

Biblioteka AVRUSB

Programowy interfejs USB w AVR

Interfejs USB już na stałe zagościł w prawie każdym urządzeniu, które ma komunikować się z komputerem PC. Do implementacji interfejsu USB we własnych urządzeniach dostępne są różne układy konwerterów pośredniczących w wymianie informacji między mikrokontrolerem a komputerem z interfejsem USB. Dostępne są również mikrokontrolery z już wbudowanym interfejsem USB. A jeśli oba z omawianych rozwiązań są z jakiegoś powodu niemożliwe do zastosowania, to również można zaimplementować USB w sposób programowy. Właśnie w tym celu powstała biblioteka AVRUSB przeznaczona dla mikrokontrolerów z rodziny AVR.

Bibliotekę AVRUSB można pobrać ze strony <http://www.obdev.at/products/avrusb/index.html>. Umożliwia ona emulację interfejsu USB 1.1 i przeznaczona jest dla dowolnych mikrokontrolerów AVR posiadających co najmniej 2 kB pamięci FLASH i 128 bajtów pamięci RAM. Co ważne, do emulacji interfejsu nie są potrzebne żadne sprzętowe UART-y czy Timery. Należy tylko zapewnić taktowanie mikrokontrolera częstotliwością co najmniej 12 MHz. Mikrokontroler może również być taktowany wewnętrznym oscylatorem o częstotliwości 12,8 MHz lub 16,5 MHz. Oprogramowanie wykonano w języku C.

Implementacja interfejsu USB

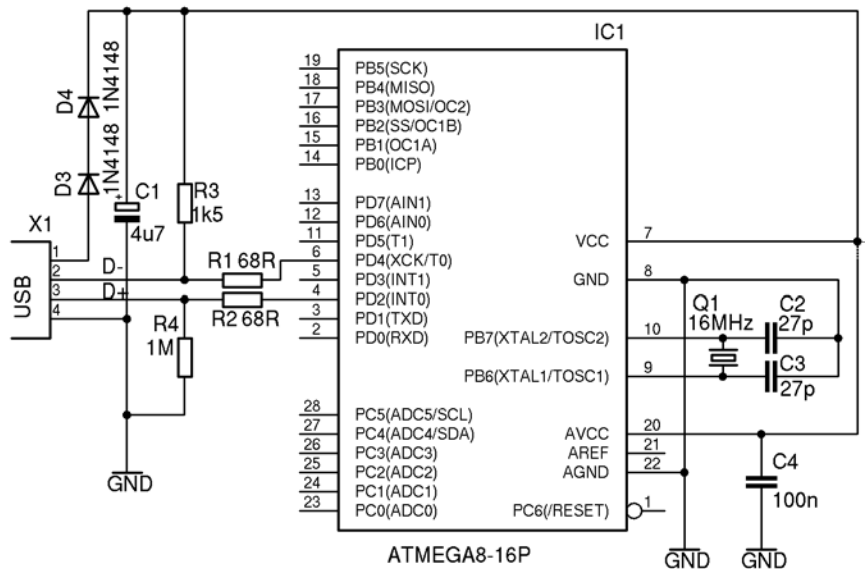
Na rys. 1 pokazano przykład elektrycznej implementacji programowanego interfejsu USB dla popularnego mikrokontrolera ATmega8. Diody D1, D2 zmniejszają napięcie zasilające mikrokontroler do wartości około 3,8 V. Jest to niezbędne dla poprawnej pracy emulowanego interfejsu USB. Zamiast diod można również zastosować stabilizator o napięciu wyjściowym 3,3 V. Jeśli mikrokontroler musi być zasilany napięciem 5 V, to należy wykonać odpowiedni konwerter napięć. Proste rozwiązania często wykorzystują diody Zenera (o napięciu stabilizacji 3,6 V), jednak trzeba powiedzieć, że nie jest to profesjonalny sposób konwersji. Co ciekawe, programowy interfejs USB można również zaimplementować w mikrokontrolerach AVR mających tylko 8 wyprowadzeń (np. ATtiny45), co pokazano na rys. 2. Zaimplementowany programowy interfejs USB może poprawnie współpracować z różnymi systemami operacyjnymi, takimi jak Windows, MAC OS oraz Linux. Na stronie in-

ternetowej producenta biblioteki AVRUSB dostępne są liczne przykłady projektów z zaimplementowanym programowym interfejsem USB. Również w spakowanym archiwum biblioteki można znaleźć kilka ciekawych przykładów jej użycia, a wśród nich wymienić można przykład nadawania oraz odbierania danych przez USB, urządzenia HID (np. klawiatury czy myszki).

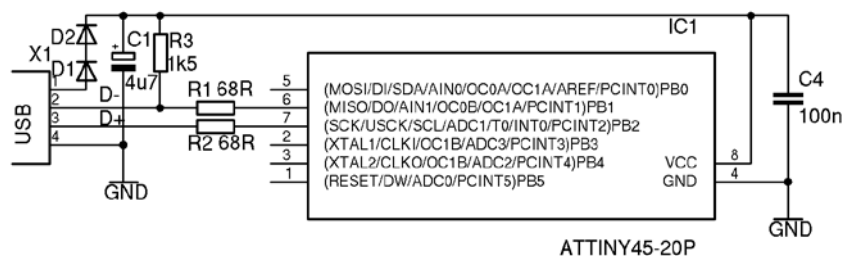
Podsumowanie

Dużą zaletą implementacji programowego interfejsu USB jest jego niski koszt implementacji przy dość dużych możliwościach wykorzystania. Dzięki bibliotece AVRUSB można interfejs USB zaimplementować nawet już w działających urządzeniach, rozbudowując ich możliwości komunikacyjne. Programowy interfejs USB można wykorzystać w wielu urządzeniach do akwizycji danych, w programatorach czy innych urządzeniach, które muszą się komunikować z komputerem. Co ważne, biblioteka dostępna jest na licencji GPL i można ją dostosować do własnych potrzeb. Moim zdaniem, przy implementacji interfejsu USB we własnym urządzeniu warto rozważyć to tanie i efektywne rozwiązanie.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl



Rys. 1. Schemat połączeń programowego interfejsu USB



Rys. 2. Programowy interfejs USB w ATtiny45