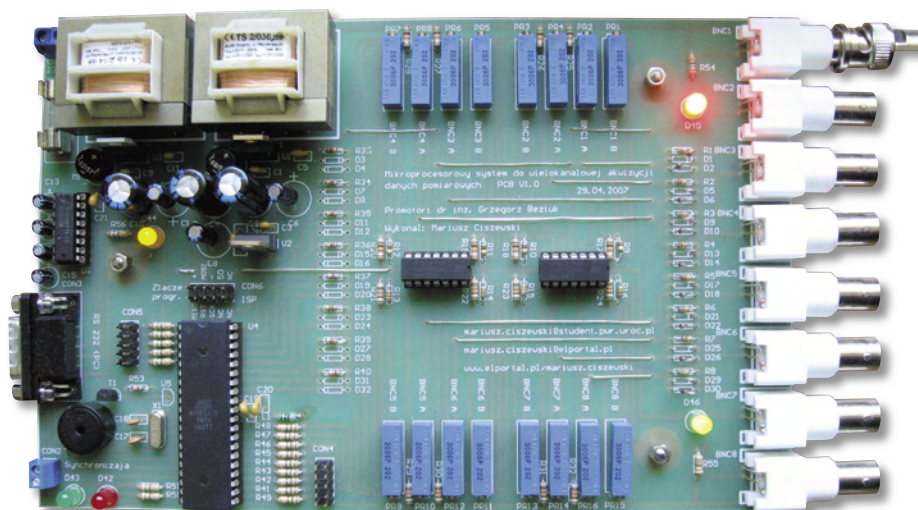




Wielokanałowy rejestrator danych pomiarowych A/C (2)

Kończymy opis rejestratora. W części drugiej przedstawiamy analizę prędkości transmisji danych pomiędzy kartą pomiarową a komputerem PC. Ponadto opisujemy program mikrokontrolera, aplikację sterującą dla systemu Windows oraz montaż i proces uruchamiania.



Transmisja danych pomiędzy kartą pomiarową a komputerem PC

Do przesyłania danych pomiędzy urządzeniem pomiarowym a komputerem osobistym wykorzystano dwie linie interfejsu szeregowego RS232: nadawczą TxD (*Transmitted Data*) i odbiorczą RxD (*Received Data*). Dla dopasowania układu USART, za pomocą którego mikrokontroler komunikuje się z innym urządzeniem, do standardu przyjętego w interfejsie RS232

niezbędny jest konwerter poziomów napięć TTL do napięć wymaganych przez standard interfejsu RS232. Taką rolę spełnia układ MAX232 (U6). Pomiary dokonywane przez mikrokontroler będą przesyłane do komputera osobistego i przechwytywane przez stworzoną w tym celu aplikację. Ponieważ założono, że zostanie wykorzystany popularny port komunikacyjny typu COM zgodny ze standardem RS232 o stosunkowo niedużej przepustowości, należało upewnić się, czy oferowana przezeń szybkość transmisji jest wystarczająca dla realizacji założeń projektowych oraz ustalić, jaką będzie można zadeklarować optymalną prędkość transmisji. Na potrzeby urządzenia przygotowano strukturę danych, która przyjęła następującą formę: Dana1:Dana2:Dana3:Dana4:Dana5:Dana6:Dana7:Dana8

Za jednym razem wysyłana jest porcja informacji, zawierająca wartości liczbowe dla wszystkich dostępnych kanałów pomiarowych. Wartość liczbową każdej z ośmiu danych rozdzielonych przecinkami jest dowolną liczbą całkowitą z przedziału 0...1023 i wynika z rozdzielczości zastosowanego w urządzeniu przetwornika A/C (w przypadku zastosowanego mikrokontrolera wynosi ona 10 bitów). Średniki pełnią funkcję separatorów, dzięki czemu aplikacja na PC bez trudu wydzieli właściwe ciągi znaków reprezentujące wartości liczbowe oraz przydzieli je do właściwych zmiennych w celu ich dalszej obróbki. Minimalna długość przygotowanej w ten sposób porcji informacji (przypadek 0:0:0:0:0:0:0) wyniesie 16 znaków, natomiast maksymalna (przypadek 1023:1023:1023:1023:1023:1023:1023:1023) to 40 znaków. Do tych

AVT-5149

W ofercie AVT:
AVT-5149A – płytką drukowaną

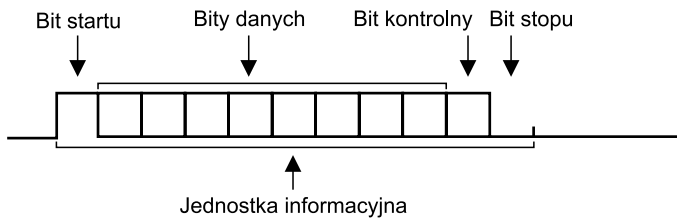
PODSTAWOWE PARAMETRY

- Płytką o wymiarach 215x140 mm
- Zasilanie 230 VAC
- Zakres napięć pomiarowych: -5...+5 V
- Rozdzielczość przetwornika A/C: 10 bitów
- Maksymalna częstotliwość pomiaru: 10 Hz (większa częstotliwość możliwa po zmianie oprogramowania)
- Liczba wejść pomiarowych: 8 niezależnych (gniazda BNC)
- Taktowanie pomiarów: wewnętrzne (z aplikacji) lub zewnętrzne (z odpowiedniego wejścia)
- Rejestracja pomiarów na wykresach, zapis do pliku BMP, eksport wyników pomiarów do pliku tekstowego
- Śledzenie przebiegu transmisji (opcja logowania do pliku)
- Pauzowanie, wznawianie i zawieszanie rozpoczętego pomiaru

PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Miniaturowy przetwornik A/C	EP 8/1996	AVT-1085
Rejestrator przebiegu cyfrowego – przystawka do TV	EP 10/1999	AVT-831
Miniaturowy przetwornik A/C do PC	EP 8/2001	AVT-1315
Amatorski oscyloskop cyfrowy	EP 9/2003	AVT-527
Cyfrowy oscyloskop/analizator stanów logicznych	EP 10-11/2003	AVT-529
Prosty analizator stanów logicznych do PC-Simple LOGIC Analyzer(SLA)	EP 4/2005	AVT-389
Bezprzewodowy DAQ z interfejsem IrDA	EP 11/2006	AVT-954
Analizator stanów logicznych	EP 4/2007	AVT-976
Niezwykły oscyloskop cyfrowy	EdW 6/2007	AVT-2828
4-kanałowy rejestrator analogowy Velleman	EP 10/2005	---
Uniwersalny rejestrator danych	EdW 10/2007	---
Mikroprocesorowy rejestrator zdarzeń	EP 5/2008	---



Rys. 5. Format jednostki informacyjnej

wartości trzeba dodać jeszcze po dwa znaki sterujące: CR (*Carriage Return*) odpowiadający za przejście karetki do początku linii oraz LF (*NL Line Feed, New Line*) odpowiadający za przejście do następnej linii. Pod uwagę należało wziąć ten drugi przypadek oraz ustalić prędkość transmisji dla portu COM, przy której możliwe będzie wysłanie przynajmniej kilkunastu sekwencji po 42 znaki w ciągu jednej sekundy. Rozważając prędkość transmisji znakowej należy wziąć pod uwagę, iż na poziomie fizycznym, w asynchronicznej transmisji znakowej, znaki przesyłane są w postaci reprezentujących je ramek, zawierających oprócz samej informacji dane nadmiarowe z perspektywy przesyłanych informacji, a jednak niezbędne dla właściwej transmisji. Te dane to bit startu, jeden lub dwa bity stopu oraz bit parzystości. Ponieważ są przesyłane razem z właściwą informacją (tworzą tzw. jednostkę informacyjną), zwiększają czas niezbędny do transmisji informacji właściwej. Format jednostki informacyjnej przedstawiono na rys. 5. Jak widać składa się ona z bitu startu, 8 bitów danych i bitu stopu. Tak ustalona jednostka informacyjna dla przesłania jednego znaku w kodzie ASCII potrzebuje fizycznego przesłania 10 bitów.

Typowe wartości szybkości transmisji przy asynchronicznej transmisji znakowej wynoszą 4800, 9600 czy 38400 bd, co w przypadku ustalonej wcześniej 10-bitowej jednostce informacyjnej daje odpowiednie, rzeczywiste prędkości: 480, 960, 3840 znaków na sekundę. Jak napisano wcześniej, prezentowane urządzenie wysyła do aplikacji na PC ciągi o maksymalnej długości 42 znaków. Oznacza to, że gdyby urządzenie miało próbkować wejścia analogowe z częstotliwością 50 Hz, wówczas musiałoby wysłać do PC-*ta* $50 \times 42 = 2100$ znaków w ciągu sekundy. Wówczas minimalna prędkość, jaką należałoby zastosować to 38400 bd, odpowiadające przesłaniu 3840 znaków przy 10-bitowej długości jednostki informacyjnej. Tak obliczona prędkość transmisji (2100 znaków na sekundę dla pomiarów przy częstotliwości 50 Hz) byłaby prawdziwa, gdyby transmisja danych odbywała się w sposób ciągły. Tymczasem, dla zapewnienia synchronizacji oraz poprawności działania całego systemu, został on zaprojektowany w taki sposób, że urządzenie wysyła pojedynczą porcję informacji (maksimum 42 znaki) na wyraźne żądanie ze strony aplikacji. Dlatego do czasu potrzebnego na przesłanie 42 znakowego ciągu z rejestratora do PC należy dodać jeszcze czas na wysłanie przez aplikację trzy-

mamy więc: 3 znaki (komputer osobisty → rejestrator) + 42 znaki (rejestrator → komputer osobisty), co daje 45 znaków. W jednym cyklu komunikacyjnym musi być więc wystarczająco dużo czasu na przesłanie 45 znaków przy dodatkowym uwzględnieniu czasu opóźnień związanych z cyklami zegarowymi niezbędnymi na wykonanie zadań po stronie rejestratora (m.in. pomiar napięć analogowych, interpretacja komend sterujących przychodzących z komputera osobistego, przygotowanie i przekształcanie ciągów z danymi na pojedyncze znaki w celu przesłania go w asynchronicznej transmisji znakowej). W bilansie czasu należy również uwzględnić opóźnienia po stronie komputera osobistego (przygotowanie i wysyłanie komend sterujących, odbieranie danych, generowanie wykresu). Czas potrzebny na wykonanie zadań po stronie komputera osobistego zależy od jego wydajności, trudno jest zatem podać nawet przybliżoną jego wartość. Można natomiast przyjąć, że jest on pomijalnie mały w porównaniu z czasem operacji na poziomie mikrokontrolera. Trudno jest także wyliczyć czas opóźnień wynikający z zadań wykonywanych po stronie rejestratora, gdyż użyto kompilatora języka wysokiego poziomu i praktycznie nie ma możliwości jednoznacznego określenia, np. jaką liczbę cykli zegarowych zajmuje wykonywanie poszczególnych instrukcji użytego kompilatora.

W celu jak najdokładniejszego określenia odpowiedniej szybkości przesyłania znaków pomiędzy rejestratorem i PC-tem należało w tej sytuacji założyć pożądaną częstotliwość próbkowania wejść analogowych (50 Hz) oraz wyznaczyć praktyczną szczytową prędkość transmisji, zwiększając liczbę znaków niezbędną do przesłania w jednym cyklu komunikacyjnym (z 45 do 50 powinno z zapasem pokryć czas opóźnień). W tej sytuacji mamy: $50 \times 50 = 2500$ znaków niezbędnych do przesłania w ciągu sekundy. Wynika z tego, że wciąż wystarczająca okazuje się prędkość 38400 bd, pozwalająca na przesłanie 3840 znaków w ciągu sekundy, przy

znakowej instrukcji sterującej, składającej się z dwóch znaków sterujących oraz znaku powrotu karetki CR niezbędnego dla przyjęcia komendy przez rejestrator. Ostatecznie

wcześniej założeniu 10-bitowej jednostki informacyjnej.

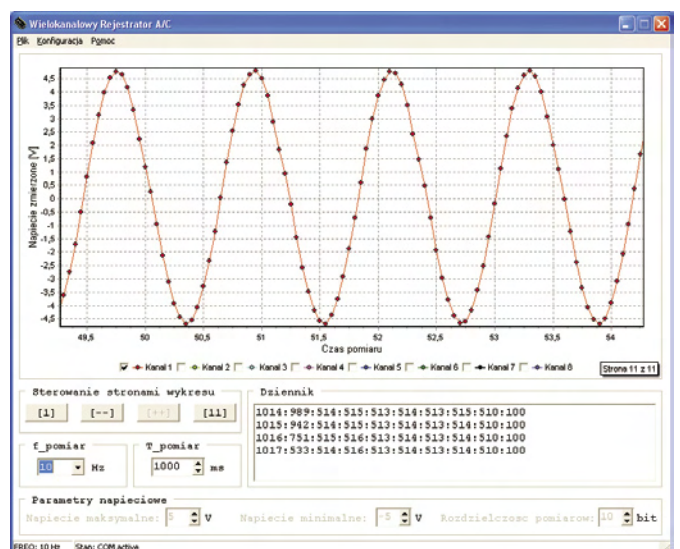
Program mikrokontrolera

Program dla mikrokontrolera pracującego w urządzeniu sprzętowym napisano w środowisku *BASCOM AVR firmy MCS Electronics*. Do kompilacji kodu źródłowego wystarczy bezpłatna wersja tego środowiska (kod wynikowy mieści się w limicie 4 kB). Kurs posługiwania się pakietem *BASCOM* został przeprowadzony na łamach „Elektronika dla Wszystkich” – numer 2/2000...1/2001. Dostępny jest również w specjalnym wydaniu „Elektronika Plus 1/2004 „Bascom”, poświęconemu w całości pakietowi *BASCOM*.

Rejestrator umożliwia pracę w dwóch trybach: aktywnym oraz pasywnym. W trybie aktywnym pomiar wejść analogowych zostaje wymuszony poprzez aplikację uruchomioną na komputerze osobistym. Okres pomiędzy kolejnymi pomiarami jest ustalany wówczas przez użytkownika. W trybie pasywnym wymuszenie pomiaru odbywa się za pomocą zewnętrznego sygnału taktującego. W tym trybie aplikacja pełni wyłącznie rolę klienta i rejestruje napływające pomiary bez wymuszania ich wysyłki przez kartę pomiarową.

Działanie programu mikrokontrolera oparte jest na pętli nieskończonej, w której odbierane są komendy sterujące, podawane przez interfejs RS232 z aplikacji działającej w komputerze osobistym. W momencie otrzymania danego polecenia przez interfejs RS232 następuje jego natychmiastowe wykonanie i przejście do żądanego trybu pracy. Po włączeniu zasilania urządzenie oczekuje na komendę sterującą ze strony aplikacji – przewidziano dwie komendy startowe: *ON_A* (uruchamia tryb aktywny) oraz *ON_P* (uruchamia tryb pasywny). Jeżeli został uruchomiony tryb aktywny (komenda *ON_A*), wówczas urządzenie oczekuje na komendy sterujące transmisją i wykonuje je w miarę ich otrzymywania:

TX – żądanie wykonania pomiarów wszystkich



Rys. 6. Okno główne programu

kanałów i przesłania komple-
tu danych do aplikacji PC,

PA – wstrzymanie transmi-
sji do momentu otrzymania
dowolnej innej komendy,

OFF – zakończenie trans-
misji i przejście do stanu
oczekiwania.

Jeśli natomiast został ur-
chomiony tryb pasywny (ko-
menda ON_P), urządzenie
na bieżąco sprawdza stan
wejścia PD3 (pin 17 mikro-
kontrolera) i po wykryciu
na nim impulsu dodatnio-
go natychmiast wysyła do
aplikacji komplet danych
pomiarowych. Wysłanie ja-
kichkolwiek znaków przez
aplikację do urządzenia
powoduje automatyczne
wyjście z trybu pasywnego,
a mikrokontroler oczekuje
na kolejną komendę ON_A,
bądź ON_P.

Wysyłane przez aplikację
PC komendy są potwier-
dzone dźwiękiem (buzzer
G1 sterowany przez pin 16
mикроkontrolera), dzięki cze-
mu użytkownik ma większą
kontrolę nad poprawnością
przebiegającej transmisji.

Pełny kod źródłowy (plik
.BAS), jak również gotowy
„wsad” (plik .BIN) zamiesz-
czono na płycie CDEP10/
2008B. Oba pliki są również
dostępne na stronie interne-
towej: www.elportal.pl/mariusz.ciszewski.

Aplikacja sterująca na
komputerze osobistym

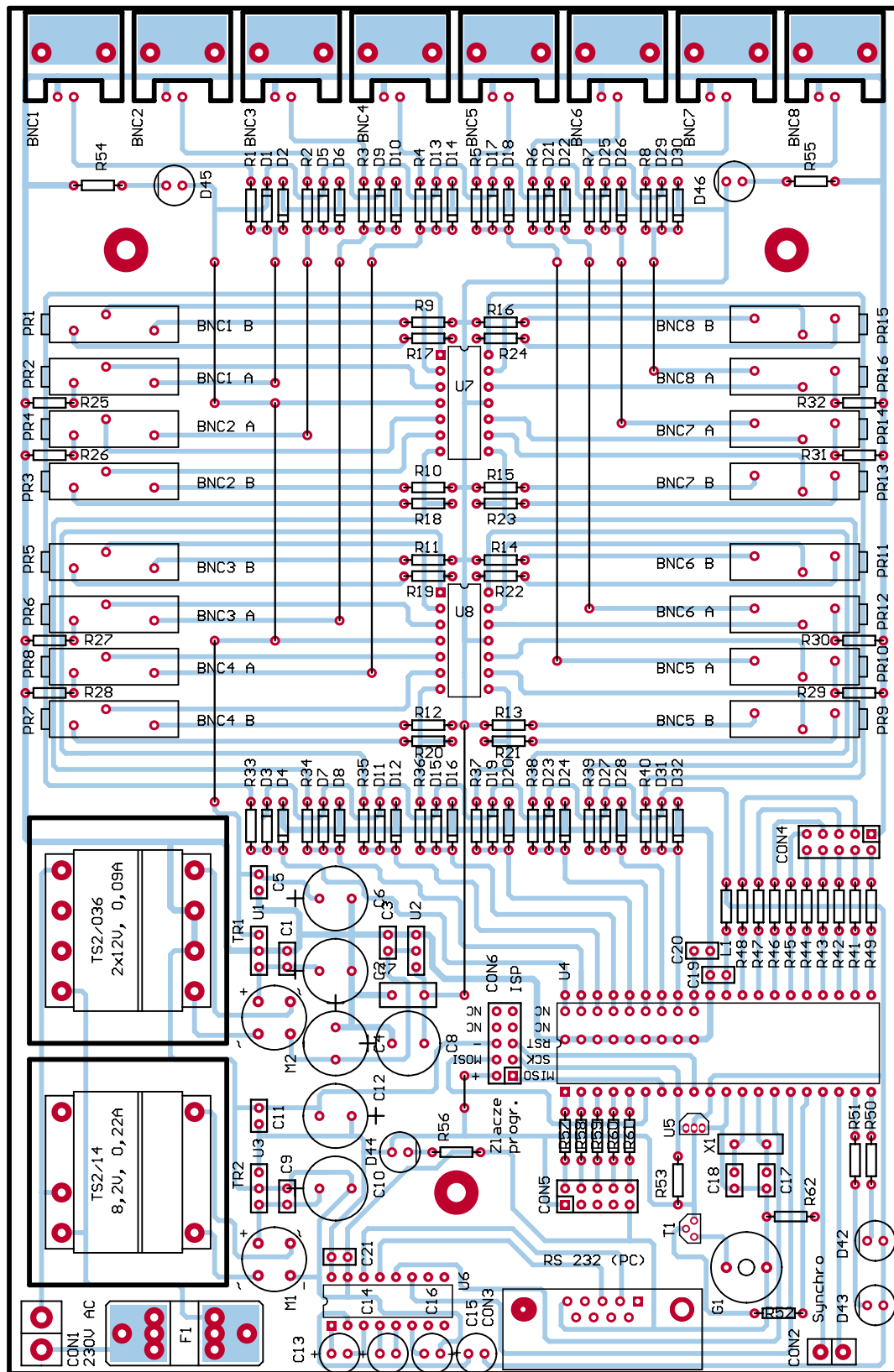
Aplikacja sterująca została
stworzona z wykorzystaniem
środowiska Borland Builder
C++ 6.0 wersja Personal
z zainstalowanym pakietem
darmowych komponentów
TeeChart firmy Steema. Ory-
ginalny kod został napisany
dla systemu operacyjnego
Windows 2k/XP/Vista, może
jednak zostać bez większe-
go problemu przetłumaczony
na dowolną dystrybucję
Linux'a lub MAC OS. Aplikacja sterująca udo-
stępnia zasadniczo dwie funkcje:
– komunikację z zewnętrznym urządzeniem
przez szeregowy port komunikacyjny COM
zgodnie ze standardem RS232,
– wizualizację danych pomiarowych na wykre-
sie.

Do realizacji komunikacji szeregowej wyko-

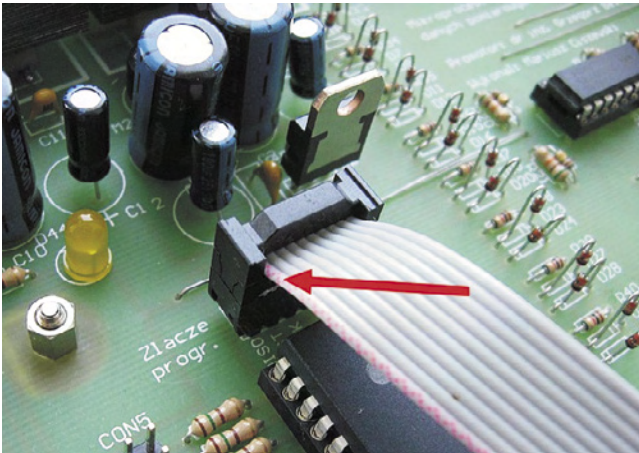
rzystano standardowe komponenty i funkcje
udostępniane przez środowisko BCB, natomiast
do rysowania wykresów wykorzystano darmo-
wą wersję pakietu zewnętrznych komponentów,
pod nazwą TeeChart, firmy Steema Software
(www.steema.com). Ze względu na specyfikę
programowania w środowiskach graficznych
typu Borland C++ Builder, szczegółowe omó-

wienie i wytłumaczenie zasad działania kodu
źródłowego wykracza poza możliwości objęto-
ściowe artykułu. Celem zapoznania się z podsta-
wami programowania środowiska Borland C++
Builder w wersji 6.0 i nowszej, zalecana jest lek-
tura [4], [5].

Na rys. 6, przedstawiono okno główne pro-
gramu, wraz z naniesionym opisem. W artykule



Rys. 7. Widok płytki rejestratora



Rys. 8. Podłączenie programatora do gniazda ISP urządzenia

Tab. 1. Ustawienia wybranych fuse bitów (CKSEL0...CKSEL3) dla poszczególnych źródeł taktowania – nota katalogowa układu ATmega16 ([4] s.23)

Opcja źródła taktującego	CKSEL3...0
External Crystal/Ceramic Resonator	1111...1010
External Low-frequency Crystal	1001
External RC Oscillator	1000...0101
Calibrated Internal RC Oscillator	0100...0001
External Lock	0000

nie są omówione wszystkie możliwości aplikacji. Informacje takie można odnaleźć w helpie programu oraz na stronie internetowej autora urządzenia. Należy jednak wspomnieć, że program umożliwia rejestrację pomiarów dokonywanych na wszystkich ośmiu kanałach, eksport wykresów do popularnych formatów graficznych, eksport wyników pomiarów do plików tekstowych, sterowanie urządzeniem pomiarowym, kontrolę przebiegu transmisji oraz inne funkcje.

Montaż i uruchomienie

Jednostronna płytką drukowaną (rys. 7) urządzenia posiada stosunkowo dużą powierzchnię (216x141 mm) Niezbędne okazało się również zastosowanie 12 zwoń, i to od nich należy rozpocząć montaż. Najlepiej wykonać je wykorzystując przewód połączeniowy, tzw. kynar. Następnie należy zamontować wszystkie rezystory

odpowiednich pinach zasilających układy scalone pojawiły się oczekiwane napięcia. Jeżeli wszystko jest w porządku, odłączamy zasilanie i umieszczamy układy scalone w podstawkach.

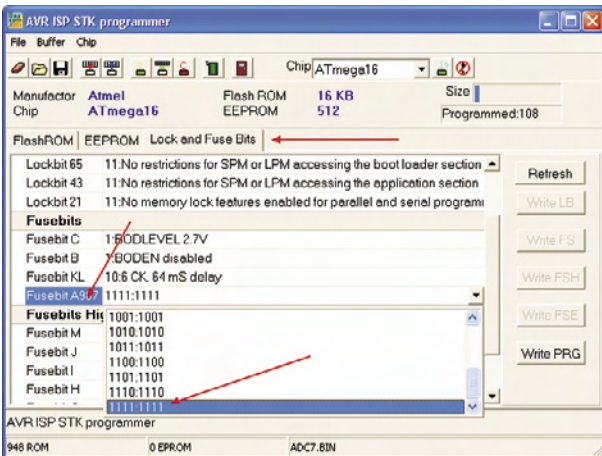
Przed następnymi pracami niezbędne jest jeszcze zaprogramowanie mikrokontrolera. Podczas uruchamiania układu użyto popularnego programatora STK200, komunikującego się z komputerem osobistym za pomocą złącza równoległego LPT i obsługiwanego bezpośrednio ze środowiska BASCOM AVR. Programator ten jest dostępny w ofercie firmy AVT jako kit AVT2550/P.

W celu dokonania ustawienia bitów kontrolnych i zaprogramowania mikrokontrolera podłączamy programator do portu drukarki komputera, a kabel ze złączem ISP podłączamy do zbudowanego urządzenia pomiarowego. Sposób podłączenia programatora do urządzenia pokazano na rys. 8 (strzałka pokazuje czerwony przewód na taśmie połączeniowej). Po prawidłowym podłączeniu urządzeń należy włączyć zasilanie układu pomiarowego oraz uruchomić środowisko BASCOM AVR na komputerze osobistym.

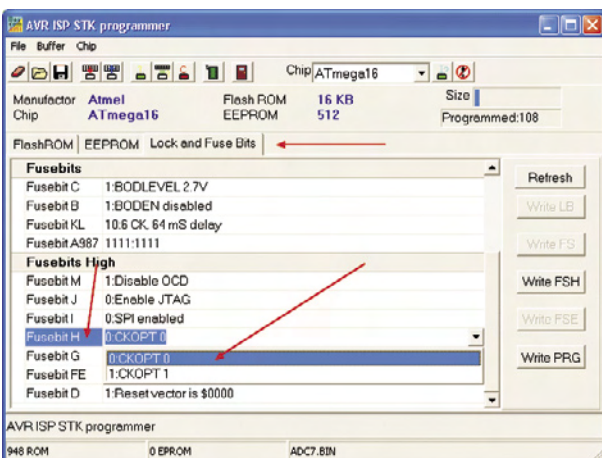
Mikrokontrolery ATmega mogą pracować z różnymi źródłami taktującymi, i aby mikrokontroler mógł działać prawidłowo (a nawet, żeby w ogóle mógł zadziałać) należy najpierw zadeklarować właściwe źródło zegarowe za pomocą bitów konfiguracyjnych (fuse bit). Podobnie jak sam proces programowania pamięci programu (Flash) i nieulotnej pamięci danych dowolnego przeznaczenia (EEPROM), tak samo proces ustawienia bitów konfiguracyjnych dokonywany jest za pomocą tego samego programatora. W pierwszej chwili należy prawidłowo ustawić bity konfiguracyjne, odpowiadające za źródło taktowania mikrokontrolera. W naszym przypadku będzie to zewnętrzny rezonator kwarcowy o częstotliwości 16 MHz. Wyboru źródła taktowania dokonuje się za pomocą odpowiedniego ustawienia bitów CKSEL0, CKSEL1, CKSEL2, CKSEL3. Opis bitów konfiguracyjnych CKSEL zawarto w tab. 1.

Fuse bity odpowiedzialne za wybór źródła taktowania mikrokontrolera są ustawiane fabrycznie na generator wewnętrzny 1 MHz o tolerancji 3%, zapewniając możliwość dalszej kalibracji. Ponieważ budowany rejestrator ma korzystać z zewnętrznego (dokładnego) rezonatora kwarcowego o częstotliwości 16 MHz (największa wartość możliwa do zastosowania przy wybraniu mikrokontrolera ATmega16), wszystkie bity CKSEL należy przestawić na „1”. Na rys. 9 zilustrowano konfigurację fuse bitów odpowiedzialnych za wybór źródła taktującego w przypadku programatora STK200 obsługiwanego wprost z pakietu Bascom AVR (po uprzednim wybraniu w opcjach BASCOM AVR tego właśnie programatora).

Dodatkowo, zgodnie z zaleceniem producenta mikrokontrolera, przy wyborze zewnętrznego rezonatora kwarcowego większego niż 8 MHz





Rys. 9. Konfiguracja fuse bitów dla zewnętrznego rezonatora kwarcowego 16 MHz



Rys. 10. Konfiguracja bitu CKOPT dla zewnętrznego rezonatora kwarcowego 16 MHz

należy ustawić jeszcze bit o nazwie CKOPT, odpowiedzialny za zwiększenie napięcia oscylacji do wartości międzyszczytowej bliskiej lub równej napięciu zasilania mikrokontrolera (*rail to rail*), dzięki czemu wzrasta odporność generatora na zakłócenia. Prawidłową konfigurację bitu CKOPT za pomocą programatora STK200 obsługiwane w środowisku BASCOM AVR pokazano na rys. 10. Po odpowiednim ustawieniu fuse bitów należy te ustawienia zapisać. Służy do tego przycisk *Write FSH* widoczny po prawej stronie rys. 10.

Po prawidłowym ustawieniu fuse bitów należy zaprogramować mikrokontroler programem (*firmware*), który został opisany wcześniej. W tym celu, po uprzednim skompilowaniu programu , należy kliknąć w ikonę programatora . Otworzy się okno programatora, gdzie w polu *Chip* powinna wyświetlić się nazwa użytego mikrokontrolera. Jeżeli proces autodetekcji nie przebiega prawidłowo, oznacza to, że programator nie działa lub jest źle podłączony. Jeżeli wszystko przebiegło prawidłowo, wystarczy teraz kliknąć w ikonę *Auto program chip*. Po chwili urządzenie zostanie zaprogramowane. Z głośniczka piezo

powinien wydobyć się krótki dźwięk oznaczający zgłoszenie się urządzenia po podłączeniu do zasilania lub po wyzerowaniu mikrokontrolera. Od tej chwili urządzenie jest gotowe do pracy. Możemy odłączyć programator i zamknąć środowisko BASCOM AVR.

Na tym etapie należy podłączyć urządzenie do zasilania i połączyć je z komputerem za pomocą kabla RS232, następnie uruchomić program instalacyjny *Rejestrator-Setup.exe* załączony na płycie. Instalator przeprowadzi użytkownika przez krótki i typowy proces instalacji programu.

Dla dociekliwych

Na początku tej części artykułu przeprowadzono dokładną analizę wydajności transmisji danych, z której wynika, że już przy szybkości transmisji 38400 bodów możliwy jest swobodny pomiar z częstotliwością 50 Hz, a nawet większą. W praktyce, mimo wielu prób i starań nie udało się uzyskać prędkości próbkowania większej niż 10 Hz i to przy znacznie większej szybkości transmisji, równej 128000 bodów! Przy próbie wykonywania pomiarów z większą częstotliwością, urządzenie pomiarowe nie zdąża

z wysłaniem pełnego ciągu znaków do aplikacji uruchomionej na PC, przez co nie jest możliwe prawidłowe kreślenie wykresu. Urządzenie spełnia założone wymagania (pomiar miał się odbywać przynajmniej kilka razy w ciągu sekundy). W aplikacji pozostawiono możliwość wyboru częstotliwości maksymalnie do 100 Hz. Aktualizacja firmware w czasie późniejszym (nowa wersja firmware, może być stworzona za pomocą innego języka programowania i kompilatora) powinna rozwiązać problem. I to wyzwanie pozostawiam już prawdziwym specom od programowania – Czytelnikom.

Artykuł powstał w oparciu o pracę dyplomową autora.

Mariusz Ciszewski
mariusz.ciszewski@elportal.pl

Literatura

- [4] Maciej Dorobek, *C++ Builder. Podręcznik*, MIKOM, Warszawa 2002
- [5] Andrzej Daniluk, *RS 232C. Praktyczne programowanie. Od Pascala i C++ do Delhi i Buildera*, HELION, Gliwice 2002

R E K L A M M A



Kolorowe koguty policyjne

Obejźj efekt na www.sklep.avt.pl

AVT 760

www.sklep.avt.pl

Producent: AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11
tel. 022 257 84 50, fax 022 257 84 55
e-mail: handlowy@avt.pl