomy logiczne. Zapewne użycie zasilania 3,3V zamiast 5V i dla procka, i dla LCD ułatwiłoby montaż oraz wpłynęło na szybkość odświeżania ekranu, lecz „z rozpędu przy projektowaniu” zostało 5V. Sugerowałem się także możliwością pracy procesora z kwarcami wyższej częstotliwości, do czego wyższe napięcie 5V jest niezbędne.

Rys. 1



Biblioteka obsługująca LCD, zawierająca potrzebne instrukcje dla sterownika ILI9341, została znaleziona na oficjalnym forum BASCOM. Takich skarbów jest tam sporo, trzeba tylko poszukać. Krótkie gotowe polecenia ułatwiają korzystanie z tego „gotowca”. Natomiast specjalistyczne ikonki, które się wyświetlają w tym komputerku, np. stacja paliw, klucz mówiący o inspekcji, to zmodyfikowana czcionka. A konkretnie nieużywane znaki w pliku czcionki, które zostały zmodyfikowane – przerobione na potrzeby tego projektu.

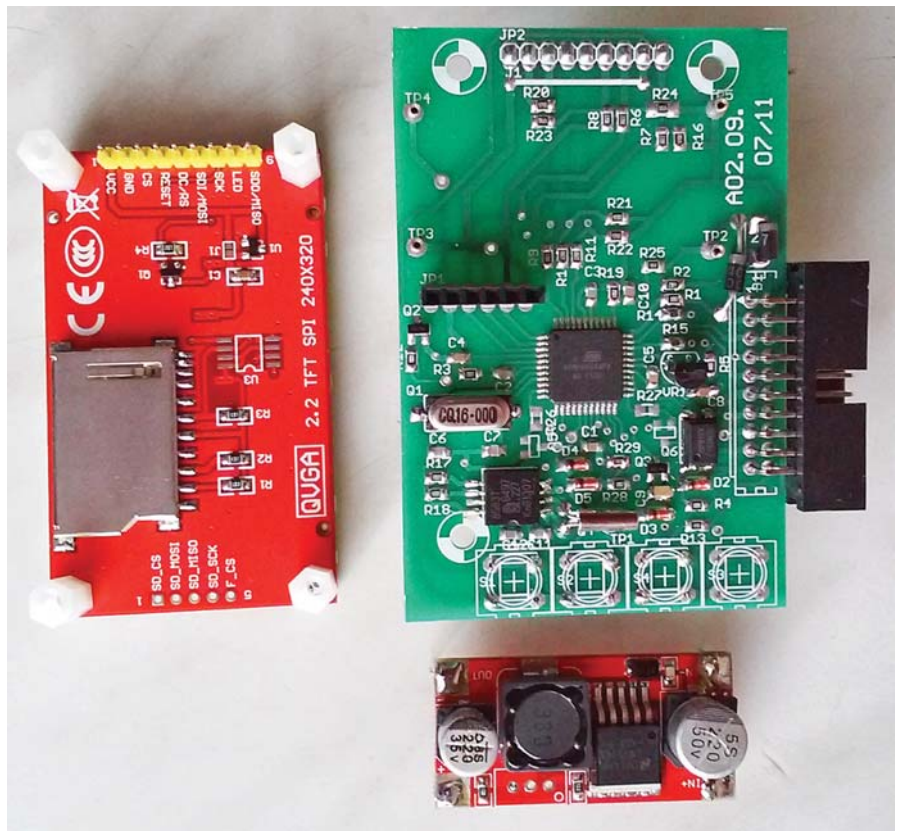
Opis układu

Rysunek 1 pokazuje schemat ideowy układu. Sercem projektu jest pojemny procesor ATmega644 w obudowie TQFP44 z 64-kilobajtową pamięcią programu. Program w chwili obecnej zajmuje ok. 45kB, więc można go jeszcze rozbudować, jeśli ktoś zechce.

Diody D1 i DX (DX została wprowadzona już po zaprojektowaniu płytki i należy ją przylutować, kierując się schematem) doprowadzają zasilanie do przetwornicy ustawionej na 5V.

Rezystory R1, R2 to dzielnik, który dopasowuje wartość napięcia podawanego do procesora i wykrywa włączenie świateł. R25, R19, C10 obniżają napięcie zasilania do poziomu nie większego niż napięcie referencyjne 2,5V i dzięki nim procesor mierzy wartość napięcia zasilania. R5, VR1 (TL431), C8, Cx (poprawia stabilność ADC) to obwód napięcia odniesienia 2,5V. R4, D2, OK1 (PC817) to obwód wykrywający włączenie wtryskiwacza paliwa.

Procesor w przerwaniu pojawiającym się co 0,25ms sprawdza stan pinu PC.5, a niski stan inkrementuje zmienną. Po odmierzeniu sekundy zmienna zawiera sumaryczny czas otwarcia wtryskiwacza paliwa. Przemnożenie przez wartość „Stałej paliwa” (liczba wprowadzana przez użytkownika w menu ustawień) oraz okresu wystąpienia przerwania da wynik w postaci zużytego paliwa.

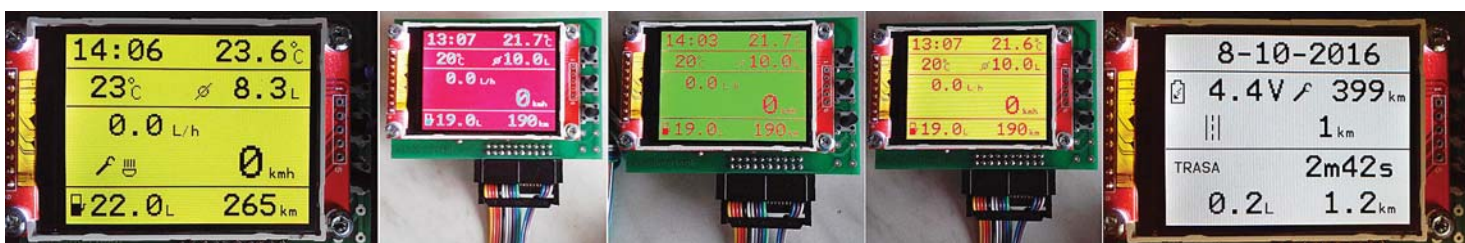


Elementy R13, C9, D3, Q1 dbają o to, żeby podanie sygnału prędkości auta powodowało uruchomienie przerwanienia INTO, zliczającego impulsy z czujnika. Liczba zliczonych impulsów jest przemnażana przez „Stałą drogi” – jest to liczba charakterystyczna dla danego samochodu, która wskazuje, co ile milimetrów przejechanej drogi występuje impuls z czujnika. Kolejne elementy: PCF8583, C11, Q4, R17, R18, C12, Cy, D4, to zegar z datownikiem. R28, R29, D5 to obwód, który powinien być stale pod napięciem 12V w celu podtrzymania pamięci zegara.

R6...11, R16, R20...R24 tworzą translator poziomów sygnału 5V/3,3V dla wyświetlacza podłączanego do złącza JP2. Im mniejsza wartość rezy-

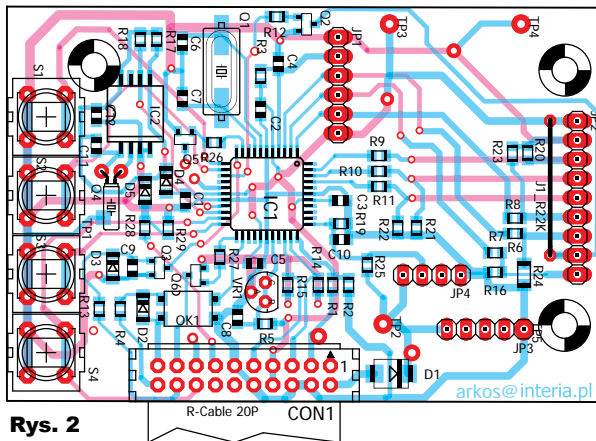
stancji, tym szybsza transmisja danych na LCD, ale też większy pobór prądu. Elementy oznaczone RC14, RC15 to kondensator 100nF równoległe z rezystorem 100kΩ. Kondensator filtruje sygnał z termometru, ogranicza „pływanie” odczytanej temperatury. Rezystor 100kΩ ściąga do masy wejście pomiarowe. Jest to przydatne, kiedy jest przerwa w połączeniu termometru z procesorem – wtedy pokazywany wynik pomiaru wyniesie -50.0°C. I wreszcie R12, Q2, J1_R22k to obwód sterujący jasnością wyświetlacza, gdzie sygnał PWM z procesora włącza tranzystor Q2.

A oto garść szczegółów dotyczących sterowania wyświetlacza. Oto komendy obsługujące LCD.



W pierwszej kolejności wybieramy tryb pracy: *widok, portret*.

1. Call Lcd_init() – inicjalizacja wyświetlacza, niezbędna komenda do wystartowania.
2. Call Lcd_clear(black) – czyszczenie całego LCD z wybranym zdefiniowanym wcześniej kolorem – tutaj na czarno.
3. Call Lcd_text(„Hello World” , 30 , 20 , 2 , Blue , Black , 2) – wyświetla tekst, zmienne zamienione na tekst poleceniem a = str(zmienna liczbowa).



po poleceniu a = str(zmienna liczbowa). To bardzo ciekawa funkcja: 30, 20 – to orientacja na lcd wyświetlenia tekstu, 2 – to wybór typu czcionki, blue – to kolor czcionki, black – kolor tła pod napisem, 2 – tu ciekawe, powiększenie wybranej czcionki. Przy wybranej czcionce 12x16, wybranie ostatniego parametru 1 spowoduje wyświetlenie znaków właśnie rozmiarem 12x16. Natomiast parametr 2 spowoduje powiększenie dwukrotne czyli 24x32. Przy parametrze równym 3 będzie 36x48. Zaleca: używanie wielu różnych czcionek zapycha pamięć flash procesora. Jedna czcionka programowo powiększana daje oszczędność miejsca.

4. Restore Mese, Call Lcd_show_bgc(95 , 100) – wyświetla kolorowy obrazek, wcześniej sformatowany w pliku .bgc i dołączony na końcu listingu: Mese:, \$bgf „mcse.bgc”
 5. Call Lcd_line(0 , 0 , 319 , 0 , 2 , Red) – rysuje linię o początku 0,0 końcu 319,0 grubości 2, w kolorze red (czerwonym).
 6. Call Lcd_box(60 , 50 , 200 , 140 , 1 , Red , Cyan0) – rysuje prostokąt, 1 – pełny, 2 – pełny z innym kolorem ramki, 3 – sama ramka.
 7. Call Lcd_circle(160 , 120 , 20 , Red) – okrąg o środku 160, 120, promieniu 20 kolor red.
 8. Call Lcd_fill_circle(60 , 60 , 30 , Green) – okrąg wypełniony kolorem.
- Tyle o układzie i jego składnikach.

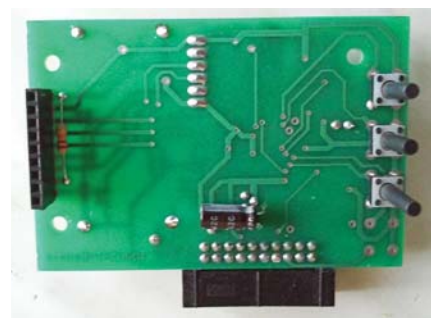
Montaż i uruchomienie

Układ można zmontować na płycie głównej, pokazanej na rysunku 2. Montaż w znacznej większości to elementy SMD w obudowach 0805. Montaż wbrew pozorom nie jest trudny. Nawet dla osób słabo zaawansowanych wlutowanie procesora nie powinno być trudne, jeśli podejdziesz do tego następująco: ustawiamy procesor na padach, lutujemy jego jedną nogę i sprawdzamy poprawność ustawienia. Wtedy

zalewamy cyną wszystkie nogi (powodujemy totalne zwarcie). Następnie używamy topnika w postaci pasty (a nie krystalicznej kalafonii) i posługujemy się cienkim miedzianym przewodem – linką (lub miedzianym ekranem z przewodu ekranowanego) do usunięcia nadmiaru cyny. Taką linkę (ekran) najpierw maczamy w paście, a następnie grzejąc ją grottem, przykładamy do wcześniej zalanych nóg procesora. Nadmiar cyny pobieli przewód miedziany, ściągając z pinów ATmegi większość cyny. To bardzo łatwa i skuteczna metoda montażu kostek SMD. Należy jedynie pilnować, aby nie przegrzać układu.

Pozostałe elementy zapewne też nie sprawią trudności. Pomocą będą fotografie oraz rysunki. W Elportalu, wśród materiałów dodatkowych do tego numeru, można znaleźć obszerną dokumentację płytki oraz definicję specyficznych znaków czcionki.

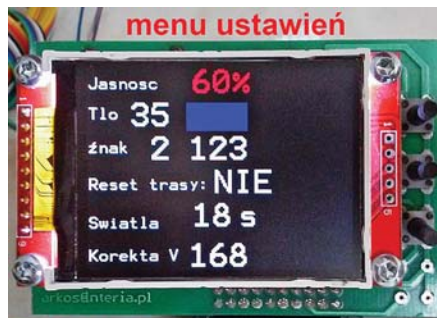
Uwaga! Na płycie są przewidziane elementy, które na razie nie są wyko-



rzystane – można je w przyszłości zastosować i wykorzystać według swoich potrzeb czy pomysłów

Na koniec zostawiamy montaż modułu przetwornicy na gólinach. Wcześniej, przed zamontowaniem w komputerku, należy ją podłączyć do zasilacza i posługując się multimetrem, ustawić na jej wyjściu 5,0V. Dopiero po upewnieniu się o prawidłowym napięciu wyjściowym montujemy ją na płycie komputera.

Po poprawnym zmontowaniu i zaprogramowaniu procesora układ powinien pracować od razu. Można wtedy ustawić kolory i jasność, zresetować trasę. Kalibracja spalania jest opisana dalej.



Korzystanie z komputerka wygląda następująco. Do obsługi wykorzystane są 3 przyciski zwierne.

Krótkie naciśnięcie na S1 zmienia ekran danych. Są dwa ekrany predefiniowane z wyświetlanymi informacjami. Pokazują to zamieszczone fotografie.

Krótkie naciśnięcie na S2 to wejście w menu wyboru parametrów: kolor tła, kolor znaków, jasność LCD, reset trasy (spalone paliwo, czas jazdy, dystans, średnie spalanie), korekta mnożnika wyświetlania zmierzonego napięcia, okres alarmu o niewłączonych światłach. Zmiany nastaw dokonuje się przyciskami S1, S3.

Krótkie naciśnięcie na S3 to wejście w menu wprowadzania zatankowanego paliwa. Zmiana za pomocą S1, S2.

Jeśli chodzi o ustawienia zegara, adaptację stałej drogi i stałej paliwa, to należy wyłączyć zasilanie, nacisnąć i trzymać S1, a następnie włączyć zasilanie – wtedy nastąpi wejście do ukrytego menu nastaw. Dostępne będą nastawy: godzin, minut, dnia, miesiąca, roku, stałej paliwa, stałej drogi, licznika kilometrów oraz okresu inspekcji.

A teraz ważna **kalibracja**: po uruchomieniu komputerka, wstępne ustawienie stałej drogi i stałej paliwa pozwala na test pomiaru spalania. Najpierw należy wyregulować (zmniejszyć lub zwiększyć) stałą drogi tak, aby prędkość na komputerku

Wykaz elementów

R1,R3,R19	10kΩ
R2	5kΩ
R4,R5,R12	1kΩ
R6...R11	240Ω
R13,R25	56kΩ
R16, R20...24	470Ω
R17,R18	3,3kΩ
C1...C5, C8,C10,C12	100n
C6,C7,C11	22p
C9	10n
Cx,Cy	100u/16V
RC14,RC15	patrz tekst
IC1	ATmega644
IC2	PCF8583

Q1,Q2	BC817
OK1	PC817
D1,Dx	4007
D2,D3,D4,D5	1N4148
VR1	TL431
MCP9700A	
CON1	IDC20 na taśmę, IDC20 kątowe
JP1,JP2	gniazdo goldpin
Q1	16MHz
Q4	32768Hz
S1,S2,S3	mikroswitch 6x6
Taśma 20 żył ok. 3 metrów	(wystarczy 14 żył, patrz fotki)

Płytką drukowaną jest dostępna w Sklepie AVT jako AVT3164

zgadzała się z prędkością auta wskazywaną przez fabryczny prędkościomierz.

Następnie kalibrujemy stałą paliwa. Tankujemy np. 10 litrów LPG lub benzyny, wprowadzamy w komputerze ilość zatankowaną, następnie korzystając z auta, obserwujemy wyświetlacz. Kiedy paliwo się skończy, to jeśli komputer pokaże, że paliwo powinno jeszcze być, a faktycznie go nie ma – wtedy musimy zwiększyć stałą paliwa np. o 50. Jeśli paliwo się jeszcze nie

skończyło, a komputer już pokazuje zero – wtedy zmniejszamy stałą paliwa. Operację taką przeprowadzamy kilkakrotnie do uzyskania jak najwiernejszego wyniku pomiaru spalania. Mnie udało się uzyskać dokładność lepszą niż 0,5 litra „na setkę”.

W razie potrzeby mogę wyjaśnić wątpliwości lub udzielić dodatkowych informacji, czy to na łamach EdW, czy indywidualnie.



Arkadiusz Krzyjszczyk
arkos@interia.pl