

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:

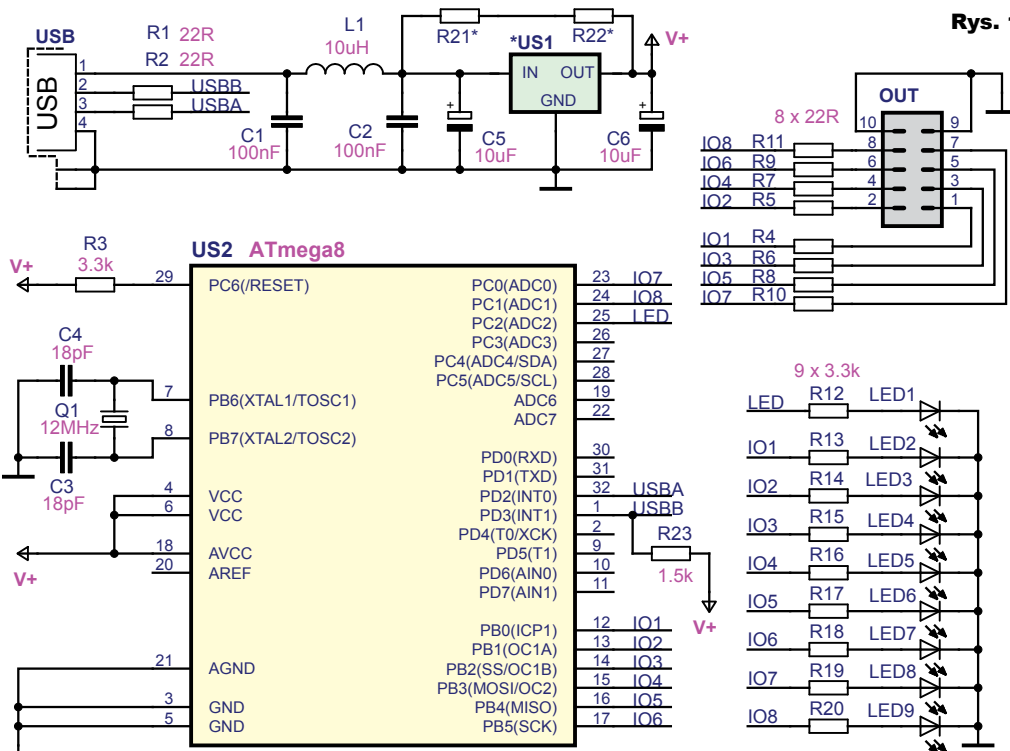
- 8 wprowadzeń, każde z nich może pracować jako wyjście lub wejście cyfrowe,
- sterowanie i zasilanie z portu USB,
- wykorzystuje klasę komunikacyjną CDC,
- emuluje port szeregowy COM, parametry komunikacji 9600, 8, 1, none,
- sterowniki dla systemów Win XP, Win7 (32 bity).
- dedykowana aplikacja dla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

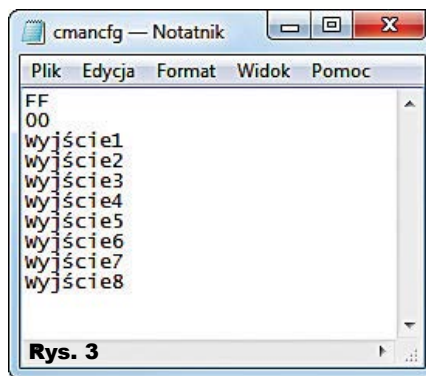
Po podłączeniu modułu wszystkie rejestry mają wartość 00, więc wszystkie wyprowadzenia są skonfigurowane jako wejścia. Aby skonfigurować wyprowadzenia jako wyjścia, należy wysłać dwie komendy: „3F ddrb =” oraz „07 ddrc =”. Po takiej operacji wszystkie 8 wyprowadzeń oraz sterowanie diodą LED1 są gotowe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



Rys. 2

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-



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

R E K L A M A

