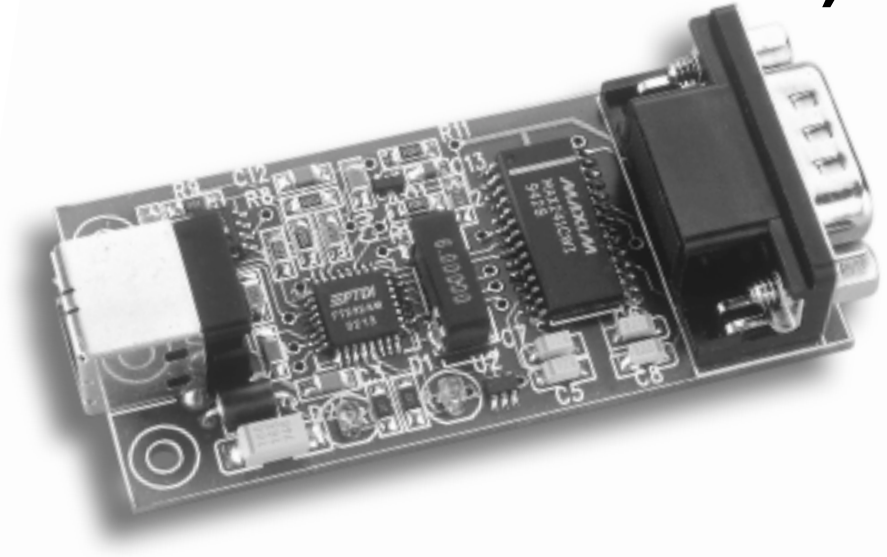


Konwerter USB<->RS232, część 2

AVT-5080



Drugą część artykułu poświęcamy przedstawieniu budowy konwertera USB<->RS232.

Zawarte w niej informacje wystarczą do samodzielnego wykonania i uruchomienia konwertera.

Czytelników zainteresowanych tym tematem zachęcamy do sięgnięcia po wrześnieowy numer EP.

Rekomendacje: *jest to wzorcowe opracowanie dla wszystkich konstruktorów zamierzających stosować we własnych konstrukcjach interfejs USB.*

W celu nabrania wprawy w korzystaniu z układów firmy FTDI oraz uniknięcia ewentualnych niespodzianek, najpierw wykonano podstawową, dokładnie opisaną aplikację konwertera USB - RS 232.

Schemat elektryczny konwertera pokazano na **rys. 7**. Jest on prawie identyczny ze schematem konwertera zalecanym przez producenta. Różni się od niego tylko typem interfejsu RS232, co wynikało z posiadanych akurat zapasów.

Sygnał różnicowy oraz zasilanie doprowadzone są przez gniazdo USB typu B. Zasilanie jest filtrowane za pomocą obwodu zbudowanego z elementów C1, L1, C9 oraz kilku rozmieszczonych na płytce kondensatorów 100 nF. Zasilanie części analogowej układu FT8U232 jest dodatkowo filtrowane przez filtr dolnoprzepustowy zbudowany z elementów R5, C10 (filtr służy do zasilenia analogowego układu powielacza częstotliwości 8x).

Rezystory R1, R2 zapewniają dopasowanie do impedancji wejściowej transceivera. Podanie napięcia 3,3 V (z wewnętrznego stabilizatora) poprzez rezystor R3 na linię D+ informuje hub o dołączeniu konwertera do magistrali.

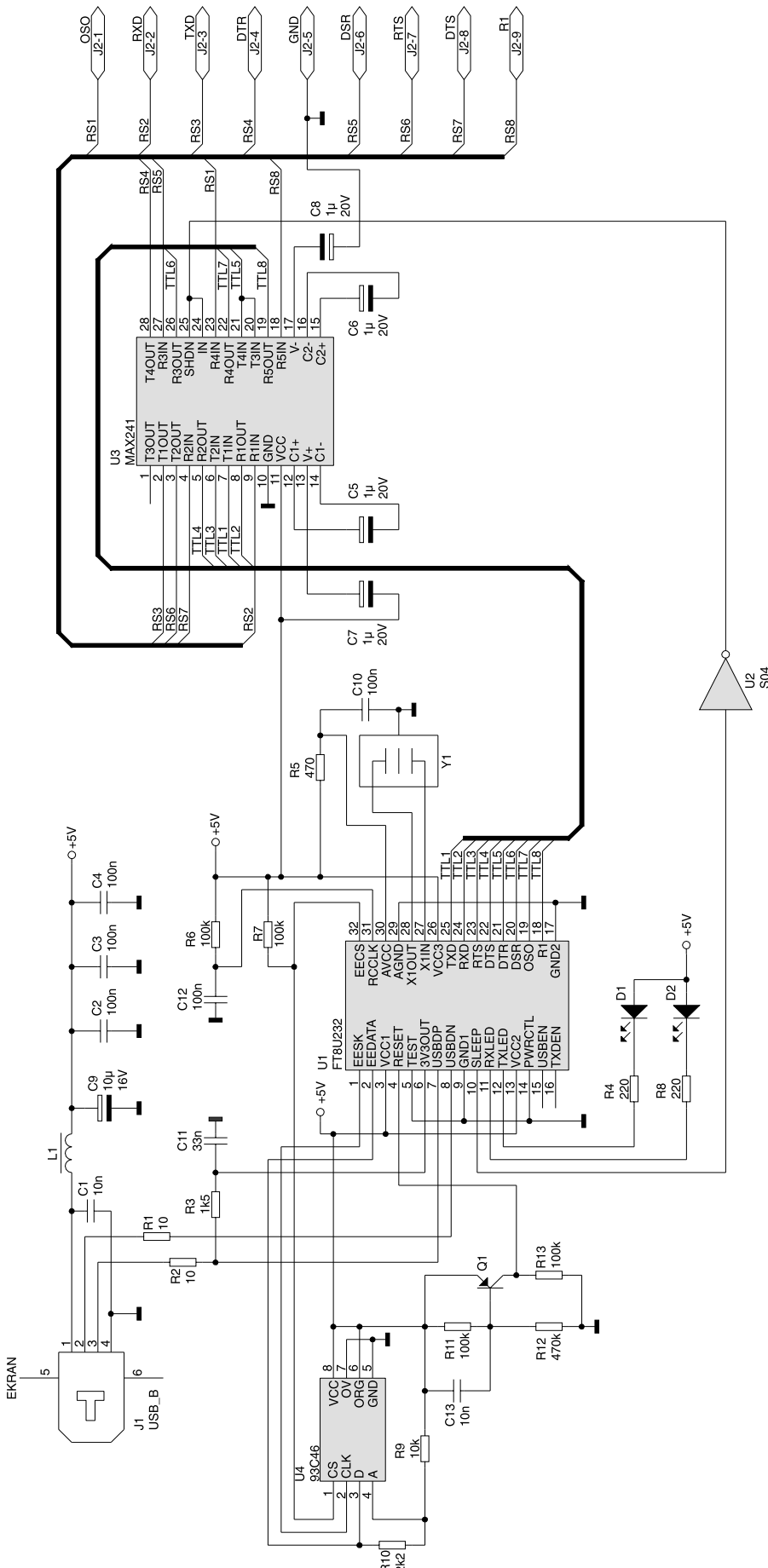
Linie EECS, EESK, EEDATA służą do obsługi opcjonalnej pamięci EEPROM 93C46. Dodatkowo EECS, jako linia wejściowa, pozwala na wybór zastosowanego kwarcu. Podciągnięcie do za-

silania (R7) odpowiada pracy z oscylatorem o częstotliwości 6 MHz i z 8-krotnym powieleniem. Dołączanie linii do masy przełącza oscylator na docelowe 48 MHz.

Na uwagę zasługuje dosyć rozbudowany układ zerowania. Po włączeniu zasilania na wyprowadzeniu RESET utrzymuje się niski poziom napięcia. Tranzystor Q1 jest włączany (dzięki dzielnikowi R11, R12) dopiero po osiągnięciu napięcia $V_{cc}=3,6$ V (U4). Kondensator C13 wprowadza niewielkie opóźnienie ok. 2 ms (zauważmy, że czas zerowania układu powinien być krótszy niż 10 ms - po tym czasie wystawiany jest sygnał zerowania magistrali przez hub - aby układ był od razu gotów do obsługi transferu kontrolnego).

Oddzielnego zerowania wymaga powielacz częstotliwości. Powinien on być zablokowany przez kilka milisekund zanim oscylator nie zacznie stabilnie pracować. Uzyskuje się to poprzez podanie poziomu niskiego na wyprowadzeniu RCCLK. Jednocześnie układ wymaga, aby dla stabilnego uruchomienia powielacza jego odblokowanie następowało przy wysokim poziomie na wejściu RESET. W konwerterze osiąga się to przez dobór odpowiedniej stałej czasowej obwodu R6, C12.

Na wyjściach RXLED i TXLED występują przebiegi impulsowe podczas napływu danych do bufora odbiorczego oraz ich wysy-



Rys. 7. Schemat elektryczny konwertera USB<->RS 232C

łania z bufora nadawczego. Diody LED dołączone przez R4 i R6 pozwalają na optyczną kontrolę transmisji. Nie będziemy ich montować na płytce schowanej w obudowie, jednak na etapie testowania są bardzo przydatne.

Wyjście USBEN służy do sygnalizacji dołączenia do magistrali - może być wykorzystane np. do przełączenia kanału transmisyjnego w urządzeniach wyposażonych zarówno w USB, jak i zamienny RS232C.

Wyjście TXDEN jest aktywne przy wysyłaniu danych aż do chwili całkowitego opróżnienia bufora nadajnika. Służy do półduplexowych połączeń RS485 i eliminuje programowe przełączanie kierunku transmisji, konieczne przy korzystaniu ze zwykłego portu.

Wyjście SLEEP informuje o trybie *suspend* (zawieszenia) magistrali. Tryb ten (o czym nie wspomniano wcześniej) dotyczy zachowania się konwertera podczas przełączenia hosta w uśpienie (*stand-by*). Wówczas host przestaje wysyłać pakiety SOF. Brak 3 pakietów (czyli przerwa 3 ms) jest wykrywany przez układ FT8U232, który także przechodzi w stan uśpienia z maksymalnie zmniejszonym poborem mocy. W trybie *suspend* hub może dostarczyć do konwertera co najwyżej 0,5 mA. Sam FT8U232 ma oczywiście wbudowane odpowiednie mechanizmy redukcji mocy, jednak pozostała część układu musi być sprzętowo wyłączona. Jest to zapewnione poprzez podłączenie wyjścia SLEEP do wejścia *shutdown* interfejsu RS232C. Oczywiście, najlepiej jest, jeśli aktywne poziomy napięć są całkowicie zgodne. Ponieważ jednak zamiast MAX213 zastosowano posiadany MAX241, konieczne było dopasowanie poziomów za pomocą jednobramkowego inwertera U2. Układ MAX241 ogranicza też szybkość transmisji do tradycyjnych dla portu szeregowego 115 kbd, podczas gdy FT8U232 oferuje transfer do 920 kbd - w razie potrzeby można wlotować nowocześniejszy interfejs - jest zachowana zgodność wyprowadzeń.

Ponieważ FT8U232 występuje tylko i wyłącznie w obudowie

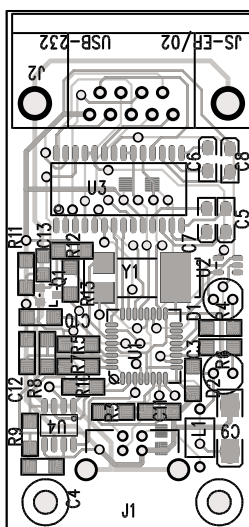
TQFP, w pozostałej części płytki także użyto głównie elementów SMD.

Nie ma raczej odwrotu od upowszechniania się tej technologii również w ręcznie lutowanych urządzeniach amatorskich. Po zdobyciu pewnej wprawy stwierdzimy, że montaż jest częstokroć szybszy i łatwiejszy niż przy sposobach tradycyjnych. Także kompletacja takich elementów sprawia coraz mniej kłopotu.

Elementy są wlotowane na płytkę drukowaną zgodnie z rys. 8. Kolejność lutowania ustalamy tak, aby zachować jak najlepszy dostęp do elementów (rezystory i kondensatory, półprzewodniki, kwarc, na końcu elementy przelewiane i gniazda). Kondensatory C1, C2, R1, R2 są montowane na spodzie płytki. Podczas początkowych testów można pominąć pamięć EEPROM.

Układ prototypowy został złożony bez pomocy specjalnych akcesoriów (jak np. grot *wave* i odpowiednie topniki) - wystarczył cienki grot i tinol 0,5 mm oraz taśma rozlutownicza *Wick*. Należy jednak zwracać uwagę na kolejność wyprowadzeń - wylutowanie np. obróconego o 180° układu będzie już wymagało bardziej wyrafinowanego sprzętu.

Z tego też względu warto sprawdzić przed montażem prawidłowość wykonania metalizacji na płytce - zajmie to chwilę, a może oszczędzić poważnych kłó-



Rys. 8. Rozmieszczenie elementów na płytce konwertera

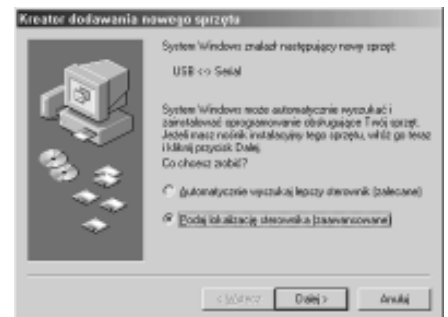
potów. Po usunięciu resztek topnika preparatem Kontakt PCC i wysuszeniu płytka jest gotowa do uruchomienia.

Przygotowanie sterowników

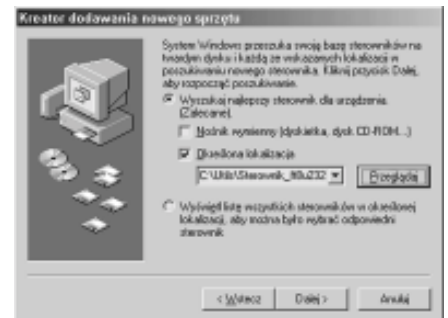
Przed pierwszym podłączeniem konwertera do komputera musimy się wyposażyć w odpowiedni sterownik. Na stronie producenta mamy do dyspozycji sterowniki dla różnych systemów operacyjnych (prototyp był sprawdzany pod Windows 98 oraz ME). Wybieramy odpowiedni i zapisujemy go na dysk. Należy też zwrócić uwagę, że FT8U232 może być obsługiwany (pod Windows) za pomocą kilku sterowników:

- podstawowy sterownik portu szeregowego, który przechwytyje wywołania funkcji API i przekierowuje je do stosu,
- USB (w komunikacji korzystamy wtedy z typowych funkcji Windows obsługi portu, w Delphi bez problemu można użyć dotychczasowe komponenty RS, np. *TComPort* lub *TRsPort* ze strony EP),
- rozszerzony sterownik portu - posiada możliwości jak wyżej, ale dodatkowo obsługuje mechanizm *Plug & Play* (wykrywanie przy starcie Windows nowych urządzeń - jeśli nie budujemy urządzenia, które zareaguje na wywołanie PnP, lepiej tego sterownika nie stosować, gdyż system będzie dłużej czekać na odpowiedź, co spowolni uruchamianie komputera),
- sterownik bezpośredni (*direct*), który wymaga oddzielnych funkcji do obsługi portu (funkcje te są udostępniane w dołączonej bibliotece *dll*) - nie można zatem korzystać z gotowych komponentów, ale za to mamy dostęp do rozszerzonego zestawu operacji (ustawianie nietypowych szybkości transmisji, dostęp do zawartości EEPROM).

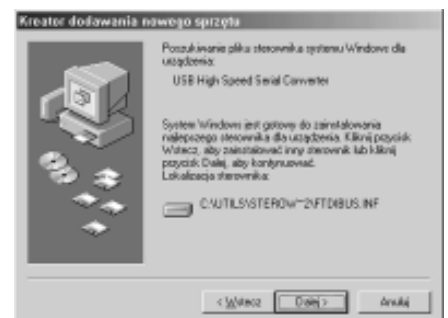
Dla potrzeb naszego projektu będą nam potrzebne dwa sterowniki: podstawowy oraz *direct*. Po ściągnięciu ze strony FTDI (www.ftdichip.com, publikujemy je także na naszej płytce CD-EP10/2002B) potrzebnych plików (dla



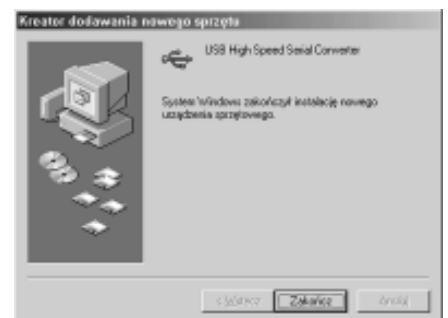
Rys. 9. Wykrycie przez Windows płytki konwertera



Rys. 10. Podanie lokalizacji potrzebnego sterownika

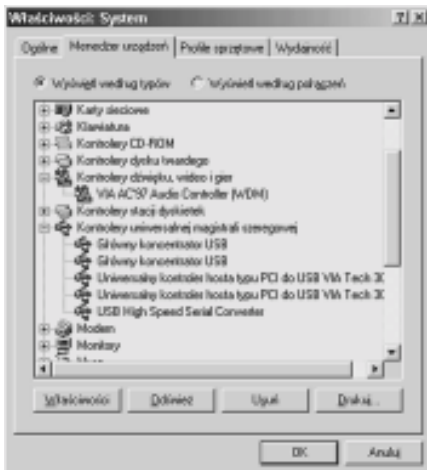


Rys. 11. Potwierdzenie odszukania sterownika



Rys. 12. Zakończenie instalacji konwertera jako wirtualnego portu COM

Windows 98/ME są to - w chwili pisania artykułu - *N8002101.zip* oraz *D2XX10401.zip*) rozpakowujemy je do oddzielnych folderów (o dowolnych nazwach) utworzonych na twardym dysku.



Rys. 13. Kontroler USB w oknie menedżera sprzętu Windows

Uruchomienie układu

Teraz nadchodzi chwila prawdy. Podłączamy kablem naszą płytkę do gniazda *downstream* komputera (lub huba). Poprawnie zmontowane urządzenie jest od razu wykryte przez system, który prosi o wskazanie lokalizacji sterownika (rys. 9) - zaznaczamy opcję *Podaj lokalizację sterownika*.

Po potwierdzeniu otwiera się okienko lokalizacji (rys. 10) - zaznaczamy opcję *Określona lokalizacja* i wybieramy folder, do którego rozpakowaliśmy pliki podstawowego drivera wirtualnego portu COM.

Windows potwierdza znalezienie prawidłowego sterownika (rys. 11), a następnie informuje o poprawnym zakończeniu instalacji nowego sprzętu (rys. 12).



Rys. 14. Wirtualny port COM w oknie menedżera sprzętu Windows

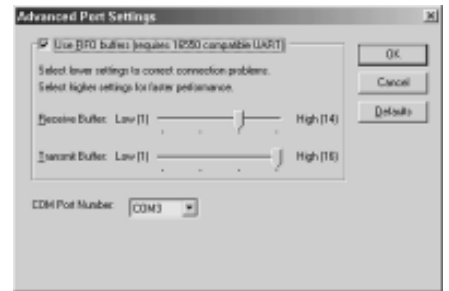
Po załadowaniu sterownika Windows umieszcza konwerter na swojej liście sprzętu i przydziela mu numer portu szeregowego - jest to widoczne w oknie menedżera urządzeń pod pozycjami *kontroler uniwersalnej magistrali szeregowej* oraz *porty* - rys. 13 i 14. Zwróćmy uwagę, że zapis na liście menedżera jest każdorazowo aktualizowany przy odłączaniu i dołączaniu płytki - możemy się naocznie przekonać, jak działa mechanizm wykrywania urządzeń przez magistralę USB. Numer portu jest wybrany jako pierwszy wolny, ale możemy go dowolnie zmienić, korzystając z okna dialogowego *Port>Właściwości>Zaawansowane* - rys. 15.

W tej chwili konwerter jest już praktycznie gotowy do działania. Jedyny problem to podłączenie dwóch płytek (bez ustawionych numerów seryjnych) do jednego komputera - wystąpią kłopoty z identyfikacją (mankament ten jest usunięty w wersji układu FT8U232BM).

Programowanie pamięci EEPROM

Dla zaprogramowania pamięci 93C46 na płytce służy bezpłatny program *ftd2xxst.exe* (pobrany ze strony FTDI wraz z opisem w formacie pdf). Wymaga on jednak zainstalowania wspomnianego wcześniej sterownika *direct*. Aby nie usuwać użytego do tej pory sterownika portu wirtualnego, wykorzystamy standardowy mechanizm aktualizacji sterowników w Windows. W oknie menedżera sprzętu wybieramy nasz kontroler USB (rys. 16) i otwieramy jego właściwości, w których uruchamiamy opcję *Aktualizuj sterownik* (rys. 17).

Przejdziemy wtedy przez szereg okien podobny do pierwszej instalacji - ale jako lokalizację podajemy folder z rozpakowanymi plikami sterownika *direct*. Po zakończeniu przeinstalowania w menedżerze sprzętu zmienia się opis konwertera a także znika wirtualny port szeregowy (rys. 18) - od tej chwili do obsługi UART-u musimy używać oddzielnych funkcji z dołączonej biblioteki *dll* (na stronie znajdziemy szereg przykładów dla popularnych środowisk



Rys. 15. Ustawienie numeru portu szeregowego

programistycznych, oczywiście także dla Delphi).

Teraz możemy uruchomić program serwisowy. Po jego uruchomieniu (rys. 19) należy wypełnić wszystkie pola edycyjne podstawowego opisu urządzenia (dalsze opcje uaktywniają się dopiero po ponownym przejściu do pola *Manufacturer klawiszem Tab*). Tutaj jedna bardzo istotna uwaga - dla własnych warsztatowych (a nie produkcyjno-komercyjnych) potrzeb nie warto zmieniać żadnych identyfikatorów poza opisem. Dotyczy to zwłaszcza numerów VID i PID, na podstawie których Windows identyfikuje potrzebny sterownik. Przy zmianie VID/PID należy wyszukać i zmienić również wpisy w odpowiednich plikach **.inf* - w przeciwnym razie system nie odnajdzie sterownika i konwerter pozostanie nieznanym urządzeniem.

Po zapamiętaniu nastaw włączamy uaktywnione wtedy *ustawienia zaawansowane (advanced setup)* - rys. 20. Umożliwiają one:

- Wybór opcji *Plug and Play* - pozostawiamy niezaznaczoną,
- Wybór ręcznego lub automatycz-



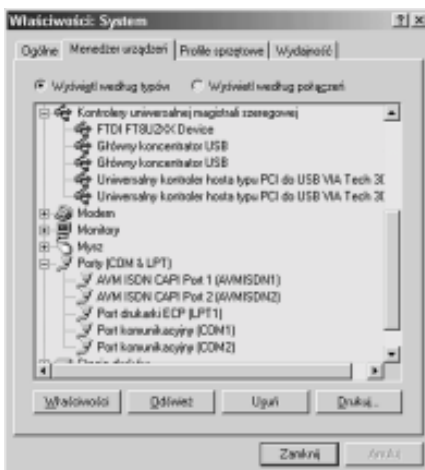
Rys. 16. Rozpoczęcie wymiany sterownika



Rys. 17. Okno umożliwiające aktualizację (wymianę sterownika)

nego przyznania numeru seryjnego (automatyczny jest wygodny, gdyż zapobiega - przy programowaniu konwertera w danym komputerze - powtórzeniu się numeru. Dla potrzeb naszego projektu obejmującego tylko kilka płytek został jednak wybrany tryb ręczny),

- Wybór sposobu zasilania: z magistrali albo samodzielne - pozostawiamy zasilanie z magistrali (nie jest to zbyt istotne, gdyż nadrzędny dla określenia sposobu zasilania jest poziom na wejściu PWRCTL kostki, co pozwala na sprzętową kontrolę aktualnie używanego źródła zasilania w bardziej rozbudowanych układach),
- Udostępnienie funkcji zdalnego budzenia *hosta* z trybu *stand-by*. Układ wykonuje to ustawiając parę różnicową w stan K (odwrócona polaryzacja) - dalszym przekazaniem tego sygna-



Rys. 18. Menedżer urządzeń po przeinstalowaniu sterownika

łu zajmuje się już hub. W naszym przykładzie opcja pozostaje nieczynna (MAX241 nie posiada funkcji zachowania aktywności wybranych linii wejściowych w trybie *shutdown*, a tylko to pozwala fizycznie dostarczyć do konwertera sygnał budzący),

- Określenie maksymalnego poboru prądu: konwerter (jak wynika z not katalogowych użytych układów) pobiera zawsze poniżej 100 mA. Umożliwia to bezproblemowe zasilanie z magistrali USB, gdyż każdy hub gwarantuje 100 mA w portach *downstream* (nawet w trakcie enumeracji). W polu edycyjnym została wpisana przykładowo wartość nieco mniejsza.
- Po wprowadzeniu danych wykonujemy programowanie i kontrolny odczyt EEPROM - wynik tej operacji widzimy na **rys. 21**.

Sprzętowy test konwertera

Program serwisowy umożliwia też przeprowadzenie kompletnego testu sprzętowego. Wymagane jest jednak do tego posiadanie w komputerze wolnych portów szeregowych COM1 i COM2 oraz przygotowanie odpowiedniego kabla połączeniowego:

Pin 3 (TXD)	-	COM2 Pin 2 (RXD)
Pin 2 (RXD)	-	COM2 Pin 3 (TXD)
Pin 7 (RTS)	-	COM2 Pin 8 (CTS)
Pin 8 (CTS)	-	COM2 Pin 7 (RTS)
Pin 6 (DSR)	-	COM2 Pin 4 (DTR)
Pin 5 (GND)	-	COM2 Pin 5 (GND)
Pin 4 (DTR)	-	COM2 Pin 6 (DSR)
Pin 1 (CD)	-	COM1 Pin 4 (DTR)
Pin 9 (RI)	-	COM1 Pin 7 (RTS)

- dla złącza DB9.

Uwaga! Test uruchamiamy dopiero po wykonaniu wszystkich połączeń - w przeciwnym przypadku program się zawiesza (w przypadku Windows 98 razem z systemem). Wynik poprawnie przeprowadzonego testu jest pokazany na **rys. 22**.

Teraz możemy - o ile zachodzi taka potrzeba - powrócić do sterownika wirtualnego portu COM. Ponieważ w systemie są już zainstalowane oba sterowniki, procedura znacznie się upraszcza: przy następnej aktualizacji wybierzemy w oknie lokalizacji opcję *Wyświetl listę wszystkich sterowników* i zaznaczymy potrzebny.



Rys. 19. Okno główne programu serwisowego

Zazwyczaj do wstępnego sprawdzenia najbardziej odpowiedni będzie pierwszy z wymienionych sterowników.

Teraz nadchodzi „chwila prawdy”. Podłączamy kablem naszą płytkę do gniazda *downstream* komputera (lub huba). Poprawnie zmontowany konwerter jest od razu wykryty przez system, który prosi o wskazanie lokalizacji sterownika. Podajemy ścieżkę do folderu, w którym ulokowaliśmy odpowiedni sterownik FTDI. Po załadowaniu sterownika Windows umieszcza konwerter na swojej liście sprzętu i przydziela mu nu-

WYKAZ ELEMENTÓW

Rezystory

- R1, R2: 10Ω 805
- R3: 1,5kΩ 1206
- R4, R6: 220Ω 1206
- R5: 470Ω 1206
- R7, R8, R11, R13: 100kΩ 1206
- R9: 10kΩ 1206
- R10: 2,2kΩ 1206
- R12: 470kΩ 1206

Kondensatory

- C1, C13: 10nF 805 ceramiczny
- C2...C4: 100nF 1206 ceramiczny
- C5...C8: 1μF/20V 3216 tantalowy
- C9: 10μF/16V 6032 tantalowy
- C10, C12: 100nF 1206 ceramiczny
- C11: 33nF 1206 ceramiczny

Półprzewodniki

- D1, D2: LED 3mm
- Q1: BC857 SOT23
- U1: FT8U232AM
- U2: NC7S04 SOT23-5
- U3: MAX241 SO28
- U4: 93C46 SO8

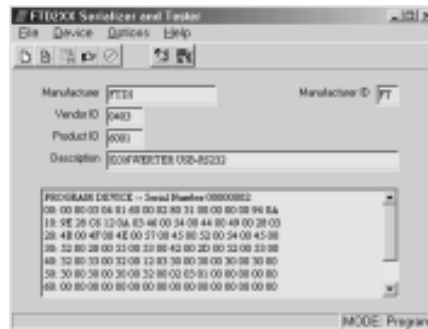
Różne

- Y1: oscylator ceramiczny 6MHz
- J1: gniazdo USB typu B
- J2: gniazdo kątowe DB-9M
- L1: koralik ferrytowy przewlekany



Rys. 20. Okno zaawansowanych ustawień układu FT8U2xxAM

mer portu szeregowego (obejrzymy to w oknie menedżera urządzeń pod pozycjami „porty“) oraz kontroler uniwersalnej magistrali szeregowej. Pierwszy szybki test możemy wykonać wysyłając cokolwiek z dowolnego programu terminala na port powiązany z konwerterem - dioda *TxLed* mruga potwierdzając wysyłanie, a na wyprowadzeniu 3 gniazda pojawia się przebieg impulsowy. Aby w pełni sprawdzić poprawność działania konwertera, musimy zestawić kompletny tor transmisyjny i skontrolować obustronne nadawanie/odbiór oraz ustawianie i odczyt linii kontrolnych. Można też przeprowadzić firmowy test, ale wymaga to posiadania wolnych portów COM1 i COM2, przygotowania kabla testowego oraz przeładowania sterownika (wymagany *direct*). Test jest wykonywany za pomocą programu dostępnego na stronie www.ftdichip.com.

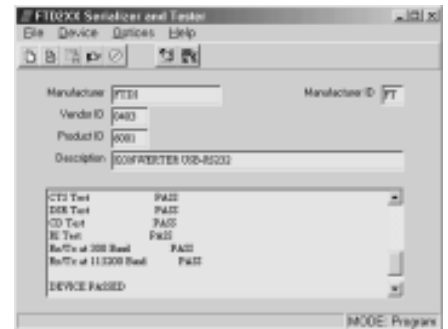


Rys. 21. Odczyt kontrolny zaprogramowanej pamięci

Znajduje się tam też PDF z dokładnym opisem obsługi, którego tutaj nie będziemy przytaczać. Ten sam program służy do zapisywania współpracującej pamięci EEPROM - jest to łatwe dzięki załączonym szczegółowym instrukcjom. Należy jedynie pamiętać o korekcie w plikach **.inf* sterowników w razie zmiany numerów VID lub PID (system przeszukuje pliki *inf* i szuka zgodnych numerów - jeśli nie znajdzie, to urządzenie pozostanie nieznanne).

Co zrobić, gdy układ nie daje znaku życia? Cóż - nie ma zbyt wielkiego pola manewru. Sprawdźmy dokładnie płytkę: przejścia, zvarcia, kolejność wyprowadzeń, zimne luty, wartości pomocniczych elementów. Spróbujmy uruchomić na innym komputerze i kablu. O ile to wszystko nie pomoże - pozostaje niestety wymiana FT8U232 lub złożenie następnego egzemplarza konwertera.

Materiały pomocnicze FTDI zawierają wiele dodatkowych informacji na temat zastosowanego



Rys. 22. Wynik poprawnego testu sprzętowego konwertera

układu oraz zalecenia do programowania transmisji wynikające ze specyfiki transferów *bulk* używanych od strony magistrali USB. Warto przynajmniej częściowo zapoznać się z nimi przed pisaniem własnych procedur obsługi komunikacji z użyciem konwertera oraz budową innych urządzeń korzystających z FT8U232.

Jerzy Szczesiul, AVT
jerzy.szczesiul@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/pazdziernik02.htm> oraz na płycie CD-EP10/2002B w katalogu PCB.

W przygotowaniu artykułu wykorzystałem:

- specyfikację standardu USB rev. 1.0 (m.in. na stronie <http://www.ep.com.pl>),
- USB Design by Example, John Hyde, Wiley Computer Publishing 1999,
- materiały ze strony FTDI <http://www.ftdichip.com/>,
- materiały ze strony <http://www.beyondlogic.org/>,
- katalogi podzespołów Maxim oraz Elfa.