

Interfejs USB On-The-Go w mikrokontrolerach Kinetis

Uniwersalna magistrala szeregową (USB) jest standardem interfejsu komunikacyjnego także w systemach mikrokontrolerowych. Mikrokontrolery Kinetis wyposażono w interfejs USB-OTG 2.0 Full-Speed, który umożliwi komunikację z otoczeniem z prędkością 12 Mb/s. Przedstawione w artykule przykładowe aplikacje wykorzystujące różne tryby pracy USB przygotowano dla mikrokontrolerów z podrodzin K40, K53, K60, m.in. dla zestawu KwikStik.

Tryby pracy interfejsu USB: *device*

Interfejs USB w tym trybie jest konfigurowany do pracy jako urządzenie podrzędne, odpowiadające na żądania urządzenia *host*. Wszystkie transmisje są rozpoczynane przez kontroler *host*, który zapewnia także napięcie VBUS. Zintegrowany z interfejsem USB, wbudowany w mikrokontrolery Kinetis moduł DCD pracuje równolegle z USB: po dołączeniu do *hosta* DCD wykrywa jego typ, następnie USB przejmuje kontrolę nad sygnałami D+ oraz D-.

Tryby pracy interfejsu USB: *host*

W trybie tym interfejs USB pracuje jako *host* posiadający pełną kontrolę nad magistralą USB. Sprzętowy silnik interfejsu USB zajmuje się synchronizacją czasową transmi-

sji, programowy stos jest odpowiedzialny za zarządzanie transferami.

Wbudowany w mikrokontroler stabilizator napięcia jest złożony z dwóch niezależnych części: regulatora *RUN* i regulatora *STANDBY*. Dzięki wykorzystaniu bitu czuwania w module integracji systemu mikrokontrolera, użyt-

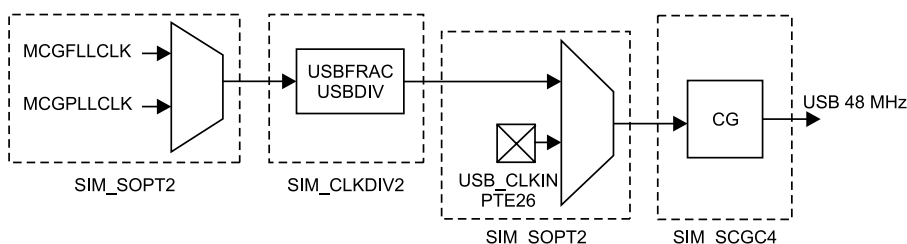
Dodatkowe informacje...
 ...są dostępne pod adresem:
www.freescale.com/medicalusb
 oraz www.kinetis.pl

Dodatkowe materiały na CD/FTP:
<ftp://ep.com.pl>, user: 16344, pass: 52qof741

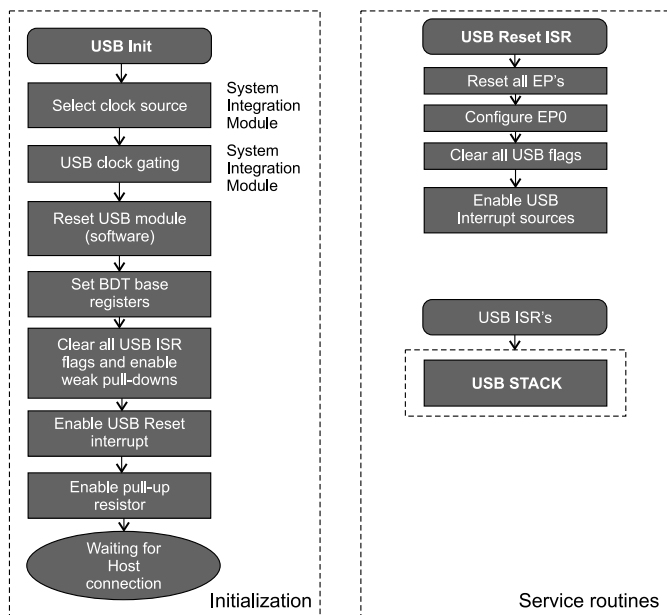
- Pliki programów przykładowych są dostępne na portalu www.KINETIS.pl i serwerze redakcyjnym EP.

kownik może wybrać, który regulator będzie używany. Wejściowe wyprowadzenie regulatora jest oznaczone jako VREGIN, a wyprowadzenie wyjściowe jako VOUT33.

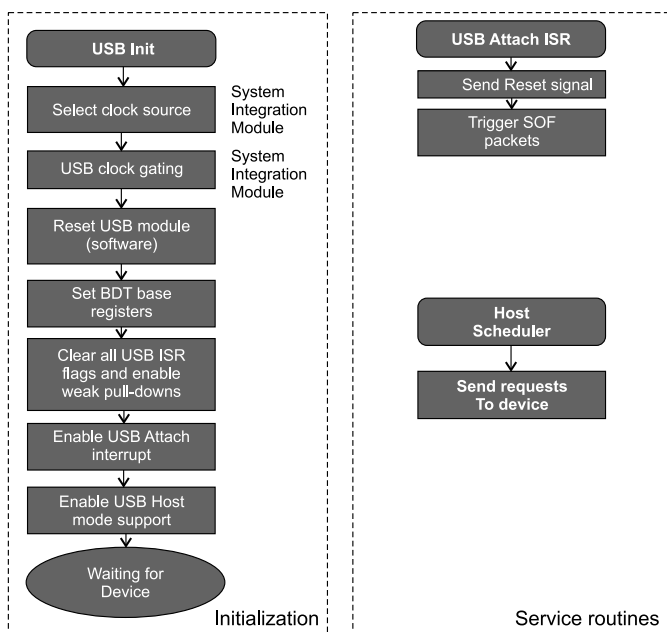
W trybie *RUN* regulatory *RUN* oraz *STANDBY* są aktywne, ale przełącznik łączący wyjście regulatora *STANDBY* z zewnętrznym wyprowadzeniem jest otwarty.



Rys. 1. Schemat blokowy systemu taktującego interfejsu USB w mikrokontrolerach Kinetis



Rys. 2. Przebieg procesu inicjalizacji trybu device



Rys. 3. Przebieg procesu inicjalizacji trybu host

W trybie *STANDBY* regulator *RUN* jest wyłączony, a regulator *STANDBY* jest aktywny. Przełącznik łączący wyjście regulatora *STANDBY* z zewnętrznym wyprowadzeniem jest zamknięty. W trybie *SHUTDOWN* obydwa stabilizatory są wyłączone.

Interfejs USB potrzebuje do prawidłowego działania sygnału zegarowego o częstotliwości 48 MHz. Są trzy możliwe źródła sygnału dla zegara USB: PLL, FLL oraz zewnętrzne wyprowadzenie oznaczone jako *USB_CLKIN*. W przypadku PLL i FLL po multiplekszerze znajduje się dzielnik częstotliwości. Dzieli on częstotliwości pętli PLL oraz FLL, aby umożliwić mikrokontrolerowi pracę przy częstotliwościach wyższych niż 48 MHz. Wyjście dzielnika częstotliwości doprowadzono do multipleksera (**rysunek 1**), po czym następuje wybór pomiędzy tym sygnałem oraz wyprowadzeniem *USB_CLKIN*. Wartość współczynnika podziału dzielnika częstotliwości może być konfigurowana w rejestrze *SIM_CLKDIV2* wewnątrz modułu integralności systemu (*SIM*).

Inicjalizacja USB

Interfejs USB wbudowany w mikrokontrolery Kinetis może pracować w trybie *device* jak i w trybie *host*. Podczas inicjalizacji oba

REKLAMA



Electronics Worldwide

Consult | Components | Logistics | Support

Jako znaczący dystrybutor elementów elektroniki na rynku światowym, kierujemy ofertę dla firm wymagających pełnego zakresu produktów, jak również świadczymy szeroko rozumianą pomoc techniczną dla rozwoju produktu oraz Design-In,

Dla naszego klienta dostosowujemy rozwiązania, w pełni uwzględniające jego specyficzne potrzeby produkcyjne i logistyczne.



tryby przebiegają podobnie, drobne różnice pomiędzy nimi omówimy w dalszej części artykułu.

W trybie inicjalizacji trybu *device* (rysunek 2) interfejs USB aktywuje rezystor *pull-up*, gdy inicjalizacja jest zakończona, aby został wykryty przez *hosta*, do którego dołączono urządzenie USB.

W celu uruchomienia trybu *host* (rysunek 3) oprogramowanie inicjuje generację sygnału SOF (*Start Of Frame*). Po wykryciu dołączenia do linii D+ lub D- rezystora *pull-up*, interfejs generuje powiązane przerwanie sygnalizujące CPU, że urządzenie *device* zostało dołączone do magistrali i musi zostać rozpoczęty proces wykrywania.

Interfejs USB 2.0 składa się z sygnałów D+ i D-, linii zasilającej VBUS (linia zasilania 5 V), linii masy (referencyjnej) oraz w niektórych przypadkach dodatkowego wyprowadzenia oznaczonego ID. Wyprowadzenie ID jest uwzględnione w specyfikacji OTG i jest wykorzystywane, gdy jedno urządzenie może spełniać rolę zarówno *hosta*, jak i *device*, zależnie od tego, jaka wtyczka jest połączona ze złączem na płytce. Wtyczka mini-A, która oznacza, że ta część jest *hostem*, posiada uziemione wyprowadzenie ID, podczas gdy wyprowadzenie ID we wtyczce mini-B jest niepodłączone, co oznacza, że ta strona komunikacji będzie pracowała jako *device*.

Jeśli aplikacja wspiera jedynie tryb *host*, nie jest konieczne sprzętowe zwieranie linii ID. Ponieważ jest to *host*, konstrukcyjnie musimy zapewnić możliwość dostarczenia napięcia 5 V i odpowiednią moc dla urządzenia *device*. Napięcie to jest zazwyczaj zapewniane przez zewnętrzny układ scalony (klucz prądowy) sterowany przez mikrokontroler.

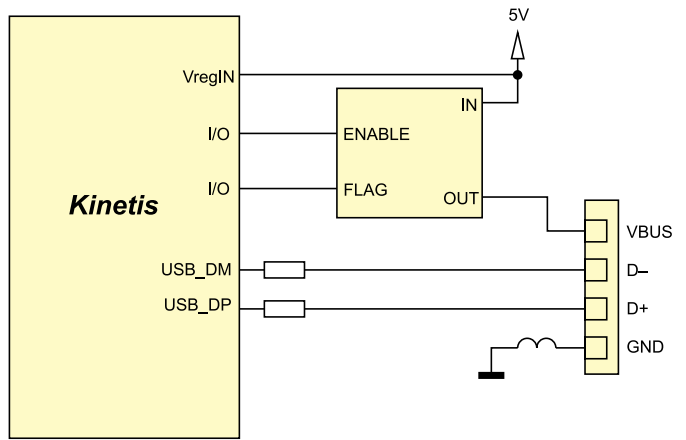
W wielu przypadkach konieczna jest komunikacja z aplikacją działającą na komputerze osobistym. W tym przypadku, aplikacja zaimplementowana na mikrokontrolerze obsługuje jedynie tryb *device*. Urządzenie może być zasilane z własnego zasilacza, źródła zewnętrznego lub z magistrali (5 V pochodzące od *hosta*). W obu przypadkach wewnętrzny stabilizator napięcia z USB musi być włączony aby zasilac transceiver USB. Przy takim rozwiązaniu nie jest wymagana obsługa linii ID.

Tryb *host/device* jest wykorzystywany, kiedy aplikacja może być podłączona do komputera osobistego lub jest zdolna do obsługi zewnętrznych urządzeń USB, takich jak czytniki odcisków palców, myszki, pamięci Flash USB i inne. Aplikacja działająca na mikrokontrolerze będzie skonfigurowana w trybie urządzenia (bez dostarczania 5 V na linię VBUS), dopóki sygnał na linii ID nie przejdzie w stan niski. Oznacza to, że potrzebna jest ponowna konfiguracja *hosta*, a napięcie 5 V jest podłączane do sygnału VBUS przy wykorzystaniu zewnętrznego scalonego klucza prądowego.

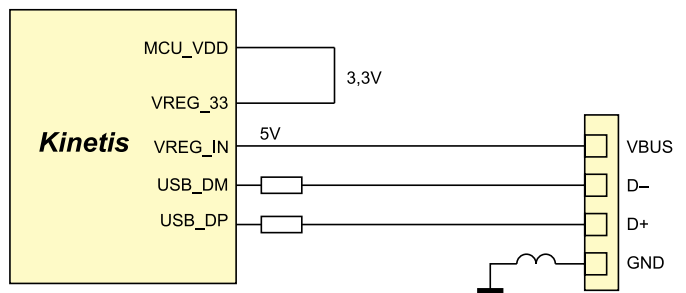
Przykładowa aplikacja

Program demonstracyjny spełnia rolę terminala wykorzystującego klasę komunikacyjną CDC interfejsu USB. Interfejs USB wbudowany w mikrokontroler jest rozpoznawany jako standardowy port COM, który może zostać wykorzystany przez *HyperTerminal* lub jakkolwiek inny program terminalowy. Źródła programu są dostępne na portalu www.KINETIS.pl, a także na redakcyjnym serwerze FTP.

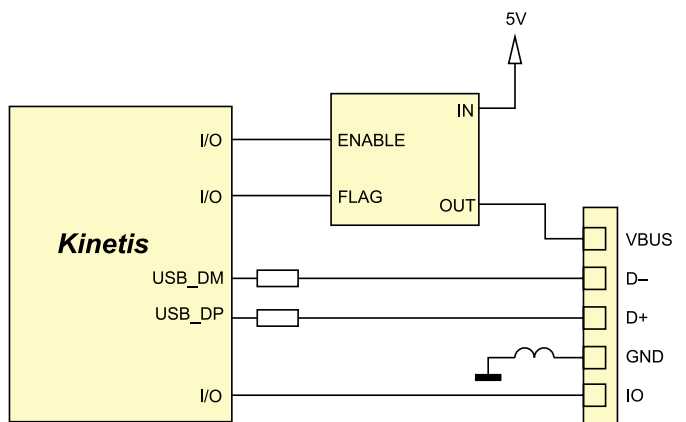
Po podłączeniu płytki KwikStik lub innej z mikrokontrolerem Kinetis i uruchomieniu *HyperTerminala* należy wybrać odpowiedni interfejs COM, ramkę danych 8 bitów, 1 bit stopu, brak kontroli przepływu danych i prędkość transmisji 9600 bodów.



Rys. 4. Schemat blokowy interfejsu w trybie host



Rys. 5. Schemat blokowy realizacji trybu device



Rys. 6. Schemat blokowy realizacji trybu host/device

Przykładowa aplikacja *host* wykrywa mysz HID USB i wysyła informację do terminala wykorzystując port szeregowy. Raportuje on także bezpośrednio do terminala o wszystkich ruchach oraz zmianach pozycji przycisków. W tym trybie terminal należy skonfigurować do pracy w trybie 8 bitów, 1 bit stopu, brak kontroli przepływu, 115200 bodów. Po dołączeniu płytki z mikrokontrolerem i zainicjowaniu USB aplikacja wyśle komunikat informujący, że na dołączenie myszki USB HID. Po pojawieniu się tej wiadomości należy podłączyć myszkę USB do złącza USB, co zostanie potwierdzone komunikatem o typie dołączonego urządzenia.

Andrzej Gawryluk

REKLAMA

Softstart do żarówek samochodowych
AVT 1599
www.sklep.avt.pl