

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji. Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

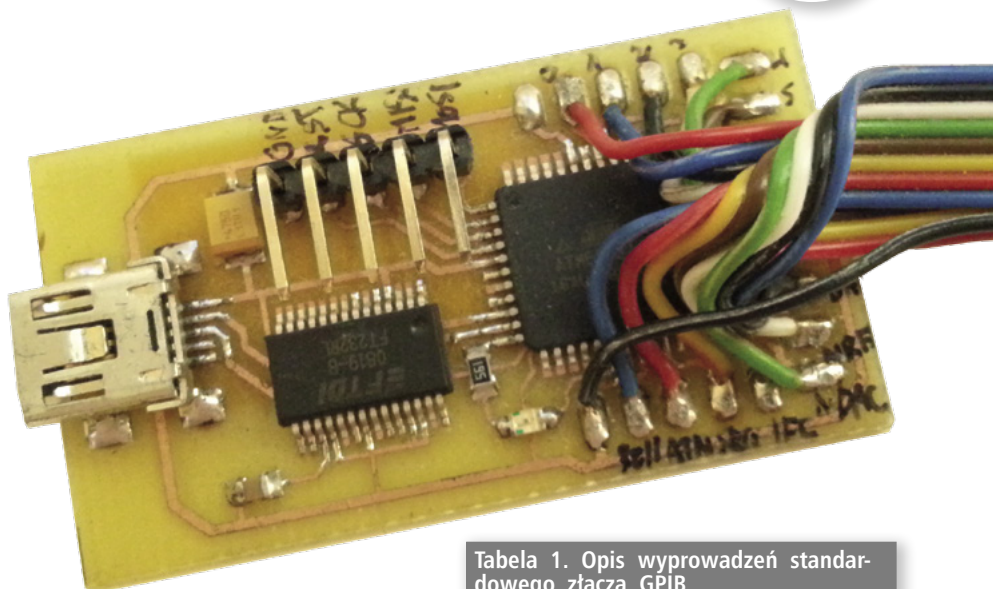
Projekt
207

Konwerter GPIB/USB

W ręce hobbystów trafiają coraz częściej przyrządy pomiarowe starszej generacji. W latach ich świetności podstawowym interfejsem komunikacyjnym był GPIB (IEEE488, w wariancie firmy Hewlett-Packard, obecnie Agilent, oznaczony HPIB). Aparatura ta, mimo, że jej parametry ciągle przewyższają popularne i niedrogie chińskie przyrządy, ma poważną wadę, którą jest brak złącza USB lub RS232. Użytkownik nie ma praktycznie żadnej możliwości zachowania wyniku pomiaru, poza zapisaniem w pamięci wewnętrznej. Konwerter opisywany w artykule konwerter powstał w celu umożliwienia udokumentowania wyników pracy.

IEEE488 pozwala na połączenie do 15 urządzeń na wspólnej magistrali. Dane są przesyłane równolegle za pomocą szyny 8-bitowej. Maksymalna prędkość transmisji to 1 MB/s, jednak zwykle w praktyce jest stosu się mniejszą. Transmisja jest typu *half duplex*, w danym momencie na magistrali może być aktywny tylko jeden nadawca (urządzenie w stanie *Talk*) i do czternastu odbiorców (w stanie *Listen*). Komunikacją zarządza kontroler magistrali i jest to jedyne urządzenie, które ma możliwość specyfikowania, kto na magistrali ma możliwość nadawania i kto może te dane odbierać. Każde urządzenie ma swój unikalny, 5-bitowy adres ustawiany w menu urządzenia lub przełącznikami typu dip-switch. Od strony elektrycznej magistrala GPIB to 8 bitów danych, 3 linie do kontroli przepływu danych oraz 5 linii do zarządzania magistralą. Wszystkie poziomy napięcie są zgodne z TTL. Typowym złączem jest 24-pinowy wtyk podobny do złącza Centronics używanego w drukarkach. W tabeli 1 umieszczono listę wyprowadzeń złącza używanego przez IEEE488. Z opisem magistrali HPIB można zapoznać się w dokumencie [1].

Wszystkie linie mają logikę ujemną. Wartość „prawda” danej linii jest oznaczony



przez poziom logiczny niski, a „fałsz” przez poziom wysoki. Również linie danych są znegowane. Do kontroli przepływu danych na magistrali używane są 3 linie:

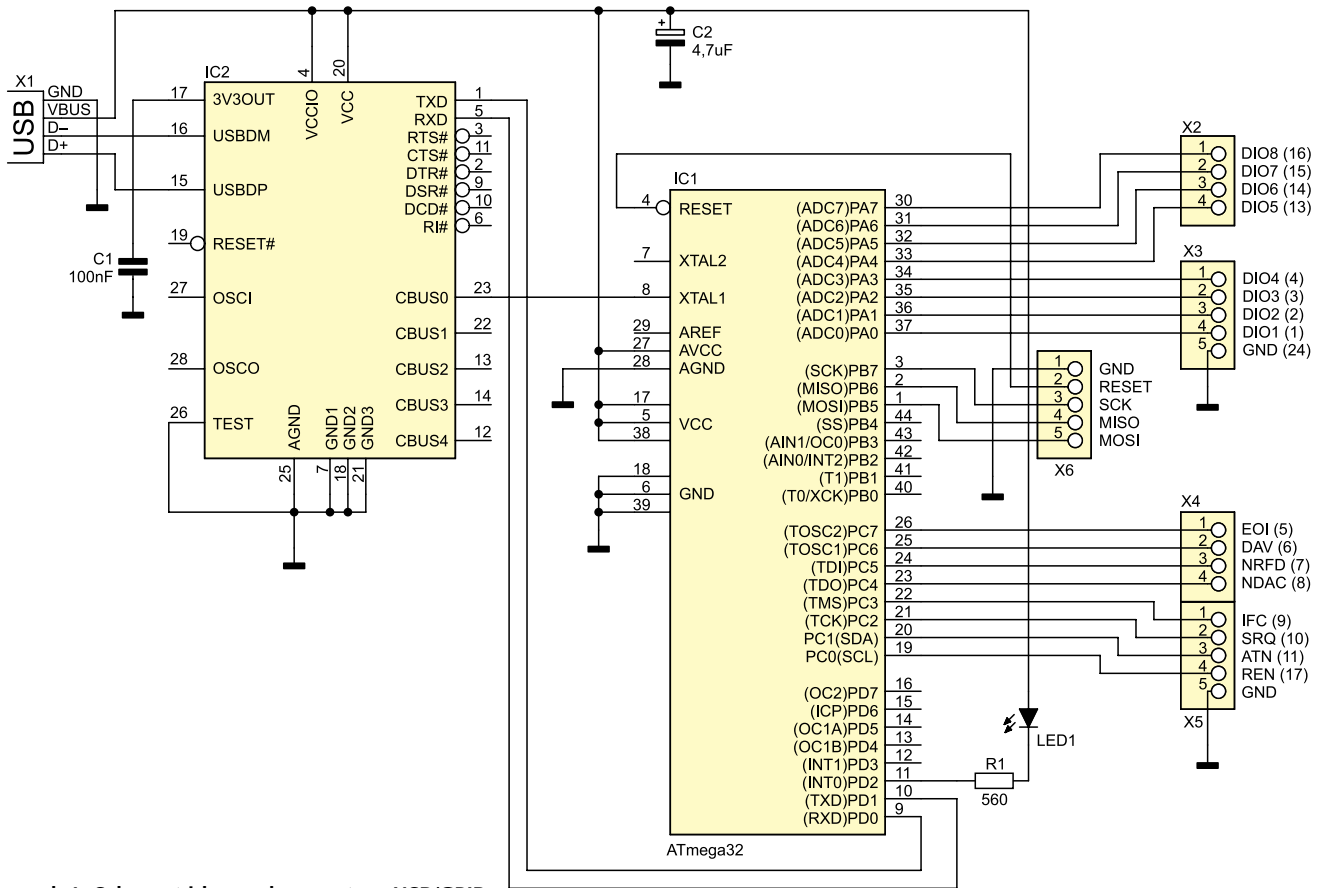
- *DAV (Data Valid)*. Stan „prawda” linii sygnalizuje, że dane na magistrali są ustalone i można je odczytać. Linia sterowana jest przez urządzenie zaadresowane jako Talker.
- *NRFD (Not Ready For Data)*. Linia NRFD steruje urządzenie będące w trybie Listen. Stan „prawda” tej linii oznacza, że urządzenie nie jest gotowe do odebrania danych. Stan gotowości do odbierania danych jest ustawiany poprzez przestawienie linii w stan false czyli ustawienie stanu wysokiego. Ponieważ linie wyjściowe urządzeń GPIB pracują jako wyjścia typu otwarty kolektor, dopiero gdy wszystkie urządzenia zasygnalizują gotowość do odbierania danych, linia NRFD fizycznie przyjmie poziom logiczny wysoki.
- *NDAC (Not Data Accepted)*. Urządzenie odbierające dane zmienia poziom tej linii ze stanu logicznego 0 na jedynkę (fałsz) w celu poinformowania nadawcy, że dane z linii DIO zostały przez nie odczytane. Dopiero wtedy, gdy wszystkie urządzenia *Listen* przestawią linie w stan „fałsz”, stan logiczny linii zmieni się na wysoki, a nadawca uzna, że wysłane przez niego dane zostały odczytane i można przesłać następne.

Jak wspomniano, 5 dodatkowych linii magistrali jest używanych do kontroli:

Tabela 1. Opis wyprowadzeń standardowego złącza GPIB

Pin	Nazwa	Opis	Źródło danych
1	DIO1	Data bit 1 (LSB)	Talker
2	DIO2	Data bit 2	Talker
3	DIO3	Data bit 3	Talker
4	DIO4	Data bit 4	Talker
5	EOI	End Or Indentity	Talker/Controller
6	DAV	Data Valid	Controller
7	NRFD	Not Ready For Data	Listener
8	NDAC	No Data Accepted	Listener
9	IFC	Interface Clear	Controller
10	SRQ	Service Request	Talker
11	ATN	Attention	Controller
12		Ekran	
13	DIO5	Data bit 5	Talker
14	DIO6	Data bit 6	Talker
15	DIO7	Data Bit 7	Talker
16	DIO8	Data bit 8 (MSB)	Talker
17	REN	Remote Enabled	Controller
18		GND DAV	
19		GND NRFD	
20		GND NDAC	
21		GND IFC	
22		GND SRQ	
23		GND ATN	
24		GND data	

- *ATN (Attention)*. Poziom wysoki na tej linii oznacza, że magistrala jest w trybie danych. W tym trybie dane są przesyła-



Rysunek 1. Schemat ideowy konwertera USB/GPIB

ne między zaadresowanymi urządzeniami. Poziom niski wymusza tryb komend, dzięki czemu kontroler ma możliwość wyboru adresu urządzeń, które mogą wysyłać dane (Talk, w danym momencie tylko jedno urządzenie może być w tym trybie) a które dane będą odbierać (Listen).

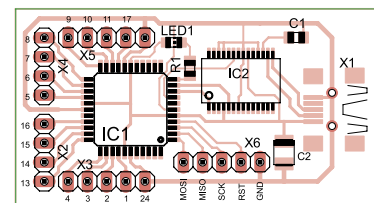
- *REN (Remote Enable)*. Kontroler ustawiając tą linię w stan „prawda” nakazuje wszystkim zaadresowanym urządzeniom przejście w tryb sterowania zdalnego (remote) – zwykle urządzenie w tym trybie ma zablokowany panel operatora, a użytkownik może dokonać wymuszenia przejścia w tryb lokalny przy użyciu dedykowanego przycisku na urządzeniu (*Local*). Przejście urządzeń w tryb lokalny nastąpi także, gdy linia REN zostanie ustawiona w stan „fałsz”.
- *IFC (Interface Clear)*. Kontroler ustawiając tą linię w stan true (poziom logiczny 0) wymusza przejście wszystkich urządzeń na magistrali w stan nieaktywny. Powoduje to skasowanie wszystkich urządzeń wcześniej zaadresowanych urządzeń. Domyślnie ta linia jest w stanie „fałsz”.
- *EOI (End of identity)*. W przypadku występowania stanu „fałsz” w linii ATN stan linii EOI jest kontrolowany przez aktywnego nadawcę i sygnalizuje wysłanie ostatniego bajtu komunikatu.
- *SRQ (Service Request)*. Linia ta jest wykorzystywana przez urządzenia w stanie

Listen w celu zasygnalizowania kontrolerowi magistrali gotowość do wysłania danych (przykładowo jeżeli zakończył się proces pomiaru) lub wystąpienie błędu.

Opis konwertera

Schemat konwertera pokazano na **rysunku 1**. Zbudowano go z użyciem mikrokontrolera ATmega32 połączonego z komputerem PC za pomocą konwertera FT232RL. Dzięki temu system operacyjny komputera PC „widzi go” jako wirtualny port COM. Konwerter jest zasilany ze złącza USB.

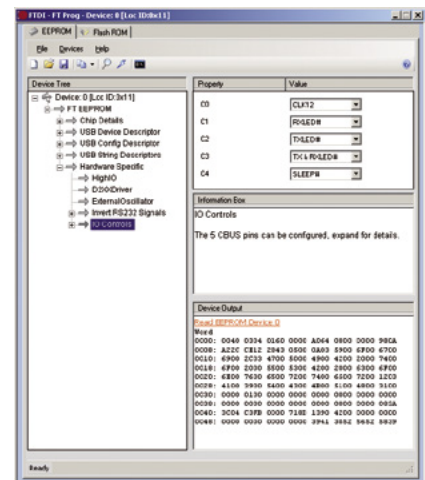
Konwerter zmontowano z użyciem elementów SMD na płytce jednostronnej (**rysunek 2**). Jego konstrukcja jest jednak tak mało skomplikowana, że można go z z powodzeniem zmontować na płytce uniwersalnej używając mikrokontrolera w obudowie DIP-40. Należy przy tym pamiętać o innym rozmieszczeniu wyprowadzeń, niż w wersji w obudowie TQFP44. Jako konwerter RS232/USB można zastosować jakkolwiek płytkę z układem FT232, PL2303 lub podobnym. Ponieważ sygnał zegarowy dla mikrokontrolera jest pobierany z FT232RL, montując układ na płytce uniwersalnej można dołączyć do mikrokontrolera kwarc 12 MHz oraz dwa kondensatory 22...27 pF. Stosując zewnętrzny kwarc należy pamiętać o odpowiedniej zmianie fusebits w mikrokontrolerze. Montaż układu z racji małej ilości elementów jest prosty. Należy go wykonać zgodnie z rys. 2.



Rysunek 2. Schemat montażowy konwertera USB/GPIB

Uruchomienie konwertera

Uruchomienie należy rozpocząć od podłączenia konwertera do komputera PC oraz zainstalowania sterowników dla układu FT232RL. Sterowniki można pobrać ze strony producenta [2]. Kolejnym etapem jest skonfi-



Rysunek 3. Konfiguracja układu FT232RL



Rysunek 4. Ustawienie bitów bezpieczników mikrokontrolera

gurowanie układu FT232RL. Należy przy tym wykorzystać program FT_PROG (do pobrania ze strony [3]). Konfigurację układu FT232RL pokazano na **rysunku 3**. Jedyną zmianą z stosunku do konfiguracji domyślnej jest przełączenie sygnału zegarowego 12 MHz na wyjście CBUS0 układu FT232RL.

Następnie należy zaprogramować mikrokontroler zawartością pliku `gpi_conv_v4.hex`. Ja użyłem do tego celu programu ISP Programmer [4]. Należy pamiętać o prawidłowym ustawieniu fusebits mikrokontrolera zgodnie z **rysunkiem 4**. Po wykonaniu powyższych czynności konwerter jest gotowy do pracy.

Jak wcześniej wspomniano konwerter GPIO-USB jest widziany w systemie jako wirtualny port COM. Do komunikacji z konwerterem można wykorzystać dowolny program terminalowy taki jak Hyper Terminal lub Putty [5]. Prędkość transmisji to 115200 bps (8 bitów danych, 1 bit stopu, bez kontroli parzystości). W przypadku putty, w opcjach konfiguracji klawiatury należy ustawić obsługę klawisza Backspace jako Control-H oraz obsługę sekwencji Esc[n~.

Komendy konwertera

Konwerter obsługuje komendy o maksymalnej długości 63 znaków. Ich listę zamieszczono poniżej.

Dxyz... Wysłanie łańcucha znaków „xyz” o minimalnej długości 1 znak w trybie danych. Funkcja ta jest zablokowana gdy konwerter jest w stanie Listen – zwraca komunikat ERROR. Ponieważ maksymalna ilość danych możliwa do wysłania to 62 bajty jest też drugi wariant tej funkcji: „M”. Składnia jest identyczna ale różnica jest taka, że komenda D wysyłając ostatni bajt ustawia linię EOI oznaczającą koniec komunikatu a komenda M tego nie robi wobec czego możliwe jest wysłanie danych do urządzenia o długości większej niż 62 bajty – w tym wypadku ostatni blok danych musi zostać wysłany komendą „D” w celu ustawienia linii EOI.

Przykład:

```
<GPIB> D*IDN?
```

Odpowiednikiem powyższej komendy jest:

```
<GPIB> M*1
```

```
<GPIB> DDN?
```

Cxyz... Wysłanie łańcucha znaków „xyz” o minimalnej długości 1 znak w trybie

komend. Komenda używana w celu zaadresowania urządzeń, które mają zostać przełączone w tryb nadawania (*Talk*) oraz w tryb odbioru (*Listen*). Urządzenia mają przydzielone unikalne, 5-bitowe adresy z zakresu 0...30; adres 31 jest używany do usunięcia adresowania urządzenia, które jest w stanie *Talk* lub *Listen*. Przykładowo aby urządzenie o adresie 14 zostało przedstawione w tryb *Talk* należy wysłać znak ASCII o kodzie będącej sumą adresu plus 64 (linia DIO7 ustawiona, $14+64=78$, jest to kod znaku „N”). Zaadresowania urządzenia w tryb *Listen* można dokonać ustawiając stan wysoki na linii DIO6 podczas wysyłania adresu co jest równoważne wysłaniu znaku ASCII o kodzie będącym adresem urządzenia powiększonym o 32. Zatem dla urządzenia o adresie 14 będzie to znak o kodzie 46 czyli „.”.

Txyz... wysłanie komend (xx=0C) lub danych (xx=0D) w postaci szesnastkowej. Minimalna ilość wysłanych danych to jeden bajt. Gdy konwerter jest w stanie *Listen* wysłanie danych jest zablokowane a komenda zwraca komunikat ERROR. Podobnie jak w przypadku pary komend D/M także ta komenda umożliwia wysyłanie bloków dłuższych niż 30 bajtów (wartość ta wynika z maksymalnej długości komendy pomniejszonej o bajt typu komendy i dwa bajty typu komunikatu, trzeba także uwzględnić, że każdy bajt jest kodowany dwoma znakami ASCII). Można tego dokonać umieszczając średnik na końcu komendy – w takim wypadku nie następuje sygnalizacja końca komunikatu linią EOI.

Przykład:

```
<GPIB> d*IDN?
```

Odpowiednikiem powyższej komendy będzie:

```
<GPIB> t0d2a49444e3f
```

Lub podzielone na dwie części:

```
<GPIB> t0d2a49444e;
```

```
OK
```

```
<GPIB> t0d3f
```

```
OK
```

Komendy używane do odbioru danych z urządzenia w trybie *Talk*. Jednorazowo odbierane jest maksymalnie 126 bajtów lub do odebrania sygnału zakończenia komunikatu (zmianą stanu linii EOI).

X. Odebranie i wyświetlenie łańcucha znaków w formacie ASCII. Komenda używana najczęściej przy komunikacji z konwerterem przy pomocy programu terminalowego.

Y. Odbiór binarny z urządzenia. W odpowiedzi na tą komendę pierwszy wysłany przez konwerter bajt to rozmiar odebranych danych a kolejne to odebrane dane z urządzenia. Tryb przeznaczony do komunikacji przy użyciu własnego oprogramowania zamiast programu terminalowego.

Z. Funkcja działa analogicznie do komendy Y tyle, że każdy bajt jest wysyłany jako liczba w postaci szesnastkowej. Komenda użyteczna w przypadku gdy odebrane

dane z urządzenia zawierają znaki specjalne (o kodach od 0 do 31) np. status urządzenia.

P. Przejście w tryb bezpośredni. Po wykonaniu tej komendy wszystkie odebrane dane po GPIB są bezpośrednio wysyłane przez terminal. Nie obowiązuje limit rozmiaru danych jak w przypadku komend X, Y lub Z. Wyjście z tego trybu jest możliwe poprzez wysłanie znaku o kodzie 27 dziesiętnie (kod klawisza ESC). Praca w tym trybie jest sygnalizowana pulsowaniem diody LED z częstotliwością 2 Hz.

Axx. Ustawienie lub odczytanie adresu konwertera. Domyślnie jest to wartość 21. Wywołanie bez parametru powoduje zwrócenie adresu konwertera. Ustawienie adresu poprzez wywołanie komendy „A” z 2-cyfrowym parametrem w formacie dziesiętnej.

Przykład użycia:

```
<GPIB> A
```

```
21
```

```
<GPIB> A07
```

```
OK
```

```
<GPIB> A
```

```
07
```

?. Wyświetlenie pomocy (lista dostępnych komend).

H. Wyświetlenie historii ostatnio wydanych komend.

Ex. Włączenie (E1), wyłączenie (E0) lub pobranie statusu (E, wywołanie bez parametru) lokalnego echa wysyłanych znaków. Domyślnie echo jest włączone a konwerter gotowy do przyjęcia nowej komendy wyświetla znak zachęty <GPIB>. Tryb wygodny w przypadku komunikacji przy użyciu programu terminalowego ale korzystający z konwertera pod własną aplikacją nie jest potrzebny. Inną metodą wyłączenia lokalnego echa jest założenie zwory między linie SCK (PB7) a MISO (PB6).

S. Wyświetlenie stanu, kolejno, linii REN, SRQ oraz stanu konwertera; znak „1” oznacza, że linia REN/SRQ jest w stanie „prawda”. „1” dla stanu konwertera oznacza, że został on zaadresowany w tryb *Listen*. Dodatkowo, stan *Listen* konwertera jest sygnalizowany miganie diody LED z częstotliwością 5 Hz.

R. Ustawienie linii REN w stan true, wymusza na zaadresowanych urządzeniach przejście w tryb sterowania zdalnego.

Wykaz elementów

Rezystory:

R1: 560 Ω (SMD 0805)

Kondensatory:

C1: 100 nF (SMD 0805)

C2: 4,7 μF/10 V (tantalowy „B”)

Półprzewodniki:

IC1: ATmega32 (TQFP44)

IC2: FT232RL

LED1: dioda LED (SMD 0805)

Inne:

X1: gniazdo mini-usb (TME: ESB34101000Z, Farnell: 2112374)

Wtyk 24-pin Centronics (TME: CENTR-224, Farnell: 2112386)



L. Ustawienie linii REN w stan „fałsz”, wymusza na zaadresowanych urządzeniach przejście w tryb sterowania lokalnego.

I. Wygenerowanie impulsu na linii IFC w celu usunięcia adresacji w urządzeniach połączonych do magistrali.

Wydanie komend D, M, C, T, R, L, I jest potwierdzane przez konwerter odesłaniem komunikatu OK. Dodatkowo w przypadku komend D, M, C oraz T brak reakcji przez określony czas, ze strony urządzeń, na chęć wysłania przez konwerter danych jest sygnalizowane komunikatem TIMEOUT.

Przykład komunikacji z multimetrem HP3478A

W poniższym przykładzie konwerter jest połączony do multimetru HP 3478A. Multimetr ma ustawiony adres 23 dziesiętnie co odpowiada znakowi ‚W’ w trybie Talk i ‚7’ w trybie Lister. Adres konwertera jest domyślny (21, Talk ‚U’, Listen ‚5’).

Ustawienie trybu remote:

< GPIB > R

OK

Multimetr zostaje ustawiony na odbiór (Listen) a konwerter jako nadajnik (Talk):

< GPIB > C_?7U

OK

Należy skonfigurować multimetr do pracy jako woltomierz DC (F1), zakres 30 V (R1), pomiary wyzwalane wewnętrznie (T1):

< GPIB > DF1R1T1

OK

Ustawienie woltomierza jako nadajnik (Talk) i konwertera jako odbiornik (Listen):

< GPIB > C_?W5

OK

Odczytanie danych z woltomierza:

< GPIB > X

+04.9039E+0

Instrukcje odczytu danych X należy wywoływać sekwencyjnie w celu odbierania kolejnych pomiarów. Można również skorzystać z trybu bezpośredniego:

< GPIB > p

+04.9041E+0

+04.9041E+0

+04.9040E+0

(...)

Przykład komunikacji z oscyloskopem Tektronix TDS3034

Adresy urządzeń są identyczne jak w poprzednim przykładzie.

< GPIB > r

OK

< GPIB > c_?U7

OK

Zapytanie o nazwę urządzenia (standardowe polecenie dostępne większości urządzeń):

< GPIB > d*IDN?

OK

Ustawienie oscyloskopu w tryb Talk a konwertera w tryb Listen:

< GPIB > c_?W5

OK

Odczytanie danych z urządzenia:

< GPIB > x

TEKTRONIX,TDS 3034.0,CF:91.1CT FV:v3.41 TDS3GM:v1.00 TDS3FFT:v1.00 TDS3TRG:v1.00

< GPIB > x

TIMEOUT

Jak widać powyżej kolejna próba odczytu z oscyloskopu kończy się komunikatem TIMEOUT, ponieważ urządzenie nie ma więcej danych do wysłania. Wywołanie funkcji Autoset poprzedzone zaadresowaniem oscyloskopu w trybie Listen:

< GPIB > c_?U7

OK

< GPIB > dAUTOSSET EXECUTE

OK

Odczytanie ustawień podstawy czasu:

< GPIB > dHOR?

OK

< GPIB > c_?W5

OK

< GPIB > x

HIGH;1.0E1;4.0E-4;1;0.0E0

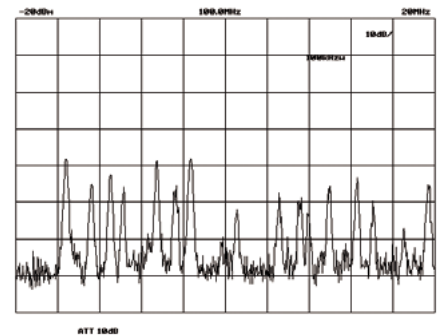
< GPIB >

Emulowanie plotera HP7470A

Głównym celem zaprojektowania tego konwertera była możliwość pobierania zrzutów ekranów z oscyloskopu oraz analizatora widma. Większość starszej generacji przyrządów pomiarowych obsługiwała ploter typu HP7470a. Urządzenia używające magistrali GPIB mają zwykle opcję Talk Only dzięki czemu nie trzeba było stosować kontrolera magistrali a ploter lub drukarka nie musiały być adresowane jako urządzenie w stanie Listen.

Przejścia konwertera w tryb bezpośredni można dokonać na dwa sposoby: wydając komendę „P” lub zakładając zworkę między piny MOSI (PB5) i MISO (PB6) złącza X6 używanego do podłączenia programatora ISP oraz resetując konwerter (stan zwory jest sprawdzany jest tylko w momencie startu konwertera). Druga metoda jest przydatna w przypadku korzystania z komercyjnych emulatorów plotera/drukarki takich jak PrintCapture [6]. Opisana poniżej aplikacja GPIBreceive podczas startu wysyła do konwertera komendę przejścia w tryb bezpośredni.

Konwerter umożliwia odebranie surowych danych, które w przypadku wybrania plotera HP7470A są komendami języka HPGL. Należy zatem odebrane dane w odpowiedni sposób zinterpretować. Z pomocą przychodzi darmowy emulator 7470a.exe autorstwa amerykańskiego krótkofalowca KE5FX [7]. Program ten będący elementem pakietu KE5FX GPIB Toolkit interpretuje komendy HPGL i wyświetla odebrany obraz. Umożliwia on także zapisanie obrazu do pliku graficznego typu GIF lub BMP. Przykład zrzutu



Rysunek 5. Zrzut ekranu z analizatora widma Anritsu

ekranu z analizatora widma Anritsu przekonwertowanego do pliku gif przy użyciu 7470a.exe przedstawiono na rysunku 5.

Ponieważ odbieranie danych przy użyciu programu terminalowego, zapisywanie ich do pliku tekstowego i kolejno wywołanie emulatora 7470a.exe jest dosyć kłopotliwe, została przygotowana prosta aplikacja GPIBreceive.exe. Aplikacja wymaga zainstalowania .Net Framework w wersji 4 – do pobrania ze strony Microsoftu [8]. Pracuje poprawnie z systemami Windows XP oraz 7.

GPIBreceive umożliwia zdefiniowanie czasu po którego upłynięciu, od momentu odebrania ostatniej danej z konwertera, dane zostaną zapisane do pliku. Możliwe jest także zaznaczenie opcji automatycznego uruchomienia emulatora 7470a.exe (należy wcześniej wskazać do niego ścieżkę przy użyciu przycisku Select app) w celu interpretacji odebranych danych.

Niektóre starsze oscyloskopy firmy HP nie obsługiwały plotera HP7470A a jedyną opcją był wydruk na drukarce zgodnej z HP2225A.

Podsumowanie

Przedstawiony układ pozwala na zwiększenie funkcjonalności starszej generacji przyrządów. Autor z powodzeniem używał go z różnego typu aparaturą pomiarową taką jak oscyloskopy HP oraz Tektronix, multimetrem HP3478A oraz analizatorem widma Advantest R4131. Dużą zaletą układu jest niski koszt oraz łatwość montażu.

Jacek Greniger
jacek.greniger@gmail.com

Bibliografia:

- [1] <http://www.bitsavers.org/pdf/hp/hpib/TutorialDescrOfHPIB.pdf>
- [2] <http://www.ftdichip.com/Drivers/VCP.htm>
- [3] http://www.ftdichip.com/Support/Utilities/FT_Prog%20v2.6.8.zip
- [4] <http://dybkowski.net/content/pl/node/16>
- [5] <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- [6] <http://www.printcapture.com/>
- [7] <http://www.ke5fx.com/gpib/readme.htm>
- [8] <http://www.microsoft.com/pl-pl/download/details.aspx?id=17851>