



kit

2920

AVT



# Analizator stanów logicznych

Analizator stanów logicznych, jak sama nazwa wskazuje, służy do analizowania przebiegów cyfrowych. Może nie jest to przyrząd tak ważny jak oscyloskop, niemniej jednak przyda się każdemu elektronikowi zajmującemu się cyfrówką oraz mikrokontrolerami. Za dobry analizator trzeba trochę zapłacić, na co nie zawsze stać początkujących elektroników. Na szczęście bardzo łatwo można wykonać takie urządzenie samemu, w przeciwieństwie do większości przyrządów pomiarowych oraz innych precyzyjnych układów analogowych. Te z reguły potrzebują żmudnego uruchamiania i regulacji. Dlatego analizator stanów logicznych może zbudować nawet początkujący elektronik, który dopiero zaczyna swoją przygodę w tej dziedzinie. Mimo prostoty i niskiej ceny dzięki zastosowaniu mikrokontrolera AVR można uzyskać imponujące parametry. Prezentowane urządzenie ma 16 kanałów pomiarowych i umożliwia pomiar z częstotliwością próbkowania maksymalnie 16MHz. Co prawda, ze względu na możliwości procesorów AVR, przy tej częstotliwości ograniczona jest liczba kanałów i można pobrać tylko 32 próbki na kanał. Ale za to dla częstotliwości 5,33MHz można już pobrać 2000 próbek – o tym dokładnie w dalszej części artykułu. Przyrząd ma także dodatkowe funkcje, takie jak rejestrowanie przebiegów analogowych, sonda logiczna, pomiar częstotliwości oraz generowanie zapamiętanego przebiegu. Może więc służyć też do testowania interfejsów i pomiaru innych wartości.

## Funkcje i parametry urządzenia

### Analizator:

- 10–16 kanałów: 1000 próbek/kanał, 2MS/s max próbkowanie,
- 6–8 kanałów: 2000 próbek/kanał, 5,33MS/s max (32 próbki/kanał, 16MS/s max),
- 4 kanały: 4000 próbek/kanał, 2,67MS/s max,
- 2 kanały: 8000 próbek/kanał, 2,67MS/s max,
- 1 kanał: 16000 próbek, 2,67MS/s max,
- podstawa czasu od 200ms (80S/s) do 1us (16MS/s),
- wyzwalenie: zboczem, poziomem, sygnałem.

### Rejestrator:

- 8 kanałów: 250 próbek/kanał, 1kS/s,
- 4 kanały: 500 próbek/kanał, 2kS/s,
- 1 kanał: 2000 próbek, 5kS/s,
- wyzwalenie: zboczem/sygnałem,

### Dodatkowe funkcje:

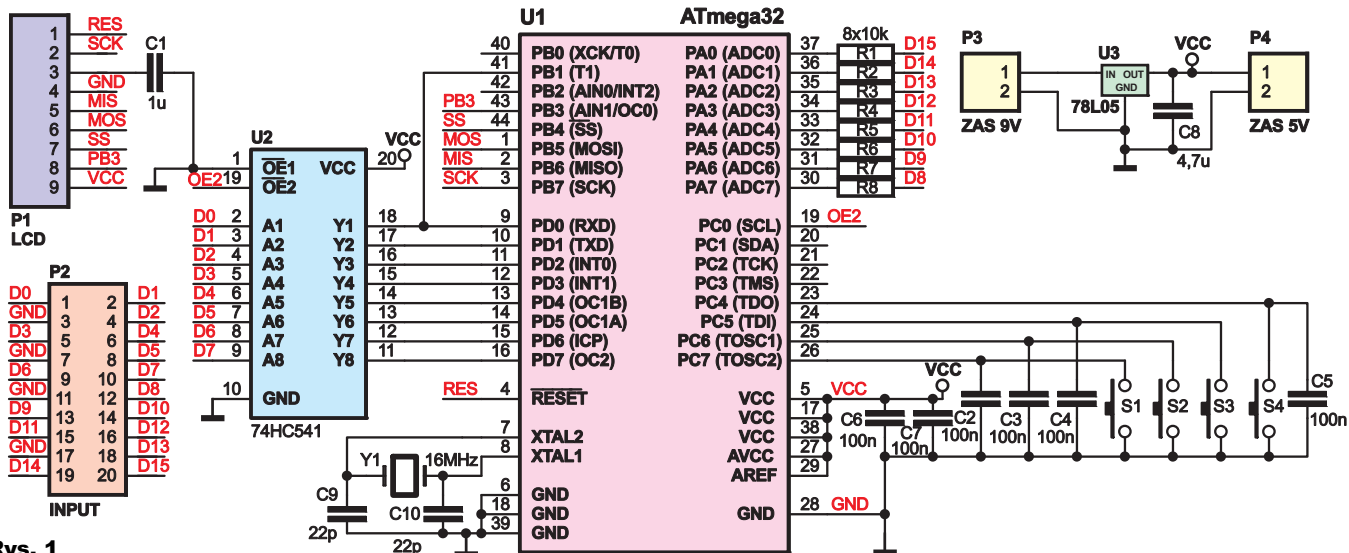
- Generator: 8 kanałów, 2000 próbek, podstawa czasu od 200ms (80S/s) do 20us, (800kS/s), wyzwalenie zboczem/sygnałem,
- sonda logiczna 16 kanałów,
- częstotściomierz do 8MHz.

Analizator, rejestrator, a także generator można wyzwalać zboczem opadającym/narastającym na wybranym kanale lub „sygnałem”. Nie chodzi tu o sygnał jako ciąg następujących po sobie stanów wysokich lub niskich, ale o równoczesne pojawienie się odpowiedniej kombinacji bitów na wybranych wejściach. Można powiedzieć, że jest to wyzwalenie wektorem, ustawianym przez użytkownika.

Jak na mały mikrokontroler AVR, parametry oraz liczba funkcji jest całkiem spora. Zastosowano w nim nietypowe rozwiązania programowe, aby uzyskać takie efekty. Gorąco zachęcam więc do lektury nawet osoby, które mają już jakiś analizator, ale zajmują się mikrokontrolerami AVR.

## Opis układu

Schemat układu znajduje się na **rysunku 1**. Jak widać, nie jest zbyt skomplikowany. Sercem analizatora jest mikrokontroler AVR ATmega32. Wybór tego procesora był podyktowany przede wszystkim jego dużą pamięcią danych oraz w miarę przyjazną obudową (łatwo przylutować). Zajmuje się on pobieraniem i przechowywaniem przebiegów cyfrowych, obsługą wyświetlacza oraz klawiatury. Układ 74HC541 pełni rolę bufora wejściowego. Nie jest on konieczny, jednak warto go zastosować, na wszelki wypadek. W razie podania zbyt wysokiego napięcia ryzykujemy spaleniem tego bufora, a nie procesora AVR, który jest nieco droższy. Natomiast kanały od 9 do 16 zabezpieczono rezystorami o wartości 10kΩ. Wprawdzie takie rozwiązanie trochę tłumi szybkie przebiegi, ale jest to nawet zaletą, ponieważ maksymalna częstotliwość próbkowania dla tych kanałów wynosi tylko 2,67MHz. Dla kanałów 1–8 wynosi ona 16MHz i dlatego zdecydowałem się tutaj na bufor. Do mikrokontrolera standardowo podłączony jest kwarc, w tym przypadku 16MHz, aby uzyskać jak największą szybkość. Wyświetlacz pochodzi z telefonu Nokia 3410 i można go kupić na Allegro za kilkanaście złotych. Ma on dużą rozdzielczość 96x65 pikseli, co zapewnia komfortową pracę z urządzeniem. Kondensator C1 współpracuje z



Rys. 1

wewnętrzna przetwornicą wyświetlacza podwyższającą napięcie. Kondensatory C2...C5 eliminują drgania zestyków przycisków S1...S4. Natomiast C6, C7 oraz C8 filtrują napięcie zasilania. Stabilizator U3 jest opcjonalny, jeśli ktoś chce zasilać układ np. z baterii 9V. Nie musi być on na napięciu 5V, wystarczy 3,3V. Przy zasilaniu analizatora z zewnątrz bądź też napięciem poniżej 5V (np. z akumulatora telefonu, baterii litowej) można zrezygnować z układu U3. Wtedy należy wykorzystać złącze P4. Zaraz, coś tu nie gra. Przecież w nocie katalogowej jest wyraźnie napisane, że procesor ATmega32 powinien być zasilany napięciem 4,5...5,5V, a tu jest mowa o napięciach rzędu 3V. No cóż, te 4,5...5,5V to raczej optymalne napięcie – jego obniżenie nie powoduje żadnych problemów, nawet w trakcie programowania. Dodam, że mój egzemplarz świetnie się sprawował nawet przy napięciu 2,7V, przy niższym zadziałał już układ BOR. Dlatego w przypadku zasilania napięciem poniżej 4V należy pamiętać, aby BOR był ustawiony na napięcie 2,7V, a nie 4V. Służy do tego fusebit o nazwie *BODLEVEL*, który standardowo ustawia niższą wartość napięcia zadziałania. Złącze P2 oprócz właściwych wejść analizatora ma także wyprowadzone 4 piny dla masy. Są one tak rozmieszczone, aby trochę wytłumić wzajemne zakłócanie się wejść. Pozwala to na zastosowanie dłuższego przewodu.

## Program

Teraz chyba najciekawsza część artykułu, czyli jak to wszystko funkcjonuje. Program zajmując

30KB pamięci oraz korzysta z prawie całej dostępnej pamięci RAM – można go ściągnąć z Elportalu. Odbiega „trochę” od standardów pisania w języku C, za to działa w miarę szybko i poprawnie. Nie obeszło się także bez wstawek assemblerowych. Dotyczy to w szczególności pliku *pomiar.h*, gdzie większość funkcji napisanych jest właśnie z wykorzystaniem assemblera. Ale zacznijmy od omówienia podstaw, czyli głównego pliku programu *main.c*. Zawiera on deklaracje dwóch dużych tablic do zapisywania wyników pomiarów oraz kilku tablic pomocniczych. Poniżej są przerwania odpowiedzialne za działanie częstotściomierza. Dalej znajduje się właściwa funkcja *main*, na początku której inicjowane są wszystkie porty, zmienne i wyświetlacz. Główna część programu składa się z kilku pętli typu *while*. Każda pętla odpowiada za obsługę poszczególnych podprogramów, a więc analizatora, ustawień, rejestratora, sondy logicznej, częstotściomierza oraz generatora. Takie podejście pozwala przyspieszyć nieco działanie programu oraz nie zaciemnia go zbyt. Funkcje wyświetlające tekst oraz wszelkie funkcje grafiki wprowadzają dane od razu na LCD, bez korzystania z dodatkowego bufora programowego.

Niestety powoduje to znaczne skomplikowanie programu oraz mniejszą elastyczność tych funkcji. Ale za to możemy cieszyć się całymi 780 bajtami wolnej pamięci RAM. To pozwala zarezerwować dużo większy obszar na próbki danych analizowanego sygnału. W ten sposób mamy do dyspozycji całe 2000 bajtów, a nie skromne 1220 bajtów. ATmega32 ma 32KB pamięci programu i można sobie pozwolić na dłuższy kod wynikowy funkcji graficznych, otrzymując w zamian więcej miejsca na próbki sygnału. Pewną wadą takiego rozwiązania jest dość trudna modyfikacja tak napisanego programu. Wystarczy spojrzeć, jak wyglądają funkcje wyświetlające przebiegi analizatora. Znajdują się one w pliku *Grafika.c* pod nazwami *lcd\_w16..lcd\_w8* oraz *lcd\_w*. Fragment przedstawia **listing 1**. Jest to wysłanie jednej linii bajtów na wyświetlacz z narysowanymi dwoma przebiegami analizatora. Wprawdzie całą funkcję można by uprościć, jednak zależało mi bardziej na szybkości wykonywania niż na objętości kodu. Dodatkowo pojawił się problem przy wyświetlaniu 10 i 12 przebiegów, gdyż każda pętla wysyłania bajtów ma tu inną budowę. Najciekawsze w tych funkcjach jest wyświetlanie napisów na dole ekranu.

```

lcd_GoTo(0,0); //ustaw kursor na pozycji początkowej
uint8_t rej;
lcd_Send(0); //wyświetlenie cyfry "1" określającej numer kanału
lcd_Send(81);
lcd_Send(103);
lcd_Send(0);
lcd_Send(0);
lcd_Send(34); //koniec wyświetlania cyfry "1"
for(uint8_t i=0; i<90; i++) //kan0,1 wyświetlanie 1 linii na wyświetlaczu
{
    rej=0;
    if((*wsk)&1)rej=1; else rej=4; //jeśli na kan0 '1' to ustaw
    punkt w stan wysoki, jeśli nie, to w stan niski
    if((*wsk++)&2)rej|=16; else rej|=64; //jeśli na kan1 '1' to
    ustaw punkt w stan wysoki, jeśli nie, to w stan niski
    if((*wsk)&1) && (rej&4)rej|=3; //jeśli na kan0 przejście
    z 0->1, to ustaw punkt przejściowy
    if(!((*wsk)&1) && (rej&1))rej|=6; //jeśli na kan0 przejście
    z 1->0, to ustaw punkt przejściowy
    if((*wsk)&2) && (rej&64)rej|=48; //jeśli na kan1 przejście
    z 0->1, to ustaw punkt przejściowy
    if(!((*wsk)&2) && (rej&16))rej|=96; //jeśli na kan1 przejście
    z 1->0, to ustaw punkt przejściowy
    if(!(i&15))rej|=136; //wyświetl punkt podziałki
    lcd_Send(rej); //wyslij gotowy bajt na wyświetlacz
}
    
```

Listing 1

Każdy bajt przedostatniej linii składa się z 4 bitów sygnału oraz 4 bitów znaku. Ostatni wiersz zawiera po jednym bicie znaku. Pokazuje to **rysunek 2**. Organizacja pamięci wyświetlacza zmusza do takiej, a nie innej realizacji. Najtrudniej jest pogodzić ze sobą sygnał ze zna-

Rys. 2





kami.  
Każdy  
znak ma  
inną treść i  
dodatko-

```
if(rozdziel) //rozdzielenie kanałów na 4 osobne wykresy
{
    rej=(rej>>2)+(iii<<6); //zmniejsz amplitudę oraz dodaj
    stała wartość dla danego przebiegu
    rej2=(rej2>>2)+(iii<<6); // amp/4+ nr.przebiegu*64
}
```

**Listing 2**

wo sposób wyświetlania zależy od tego, czy pomiar wyzwalany jest zbroczem czy sygnałem. Najgorsze w tym wszystkim jest jeszcze to, że dane muszą zostać przetworzone w „locie” bez żadnego pomocniczego bufora. Dlatego jest to tak skomplikowane. O wiele lepiej przedstawiają się już funkcje wyświetlające przebiegi analogowe w rejestratorze *lcd\_wa8...lcd\_wa1*. Są krótkie, ale za to trochę wolniejsze. Jako że priorytetem był analizator, to postanowiłem nie marnować na te funkcje zbyt wiele pamięci programu i wykorzystać ją do bardziej pożytecznych celów. Można odnieść wrażenie, że faktycznie są one wolne, ale od razu uprzedzam, że budowa głównej pętli rejestratora jest inna niż analizatora. Tutaj z każdym przebiegiem pętli uruchamiana jest procedura wyzwalania i pobierania danych, co właśnie powoduje powolne przesuwanie wykresu. To, że funkcje wyświetlające przebiegi analogowe są krótkie nie znaczy jeszcze, że łatwe do zrozumienia. Jest wręcz przeciwnie, trudno się w tym połapać. Cała procedura jest za długa, aby ją tu prezentować, a pokazanie tylko fragmentu nie ma sensu. Dlatego odsyłam do pliku *Grafika.c*. Dla ułatwienia weźmy do przestudiowania funkcję *lcd\_wa1*, ponieważ wyświetla ona jeden kanał. A więc tak: pierwsza pętla *for* wykonywana jest 8 razy i odpowiada to liczbie wyświetlanych linii bajtów. Przy okazji zostaje wysłany dodatkowy znak podziałki na LCD. Dalej znajduje się kolejna pętla *for*, która odpowiada za wyświetlenie jednej linii danych. Na początku tej pętli ładowane są dane o przebiegu poprzez wskaźniki. Jako że adresowanie wyświetlacza zaczyna się od góry, należy odwrócić wartości danych, aby prawidłowo wyświetlić przebieg. Następnie sprawdzana jest wartość amplitudy, aby potem narysować przebieg w prawidłowej linii. Jeżeli wartość amplitudy znajduje się w tym przedziale, to obliczany jest bit, na którym zostanie wyświetlony punkt. Kolejne dwie instrukcje *if* sprawdzają sąsiednie dane, aby narysować odpowiednią pionową linię. W efekcie zamiast kropek na wykresie mamy piękny ciągły przebieg. Funkcje rysujące 4 lub 8 przebiegów mają jeszcze jedną pętlę. Wykonywana jest ona odpowiednio 4 albo 8 razy, w zależności od liczby wyświetlanych przebiegów. Przebiegi te można nałożyć na siebie lub przedstawić jeden pod drugim. Aby je rozdzielić, wystarczy prosta transformacja. Wystarczy zmniejszyć wartości amplitud oraz dodać do nich pewne stałe wartości, zależnie od numeru przebiegu. Pokazuje to **listing 2**. Myślę, że na temat grafiki wystarczy już tych opowieści. Teraz trzeba trochę powiedzieć o zbieraniu próbek sygnału. Otwieramy więc plik *Pomiar.h* i tu już króluje assembler. Ale spokojnie, funk-

cje nie są zbyt skomplikowane. Już na samym początku w oczy rzuca się dziwne makro *TAKE\_OSC*. Za jego pomocą pobierane i zapisywane są 4 próbki ośmiobitowe z częstotliwością 5,33MHz. Aby pobrać 2000 próbek, potrzebnych jest „tylko” 500 takich makr. Aż strach pomyśleć jak wyglądałaby taka procedura bez użycia makra. Wprawdzie kod wynikowy będzie taki sam, ale większą część pliku źródłowego będzie zajmowała właśnie ta procedura odczytu. Niżej jest funkcja pobierająca dane z prędkością 16MHz. Niestety, dane muszą być przechowywane w rejestrach, a więc można zapisać tylko 32 próbki ośmiobitowe. Jest to bardzo mało, ale można zaobserwować na przykład, czy nie występują drgania przy zmianie stanów. Fragmenty tej procedurki przedstawiam na **listingu 3**. Kolejna funkcja była już omówiona wcześniej, korzysta ona z makra *TAKE\_OSC*. Poniżej mamy jeszcze kilka procedur napisanych w assemblerze dla uzyskania prędkości próbkowania odpowiednio 2,67MHz, 1,6MHz oraz 32-800kHz. Te funkcje mogą już pracować z różną liczbą kanałów, w przeciwieństwie do dwóch pierwszych, które korzystają zawsze z ośmiu. Ostatnią funkcję, pobierającą dane cyfrowe, można było spokojnie napisać w C. Nieco wyżej można zobaczyć, jak wygląda procedura wyzwalania zbroczem lub sygnałem. Funkcje pisane w assemblerze mają swoje własne procedury wyzwalania, aby było szybciej. Tutaj jedna uwaga. Dla nich, przy wyzwalaniu zbroczem, nie ma możliwości opuszczenia pętli przyciskiem. Jeśli więc nie pojawi się odpowiednie zbrocze, trzeba zrestartować analizator. Zaletą takiego rozwiązania jest większa precyzja, przez co wykres jest stabilny przy szybkich sygnałach. Wyzwalanie jest dostępne również dla rejestratora oraz generatora. Dodam, że sprawdzany jest stan wejść 1 do 8, a rejestrator i generator pracują na dalszych kanałach. Do omówienia pozostała jeszcze sonda logiczna, częstotściomierz, no i wyżej wymieniony generator. Pętla realizująca sondę jest bardzo prosta, wystarczy

```
asm volatile(\
    "push r0"        "\n\t"
    "push r1"        "\n\t"
    ...
    "push r30"       "\n\t"
    "push r31"       "\n\t"
    //wyzwalanie
    ...
    //pomiar
    "in r30, %[aPORT_OSC]"        "\n\t"
    //pobierane próbek
    "in r31, %[aPORT_OSC]"        "\n\t"
    "in r0, %[aPORT_OSC]"        "\n\t"
    "in r1, %[aPORT_OSC]"        "\n\t"
    ...
    "in r28, %[aPORT_OSC]"       "\n\t"
    "in r29, %[aPORT_OSC]"       "\n\t"
    "sts kan2, r30"               "\n\t"
    "sts (kan2+1), r31"           "\n\t"
    "ldi r30, lo8(kan2+2)"        "\n\t"
    "ldi r31, hi8(kan2+2)"        "\n\t"
    "st Z+, r0"                   "\n\t"
    "st Z+, r1"                   "\n\t"
    ...
    "st Z+, r28"                  "\n\t"
    "st Z+, r29"                  "\n\t"
    "pop r31"                      "\n\t"
    "pop r30"                      "\n\t"
    ...
    "pop r1"                       "\n\t"
    "pop r0"                       "\n\t"
    :
    [rej] "a" (rej),
    [rej2] "a" (rej2)
    :
    [wyzw] "a" (wyzw),
    [wyzw] "a" (wyzw),
    [mask] "a" (mask),
    [PR1_PIN] "M" (1<<Pp1),
    [PR1_PORT] "M" (_SFR_IO_ADDR(PIN(APORT1))),
    [aPORT_OSC] "M" (_SFR_IO_ADDR(PIN(APORT1)));
}
```

**Listing 3**

stan wejść i wysłać na wyświetlacz pusty znak, czyli spację. W zależności od odczytanego poziomu logicznego jest ona zwykła (same zera) bądź też zanegowana (same jedynki). Tak więc jeżeli na danym wejściu pojawi się stan wysoki, to nad numerem kanału zaświeci się czarna kratka. Częstotściomierz zrealizowany jest tradycyjnie, czyli *TIMER0* odlicza 1 sekundę, a *TIMER1* zlicza impulsy wejściowe. W przerwanach wykorzystano dodatkowe zmienne dla zwiększenia pojemności liczników. Generator jest jakby odwrotnością analizatora, zamiast pobierać dane i zapisywać w pamięci, wysyła jednorazowo cały bufor na wyprowadzenia od 9 do 16. Zresztą obsługuje się go tak samo jak analizator, ponieważ korzysta z tych samych funkcji graficznych. Nie ma tylko możliwości wyboru liczby kanałów.



Wyzwalanie generatora można zaprogramować w ustawieniach analizatora bądź też rejestratora. Myślę, że chociaż z grubsza udało mi się opisać działanie programu. Powstawałem też trochę komentarzy, więc nie jest najgorzej©.

## Montaż i uruchomienie

Układ udało się zmontować na jednostronnej płytce drukowanej, pokazanej na **rysunku 3**. Niestety nie obszło się bez kilku zworek. Proponuję wlotować je na końcu. Zaczynamy od wlotowania mikrokontrolera. Najlepiej spisać się przy tym stacją na gorące powietrze, choć zwykłą lutownicą też można sobie poradzić bez większego problemu. Na przykład grotom minifala lutujemy wszystkie końcówki naraz. Tak nawiasem, zwykłym grotom też się da... Jak ktoś ma czas, może lutować nóżka po nóżce, są tylko 32. W następnej kolejności kładziemy kondensatory, potem rezystory i drugi układ scalony. Na samym końcu montujemy przyciski oraz gniazda wejściowe i zasilania. Tu małym problemem będzie lutowanie od strony elementów. Trzeba wysunąć do oporu goldpiny ze złącza i dopiero potem lutować. Na końcu przesuwamy obudowę gniazda z powrotem na miejsce. Z kwarcem może być więcej problemu. Nie należy dociskać go do płytki, rezonator lutujemy kilka milimetrów nad nią. Da się go delikatnie wygiąć tak, aby dotrzeć grotom do jego wyprowadzeń. Przyda się tutaj cienki grot. Dobrze jest przylutować kawałki goldpinów na złącze wyświetlacza. Wtedy będzie ładnie przylegał do płytki. Mam na myśli lutowanie na płytce, a nie na wyprowadzeniach LCD. Lutowany jest już do niej kabelek do programowania. Układ U2 jest trochę za wysoki. Ale wystarczy wyciąć w LCD plastik w miejscu, gdzie znajduje się układ i problem z głowy. Podczas wycinania lepiej wyciągnąć szybkę ze sterownikiem, aby jej nie uszkodzić. W wyświetlaczu trzeba też wywiercić dwie dziury na śrubki mocujące. Dobrze jest spiłować szkiełko w miejscu gdzie będą otwory, a więc jakieś 0,5–1 mm. To także zapobiegnie jego uszkodzeniu. Teraz czas, aby zaprogramować mikrokontroler. Można to było zrobić specjalnym programatorem, przed

wlotowaniem w płytkę, ale także jest na niej wyprowadzony interfejs SPI. Wystarczy przyłożyć bądź przylutować kabelki do odpowiednich pinów złącza wyświetlacza, co pokazuje **rysunek 4**. Łatwo zauważyć, że LCD ma 8 wyprowadzeń, a na płycie jest 9. Ten ostatni pin to reset procesora i jest potrzebny właśnie przy programowaniu. Do załadowania programu można wykorzystać na przykład darmowy ISPprog. Na koniec montujemy wreszcie wyświetlacz i podłączamy nasz analizator. Układ jest w miarę prosty, jednak mimo wszystko czasem przy montażu zdarzy się błąd i układ nie ruszy. Nie trzeba od razu panikować. Może gdzieś, jest zwarta nóżka mikrokontrolera albo źle przylutowana. Jeżeli programowanie przebiegło bez problemu, a wyświetlacz nie daje znaku życia, trzeba sprawdzić, czy złącze dobrze przylega i nie ma na nim pyłu. Podczas wiercenia mógł się tam dostać. Teraz z czystym sumieniem mogę stwierdzić, że: *układ zmontowany poprawnie i ze sprawnych elementów nie wymaga uruchamiania i działa od razu po włączeniu zasilania*. Do analizatora przyda się kabelek z chwytakami. Takie chwytaki można kupić po kilka złotych za sztukę. W tym momencie koszt urządzenia wzrośnie kilkakrotnie, więc jest to opcjonalny dodatek. Jednak mimo wszystko warto w to zainwestować, gdyż taki kabelek znacznie poprawia komfort pracy z urządzeniem. Warto też zastanowić się nad zasilaniem baterijnym. Świetnie nadaje się do tego akumulator z telefonu Siemens C65. Jest płaski i wymiarami pasuje do płytki analizatora. Trzeba wtedy pomyśleć o zabezpieczeniu przed rozładowaniem i ładowaniem takiego akumulatora. Można oczywiście ładować go w telefonie, ale wtedy potrzebne będzie złącze na płytce pod ten akumulator. Lepiej zamocować go na stałe w urządzeniu. Układ zabezpieczający zmieści się na spodzie płytki obok baterii. Umieszczamy go pod złączem P4 i lutujemy dwie przelotki na zasilanie. Można także wykorzystać złącze P3, a w miejsce stabilizatora wmontować przełącznik, który będzie odłączał akumulator w trakcie ładowania. Sposobów jest wiele, dlatego tę kwestię pozostawiam do rozstrzygnięcia Czytelnikom. W miarę możliwości układ warto zasilac z mierzonego urządzenia. Należy tylko pamiętać aby nie przekraczać 5,5V. Dotyczy to również napięcia mierzonego. Nie powinno być także wyższe od zasilającego, jeżeli to wynosi chociażby tylko

3V. Może to spowodować przepływ prądu przez diody zabezpieczające, wbudowane w strukturę układu. Prowadzi to do pojawienia się szpilek na linii zasilania, co w najgorszym wypadku wprowadzi zakłócenia w pomiarze, a nawet zawie-

## Obsługa urządzenia

Samo uruchomienie układu za wiele nie da, trzeba jeszcze umieć się nim posługiwać. Do poruszania się po analizatorze służą przyciski S1...S4. Ich funkcje są następujące:

S1 – przycisk powrotu, służy do opuszczania podprogramów.

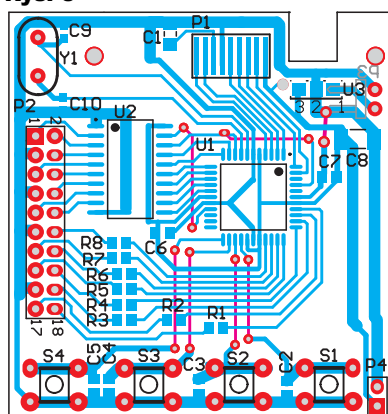
S2 – przycisk nawigacyjny, służy do poruszania się w lewo bądź do góry, w zależności od tego, co robimy w danej chwili.

S3 – przycisk nawigacyjny, jak powyżej, tylko poruszamy się w prawo bądź w dół.

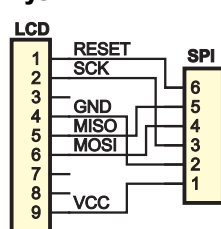
S4 – przycisk wyboru, zmiany parametru. Na przykład w ustawieniach służy do zmiany liczby kanałów lub sposobu i kombinacji wyzwalania. Natomiast w analizatorze służy do przełączania pomiędzy wykresem a podstawą czasu.

Co trzeba zrobić, aby rozpocząć pomiar? Teoretycznie wystarczy po włączeniu zasilania nacisnąć S4, by wejść w analizator, a następnie znowu S4, aby rozpocząć pomiar. To wszystko. Dostaniemy wykres danych próbkowanych z częstotliwością 16MHz, wyzwalanie dowolnym sygnałem. Zatem proponuję najpierw ustawić sobie sposób pomiaru. Po włączeniu zasilania przechodzimy niżej przyciskiem S3 na ustawienia. Wchodzimy S4 i możemy ustawić liczbę kanałów pomiarowych. Do zmiany również wykorzystujemy S4. Poniżej można ustawić sposób wyzwalania. Tak jak wyżej, poruszamy się klawiszami S2 i S3, a zmian dokonujemy, naciskając S4. Możemy wybrać zboczne narastające lub opadające, a następnie ustawić kanał, który będzie na nie reagował. Dostępne jest także wyzwalanie kombinacją stanów, które pojawiają się jednocześnie na wejściu. Przyciskiem S4 wybieramy stan dla każdego z ośmiu wejść. Może to być 0, 1, lub X, a więc dowolny stan, jaki pojawi się na tym kanale będzie akceptowany. Gdy już skończymy, możemy przystąpić do normalnego pomiaru. Przyda się jakiś sygnał testowy, na przykład z licznika binarnego. Po wejściu w analizator przyciskami S2 i S3 możemy zmieniać czas próbkowania. Na LCD podany jest czas na działkę, a więc co 16 wyświetlonych punktów. S4 rozpoczyna pomiar. Po załadowaniu przebiegu możemy się po nim poruszać. Aby zmienić podstawę czasu, znowu naciskamy S4 i tak dalej. W rejestratorze ten przycisk służy do wyświetlania przebiegów analogowych na wspólnym wykresie lub osobno jeden pod drugim. Dla

Rys. 3



Rys. 4



sondy logicznej i częstotściomierza nie ma żadnych ustawień. Jedynym działającym tu klawiszem jest S1. Generator obsługuje się tak samo jak analizator. Przycisk S4 przełącza pomiędzy wykresem a podstawą czasu oraz rozpoczyna jednorazowe generowanie sygnału. Kompletny interfejs użytkownika pokazany jest na **rysunku 5**. Rysunek ilustruje również rozmieszczenie poszczególnych kanałów na złączu wejściowym. Będzie to pomocą podczas budowy kabelka pomiarowego lub jeśli ktoś się na niego nie zdecyduje, pomoże po prostu w wybraniu dobrego kanału podczas testów.

## Wykonywanie pomiarów

Co można tym mierzyć? Przede wszystkim przebiegi cyfrowe, głównie te niezbyt skomplikowane i nie za szybkie. Na przykład protokołu USB nie radzę testować. Natomiast interfejsy SPI, RS232, JTAG, PS2, I<sup>2</sup>C, wyświetlaczy, pamięci i wiele, wiele innych spokojnie da się podglądać. Ba, można pokusić się nawet o prostą symulację takiego interfejsu dzięki funkcji generatora. Możliwości programu pozwalają na wysłanie jednorazowej sekwencji informacji. Ustawiamy wtedy sygnał wyzwalający, po którym zostanie wygenerowany zapisany sygnał. Aby mieć ten sygnał, trzeba najpierw odczytać go analizatorem. Oczywiście należy pamiętać o dopasowaniu napięć na przykład dla RS232 a także o tym, że w niektórych interfejsach stosuje się wyjścia typu otwarty kolektor (OC), na przykład w I<sup>2</sup>C i PS2. Dla nich symulacja nie zawsze będzie możliwa. Mimo wszystko nie trzeba się bać uszko-

dzenia układu, gdyż rezystory na kanałach od 9 do 16 skutecznie zabezpieczą obydwie urządzenia w krytycznych sytuacjach. Tak więc nie ma się czym martwić. Uwaga tylko na przekroczenie napięcia na kanałach od 1 do 8! Wtedy konsekwencje mogą być poważniejsze. W prostszych zastosowaniach analizator będzie służył do sprawdzania stanów portów mikrokontrolera, a mamy w końcu do dyspozycji 16 kanałów. Do sprawdzenia stanów w czasie rzeczywistym służy sonda logiczna, jednak należy stosować tę funkcję tylko dla powolnych sygnałów.

Pozostał do omówienia jeszcze rejestrator. W żadnym wypadku nie należy traktować tego jak oscyloskopu, po prostu nie można porównywać go z oscyloskopem. Ten rejestrator pozwoli natomiast orientacyjnie stwierdzić, jak wyglądają przebiegi analogowe, na przykład na wejściach komparatora analogowego czy też przetwornika A/C lub wyjść C/A w mikrokontrolerze. Po uśrednieniu można oszacować wypełnienie sygnałów PWM. Jednak z większą dokładnością da się to zrobić za pomocą analizatora. Wystarczy wziąć linijkę i zmierzyć na wyświetlaczu długość stanu niskiego i stanu wysokiego. Na koniec jeszcze drobna uwaga. Dobrze jest podłączyć nieużywane wejścia do masy. Wprawdzie nic złego by się nie stało, gdyby wisiały w powietrzu, jednak pojawiające się przypadkowe stany na tych wejściach mogą irytować przy oglądaniu wykresu. Ma to miejsce na przykład, jeśli mierzymy na 7 kanałach, a wykres pokazuje 8.

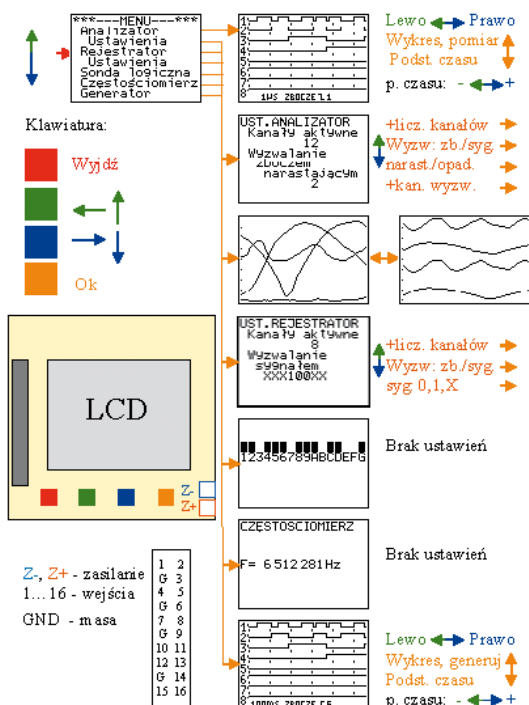
## Możliwości zmian

Jeśli chodzi o program, to zmiany są ograniczone. Można pokusić się o dodanie do częstotściomierza pomiaru okresu oraz wypełnienia sygnału. Cała procedura musiałaby być krótsza od czasu pomiędzy kolejnymi przerwaniem sygnału wejściowego... Tak, tak, przy pomiarze okresu *TIMER0* będzie liczył długość stanu, a *TIMER1* będzie zliczał tylko 1 impuls wejściowy i przerwanie będzie wykonywane. W tej sytuacji przydałoby się zamienić liczniki rolami dla uzyskania większej rozdzielczości pomiaru, ale to wymaga podłączenia wejścia pomiarowego do wyprowadzenia PB0(T0) mikrokontrolera. Na szczęście nie jest to problem, wystarczy zewrzeć ze sobą

nóżki 40 i 41. *TIMER1* będzie pracował w trybie czasomierza, więc nie będzie reagował na to, co pojawia się na wyjściu. Wtedy także dla zwykłego pomiaru częstotliwości można wykorzystać timer0 do zliczania impulsów wejściowych. Dla dużych częstotliwości wejściowych spowolni to znacznie program, ale jak łatwo zauważyć, pętla dla częstotściomierza jest prawie pusta... no więc do dzieła! Ja spróbuję wprowadzić te zmiany w ewentualnej następnej wersji programu. Okres będzie mierzony dla niskich częstotliwości, co zapewni dużą rozdzielczość i szybkość, a częstotliwość będzie mierzona dla szybszych sygnałów wejściowych. Obydwie funkcje będą się nawzajem uzupełniać. Oczywiście na LCD będzie cały czas wyświetlana informacja zarówno o częstotliwości jak i okresie. Wiadomo przecież, że częstotliwość jest odwrotnością okresu sygnału. Być może coś jeszcze nadaje się do zmiany w programie. A jeśli chodzi o samo urządzenie, to do rozstrzygnięcia pozostawiam sposób zasilania, ale to już było opisane. Do wejść można podłączyć przetwornik A/C z wyjściem równoległym i odpowiednio zmodyfikować program. Wykorzystując na przykład 8-bitowy ADS830, uzyskamy rejestrator analogowy parobkujący z częstotliwością do 3,2MHz. Potrzebne są 3 takty procesora na odczyt i zapisanie danych do pamięci RAM oraz 2 takty na wygenerowanie sygnału *clk* potrzebnego do prawidłowej pracy układu ADS830. Teraz tylko wystarczy dodać obwody wejściowe i mamy porządny oscyloskop... Jednak to wykracza już poza ten projekt. Myślę, że powyższy artykuł wystarczająco wyjaśnił, jak działa „Analizator stanów logicznych”. Wszelkie pytania dotyczące urządzenia proszę kierować na podany niżej adres mailowy.

Arkadiusz Hudzikowski  
a-r-o@o2.pl

Rys. 5



## Wykaz elementów

<b>Rezystory</b>	R1-R8	10kΩ SMD
<b>Kondensatory</b>	C1	1μF SMD
	C2-C7	100nF SMD
	C8	4,7μF SMD
	C9, C10	22pF SMD
<b>Półprzewodniki</b>	U1	ATmega32-16AU
	U2	74HC541 SMD
	U3	.78L05 (opcjonalnie)
	LCD	LCD Nokia3410
<b>Pozostałe</b>	P1	goldpin (patrz opis)
	P2	gniazdo C-Grid 2x10
	P3, P4	goldpin x2
	Y1	rezonator kwarcowy 16MHz U70

**Komplet podzespołów z płytą jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2920.**