

# Prosty analizator stanów logicznych do PC – Simple Logic Analyzer (SLA)

## AVT-389

Analizator logiczny jest dla projektantów układów cyfrowych podstawowym przyrządem pomiarowym. Niestety ceny przyrządów fabrycznych, nawet najprostszych skutecznie mogą blokować ich zakup. Dlatego też postanowiliśmy opublikować artykuł, w którym opisujemy jak wykonać prosty i tani analizator stanów logicznych samemu.

### Rekomendacje:

prezentowany w artykule projekt polecamy wszystkim fanom techniki cyfrowej i mikroprocesorowej, którzy niewielkim nakładem kosztów chcieliby zbudować praktyczny i funkcjonalny analizator przebiegów logicznych do PC.



### PODSTAWOWE PARAMETRY

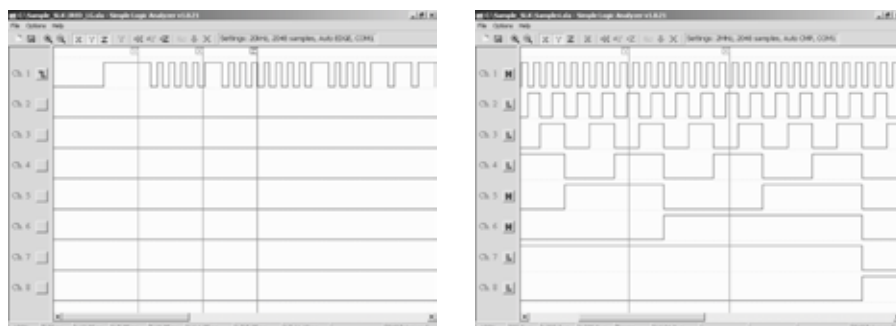
Płytką o wymiarach 80 x 94 mm  
Zasilanie +7... +18 V  
8 cyfrowych kanałów pomiarowych  
Akceptowane poziomy: TTL i CMOS (5 V)  
Częstotliwość próbkowania 2 kHz... 2 MHz  
Wyzwalania: ręczne, kombinacją stanów, zbczem  
Komunikacja z PC: RS232C  
Oprogramowanie na PC



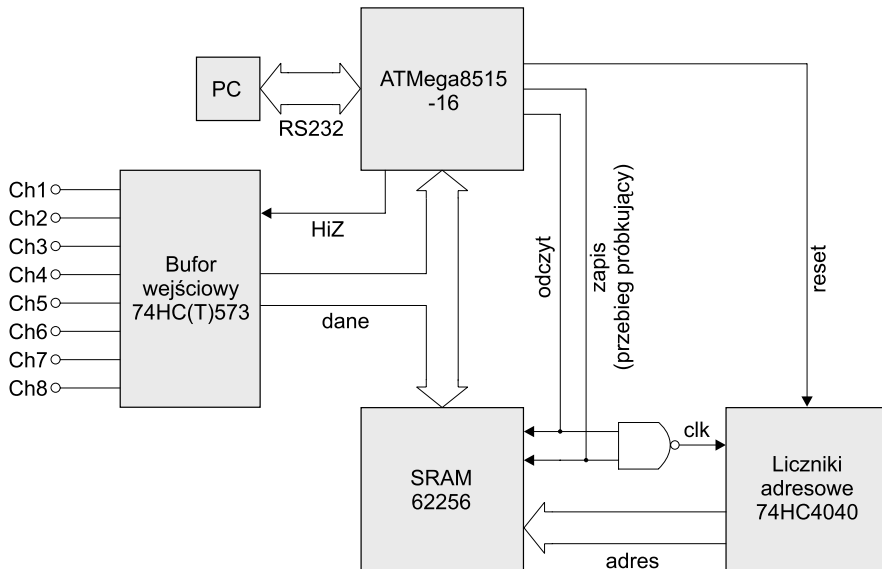
Nie wiem jak to jest w przypadku pozostałych Czytelników EP – ja przez długi czas praktyki obywatela byłem się “jakoś” bez porządnego rejestratora logicznego. Zawsze gdy trzeba było podejrzeć pracę badanego układu znajdowały się inne, zastępcze rozwiązania. Były to zwykle proste “rejestratory” na PC dołączane wprost do portu LPT do których obsługi stosowany był lepszej lub gorszej (zwykle to drugiej) jakości program. Tak było do pewnego momentu, gdy stwierdziłem, że w końcu przyszedł czas zaopatrzyć się w coś nieco bardziej wyrafinowanego. Zakup profesjonalnego analizatora nie wchodził w grę ze względu na cenę, która dla sprzętu tej klasy przekracza 3000...4000 PLN, oraz dlatego, że jego możliwości w warunkach amatorskich pozostałyby w dużej mierze nie wykorzystane.

Postanowiłem rozejrzeć się i sprawdzić, co da się znaleźć w opisach urządzeń prezentowanych w piśmie dla elektroników. Owszem – analizatorów było sporo – ale były to niemal w 100% (nazbyt) proste układy podobne do tych, którymi dotychczas się posługiwałem. Pozostałą część stanowiły skomplikowane urządzenia o możliwościach daleko wykraczających poza potrzeby amatorskie. Brakowało “złotego środka”. Postanowiłem więc wziąć sprawy w swoje ręce i stworzyć analizator “na miarę” przeciętnego amatorskiego warsztatu. Powstałe urządzenie nazwałem po prostu *Simple Logic Analyzer*, czyli w skrócie *SLA*.

Podstawowym założeniem było to, że rejestrator ma mieć ładny i funkcjonalny interfejs na komputer PC. Miałem dość liczenia jedynek, słupków itp. wynalazków jakie wy-



Rys. 1. Okno główne programu analizatora



Rys. 2. Schemat blokowy analizatora

stępowały w większości prostych konstrukcji. Interfejs miał “udawać” interfejsy programów służących do obsługi profesjonalnych rejestratorów. Stwierdziłem, że zarejestrowane przebiegi muszą być elegancko rysowane w głównym oknie aplikacji. Musi też ona umożliwiać dokładny pomiar odcinków czasu i częstotliwości za pomocą funkcjonalnych kursorów. Powinna też w miarę możliwości udostępniać różnorakie tryby wyzwalania. Co do części sprzętowej – powinna być przede wszystkim tania i zbudowana z łatwo dostępnych elementów. Szybkość (maksymalna częstotliwość próbkowania) nie jest tu najistotniejsza – w zupełności wystarczy 1 MHz. Od razu odrzuciłem koncepcję budowy prostej “wtyczki” do LPT. Rozwiązanie takie ma niestety na tyle dużo wad, że przesłaniają one niewątpliwą zaletę jaką jest niemal zerowy koszt budowy. Przede wszystkim – podczas próbkowania “przywieszane” są inne aplikacje, gdyż proces ten pochłania niemal 100% aktywności procesora komputera. Pojawiają się trudności podczas używania ich ze starszymi PC-tami. A poza tym – w układach tego typu trudno jest określić jaka jest maksymalna częstotliwość próbkowania, gdyż zależy ona bardzo od szybkości procesora. Trudno też mówić o dużej dokładności pomiaru zarejestrowanych przebiegów. W końcu zdecydowałem, że próbkowanie realizowane będzie sprzętowo, a próbki zapisywane będą w pamięci RAM. Ze względu na niską cenę i popularność wybrałem zna-

ną kostkę 62256. W naturalny sposób pociągnęło to za sobą decyzję, że projektowany analizator będzie 8-kanalowy.

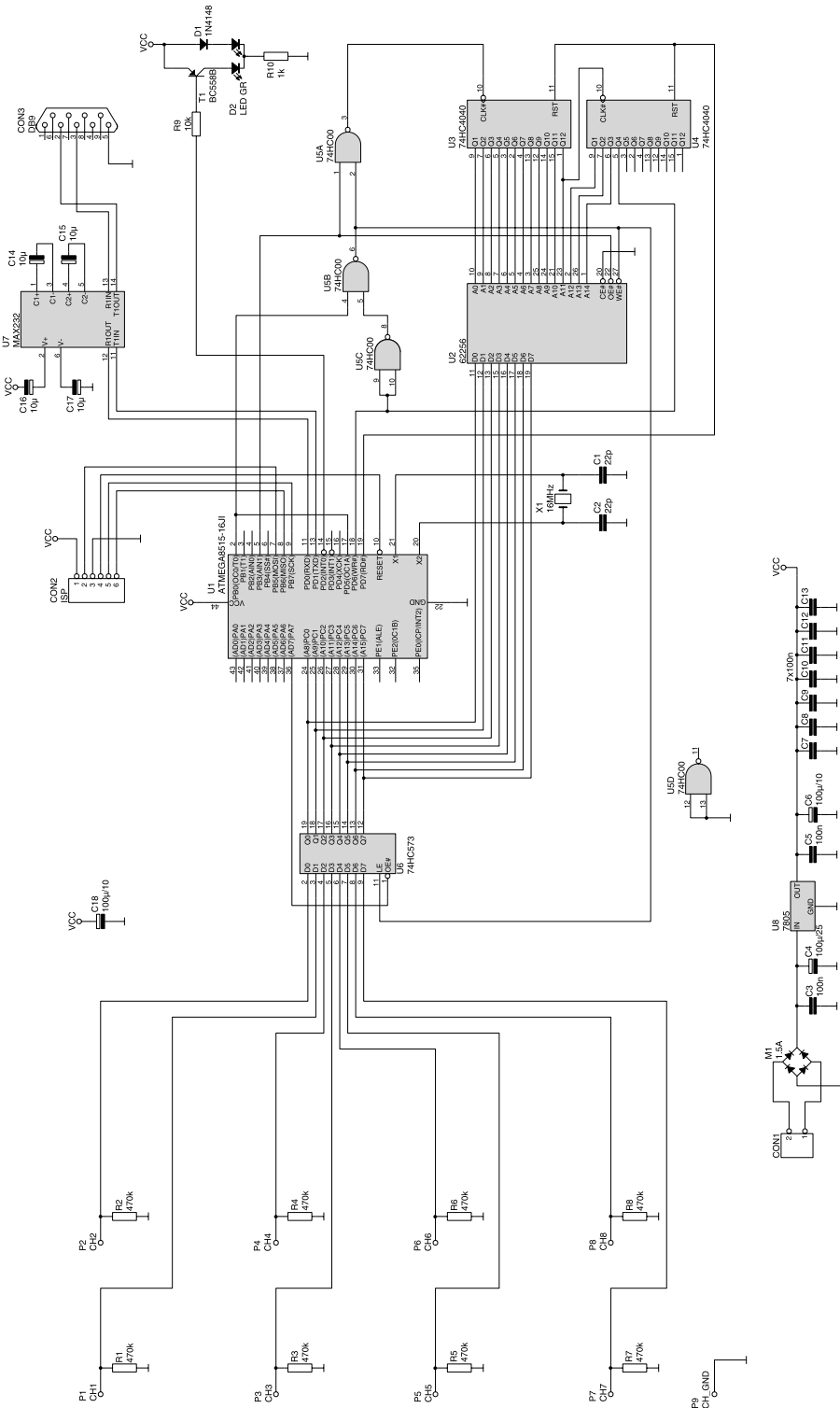
Na rys. 1 zamieściłem dwa zrzuty ekranowe aplikacji sterującej analizatorem. Jeden z nich przedstawia zarejestrowany przebieg pochodzący z pilota IRED jednej z dalekowschodnich firm (kodowanie klasy SONY), drugi – przebiegi występujące na sąsiednich wyjściach typowego licznika asynchronicznego 4020 taktowanego generatorem kwarcowym. W tab. 1 zebrane zostały parametry urządzenia. Mam nadzieję, że zarówno rys. 1 jak i tab. 1 narobią Czytelnikom “smacz-

ku” i zachęcą do budowy analizatora. Pozwala on na zarejestrowanie do 32768 próbek z częstotliwością próbkowania od 2 kHz do 2 MHz. Użytkownik ma możliwość wyboru liczby próbek (od 1024 do 32768), częstotliwości próbkowania oraz sposobu wyzwalania. Dostępne są trzy rodzaje wyzwalania:

- Manual – wyzwalanie ręczne. Akwizycja danych rozpoczyna się natychmiast po wciśnięciu przez użytkownika przycisku “GO” na pasku narzędzi programu sterującego;
- Auto CMP – wyzwalanie dowolną kombinacją stanów na wejściach. Wyzwolenie rejestracji następuje wraz z pojawieniem się na wejściach rejestratora ustawionej kombinacji stanów L i H. Ze względu na programową implementację wyzwalania (więcej piszę o niej dalej) zapewniającą prostotę konstrukcji zrezygnowałem z możliwości wyboru X – dowolnego stanu;
- Auto EDGE – wyzwalanie dowolnym zboczem na dowolnym kanale. Wyzwolenie następuje w chwili wystąpienia wybranego zbrocza (narastające/opadające) na wybranym kanale, przy czym stany na pozostałych kanałach są w tym trybie ignorowane.

Oferowane tryby wyzwalania nie mogą oczywiście konkurować z tym, na co pozwalają profesjonalne rejestratory (dowolne kombinacje zbroczy i stanów – w tym stanu dowolnego X, opcja pretrigger itd.). Jednak SLA

Tab. 1. Parametry analizatora	
Parametr	Wartość
Liczba kanałów	8
Częstotliwości próbkowania	2 MHz, 200 kHz, 20 kHz, 2 kHz.
Pojemność bufora	32 kB (możliwość wyboru liczby próbek z typoszeregu 1024, 2048, 4096 ... 32768)
Sposoby wyzwalania	Ręczne (Manual), Dowolną kombinacją stanów (Auto CMP), Dowolnym zboczem na dowolnym kanale (Auto EDGE).
Zakres napięć wejściowych	TTL lub CMOS (5 V) zależnie od zastosowanego bufora na wejściu (HCT lub HC)
Impedancja wejściowa	470 kΩ
Komunikacja z PC	RS232C
Zasilanie	7...18 VDC/100 mA
Obudowa	Z5
Aplikacja na PC	Aplikacja okienkowa SDI; System operacyjny: Windows 9x/ME/2000/XP; Ustawianie częstotliwości próbkowania, liczby próbek i sposobu wyzwalania; 3 kursory pomiarowe - pomiar czasu i częstotliwości; Możliwość powiększania przebiegów; Automatyczne dostosowanie widoku do rozdzielczości ekranu; Możliwość archiwizacji przebiegów i ustawień na dysku (pliki *.sl). Aplikacja na PC
Koszt elementów	ok. 40 PLN



Rys. 3. Schemat elektryczny

jest produktem na wskroś amatorskim i to co umożliwił wystarczy w zupełności w amatorskiej praktyce elektronicznej. Kolejnym jego atutem jest bardzo niska cena, wynosząca zaledwie kilkadziesiąt złotych.

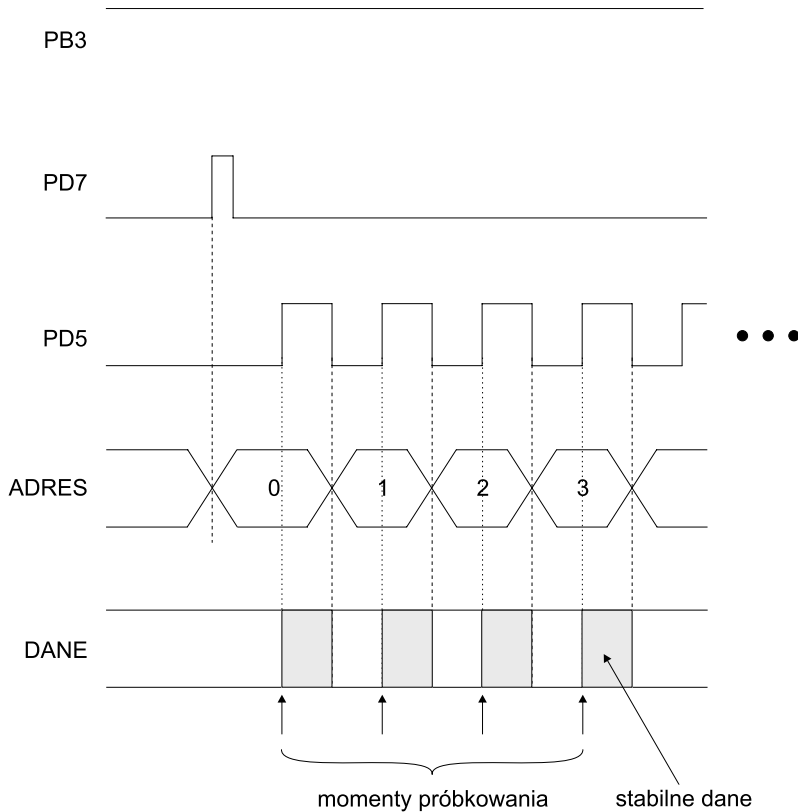
**Opis układu**

Efektom powyższych rozważań nad koncepcją budowy analizatora jest układ, którego schemat bloko-

wy przedstawiony został na **rys. 2**. Głównym elementem urządzenia jest pamięć SRAM typu 62256 mogąca pomieścić 32 kB słów ośmiobitowych. To właśnie w niej składowane są zarejestrowane podczas akwizycji dane o stanach na wejściach analizatora – każdy bajt pod kolejnym adresem to stan wejść w kolejnej chwili czasowej. Wejściami kanałów pomiarowych są punkty

Ch1...Ch8. Bajty pochodzące z tych punktów, poprzez zatrask – bufor trójstanowy typu 74HC(T)573, podawane są zarówno na wejścia danych pamięci, jak i na osiem wyprowadzeń mikrokontrolera AVR typu ATMega8515–16. Zatrask pełni trzy funkcje. Po pierwsze – stanowi prosty bufor wejściowy, oddzielający “świat” od wnętrza rejestratora. Po drugie – dzięki możliwości wprowadzenia jego wyjść w stan wysokiej impedancji – pozwala na bezkonfliktową współpracę procesora z pamięcią podczas odczytywania z niej próbek. Po trzecie wreszcie – wykorzystanie funkcji zatrasku (*latch*) sprawia, że dane na wejściach pamięci RAM są stabilne przez cały czas trwania impulsu zapisującego, co jest wyraźnie wymagane przez specyfikację tej kostki (dla uproszczenia nie zostało to pokazane na rys. 2). W chwilach pomiędzy impulsami zapisującymi zatrask jest transparentny.

Zadaniem mikrokontrolera jest odbieranie nastaw z komputera PC, sprzętowe wytwarzanie przebiegu próbkującego (za pomocą odpowiednio skonfigurowanego Timera1), odczyt zgromadzonych danych wraz z przesłaniem ich do komputera oraz realizacja wyzwalania procesu akwizycji danych. Wyzwalanie zrealizowane zostało programowo, co generalnie jest rozwiązaniem ułomnym, jednak w tym przypadku mogło zostać zaimplementowane z bardzo dobrym skutkiem. Jest tak dlatego, że stosunkowo duża częstotliwość taktowania (16 MHz) pozwala na wystarczająco szybką reakcję procesora na wystąpienie odpowiedniego zdarzenia wyzwalającego (kombinacja stanów/odpowiednie zbocze). Czas upływający pomiędzy momentem jego wystąpienia a zarejestrowaniem pierwszej próbki jest mniejszy niż okres próbkowania przy największej podstawie czasu. Rozwiązanie to uwolniło mnie od konieczności stosowania programowalnego sprzętowego bloku wyzwalania (którym w praktyce musiałby być dość szybki układ PLD) oraz dodatkowego – również programowalnego – bloku podstawy czasu. Tym samym obniżyło koszty budowy rejestratora oraz uprościło jego konstrukcję. Do adresowania pamięci RAM wykorzystałem proste liczniki binarne 74HC4040. Było to możliwe ze względu na fakt, że zarówno podczas próbkowania jak i



Rys. 4. Przebiegi w układzie podczas akwizycji danych

późniejszego odczytu zebranych danych pamięć adresowana jest w ten sam sposób – od adresu zero do adresu równego liczbie próbek pomniejszonej o 1. Układy 74HC4040 przy zasilaniu 5 V mają maksymalną częstotliwość pracy na poziomie 10 MHz, co z nawiązką pokrywa potrzeby opisywanej konstrukcji.

Działanie całości jest następujące. Gdy użytkownik wyda polecenie rozpoczęcia rejestracji danych, mikrokontroler odbiera z komputera PC rozkaz zawierający między innymi wybraną przez użytkownika częstotliwość próbkowania, rodzaj wyzwalania i liczbę próbek (sposób komunikacji został szczegółowo opisany w dalszej części artykułu). Zatrząsk wejściowy wprowadzany jest w tryb przezroczysty zaś liczniki adresowe są zerowane. Jeśli wybrany rodzaj wyzwalania to *AutoCMP* lub *AutoEDGE* mikrokontroler czeka na wystąpienie zdarzenia wyzwalającego, czyli odpowiednio kombinacji stanów lub zbocza. Należy zaznaczyć, że użytkownik może w każdej chwili przerwać działanie rejestratora wprowadzając go w stan początkowy. Dzięki temu nie ma obawy, że układ będzie czekał na zdarzenie które nigdy nie nastąpi.

Po wyzwoleniu rozpoczyna się akwizycja danych (oczywiście, je-

śli wybrano tryb *Manual* procesor rozpoczyna ją natychmiast po odebraniu rozkazu). Ze strony mikrokontrolera polega ona na podaniu przebiegu prostokątnego o wypełnieniu 50% i częstotliwości równej wybranej częstotliwości próbkowania na wejście zapisujące pamięci RAM. Za pośrednictwem bramki NAND jest on także kierowany na wejście zegarowe liczników 74HC4040. Przebieg ten wytwarzany jest sprętowo przez odpowiednio skonfigurowany (po odebraniu rozkazu) wewnętrzny licznik 16-bitowy. Dzięki temu wszystko co musi zrobić procesor aby zapoczątkować proces zbierania próbek to włączenie tego licznika, co oczywiście wymaga bardzo mało czasu. Podczas próbkowania na bieżąco zliczana jest liczba zarejestrowanych do tej pory próbek. Zajmuje się tym drugi (8-bitowy) wewnętrzny Timer mikrokontrolera. Jeśli liczba próbek osiągnie ustaloną wartość – proces akwizycji danych zostaje przerwany. Następnie procesor wprowadza bufor wejściowy w stan wysokiej impedancji, zeruje liczniki adresowe i odczytuje z pamięci RAM kolejne próbki. Dzięki obecności wyróżnionej na rys. 2 bramki NAND liczniki adresowe taktowane są automatycznie

```
List. 1. Komunikacja z PC
//*****
// RS232 Byte Receive Interrupt
//*****
void USART_REC_INT(void)
{
    static unsigned char FrameCounter;
    unsigned char data,i;

    data=UDR;

    //Quit sampling or data sending or any other
    activity
    if((data=='Q') && (fQFrameActive==0))
    {
        CommAbort();
        return;
    }

    //Allow for sending greeting to PC
    if(RecState==RECSTATE_WAITFORGREETING)
    {
        if(data=='I')
            fGreeting=1;
        else
            fGreeting=0;
    }
    return;

    //Query frame receiving
    if(RecState==RECSTATE_WAITFORFRAME)
    {
        if(fQFrameComplete==1)
            return;

        if((data==HEADER_VALUE) && (fQFrameActive==0))
        {
            FrameCounter=0;
            fQFrameActive=1;
        }
        else
        {
            if(fQFrameActive==0)
                return;

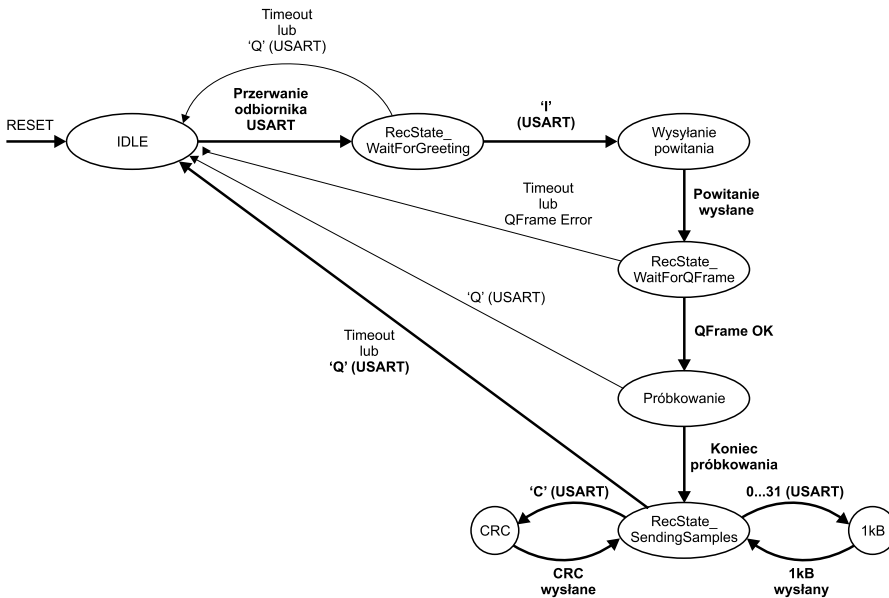
            QueryFrame[FrameCounter++]=data;
            if(FrameCounter==6)
            {
                fQFrameComplete=1;
                fQFrameActive=0;
                FrameCounter=6;
            }
        }
        return;

    //Sending samples
    if(RecState==RECSTATE_SENDINGSAMPLES)
    {
        if((data<32) || (data=='C'))
        {
            SendingBlockNumber=data;
            fSampleBlockRequest=1;
        }
        return;
    }
}
}
```

zarówno podczas zapisu jak i odczytu danych.

Każdy odczytany blok próbek o wielkości 1 kB wysyłany jest w odpowiedni sposób do komputera – jest to szczegółowo opisane w dalszej części artykułu dotyczącej komunikacji analizatora z PC. Na koniec wysyłana jest 8-bitowa suma kontrolna CRC o wielomianie generującym  $x^8+x^5+x^4+1$  – takim samym jaki wykorzystywany jest w pastylkach *iButton* firmy Dallas. Suma CRC liczona jest ze wszystkich przesyłanych bajtów.

```
List. 2. Realizacja programowego
zerowania mikrokontrolera
//*****
// Abort any activity
// SOFTWARE RESET
//*****
void CommAbort(void)
{
    //WDTimer On
    WDTCR&=~(3<<WDP0); //aprox. 16ms
    WDTCR|=1<<WDE;
    while(1); //wait for Watchdog
    reset
}
}
```



Rys. 5. Graf stanów komunikacji analizatora z PC

Schemat elektryczny przedstawiony jest na rys. 3. Całość zasilana jest napięciem 7...15 V za pośrednictwem prostego zasilacza zbudowanego na układzie U8 typu 7805. Dzięki obecności mostka M1 polaryzacja napięcia zasilającego nie ma najmniejszego znaczenia. Tak jak to przedstawia schemat blokowy, buforem wejściowym jest układ U6 typu 74HC573 lub 74HCT573. Od tego czy zastosujemy kostkę w wersji HC czy HCT zależy zakres napięć wejściowych z jakimi współpracował będzie analizator. Jeśli będzie to układ HC – zakres napięć wejściowych dla stanu niskiego wyniesie 0...1,5 V zaś dla stanu wysokiego 3,5...5 V. Ograniczy to w praktyce zakres napięć zasilania badanych układów do napięcia 5 V. W przypadku włożenia w podstawkę kostki w wersji HCT zakresy napięć będą typowe dla standardu TTL, to znaczy 0...0,8 V dla stanu niskiego i 2,0...5 V dla stanu wysokiego. Pozwoli to na badanie układów CMOS zasilanych nie tylko napięciem 5 V, ale także niższymi napięciami (w praktyce z zakresu 3...5 V). Takie “sprzężenie bezpośrednie” wyjść układów LV z układami posiadającymi wejścia TTL jest w wielu przypadkach jak najbardziej możliwe. Wystarczy aby badane układy LV przy danym napięciu zasilającym miały gwarantowane minimalne napięcie wyjściowe dla stanu wysokiego większe od 2 V, zaś maksymalne napięcie wyjściowe dla stanu niskiego mniej-

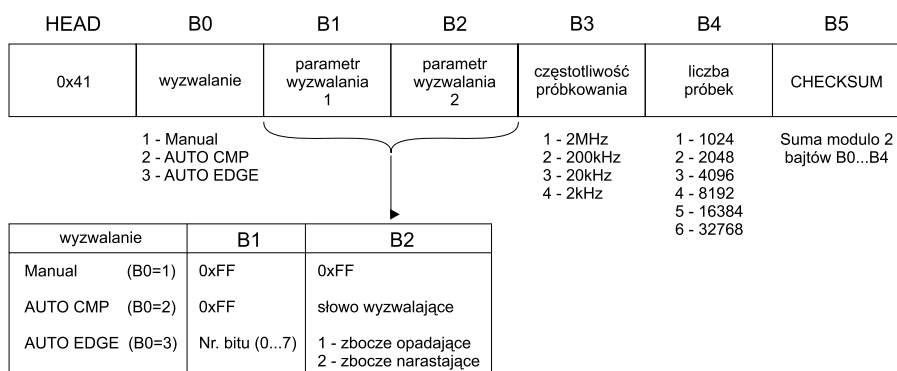
sze od 0,8 V (najlepiej z pewnymi marginesami zakłóceń).

Rezystory R1...R8 zwierają wejścia kanałów do masy ustalając tym samym impedancję wejściową analizatora równą 470 kΩ. Taka – dość wysoka – impedancja pozwoli na badanie także tych układów, które mają niewielką wydajność w stanie wysokim, jak na przykład wyjścia *open collector* podciągnięte do plusa zasilania za pomocą rezystora o stosunkowo dużej wartości. Nic nie stoi na przeszkodzie aby wartości rezystorów R1...R8 zwiększyć nawet do 2 MΩ. Dalsze zwiększanie ich rezystancji nie jest dobrym pomysłem, gdyż zbyt duża impedancja wejść może sprzyjać indukowaniu się w nich rozmaitych zakłóceń.

Głównymi elementami układu są oczywiście mikrokontroler U1 oraz pamięć RAM U2. Linia PA7 mikrokontrolera połączona jest z wejściem /OE kostki U6 – na schemacie blokowym jest ona oznaczona HiZ. Wystawienie na nią stanu wysokiego powoduje przejście wyjść U6 w stan wysokiej impedancji. Wyprowadzenie LE U6 połączony jest z wejściem zapisującym /WE pamięci, które aktywne jest w stanie niskim. Dzięki temu podczas zapisu do niej jakichkolwiek danych zatrząsk U6 jest zamknięty utrzymując na wyjściach ostatnio zapamiętany stan. Sprawia to, że podczas zapisywania dane na wejściach D0...D7 pamięci są stabilne, co warunkuje poprawny zapis. Wyjście PD5 (OC1A) procesora połączony jest z wejściem PBO (OC0/T0) oraz – poprzez U5B

– z wejściem zapisującym U2. Na PD5 wyprowadzany jest przebieg próbkujący podczas akwizycji danych, który wytwarzany jest przez Timer1. PBO jest wejściem, na którym impulsy próbkujące zliczane są przez Timer0, dzięki czemu procesor “wie” ile dotychczas zarejestrował próbek i kiedy wyłączyć Timer1 (zakończyć próbkowanie). Przebiegi występujące w tej części układu podczas próbkowania przedstawia rys. 4.

Bramka U5A (pokazana także na rys. 2) wytwarza przebieg zegarowy dla liczników adresowych U3 i U4 zarówno podczas zapisu jak i odczytu danych z pamięci. Można powiedzieć, że pełni ona rolę “bramki NOR dla aktywnych stanów niskich” (logika ujemna). Innymi słowy – wystąpienie stanu niskiego (aktywnego) na którymkolwiek z jej wejść skutkuje pojawieniem się na wyjściu stanu wysokiego, czyli przeciwnego do aktywnego. Wprowadzana przez nią negacja sprawia, że adresy zmieniają się w nieparzystych (umownie) półokresach przebiegu zapisującego lub odczytującego, zaś momenty zapisu lub odczytu występują w półokresach parzystych. Dokładnie widać to na rys. 4. Dzięki temu zarówno podczas zapisu jak i odczytu danych adres jest stabilny. Bramki U5B i U5C zapobiegają “przewinięciu” się adresu z wartości 32767 do 0 podczas rejestrowania danych z ustawioną maksymalną liczbą próbek równą 32768. Bez nich takie zjawisko wystąpiłoby, gdyż zakończenie procesu próbkowania odbywa się z chwilą stwierdzenia, że zliczający próbki Timer0 naliczył ich już wystarczającą ilość – zaś to analizowane jest w przerwaniu od przepełnienia T0. Jak wiadomo czas reakcji na przerwanie wynosi w mikrokontrolerach AVR co najmniej 4 cykle zegarowe, co w połączeniu z czasem potrzebnym na wykonanie odpowiedniej instrukcji warunkowej przekracza okres próbkowania przy najszybszej podstawie czasu równej 2 MHz. Z resztą – zjawisko rejestracji większej liczby próbek niż ustawiona (o 2..3 próbki) występuje zawsze gdy używamy podstawy czasu 2 MHz. Jednak przy ustawionej liczbie próbek mniejszej niż 32768 przewinięcie adresu nie jest groźne. Podczas odczytu danych i przesyłania ich do komputera “nadmiarowe” próbki są po prostu ignorowa-



Rys. 6. Format ramki zapytania

ne – leżą one zawsze poza zbiorem adresów 0...liczba próbek-1.

Dwukolorowa dioda LED D2 sygnalizuje stan analizatora. W spoczynku świeci się na zielono pełniąc funkcję lampki "POWER". W takiej sytuacji tranzystor T1 jest zatkany i prąd płynie przez diodę D1 oraz zieloną sekcję D2. Podczas rejestracji danych oraz oczekiwania na wyzwalanie procesor wystawia na nóżkę PD2 stan niski, co powoduje nasycenie T1. W rezultacie do plusa zasilania dołączona zostaje czerwona sekcja D2. Napięcie pracy czerwonej diody LED jest równe ok. 1,9 V i nie wystarczy do spolaryzowania połączonych szeregowo diod D1 i zielonej połówki D2. D2 świeci wtedy na czerwono sygnalizując stan zajętości analizatora ("BUSY").

## Komunikacja analizatora z PC

Komunikacja rejestratora z PC odbywa się "standardowo" za pomocą interfejsu RS232. W fazie projektowania zastanawiałem się nad zastosowaniem USB, ale zrezygnowałem z tego pomysłu ponieważ koszt dowolnego dostępnego kontrolera USB byłby większy od ceny jakiegokolwiek innego elementu urządzenia (no chyba, że podjąłbym się zaimplementowania protokołu w tanim mikrokontrolerze – byłoby to jednak wyzwanie daleko większe niż budowa samego rejestratora – polecam stronę [www.cesko.sk](http://www.cesko.sk)). Zajmujący się elektroniką posiadacze nowoczesnych laptopów, w których na płytach głównych nie zamontowano "staruszka" RS232, zapewne już dawno zaopatrzyli się w odpowiednie konwertery USB<->RS232 jakie wielokrotnie były opisywane w EP i nie tylko.

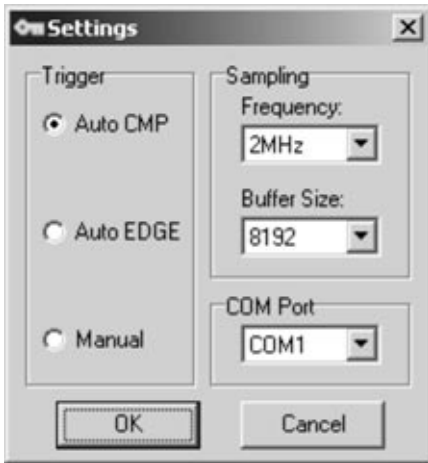
Konwersją poziomów napięć pomiędzy strefą 5 V a RS232 zajmuje się popularny układ U7 typu MAX232. Transmisja danych odby-

wa się z szybkością 38400 bodów, co przy zastosowaniu rezonatora kwarcowego 16 MHz i szybkiego trybu pracy USART-a procesora daje błąd względny równy 0,16%. Jest on oczywiście dopuszczalny – dla trybu *doublespeed* Atmel zaleca, aby całkowity błąd nie przekraczał 1,5%. W naszym przypadku warunek ten jest oczywiście spełniony, gdyż drugi co do ważności składnik błędu – tolerancja kwarcu – jest na poziomie co najwyżej 0,1% (dla rażąco niskiej jakości rezonatora).

Algorytm komunikacji jest najważniejszą rzeczą jaką zrealizowana została w programie mikrokontrolera. Opracowałem go kierując się dewizą "najważniejszy jest użytkownik". Za niedopuszczalne uznałem aby w toku komunikacji mikrokontroler bądź program sterujący oczekiwał na wystąpienie jakiegokolwiek zdarzenia "komunikacyjnego" bez objęcia tego oczekiwania odpowiednim przeterminowaniem (timeout). Poza tym, użytkownik musi mieć możliwość przerwania (z poziomu programu na PC) sesji pomiarowej w dowolnej chwili, zarówno podczas oczekiwania na wyzwalenie jak i podczas samego procesu próbkowania oraz transmisji próbek z rejestratora do PC. **Rys. 5** przedstawia graf stanów algorytmu komunikacji zaszytego w programie mikrokontrolera (linią pogrubioną zaznaczyłem przejścia między stanami podczas bezbłędnej sesji pomiarowej). Z formalnego punktu widzenia jest to prosty automat Moore'a. Stan następny określany jest na podstawie aktualnego stanu i wejścia automatu, którym jest albo informacja otrzymana z interfejsu szeregowego albo informacja o stanie próbkowania. Wyjściem jest to, co mikrokontroler wysyła przez RS232 do komputera nadrzędnego. Jak widać wyjście zależy jedynie

od stanu w jakim aktualnie znajduje się automat. Stany mające w nazwie słowo *wait* są stanami oczekiwania na określone zdarzenia i jako takie objęte są odpowiednimi timeout-ami. Zdarzeniem takim jest zwykle otrzymanie z komputera potrzebnej informacji. Pozostałe stany są tymczasowe – program "przelatuje" przez nie wykonując przypisane im czynności i na nic nie oczekuje. Nazwy kilku stanów napisane zostały po angielsku aby zachować jednolite nazewnictwo w artykule i programie (tam wszystkie zmienne i komentarze pisane są w tym języku). Układ PC – mikrokontroler jest klasycznym układem Master – Slave. Oznacza to, że mikrokontroler nie inicjuje komunikacji, a jedynie odpowiada na pytania jakie zadaje mu komputer.

Sesja pomiarowa przebiega w sposób następujący. W spoczynku procesor U1 pozostaje w stanie IDLE i oczekuje na zgłoszenie przerwania od sterownika transmisji szeregowej (USART). Gdy użytkownik wciśnie przycisk "GO" w programie sterującym na PC, komputer wysyła literę *.I* oznaczającą prośbę o przedstawienie się "rozmówcy" (*Identification*). Odebranie przez mikrokontroler jakiegokolwiek znaku przez łącze szeregowo powoduje obudzenie go ze stanu uśpienia i wejście w stan RECSTATE\_WAITFORGREETING. W tym stanie oczekuje on właśnie na nadejście litery *.I*. Jeśli sesja przebiega poprawnie tak się dzieje i program przechodzi do stanu "Wysyłanie powitania". Pozostaje w nim przez krótką chwilę i jedyne co robi to wysyła identyfikujący go napis "SLA\_ANTONIAK\_V1" zakończony bajtem o wartości zero (*null terminated string*). Następnie przechodzi do stanu RECSTATE\_WAITFORQFRAME w którym oczekuje na ramkę zapytania. Komputer odebrałszy poprawny napis identyfikujący analizator wysyła ramkę zapytania (*Query Frame*). Jej format przedstawiony został na **rys. 6**. Ramka składa się z siedmiu bajtów. Pierwszy z nich (*HEAD*) ma stałą wartość odpowiadającą literze *.A*, co łączy go z dużą grecką literą  $\alpha$  symbolizującą początek. Następnie występują zakodowane liczbami informacje o rodzaju wyzwalania (oraz jego parametrach), częstotliwości próbkowania i liczbie próbek. Ostatni bajt zawiera sumę kontrolną równą sumie modulo 2



Rys. 7. Okno wyboru parametrów próbkowania

wszystkich bajtów ramki z wyjątkiem pierwszego i ostatniego (czyli samej sumy). Bajty B1 i B2 zawierają parametry wyzwalania opisane w tabelce na rys. 6. W trybie *Manual* nie są one używane i mają wartość 255.

Po odebraniu poprawnej ramki zapytania mikrokontroler “wyciąga” z niej informacje o częstotliwości próbowania, liczbie próbek, a także o rodzaju i parametrach wyzwalania. Na ich podstawie konfiguruje odpowiednio swoje peryferia (np. Timery) i wchodzi w stan “Próbkowanie”. Jeśli wybrano któryś z automatycznych trybów wyzwalania procesor oczekuje w pętli na spełnienie warunku wyzwolenia. Następnie włącza Timer1 rozpoczynając akwizycję próbek. Program napisałem w C przy użyciu kompilatora ICCAVR ver. 6.29 (trial). Posiada on możliwość “podglądania” w assemblerze skompilowanego kodu. Dzięki temu jego fragmenty krytyczne pod względem czasowym (np. wyzwalanie) mogłem zoptymalizować na maksymalną szybkość wykonania bez konieczności pisania wstawek assemblerowych.

Po zakończeniu procesu próbkowania (zliczenie przez Timer0 ustawionej ilości próbek) program przechodzi do stanu RECSTATE\_SENDINGSAMPLES. Wejście do niego wiąże się z wysłaniem pierwszego kilobajta próbek. Dla PC-ta jest to znak, że proces akwizycji danych dobiegł końca – formalnie w kontekście architektury Master – Slave stanowi to odpowiedź na ramkę zapytania. Jeśli użytkownik wybrał więcej niż 1024 próbki, pozostałe bloki o wielkości 1 kB są prze-

syłane do PC-ta na żądanie. Żądaniem jest wysłanie przez niego bajtu o wartości równej numerowi bloku jaki chce otrzymać w odpowiedzi. Może on mieć ogólnie wartość 0...31. Dzięki takiej organizacji transmisji wszelkie błędy jakie mogą pojawić się w jej trakcie są bardzo łatwe do naprawienia. Wystarczy, że komputer ponownie poprosi o jednokilobajtowy blok próbek w którym doszło do przekłamań. Po poprawnym odebraniu wszystkich bloków komputer prosi mikrokontroler o wysłanie sumy kontrolnej CRC liczonej ze wszystkich przesyłanych bajtów. Dokonuje tego wysyłając literę ‘C’. Suma CRC stanowi ostateczne zabezpieczenie poprawności danych. Następnie komputer kończy sesję wysyłając znak ‘Q’ (*Quit*). Stan RECSTATE\_SENDINGSAMPLES objęty jest oczywiście odpowiednim timeout-em.

Tak przebiega komunikacja. Na list. 1 zamieściłem procedurę obsługi przerwania USART-a, która w głównej mierze realizuje opisany proces. Jak widać każdemu stanowi (określonemu przez zmienną *RecState*) odpowiada oddzielna instrukcja warunkowa co czyni całość dosyć elegancką. Jeśli w pewnym stanie odebrane zostaną oczekiwane dane lub polecenia – następuje “zapalenie” odpowiedniej globalnej flagi. W pętli głównej program we właściwy sposób reaguje na ten fakt (w szczególności przechodzi do następnego stanu).

Wysłanie znaku terminującego sesję ‘Q’ może nastąpić także na żądanie użytkownika. Jego odebranie przez mikrokontroler powoduje natychmiastowe wykonanie funkcji *CommAbort()* pokazanej na list. 2. Jest to nic innego jak realizacja programowego reset-u mikrokontrolera z użyciem wewnętrznego licznika *Watchdog*. Procesor po prostu włącza go ustawiając minimalny dostępny interwał zerowania (ok. 16 ms) i czeka w nieskończonej pętli na jego przepelnienie, czyli w konsekwencji na reset.

### Obsługa programu

Program sterujący pracą rejestratora napisałem używając środowiska Borland C++ Builder 6 w darmowej wersji Personal. Dostarczany jest on w formie pojedynczego folderu, który należy skopiować w dowolne miejsce na dysku twardym. Folder ten zawiera samą

aplikację oraz niezbędne biblioteki firmy Borland, zawierające głównie komponenty *VCL* (*Visual Component Library*). Oprócz tego w podkatalogu *Samples* zamieściłem kilka zarejestrowanych przykładowych przebiegów (pliki \*.sla) – pozwoli to “pobawić” się aplikacją bez konieczności montowania części sprzętowej.

Po uruchomieniu programu ukazuje się zmaksymalizowane główne okno aplikacji. W jego lewej części widać osiem przycisków służących do ustalania parametrów wyzwalania. Prawa część to główne okno programu w którym rysowane są zarejestrowane przebiegi. Obie części połączone są dynamicznym *splitterem* dzięki czemu ich rozmiar można dowolnie zmieniać. Rodzaj wyzwalania, częstotliwość próbkowania i liczbę próbek wybieramy wciskając na pasku narzędzi przycisk *Settings* (kluczyk). Otwiera się wtedy okienko pokazane na rys. 7. Po zaakceptowaniu nastaw lewa część widoku zmienia się odpowiednio do naszego wyboru. Jeśli wybraliśmy opcję *Manual* przyciski stają się nie aktywne – w tym trybie nie ma potrzeby ustalania jakichkolwiek parametrów wyzwalania. Po wybraniu wyzwalania *Auto*

### WYKAZ ELEMENTÓW

#### Rezystory

R1...R8: 470 kΩ  
R9: 10 kΩ  
R10: 1 kΩ

#### Kondensatory

C1, C2: 22 pF  
C3, C5, C7...C13: 100 nF  
C4: 100 μF/25 V  
C6, C18: 100 μF/16 V  
C14...C17: 10 μF/16 V

#### Półprzewodniki

U1: ATmega8515-16JI (PLCC44)  
U2: 62256  
U3, U4: 74HC4040  
U5: 74HC00  
U6: 74HC573 lub 74HCT573 (patrz tekst)  
U7: MAX232  
U8: 7805  
T1: BC558B  
D1: 1N4148  
D2: LED dwukolorowa wsp. Katedoda  
M1: mostek 1,5 A

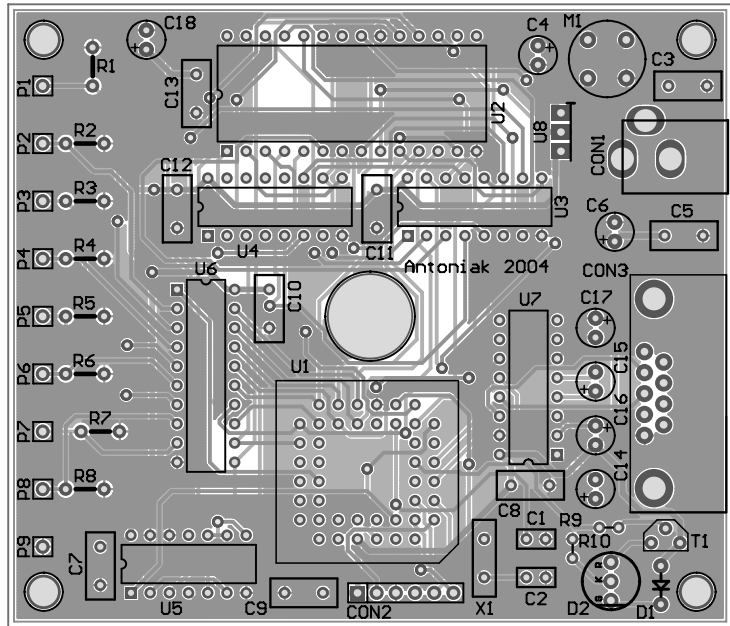
#### Inne

X1: kwarc 16 MHz  
CON1: złącze zasilania  
CON2: złącze ISP (goldpiny)  
CON3: DB9F do drukarki  
Obudowa: Z5 plastikowa  
Miniaturowe krokodyłki 9 szt.

Tab. 2. Ustawienia bezpieczników (fusebit) mikrokontrolera		
	Fusebit	Wartość
Fusebit LOW	BODLEVEL	1: BODLEVEL 2,7V
	BODEN	0: BODEN enabled
	SUT1..0	10: 65ms – slowly rising power
	CKSEL3..0	1010: Crystal oscillator
Fusebit HIGH	S8515C	1: MEGA8515 mode
	WDTON	1: Watchdog timer enabled by software
	SPIEN	0: Serial programming enabled
	CKOPT	0: Oscillator option
	EESAVE	1: Erase EEPROM when chip erase
	BOOTSZ1..0	00: 1024 words boot size
	BOOTRST	1: Reset vector is \$0000

CMP możemy ustalić słowo wyzwalające, zaś dla opcji *Auto EDGE* – zboczce.

Sesję pomiarową rozpoczynamy wciskając przycisk *GO* (zielona strzałka). Możemy ją w każdej chwili przerwać wciskając sąsiedni przycisk *STOP* (czerwony krzyżyk). Podczas trwania sesji dioda LED w rejestratorze świeci się na czerwono. Pomyślne zakończenie sesji skutkuje przesłaniem danych do komputera. Postęp transmisji wskazuje pasek postępu znajdujący się w prawej części paska statusu. Po przesłaniu wszystkich próbek zarejestrowane przebiegi rysowane są w prawej części widoku, zaś na pasku narzędzi wypisywane są wszystkie ustawione wcześniej (rys. 7) parametry próbkowania. Przebiegi możemy do woli oglądać przesuując odpowiedni *scroll* oraz powiększać i pomniejszać za pomocą przycisków *Zoom In* i *Zoom Out*. Pomiarów dokonujemy za pomocą trzech kursorów oznaczonych literami X, Y i Z. Włączamy i wyłączamy je niezależnie od siebie za pomocą odpowiednich przycisków na pasku narzędzi. Wyboru kursora którego pozycję chcemy zmienić dokonujemy przyciskiem *Set Active Cursor*. Możemy też w każdej chwili przejść w miejsce gdzie znajduje się dowolny kursor za pomocą jednego z przycisków *Goto*. Na pasku statusu podawane są w jednostkach czasu pozycje



Rys. 8. Schemat montażowy

wszystkich kursorów i wskaźnika myszki liczone względem chwili wyzwolenia. Jeśli włączone są co najmniej dwa kursory jest tam także obecna informacja o długości odcinka czasu między każdą parą aktywnych kursorów. Kliknięcie lewym przyciskiem myszy na pasek statusu w tym miejscu powoduje wyświetlenie częstotliwości odpowiadającej danemu odcinkowi czasu. Ponowne kliknięcie przywraca informację o czasie. Zarówno czas jak i częstotliwość podawane są w sformatowanych jednostkach.

Całą sesję pomiarową (tzn. zarejestrowane przebiegi wraz z informacją o podstawie czasu, liczbie próbek i wyzwalaniu) można zapisać na dysku przyciskiem *Save*. Informacje te przechowywane są w specjalnych plikach binarnych z rozszerzeniem *sla*. Poprawność danych zawartych w pliku chroni CRC.

### Montaż i uruchomienie

Schemat montażowy znajduje się na rys. 8. Montaż płytki rozpoczynamy od wlutowania podstawek pod układy scalone (polecam precyzyjne), a następnie lutujemy wszystkie elementy w kolejności od najmniejszych do największych. Zastosowany procesor zamknięty jest w obudowie PLCC44 i taką podstawkę musimy umieścić na płycie drukowanej. Złącze służące do podłączenia przewodu RS232 powinno być typu DB9F (żeńskie). Po zmontowaniu całości musimy zaprogramować mikrokontroler programem dostarczonym na CD

lub pobranym ze strony internetowej EP. Można do tego użyć popularnego programatora STK200/300 i odpowiedniego programu na PC (ja wykorzystałem pakiet Bascom AVR). Należy zadbać o właściwe ustawienie *fusebitów*. Bez tego układ nie ma szans działać ze względu na to, że fabrycznie opcje oscylatora są inne niż wymagane przez rejestrator. Pomocna będzie tab. 2. Pokazane w niej ustawienia fusebitów zapewniają także włączenie wewnętrznego układu Brown-out-Reset (z progiem działania 2,7 V). Dzięki temu wszelkie ewentualne spadki napięcia zasilającego układ nie są groźne. Wartości różniące się od domyślnych (występujących w fabrycznie nowym mikrokontrolerze) zaznaczyłem pogrubioną czcionką.

Po poprawnym zmontowaniu płytki i zaprogramowaniu procesora analizator jest gotowy do działania. Połączenie z komputerem PC powinno być wykonane przewodem "jeden do jednego". Płytkę można umieścić w obudowie Z5 pod którą została specjalnie zwymiarowana. Wejścia kanałów pomiarowych wykonujemy za pomocą dziewięciu (osiem+masa) odcinków przewodu o długości maksymalnie 0,5 m zakończonych chwytakami pomiarowymi lub niewielkimi krokodylkami.

**Arkadiusz Antoni**  
arkadiusz.antoniak@wp.pl

W ofercie handlowej AVT są dostępne:  
- [AVT-389A] płytka drukowana  
- [AVT-389B] kompletny kit