

# Moduł I/O sterowany przez USB

## Do czego to służy?

Prezentowany moduł pozwala sterować ośmioma portami mikrokontrolera za pomocą komputera. Każdy port może pracować jako wyjście cyfrowe, wtedy poprzez dowolny moduł wykonawczy może sterować pracą urządzeń, włączając je lub wyłączając. Każdy port może także pełnić funkcję wejścia cyfrowego, wtedy możliwe jest sprawdzenie stanu np. przełączników czy czujników. Cechą wyróżniającą urządzenie na tle podobnych konstrukcji jest prosta budowa

– moduł zbudowany jest z mikrokontrolera i garści elementów biernych.

Podstawowe parametry to:

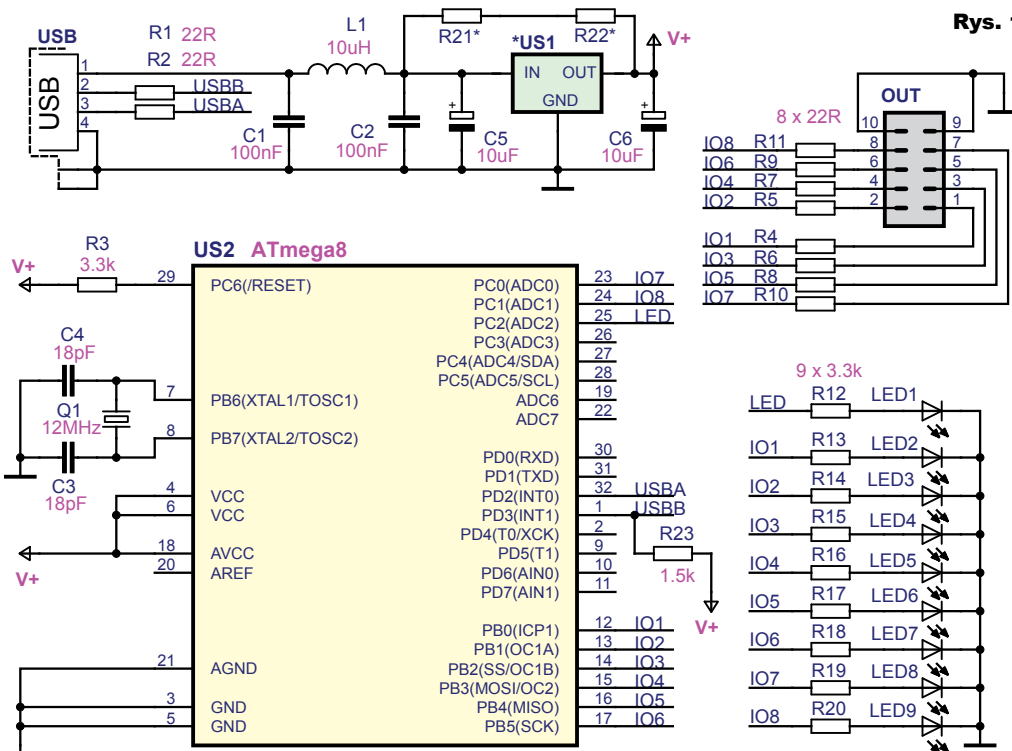
- 8 wprowadzeń, każde z nich może pracować jako wyjście lub wejście cyfrowe,
- sterowanie i zasilanie z portu USB,
- wykorzystuje klasę komunikacyjną CDC,
- emuluje port szeregowy COM, parametry komunikacji 9600, 8, 1, none,
- sterowniki dla systemów Win XP, Win7 (32 bity).
- dedykowana aplikacja dla Windows z kodem źródłowym w .NET C#,

- wymiary 79x41mm, dedykowana obudowa G403

## Jak to działa?

Urządzenie powstało na bazie projektu CDC-IO, którego źródło i dokładny opis dostępny jest na stronie <http://www.recursion.jp/avr/cdc/cdc-io.html>, a jego twórcą jest Osamu Tamura.

Schemat układu przedstawiony jest na **rysunku 1**. Zasilanie z gniazda USB najpierw zostaje doprowadzone do filtra C1, L1, C2, C5, następnie po przej-



ściu przez dwie diody prostownicze połączone szeregowo w miejscach elementów R21\* i R22\* zostaje obniżone do poziomu ok. 3,5V i dodatkowo filtrowane przez C6. Pomimo prostego rozwiązania taki „regulator” napięcia zapewnia w pełni poprawną pracę układu. Jedyną wadą jest to, że wartość tego napięcia może się zmieniać w zależności od obciążenia wyjść urządzenia. Jeśli taka cecha nie jest do zaakceptowania, można zastosować „prawdziwy” stabilizator w miejscu US1 typu LM3940-3.3V. Sygnały z gniazda USB, poprzez rezystory o niewielkiej wartości R1 i R2, trafiają bezpośrednio do mikrokontrolera, sygnał USBB (D-) jest dodatkowo podciągnięty do plusa zasilania – zgodnie ze specyfikacją standardu

Funkcja	Znak komendy	Format komendy	Przykład	Działanie przykładu
Test	@	@	@	Odpowiada „cdc-io”
Odczytanie wartości rejestru	?	adres ?	pinb ?	Zwraca wartość rejestru np.: „3F”
Ustawienie wartości rejestru	=	wartość adres =	0F portb =	Ustawia na porcie B wartość 0x0F
Operacja logiczna AND na rejestrze (wyzerowanie bitów w rejestrze)	&	maska adres &	01 portb &	Zeruje na porcie B wszystkie bity z wyjątkiem najmłodszego
Operacja logiczna OR na rejestrze (ustawianie bitów w rejestrze)		maska adres	01 portb	Ustawienie najmłodszego bitu na porcie B
Operacja Logiczna EX-OR (wyzerowanie bitów w rejestrze)	^	maska adres ^	01 portb ^	Wyzerowanie najmłodszego bitu na porcie B

odpowieź, wysyłając te dwa znaki specjalne. Wszystkie wysyłane wartości (elementy wartości oraz maska z tabeli 1) muszą być zapisane w postaci hex., czyli muszą zawierać się w przedziale 00...FF, również moduł zwraca wartości w takiej właśnie postaci. Jako *rejestr* należy podać nazwę rejestru, który chcemy zapisać lub odczytać. W prezentowanym urządzeniu na złącze wyjściowe składają się dwa porty mikrokontrolera –

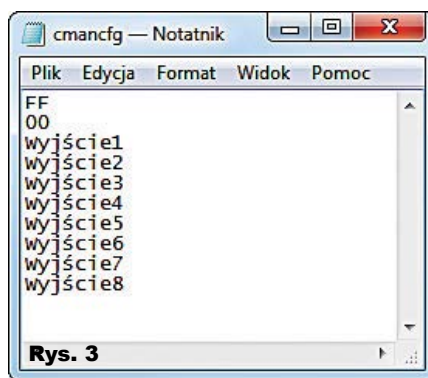
Tabela 1

6 bitów portu „B” (wyprowadzenia 1...6) oraz dwa bity portu „C” (wyprowadzenia 7 i 8). Zatem interesujące nas rejestry to: „portb” i „portc” – rejestry pozwalające ustawić odpowiedni stan na wyjściach modułu; „ddrb” i „ddrc” – rejestry konfigurujące kierunek pracy: wejście (wartość bitu = 0) lub wyjście (wartość bitu = 1); „pinb” i „pinc” – które służą do odczytania stanu panującego na złączu wyjściowym.

Po podłączeniu modułu wszystkie rejestry mają wartość 00, więc wszystkie wyprowadzenia są skonfigurowane jako wejścia. Aby skonfigurować wyprowadzenia jako wyjścia, należy wysłać dwie komendy: „3F ddrb =” oraz „07 ddrc =”. Po takiej operacji wszystkie 8 wyprowadzeń oraz sterowanie diodą LED1 są gotowe do pracy. Można teraz wysłać komendę „04 portc =” która spowoduje zaświecenie diody. Aby skonfigurować wyprowadzenia do pracy jako wejścia cyfrowe,



Rys. 2



Rys. 3

wanej aplikacji. W tabeli 1 znajduje się spis obsługiwanych komend.

A oto kilka ważnych uwag dotyczących komend sterujących: elementy komendy muszą być oddzielone znakiem spacji. Każda komenda musi kończyć się znakiem nowej linii, czyli tzw. „CR” „FL” (w praktyce chodzi o to, aby każda komenda była zakończona naciśnięciem klawisza ENTER). Również moduł potwierdza każdą komendę oraz każdą

USB w ten sposób ustalana jest prędkość komunikacji, w tym wypadku „low speed” (1,5Mbit/s). Osiem linii mikrokontrolera, poprzez rezystory zabezpieczające o niewielkiej rezystancji, doprowadzonych jest do złącza wyjściowego OUT oraz do diod LED, które mają obrazować stan wyjść. Dodatkowa dioda – LED1, nie jest połączona ze złączem i może być wykorzystana jako dioda sygnalizująca status urządzenia. Poza tymi elementami na schemacie znajduje się tylko rezonator kwarcowy z kondensatorami i pull-up sygnału reset.

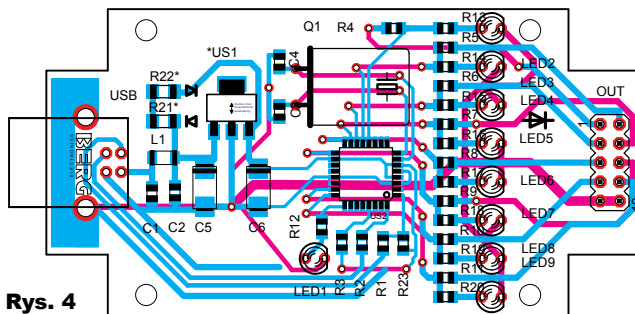
**Komunikacja z modułem.** Oprogramowanie CDC-IO implementuje w mikrokontrolerze AVR prosty interfejs USB1.1. Po dołączeniu do gniazda USB komputera i zainstalowaniu urządzenia zostanie rozpoznane jako wirtualny port szeregowy. Aby sterować pracą urządzenia, należy wysłać z komputera do tego portu proste komendy tekstowe za pomocą dowolnego programu typu terminal lub z użyciem dedyko-

R E K L A M A

odpowiednie bity rejestrów ddr muszą mieć wartość zero. Wysłanie komendy ,3C ddrb =’ ustawi wyprowadzenia 1 i 2 jako wejścia. Aby odczytać stan panujący na wejściach, należy wysłać komendę ,pinb ?’, w odpowiedzi moduł wyśle wartość hex. opisującą stan na całym porcie ,B’.

**Aplikacja sterująca.** Widok okna aplikacji sterującej, wykorzystanej w komputerze PC, przedstawiony jest na **rysunku 2**. Program został napisany za pomocą darmowego narzędzia Visual C# Express 2010, pełny kod dołączony jest w Elportalu wśród materiałów dodatkowych do projektu.

Po uruchomieniu aplikacja odnajduje wszystkie istniejące w komputerze porty COM, w górnej ramce należy wskazać numer portu, na którym został zainstalowany moduł. Po nawiązaniu komunikacji, przy pierwszym uruchomieniu programu, zostaje utworzony plik konfiguracyjny, przy kolejnych uruchomieniach ten etap jest pomijany. Zawartość pliku konfiguracyjnego pokazana jest na **rysunku 3**. W pierwszej linii zapisana jest wartość FF. Określa ona, które wyprowadzenia modułu mają być skonfigurowane jako wyjścia, w tym wypadku wszystkie. A więc program wysyła do modułu komendy: ,3F ddrb =’ oraz 07 ddrc =. Następnie program odczytuje stan wyprowadzeń modułu za pomocą komend: ,pinb ?’ oraz ,pinc ?’ i przekłada stan na obrazy kontrolki (czerwonych diod LED). Jednocześnie program zaświeca kontrolkę statusu oraz diodę statusu (LED1). Wartość w kolejnej linii ma znaczenie tylko dla wyprowadzeń pracujących jako wejścia – określa, które wejścia będą miały włączone rezystory podciągające do plusa zasilania, w tym wypadku 00, czyli żadne z wejść. Następnymi osiem linijek to nazwy, które mają być przypisane ośmiu kontrolkom w programie – ośmiu wyprowadzeniom w module. Plik konfiguracyjny znajdujący się w komputerze można edytować w notatniku, tworząc różne konfiguracje dla modułu; ważne jest tylko, aby nie zmieniać jego nazwy.

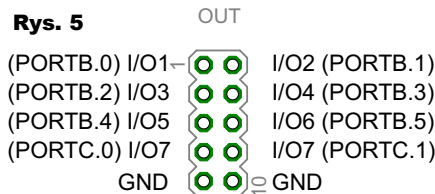


Rys. 4

Kliknięcie na kontrolkę powoduje zmianę jej stanu na przeciwny oraz wysłanie do modułu odpowiedniej komendy. Program wykorzystuje do tego komendy ,xx portb =’ oraz ,xx portc =’, a więc za każdym razem wysyła stan całego wyjścia, a nie tylko pojedynczych bitów (nie wykorzystuje komend ,|’, ,&’, ,^’). Aby sprawdzić stan na wejściach modułu, należy wysłać odpowiednie komendy i odczytać odpowiedź modułu; aplikacja uruchamia się z domyślnie włączoną funkcją ,Auto odczyt’. Oznacza to, że cyklicznie (około 2 razy na sekundę) program odpytuje moduł o stan wejść i obrazuje to na kontrolkach. Funkcję tę można wyłączyć, wtedy sprawdzenie stanu możliwe jest po kliknięciu na przycisk ,Odczyt stanu wejść’.

## Montaż i uruchomienie

Montaż wykonujemy według ogólnych zasad, zgodnie ze schematem montażowym na **rysunku 4**. Pomijamy stabilizator \*US1, zamiast niego montujemy diody w miejscu R21\* i R22\*. Po zmontowaniu należy płytkę umyć i dokładnie sprawdzić poprawność montażu. Jakikolwiek błąd może spowodować uszkodzenie portu USB. Po dołączeniu do komputera system operacyjny poprosi o sterowniki do urządzenia, sterowniki dostępne są w materiałach dodatkowych oraz na stronie: <http://www.recursion.jp/avrddc/driver.html#windows>. W systemach 64-bitowych wystąpi problem z instalacją sterowników niepodpisanych cyfrowo. Problem ten można ominąć: w Internecie można znaleźć informacje na ten temat, jednak mniej doświadczeni użytkownicy komputerów nie powinni tego robić. Po zainstalowaniu sterowników można uruchomić aplikację



## Wykaz elementów

R1, R2, R4-R11.....	22R SMD0805
R3, R12-R20.....	3,3k SMD0805
R23.....	1,5k SMD0805
R21*, R22*.....	1N4148 SMD
C1, C2.....	100nF SMD0805
C3, C4.....	15-18pF SMD0805
C5, C6.....	10uF SMD1206
L1.....	10uH SMD1206
Q1.....	rezonator 12MHz
LED1.....	LED 3mm zielona
LED2-LED9.....	LED 3mm czerwone
*US1.....	nie montować
US2.....	ATmega8 SMD
USB.....	gniazdo USB B
OUT.....	goldpin 2x5

**Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3130.**

i jeśli wszystko zostało wykonane prawidłowo, w module zaświeci się zielona dioda.

Rozmieszczenie portów na złączu wyjściowym przedstawia **rysunek 5**. Każde z wyprowadzeń modułu daje do dyspozycji napięcie około 3,5V oraz prąd około 5...10mA. Aby sterować pracą innych urządzeń, niezbędne będą układy wykonawcze np. w postaci tranzystora i przekaźnika. Ciekawym rozszerzeniem dla modułu może być zestaw AVT1560 – 8-kanalowa karta przekaźników. Jeśli wyprowadzenia będą pełniły funkcje wejść, to należy pamiętać, że maksymalne napięcie, jakie można doprowadzić, to ok 3,5V. Jeśli zajdzie potrzeba doprowadzenia sygnału o napięciu 5V, można to zrobić poprzez rezystor o wartości kilku kΩ. Jeśli stanem aktywnym na wejściu ma być zwarcie do masy, to należy na tym wejściu załączyć pull-upa (rezystor podciągający do plusa).

Oprogramowanie CDC daje dużo więcej możliwości niż to zostało opisane w powyższym artykule – zainteresowanych odsyłam do strony źródłowej projektu.

**KS**  
ksavt@wp.pl