

STM8S001J3 (4)

Blok sygnałów zegarowych mikrokontrolera

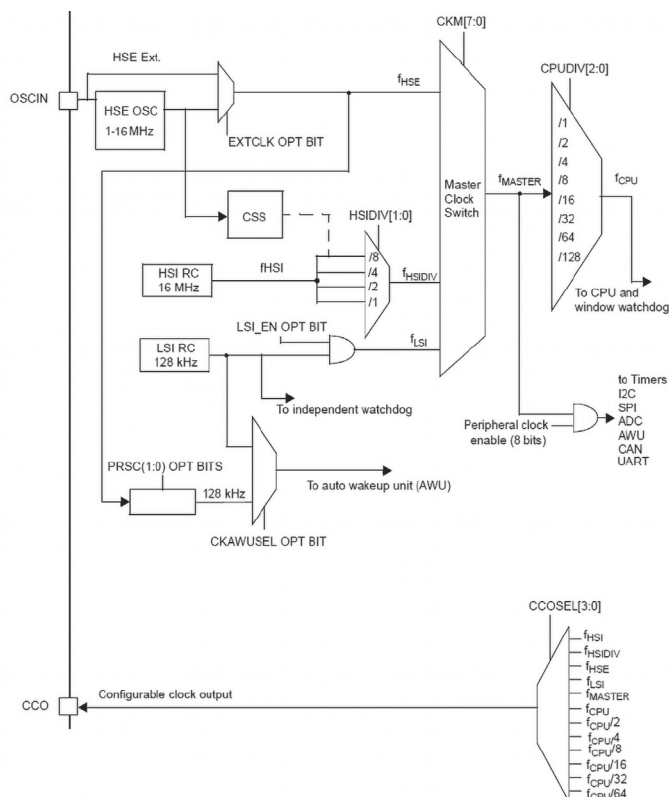
Poprzednie artykuły z serii dotyczącej 8-pinowego mikrokontrolera STM8S001J3 dostarczały informacji o tym, jak zacząć pracę z tym układem: w warstwie sprzętowej (projekt płytki PCB) oraz programowej (instalowanie narzędzi, wykonanie projektu w środowisku programistycznym). Czwarty artykuł rozpoczyna cykl dotyczący peryferii. Jako pierwszy wybrany został blok sygnałów zegarowych, dla którego przedstawiono opis oraz przykładową aplikację.



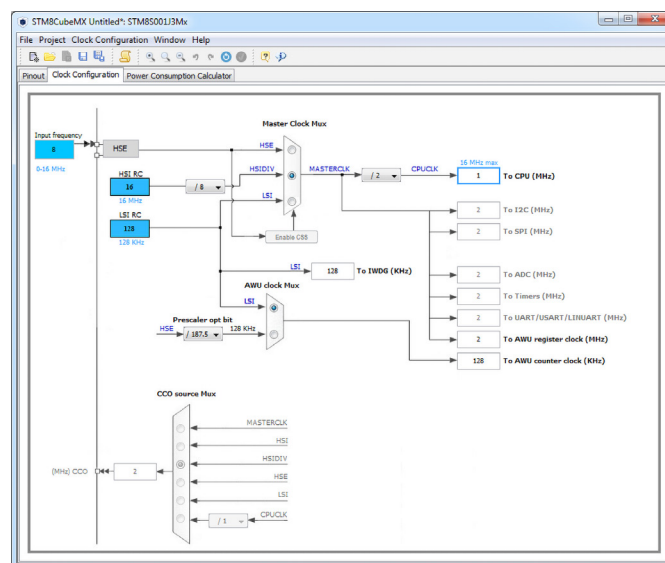
Blok sygnałów zegarowych można zdefiniować jako moduł funkcjonalny mikrokontrolera, który odpowiada za wytworzenie sygnałów zegarowych o określonej częstotliwości, przetworzenie ich przez podzielniki o edytowalnej wartości oraz doprowadzenie ich (przy udziale multiplekserów) do zasobów mikrokontrolera w celu ich takowania: rdzenia (CPU), pamięci oraz peryferii (np. przetwornika A/C, interfejsów komunikacyjnych, timerów itp.).

Schemat bloku sygnałów zegarowych w mikrokontrolerze STM8S001J3 pokazano na **rysunku 1**. Dostępne są trzy źródła sygnału zegarowego: HSE, HSI oraz LSI. HSE (*High Speed External*) to zewnętrzne źródło sygnału zegarowego. Funkcję tę pełni może oscylator, z którego sygnał zegarowy (o częstotliwości maksymalnej 16 MHz) doprowadzony jest do pinu OSCIN mikrokontrolera. Z kolei HSI (*High Speed Internal*) jest wewnętrznym źródłem zegarowym. Jest

to oscylator RC wytwarzający sygnał o stałej częstotliwości 16 MHz. Sygnał ten doprowadzony jest do podzielnika *HSIDIV*, który w zależności od konfiguracji może mieć wartość 1, 2, 4 lub 8. Tak więc na wyjściu podzielnika możliwe jest uzyskanie sygnału zegarowego (*fHSIDIV*) o częstotliwości 2...16 MHz. Trzecim i zarazem ostatnim źródłem sygnału zegarowego jest LSI (*Low Speed Internal*). Ten wewnętrzny oscylator RC generuje sygnał o częstotliwości 128 kHz. Sygnały ze wszystkich trzech źródeł (HSE do 16 MHz, HSI 2...16 MHz, LSI 128 kHz) doprowadzone są do multipleksera o nazwie *Master Clock Switch*. W zależności od jego konfiguracji jeden z trzech sygnałów zostaje wybrany i jest dystrybuowany dalej: jako *fMASTER* bezpośrednio do peryferii mikrokontrolera oraz jako *fCPU* pośrednio (przez podzielnik *CPUDIV*) do procesora mikrokontrolera. Dostępne wartości podzielnika *CPUDIV* to 1, 2, 4, 8, 16, 32, 64 i 128. Łatwo zatem zauważyć, że maksymalna możliwa do uzyskania częstotliwość dla *fMASTER* i *fCPU* to 16 MHz, natomiast wartości minimalne to odpowiednio 128 kHz dla *fMASTER* oraz 1 kHz dla *fCPU*.



Rysunek 1. Schemat bloku sygnałów zegarowych w mikrokontrolerze STM8S001J3



Rysunek 2. Okno z drzewem sygnałów zegarowych mikrokontrolera STM8S001J3 w aplikacji STM8CubeMX

Tabela 1. Zestawienie wybranych funkcji do sterowania blokiem sygnałów zegarowych STM8S

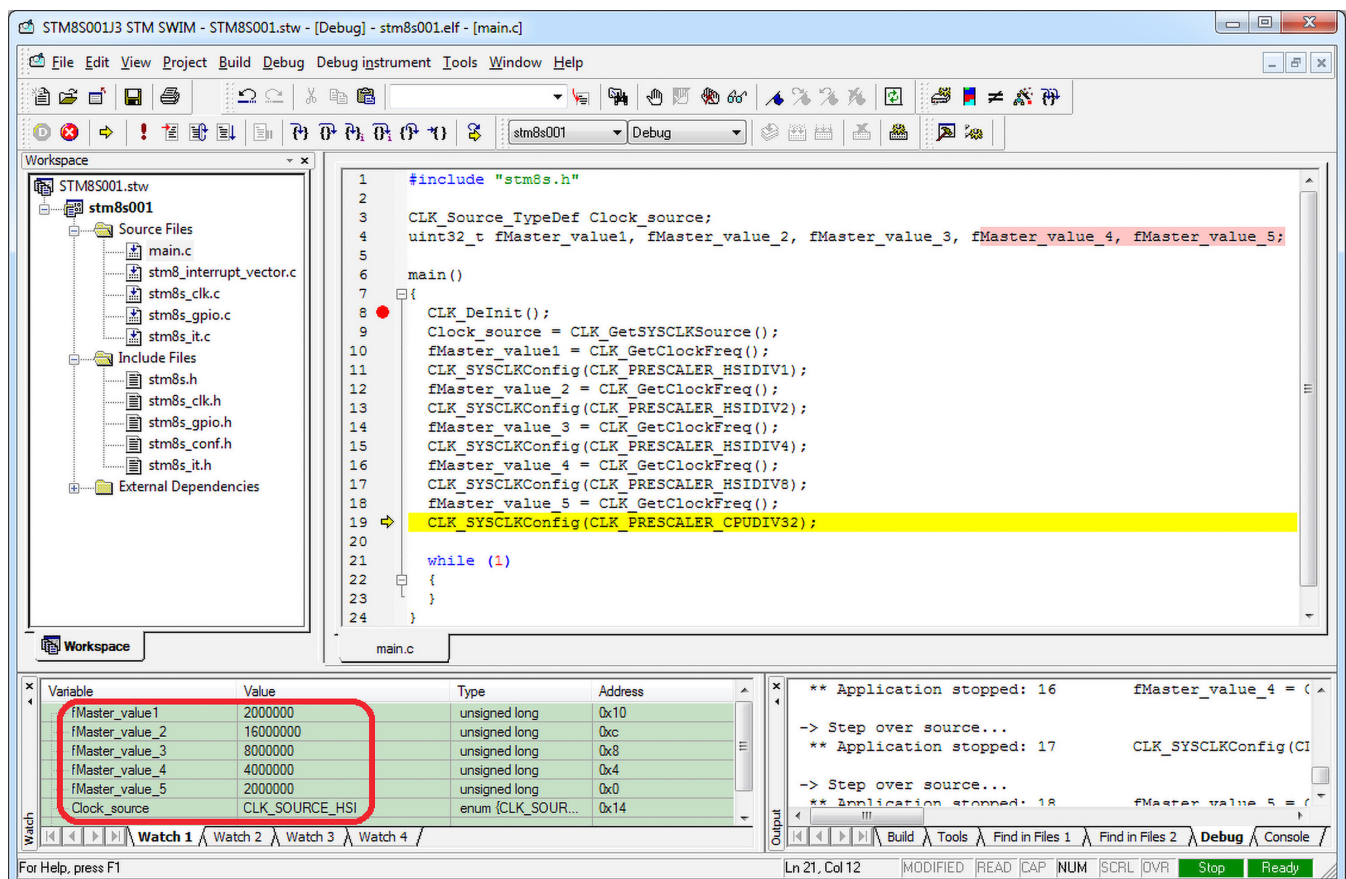
Nazwa funkcji	Opis działania funkcji
CLK_DeInit(void)	Ustawienie wartości domyślnych w rejestrach domeny zegarowej
CLK_HSECmd(FunctionalState NewState)	Włączenie lub wyłączenie źródła sygnału zegarowego HSE
CLK_HSCmd(FunctionalState NewState)	Włączenie lub wyłączenie źródła sygnału zegarowego HSI
CLK_LSCmd(FunctionalState NewState)	Włączenie lub wyłączenie źródła sygnału zegarowego LSI
CLK_CCOCmd(FunctionalState NewState)	Włączenie lub wyłączenie funkcji CCO służącej do wyprowadzenia wybranego sygnału zegarowego na pin CCO
CLK_ClockSwitchCmd(FunctionalState NewState)	Włączenie lub wyłączenie możliwości zmiany używanego źródła sygnału zegarowego
CLK_PeripheralClockConfig(CLK_Peripheral_TypeDef CLK_Peripheral, FunctionalState NewState)	Włączenie lub wyłączenie zegara dla wybranego zasobu mikrokontrolera (przetwornika A/C, interfejsu UART, timera itp.)
CLK_ClockSwitchConfig(CLK_SwitchMode_TypeDef CLK_SwitchMode, CLK_Source_TypeDef CLK_NewClock, FunctionalState ITState, CLK_CurrentClockState_TypeDef CLK_CurrentClockState);	Dokonanie zmiany używanego źródła sygnału zegarowego
CLK_HSPrescalerConfig(CLK_Prescaler_TypeDef HSPrescaler);	Ustawienie wartości podzielnika dla źródła sygnału zegarowego HSI
CLK_CCOConfig(CLK_Output_TypeDef CLK_CCO);	Wybranie sygnału zegarowego, który zostanie wyprowadzony na pin CCO
CLK_SYSCLKConfig(CLK_Prescaler_TypeDef CLK_Prescaler);	Ustawienie wartości podzielnika dla sygnału fHSIDIV lub fCPU
CLK_ClockSecuritySystemEnable(void)	Włączenie automatycznej zmiany źródła sygnału zegarowego z HSE na HSI/8 w przypadku awarii tego pierwszego
CLK_GetClockFreq(void)	Odczytanie aktualnej wartości częstotliwości sygnału fMaster
CLK_GetSYSCLKSource(void)	Odczytanie nazwy używanego źródła sygnału zegarowego

Warto w tym miejscu dodać, że domyślną konfiguracją bloku sygnałów zegarowych mikrokontrolera tuż po podłączeniu napięcia zasilania jest zawsze aktywne źródło HSI oraz podzielnik o wartości 8. Skutkiem tego mikrokontroler rozpoczyna pracę, używając sygnału o częstotliwości 2 MHz.

Interesującym rozwiązaniem zastosowanym w bloku sygnałów zegarowych jest moduł CSS (*Clock Security System*). W przypadku, gdy używane jest źródło HSE i z jakiegoś powodu przestanie ono

nagle wytwarzać sygnał zegarowy (przykładowo z powodu fizycznego uszkodzenia), moduł CSS automatycznie wykryje zanik sygnału i automatycznie przełączy mikrokontroler na taktowanie ze źródła HSI/8. Jest to zatem funkcja bezpieczeństwa, która gwarantuje ciągłość pracy mikrokontrolera w przypadku awarii HSE.

Warto również wspomnieć, że wiele spośród sygnałów bloku zegarowego można wyprowadzić na zewnątrz mikrokontrolera. Służy do tego



Rysunek 3. Środowisko STVD w trybie debugowania, udostępniające w oknie Watch wartości zmiennych kodu aplikacji sterującej blokiem sygnałów zegarowych mikrokontrolera

moduł CCO (*Configurable main Clock Output*). Wybór sygnału odbywa się za pomocą multiplexera CCOSEL. Wybrany sygnał doprowadzony zostaje do pinu CCO mikrokontrolera.

Istotną zaletą bloku sygnałów zegarowych w układach STM8 jest możliwość dynamicznej konfiguracji wszystkich jego elementów. Oznacza to, że w trakcie działania aplikacji możliwa jest zmiana używanego źródła sygnału zegarowego oraz wartości poszczególnych podzielników. Korzyścią takiego rozwiązania jest możliwość zmiany wydajności obliczeniowej i poboru prądu mikrokontrolera w taki sposób, aby wielkości te były różne (optymalne) dla różnych etapów działania aplikacji.

Blok sygnałów zegarowych w STM8CubeMX

Wartościowym dodatkiem do dokumentacji technicznej pozwalającym lepiej zrozumieć blok sygnałów zegarowych mikrokontrolera STM8S001J3 jest program komputerowy STM8CubeMX. Okno *Clock Configuration* tego programu udostępnia użytkownikowi pełne drzewo sygnałów zegarowych i co więcej, umożliwia jego graficzną edycję. Dokonując samodzielnie zmian wartości poszczególnych podzielników i zmian ustawień multiplexerów, użytkownik natychmiast widzi, jak te zmiany wpływają na częstotliwość sygnałów taktujących CPU, pamięć oraz peryferia. Widok okna *Clock Configuration* programu STM8CubeMX pokazano na **rysunku 2**.

Funkcje SPL do sterowania blokiem sygnałów zegarowych

Aby w prosty sposób skonfigurować blok sygnałów zegarowych układu STM8S001J3, warto w aplikacji użyć bibliotek SPL (*Standard Peripheral Library*) przygotowanych dla mikrokontrolerów STM8S. Pliki *stm8s_clk.h* oraz *stm8s_clk.c* udostępniają szereg funkcji do tego celu. Ich zestawienie umieszczono w **tabeli 1**.

Przykładowa aplikacja sterująca blokiem sygnałów zegarowych

W celu stworzenia przykładowej aplikacji użyte zostało środowisko programistyczne STVD (ST Visual Develop) oraz kompilator Cosmic CXSTM8. Opis tych narzędzi, jak również instrukcja, jak stworzyć za ich pomocą szablon nowego projektu wraz z dodaniem bibliotek SPL, dostępne są w poprzednim artykule (numer 3) z tej serii. Korzystając ze wspomnianego szablonu projektu, należy edytować kod pliku *main.c*, w którym umieszczony zostanie cały kod aplikacji.

Aplikacja ma na celu pokazać, w jaki sposób konfigurować blok sygnałów zegarowych. Ma ona również na celu zademonstrować możliwość dynamicznej konfiguracji sygnałów zegarowych w trakcie działania aplikacji. W tym celu wykonane zostaną następujące kroki:

- Stworzona zostanie zmienna, do której będzie można wczytać nazwę używanego aktualnie źródła sygnału zegarowego.
- Stworzone zostaną zmienne, do których będzie można wczytać częstotliwość sygnału fMASTER po różnych konfiguracjach tego sygnału.
- Na początku aplikacji blok sygnałów zegarowych zostanie skonfigurowany w sposób domyślny. W tym celu wywołana zostanie funkcja *CLK_DeInit()*.
- Sprawdzone zostanie, jakie źródło sygnału zegarowego jest używane. W tym celu użyta zostanie funkcja *CLK_GetSYSCLKSource()*.
- Pierwszy raz sprawdzona zostanie częstotliwość sygnału fMASTER, do czego posłuży funkcja *CLK_GetClockFreq()*.
- Czterokrotnie zmieniony zostanie podzielnik używanego źródła sygnału HSI i za każdym razem sprawdzone

```
Listing 1. Wykonany w oparciu o funkcję SPL kod pliku main.c, który implementuje aplikację demo dla bloku sygnałów zegarowych mikrokontrolera STM8S001J3
#include „stm8s.h”

CLK_Source_TypeDef Clock_source;
uint32_t fMaster_value_1, fMaster_value_2, fMaster_value_3, fMaster_value_4, fMaster_value_5;

main()
{
    CLK_DeInit();
    Clock_source = CLK_GetSYSCLKSource();
    fMaster_value_1 = CLK_GetClockFreq();
    CLK_SYSClkConfig(CLK_PRESCALER_HSIDIV1);
    fMaster_value_2 = CLK_GetClockFreq();
    CLK_SYSClkConfig(CLK_PRESCALER_HSIDIV2);
    fMaster_value_3 = CLK_GetClockFreq();
    CLK_SYSClkConfig(CLK_PRESCALER_HSIDIV4);
    fMaster_value_4 = CLK_GetClockFreq();
    CLK_SYSClkConfig(CLK_PRESCALER_HSIDIV8);
    fMaster_value_5 = CLK_GetClockFreq();
    CLK_SYSClkConfig(CLK_PRESCALER_CPUDIV32);
    while (1)
    {
    }
}
```

zostanie, czy sygnał fMASTER został adekwatnie zmieniony. W tym celu użyte zostaną funkcje *CLK_SYSClkConfig()* oraz *CLK_GetClockFreq()*.

- Zmieniony zostanie podzielnik dla sygnału fCPU.

Kod zgodny z zaprezentowanym opisem pokazano w **listingu 1**.

Działanie aplikacji łatwo jest zweryfikować. Należy skompilować projekt, a następnie wgrać plik wynikowy do pamięci mikrokontrolera i otworzyć tryb debugowania. Teraz należy dodać do okna *Watch* stworzone w kodzie zmienne (należy najechać kursorem myszy na każdą zmienną, kliknąć na niej prawym przyciskiem myszy i z menu kontekstowego wybrać opcję *QuickWatch...*). Po uruchomieniu aplikacji należy zaobserwować, jakie wartości zostały zwrócone przez funkcje i wpisane zostały tym samym do zmiennych. Proces ten pokazano na **rysunku 3**. Widać, że mikrokontroler rozpoczyna pracę, korzystając ze źródła sygnału zegarowego HSI (zmienna *Clock_source* ma wartość *CLK_SOURCE_HSI*) i częstotliwość tego sygnału ma wartość 2 MHz (zmienna *fMASTER_value_1* ma wartość 2000000). Następnie gdy zmienia się podzielnik HSIDIV kolejno na 1, 2, 4 oraz 8, częstotliwość sygnału zmienia się adekwatnie (wartości zmiennych *fMaster_value_2/3/4/5* to odpowiednio 16000000, 8000000, 4000000 oraz 2000000). Na końcu aplikacji dodatkowo zmieniany jest podzielnik CPUDIV na 32, co skutkuje zmianą sygnału fCPU na 62.5 kHz (2 MHz/8/32).

Podsumowanie

W artykule przekazano podstawowe informacje o bloku sygnałów zegarowych mikrokontrolera STM8S001J3 wraz z opisem przykładowej aplikacji demo. Osoby chcące dowiedzieć się bardziej szczegółowych informacji powinny sięgnąć do dokumentacji technicznej producenta. Parametry i charakterystyka tego zasobu dostępne są w nocie katalogowej mikrokontrolera (*datasheet*). Z kolei schematy oraz opis wszystkich funkcjonalności i rejestrów znajduje się w podręczniku użytkownika mikrokontrolera (*user manual* RM0016). Dodatkowo trzy aplikacje testowe dostępne są w podkatalogu bibliotek SPL: `...\\STM8S_StdPeriph_Lib\\Project\\STM8S_StdPeriph_Examples\\CLK`.

Szymon Panecki
szymon.panecki@st.com

REKLAMA

zawsze na czasie

ELEKTRONIKA PRAKTYCZNA 3/2018 95