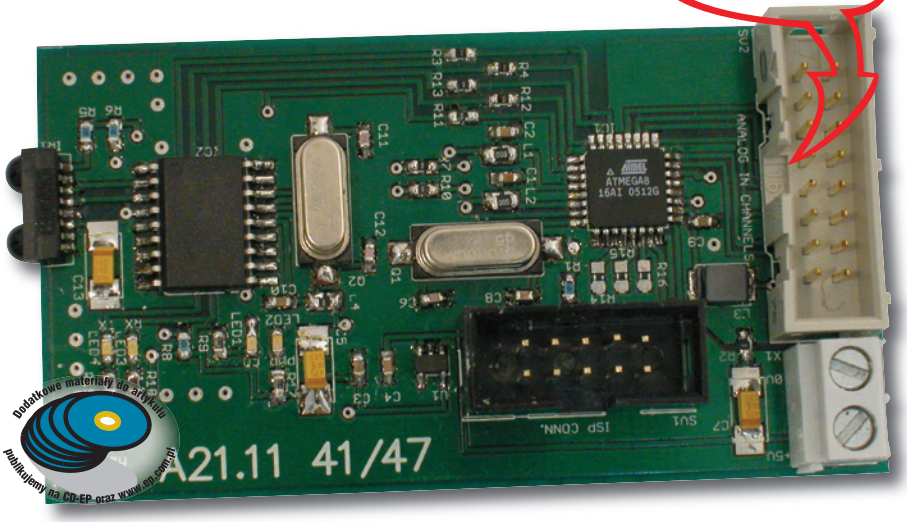
Bezprzewodowy DAQ z interfejsem IrDA

AVT–954

PROJEKT Z OKŁADKI



Urządzenie przedstawione w artykule miało w zamyśle prezentować możliwości sprzętowych kontrolerów IrDA (sukesywnie opisywanych w EP) – jest to ośmiokanałowy rejestrator analogowy z transmisją danych w podczerwieni. Podczas projektowania urządzenia wpadłem na pewne interesujące zastosowanie tego urządzenia: wyobraźmy sobie „maszynę" pracującą w ciężkich warunkach przemysłowych, w dużym zapyleniu bądź wilgotności. Wyobraźmy sobie, że mamy 40 takich maszyn, a z każdej raz dziennie chcielibyśmy odczytywać wskazania dotyczące stanu jej pracy. Możemy wziąć zeszyt i podchodzić do maszyn po kolei zapisując interesujące nas informacje, możemy też wziąć palmtopa (lub laptopa), podchodzić do maszyn i jednym kliknięciem ściągać dane. Co zyskujemy? Potencjalnie niższą cenę maszyn bez wyświetlacza, ale za to z interfejsem podczerwieni oraz bezpieczeństwo – dane mogą zostać odczytane tylko przez nas, którzy dysponujemy odpowiednim oprogramowaniem (nic nie stoi też na przeszkodzie zaszyfrowania przesyłanych danych).

PODSTAWOWE PARAMETRY

- Płytka o wymiarach 73x40 mm
- Zasilanie 6...16 VDC
- Liczba kanałów analogowych 8 (wejścia nie są buforowane)
- Zakres pomiaru napięć w kanałach analogowych 0...5 V
- Rozdzielczość pomiaru 10 bitów
- Próbkowanie tylko „na życzenie" – po wysłaniu komendy (brak własnego bufora i trybu pracy autonomicznej)
- Transmisja danych przez port IrDA
- Prędkość transmisji 9600 b/s

Opis funkcjonalny

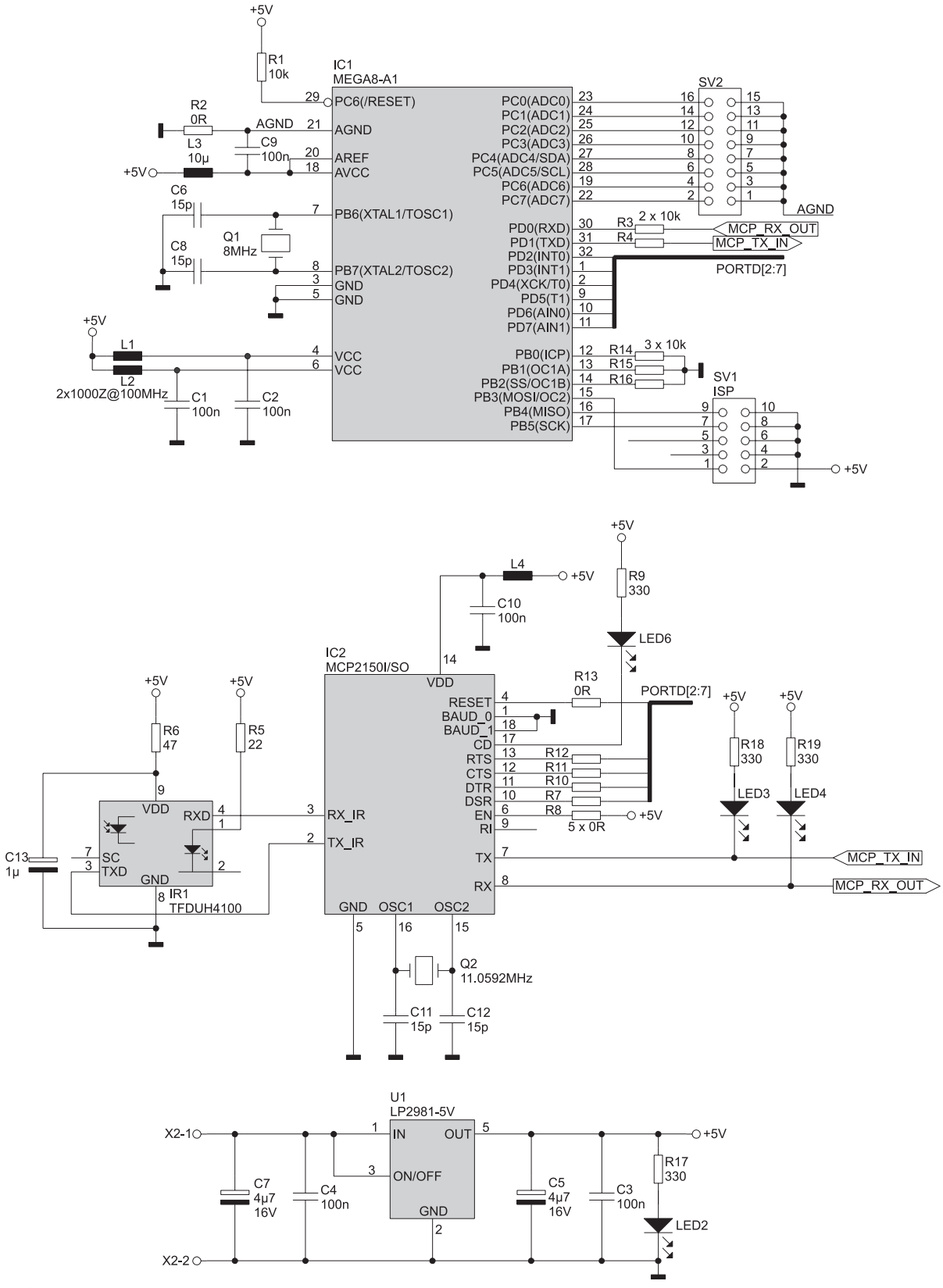
Urządzenie zapewnia pomiar napięcia z 8 niezależnych kanałów w zakresie 0…5 V każdy. Używa 10–bitowego przetwornika (z multipleksowanymi wejściami) wbudowanego w mikrokontroler ATmega 8, który steruje też pracą całego rejestratora. Tytułowe określenie DAQ (od *Data AcQuisition* – akwizycja danych) zostało użyte trochę na wyrost. Po takim sformułowaniu nazwy sprzętu należałoby się spodziewać przetwornika A/C z lokalnym buforem zdolnym próbkować sygnał analogowy z żądaną częstotliwością przez określony czas. Po wydaniu odpowiedniej komendy zgromadzone próbki zostałyby przesłane do komputera. Opisywane urządzenie jest uproszczoną wersją „prawdziwego" DAQ: mianowicie po wydaniu komendy zawierającej numer kanału, rejestrator odpowiada zmierzoną w nim wartością napięcia. Zatem rejestrator w prezentowanej wersji nie posiada żadnych funkcji „autonomicznych", co jednak nie oznacza, że nie można go o te funkcje rozbudować.

Budowa

Konstrukcyjnie rejestrator podzielono na 3 bloki: blok mikrokontrolera, układ MCP2150 oraz blok zasilania wykonany na stabilizatorze LDO o napięciu wyjściowym 5 V (LP2981). Mikrokontroler ATmega 8 pracuje w konfiguracji z zewnętrznym kwarcem 8 MHz. Ponieważ istotą działania urządzenia jest pomiar napięcia, to masy: cyfrowa (GND) i analogowa (AGND) zostały na płytce rozdzielone i łączą się ze sobą tylko w jednym punkcie. Jest nim rezystor „zerowy" R2.

Dla zredukowania szumów pomiaru (a więc również zwiększenia dokładności) zastosowano filtr L3, C9 dostarczający napięcia do analogowej części mikrokontrolera (piny AREF i VCC). Pewną niedoskonałością prezentowanego rozwiązania jest to, że wejścia analogowe mikrokontrolera nie posiadają układów buforujących. Dodanie wtórników napięciowych, zbudowanych w oparciu o wzmacniacze operacyjne na wejściach zwiększyłoby impedancję wejściową kanałów analogowych, a także stanowiłoby zabezpieczenie przed podaniem zbyt wysokiego napięcia na wejścia. Wszystkie kanały analogowe doprowadzono do złącz 2x8 (SV2) w ten sposób, że każda linia analogowa oddzielona jest od następnej stykiem z doprowadzoną masą AGND. Na płytce znajduje się również złącze programowania ISP. Jego dość duży rozmiar (2x5) to pewna zaszłość wynikająca z przyjętego standardu dla programatorów zgodnych z STK200/300 (taki posiadam, stąd wybór konektora).

Jako interfejs podczerwieni zastosowano sprzętowy kontroler IrDA MCP2150 zapewniający obsługę sto-



Rys. 1. Schemat elektryczny prezentowanego urządzenia

su IrDA, począwszy od warstwy sygnałów fizycznych (PHY) poprzez warstwę kontroli dostępu do łącza i jego zarządzania (IrLAP i IrLMP) aż po warstwę transportową (TinyTP) i najbardziej interesujący nas (a zarazem najbardziej użyteczny) protokół IrCOMM. Układ współpracuje z optycznym transceiverem IrDA TFDU4100. Do poprawnej pracy układ MCP2150 wymaga zewnętrznego kwarcu 11,059 MHz. Sygnał zegarowy uzyskany przy jego pomocy służy m.in. do wytworzenia podstawy czasu dla zintegrowanego układu UART. Ta właśnie częstotliwość pozwala na pracę UART–u ze standardowymi prędkościami trans-

Rys. 2. Komunikat wyświetlany w Windows XP po wykryciu IrDA DAQ

misji. Wynika stąd, że nie mamy zbyt wielu możliwości manewru przy wyborze kwarcu. Wszystkie linie kontroli przepływu, jak i linia zerująca (RESET) zostały podłączone do mikrokontrolera. Linie BAUD_0 i BAUD_1 podłączono do masy w celu wymuszenia prędkości transmisji UARTU 9600 b/s. Do linii CD (*Carrier Detect*) podłączono diodę LED wraz rezystorem podciągającym. Stanowi ona źródło informacji o otwarciu bądź zamknięciu połączenia Ir. Linia EN (*enable*) została na stałe podłączona do plusa zasilania (poprzez rezystor 10 kΩ). Linie RX i TX podłączone są do odpowiednich linii UART–u ATmega 8 i są wyposażone w diody LED sygnalizujące przepływ danych.

Jedynym sposobem komunikacji z urządzeniem jest zatem połączenie w podczerwieni. MCP2150 zapewnia, że dla ATmega jego UART jest niemal przeźroczysty (tak jakbyśmy podłączyli go do portu COM komputera). „Niemal", bo musimy zapewnić obsługę lokalnie generowanych (przez MCP2150) sygnałów przepływu danych. Omawiając normalny przypadek, a więc odbieranie komend i nadawanie odpowiedzi, interesujące są jedynie dwa sygnały: CTS (*Clear to Send*) – wyjście, informujące mikrokontroler o tym, że MCP2150 jest gotowy na przyjęcie danych. Jest to o tyle ważne, że, pomimo iż układ posiada osobne 64–bajtowe bufory na dane z linku podczerwieni, jak i na dane

Tab. 1. Opis wyprowadzeń złącza SV2

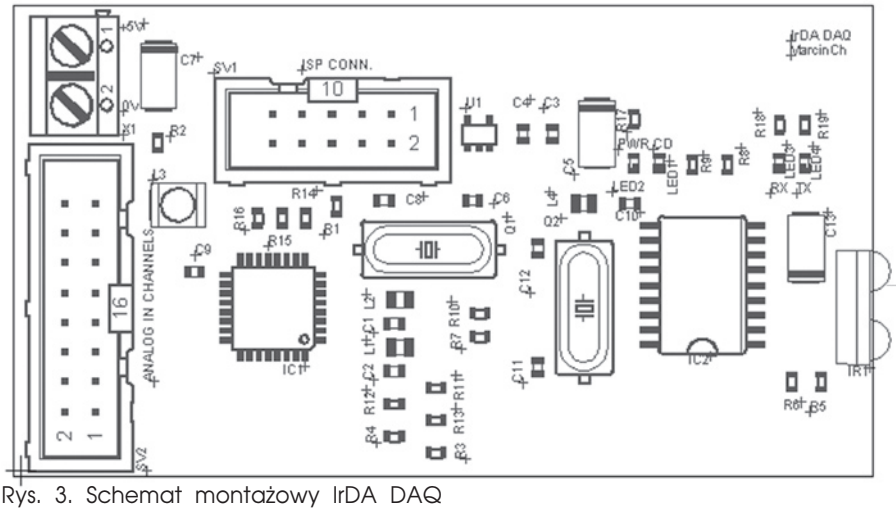
1.	AGND
2.	AIN 7
3.	AGND
4.	AIN 6
5.	AGND
6.	AIN 5
7.	AGND
8.	AIN 4
9.	AGND
10.	AIN 3
11.	AGND
12.	AIN 2
13.	AGND
14.	AIN 1
15.	AGND
16.	AIN 0

Tab. 2. Opis wyprowadzeń złącza ISP – SV1

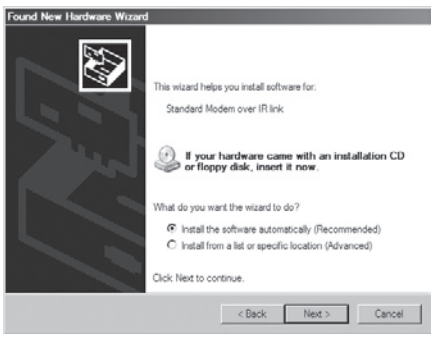
1.	MOSI
2.	+5V
3.	NC
4.	GND
5.	RESET
6.	GND
7.	SCK
8.	GND
9.	MISO
10.	GND

odbierane od strony UART–u, to jednocześnie obsługuje tylko jeden z tych buforów. Zatem w przypadku, gdy *Host* (PC) rozpoczyna transmisję Ir, mikrokontroler nie powinien nadawać żadnych znaków do MCP2150. Zakaz ten będzie sygnalizowany stanem wysokim na pinie CTS. Jeśli zignorujemy ten sygnał, MCP2150 odbierze jedynie dwa znaki (taki jest rozmiar wewnętrznej kolejki FIFO UART–u). Drugi sygnał – RTS (*Request to Send*) informuje z kolei układ MCP2150 o tym, czy kontroler zdolny jest odebrać od niego dane (stan niski). Jeśli odbiór danych będzie odbywał się w przerwaniu (tak właśnie jest w opisywanym urządzeniu), nie powinniśmy mieć żadnych problemów, jeśli na stałe ustawimy na niej stan niski.

Linia RESET MCP2150 jest nadzorowana przez mikrokontroler, co umożliwia wprowadzanie układu w tryb programowania, w którym



Rys. 3. Schemat montażowy IrDA DAQ



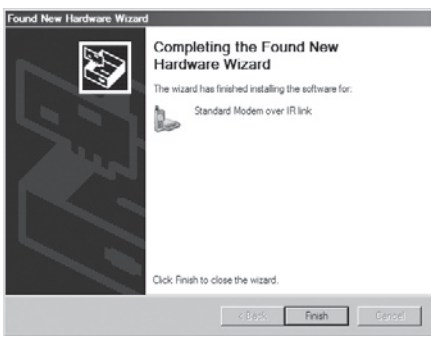
Rys. 4. Kreator dodawania nowego sprzętu po wykryciu MCP2150

możemy przekazać (poprzez transmisję do UART) ciąg 19 znaków stanowiących ID urządzenia. String ten jest wyświetlany (w WinXP) na etapie wykrywania (**rys. 2**).

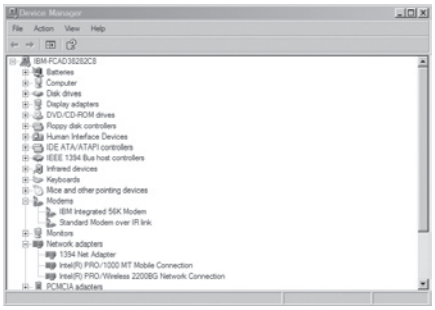
Device ID (tak w dokumentacji nazywany jest omawiany ciąg znaków) jest zapisany w pamięci EEPROM i może zawierać znaki ASCII z przedziału od 20 h do 7 Ah. Domyślnie *Device ID* ma wartość „*Generic IrDA*". Aby zapisać nową wartość *Device ID* do MCP2150, mikrokontroler musi wykonać następującą sekwencję działań: ustawić na linii RESET stan niski, linie DTR i RTS wprowadzić odpowiednio w stan wysoki i niski, po czym uwolnić układ ze stanu zerowania, następnie wysłać ciąg znaków poprzedzony bajtem zawierającym długość całego ID. Należy być tutaj niezwykle ostrożnym: jeśli podany ciąg znaków będzie zawierał kod ASCII spoza podanego zakresu, MCP2150 nie będzie w stanie utworzyć linku podczerwieni z *Hostem*, a tym samym stracimy możliwość komunikacji z IrDA DAQ!

Montaż

Schemat montażowy urządzenia przedstawiono na **rys. 3**. Jedyną trudność mogą sprawić elementy SMD w rozmiarze 0603. Układ zasilamy doprowadzając do złącza X1



Rys. 5. Kreator dodawania sprzętu po pomyślnej instalacji urządzenia



Rys. 6. Manager urządzeń po zainstalowaniu MCP2150

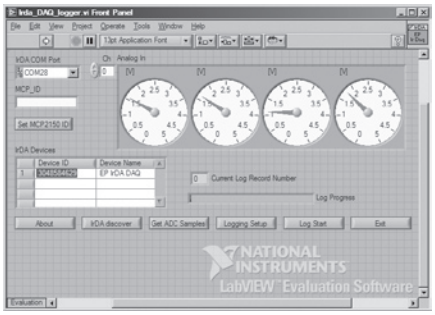
około 6 V (max. 16 V). Pobór prądu nie przekracza kilkudziesięciu mA. Sygnały analogowe najwygodniej doprowadzić płaskim kablem 16–żyłowym. Opis złącza kanałów analogowych pokazano w **tab. 1**, a rozmieszczenie wyprowadzeń złącza ISP w **tab. 2**.

Uruchomienie

Po dołączeniu napięcia zasilającego w pierwszej kolejności powinniśmy zobaczyć świecącą się diodę LED2 (PWR). Będzie to świadczyło o prawidłowej pracy bloku zasilania. W tym momencie możemy ustawić płytkę w obszar widoczny przez transceiver IrDA w komputerze. Jeśli część układu z MCP2150 działa poprawnie, po chwili powinniśmy dostrzec komunikat widoczny na rys. 2. Następnie system uruchomi kreatora dodawania nowego sprzętu. Jak widać na **rys. 4**, układ MCP2150 identyfikuje się w systemie jako modem, ma to stanowić informację dla systemu, że jest urządzeniem komunikującym się szeregowo z ograniczonymi zasobami pamięci. Naturalnie MCP2150 nie

Rys. 7. Monitor IrDA z informacją o urządzeniu w zasięgu

Rys. 8. Aplikacja instalująca sterowniki wirtualnego portu szeregowego

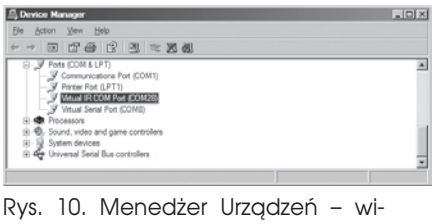


Rys. 9. Główne okno aplikacji Irda_DAQ_logger

jest „prawdziwym" modemem i np. nie odpowiada na komendy AT.

Na ekranie z rys. 4 klikamy *Next* pozwalając na automatyczną instalację. Po chwili kreator pokaże okno widoczne na **rys. 5** – klikamy *Finish*. Przechodząc teraz do windowsowego *Menedżera Urządzeń* (**rys. 6**) powinniśmy dostrzec w sekcji modemy: *Standard Modem over IR link*, a po najechaniu wskaźnikiem myszy na ikonę monitora portu podczerwieni ukaże się informacja o obecności IrDA DAQ w zasięgu (**rys. 7**).

Kolejnym krokiem będzie zainstalowanie sterowników VCP (*Virtual COM Port*). Darmowe sterowniki (licencja *GNU General Public License*) możemy pobrać np. ze strony *http://www.ircomm2k.de/download.html* w sekcji *Download*. Interesujący nas plik to *IrCOMM2k–1.2.1–eng.zip* Obecnie jest już dostępna wersja *2.0 beta3* tego sterownika, jednak nie udało mi się nawiązać poprawnej komunikacji z urządzeniem po upgradzie sterownika. Być może należy poczekać na wersję końcową. Po rozpakowaniu uruchamiamy *Setup.exe*. Program poprosi nas o wskazanie numeru portu, który chcemy przyporządkować sterownikowi oraz wyświetli komunikat o tym, iż sterownik nie jest podpisany cyfrowo, mimo to kontynuujemy jego instalację. Następnie program zarejestruje i uruchomi usługę w systemie o nazwie *ircomm2k.exe*. Takie postępowanie jest konieczne ze względu na



Rys. 10. Menedżer Urządzeń – widoczny numer wirtualnego portu *COM port IrDA*

sposób obsługi IrDA przez system Windows. Szczegółowe informacje na temat architektury sterownika znajdziemy pod adresem: *http://www.ircomm2k.de/techdoc.html*.

W tym momencie jesteśmy gotowi rozpocząć kolejną część instalacji, a więc uruchamianie właściwej aplikacji zbierającej dane z IrDA DAQ. Aplikacja została napisana w ewaluacyjnej, 30–dniowej wersji LabVIEW 8 (do pobrania z *ftp://ftp.ni.com/evaluation/labview/pc/labview_80.exe*), która nie pozwala na budowanie projektów do plików wykonywalnych *.exe*. Jednak po zainstalowaniu LabVIEW możemy uruchamiać aplikację bez ograniczeń. Główny plik *.vi* aplikacji nosi nazwę *Irda_DAQ_logger.vi*.

Po upewnieniu się, że nasze urządzanie jest w zasięgu (rys. 7), ustawiamy w polu *IrDA COM Port* numer portu, który został przyporządkowany przez driver VCP (numer potu możemy sprawdzić w Menedżerze Urządzeń – **rys. 10**).

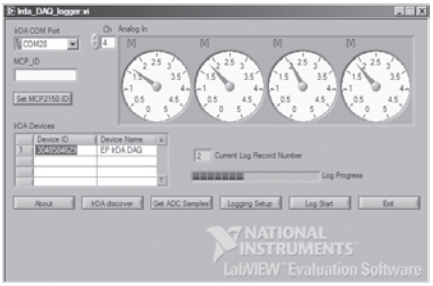
Następnie klikamy białą strzałkę jednokrotnego uruchomienia aplikacji. Po prawidłowej inicjalizacji portu program przechodzi w stan *idle*, my za to powinniśmy zaobserwować włączenie się diody LED1 – CD (*Carrier Detect*). Pierwszą funkcją programu jest wykrywanie urządzeń IrDA (przycisk *IrDA discover*). Po jego naciśnięciu na liście *IrDA Devices* pojawią się wszystkie urządzenia znajdujące się obecnie w zasięgu. W celach testowych możemy np. ustawić telefon komórkowy z włączonym portem IrDA w obszar widoczności. Po kliknięciu *IrDA Discover* powinniśmy zobaczyć na liście identyfikujący *go string* i ID (w tym przypadku jest ono numerem, natomiast *Device ID* programowane wewnątrz MCP2150 zgodnie z dokumentacją producenta występuje tutaj jako *Device Name* – jest to pewna nieścisłość, ale nie ma ona dla nas żadnego znaczenia).

Następnie możemy odczytać wartości napięcia z kanałów analo-

Rys. 11. Okno *Logging_Setup.vi*

List. 1. Przykładowy plik z wartościami zebranymi przez IrDA DAQ

Timestamp	CH 1 [V]	CH 2 [V]	CH 3 [V]	CH 4 [V]	CH 5 [V]	CH 6 [V]	CH 7 [V]	CH 8 [V]
23:51:02	1.851	1.836	1.357	1.934	1.934	1.909	2.090	1.948
23:51:07	1.836	1.719	1.421	1.919	1.875	1.958	2.090	1.875
23:51:12	1.812	1.855	1.367	1.909	1.943	1.914	2.065	1.968
23:51:17	1.909	1.797	1.362	1.948	1.909	1.909	2.109	1.934
23:51:22	1.904	1.719	1.401	1.943	1.875	1.948	2.100	1.875
23:52:23	1.885	1.719	1.421	1.943	1.875	1.953	2.104	1.875



Rys. 12. Główne okno aplikacji podczas logowania

Rys. 13. Komunikat informujący o przerwaniu połączenia IrDA po zmianie Device ID

gowych IrDA DAQ. Podłączamy zatem źródło napięcia, bądź zwieramy któryś z kanałów (w celu uzyskania wyniku 0 V) i klikamy *Get ADC Samples*. Po chwili powinniśmy ujrzeć aktualne wartości napięć w kanałach widoczne na analogowych miernikach w tablicy *Analog In*. Zmieniając wartość pola *Ch* możemy wyświetlić wszystkie 8 kanałów (wartość pola *Ch* informuje, który kanał znajduje się tuż po jego prawej stronie). W celu zarejestrowania wartości pomiarowych do pliku klikamy przycisk *Logging Setup*. Ukaże nam się okno ustawień logowania (**rys. 11**).

Przyciskiem po prawej stronie *Log Path* otwieramy okno wyboru katalogu, w jakim przechowywane będą pliki z logami. *Log Period* oznacza wartość w sekundach pomiędzy kolejnymi odpytywaniami IrDA DAQ o nowe wartości. *Record Number* – to liczba odpytań w trakcie jednego procesu logowania. W celu zachowania zmian klikamy *Save&Close*. Po powrocie do głównego okna rozpoczynamy proces logowania klikając na *Log Start*. Nowe wartości będą zapisywane kolejno do pliku tekstowego on nazwie: *IrDA_EP_DAQ_data_godzina.txt*. i formacie przedstawionym na **list. 1**.

Za pomocą aplikacji *Irda_DAQ_logger.vi* możemy też zmienić ID urządzenia. Służy do tego pole *MCP_ID* i przycisk *Set MCP2150 ID*. Po zmianie ID połączenie zostanie przerwane (**rys. 13**). Ponowne jego przywrócenie nastąpi automatycznie po chwili. Spowodowane jest to tym, że wprowadzenie MCP2150 w tryb programowania powoduje też jego wyzerowanie (tak samo podczas wprowadzania go w tryb normalny).

Podsumowanie

Punktem krytycznym całego projektu było zastosowanie interfejsu programowego do nawiązania komunikacji z MCP2150 z poziomu Windows XP. W początkach implementacji IrDA w systemach Windows (95, 98, Me) protokół IrCOMM obsługiwany był przez sterowniki zintegrowane z systemem, które automatycznie tworzyły wirtualny port szeregowy. Jednak począwszy od Windows 2000 obsługa IrCOMM została przeniesiona z API odpowiedzialnego za wirtualne porty COM (które przestały istnieć jako część systemu) do WinSock API. Zatem komunikacja z MCP2150 w systemach takich jak Windows XP czy 2000 odbywa się właśnie, poprzez *sockety*. Jeśli odpowiada nam taka form komunikacji (musimy wtedy napisać zupełnie nową aplikację *loggera*) nie musimy instalować drivera VCP opisanego w artykule (usługa *ircomm2k.exe* uruchamiana po jego instalacji spełnia rolę mostu pomiędzy WinSock API i Serial API). Z drugiej jednak strony używając wirtualnych portów COM możemy do obsługi urządzeń z interfejsem IrDA używać standardowego HyperTerminala. Takie rozwiązanie jest najwygodniejsze i najszybsze w implementacji jak również najczęściej opisywane na łamach EP. Wybór drogi – jak zwykle – zależy od Czytelnika.

Marcin Chruściel, EP
marcin.chrusciel@ep.com.pl

WYKAZ ELEMENTÓW

Rezystory (0603)
R2, R7, R10...R13: 0 Ω
R1, R3, R4, R8, R14...R16: 10 kΩ
R9, R17...R19: 330 Ω
R5: 22 Ω
R6: 47 Ω

Kondensatory
C6, C8, C11, C12: 15 pF
C1...C4, C9, C10: 100 nF
C13: 1 μF/16 V
C5, C7: 4,7 μF/16 V

Półprzewodniki
U1: LP2981 – 5 V
LED1...LED4: diody LED
IC1: ATMmega8-AI
IC2: MCP2150I/SO
IR1: TFDUH4100

Inne
SV1: gniazdo ML10
SV2: gniazdo ML16
X1: gniazdo AK2 5 mm
Q1: rezonator kwarcowy 8 MHz
Q2: rezonator kwarcowy 11,0592 MHz
L3: dławik 10 μH
L1, L2, L4: dławik 1000 μH