

Wprowadzenie

Mikrokontrolery ST7LITE są uproszczoną wersją „dużych” mikrokontrolerów ST72, z tym samym rdzeniem i nieco uboższym zestawem układów peryferyjnych. Pojemność wewnętrznych pamięci również jest mniejsza w porównaniu z mikrokontrolerami z rodziny ST72: wyposażono je w pamięci Flash o pojemności w zakresie od 1 do 8 kilobajtów.

Mikrokontrolery ST7LITE są dostępne w obudowach o liczbie wyprowadzeń od ośmiu (grupa ST7UltraLite) do 32 (grupa ST7FLITE4M). Większość mikrokontrolerów ma interfejs SPI (za wyjątkiem ST7UltraLite i ST4FLITE4M, który zawiera zamiast interfejsu SPI interfejs I²C) oraz 10-bitowy przetwornik analogowo-cyfrowy (za wyjątkiem mikrokontrolerów ST7LITE0/SUPERLITE, których wewnętrzny przetwornik ma rozdzielczość ośmiu bitów).

Tab. 1.1. Podstawowy podział mikrokontrolerów ST7Lite wraz z zestawieniem najważniejszych wewnętrznych układów peryferyjnych

Parametr	ST7 ULTRALITE	ST7 SUPERLITE	ST7 LITE0	ST7 LITE1B	ST7 LITE2	ST7 LITE3	ST7 DALI	ST7 LITE4M
Flash	1 kB, 2 kB	1 kB	1,5 kB	2 kB, 4 kB	8kB	8 kB	8 kB	4 kB
RAM	128 B	128 B	128 B	256 B	384 B	384 B	384 B	384 B
EEPROM	0 B, 128 B	0	0 B, 128 B	0 B, 128 B	0 B, 256 B	0 B, 256 B	256 B	128 B
Interfejsy	–	SPI	SPI	SPI	SPI	SPI, LINSCI	SPI, DALI	I ² C
Timer 8-bitowy	+	+	+	+	+	+	+	+
Timer 12-bitowy	+	+	+	+	+	+	+	+
Liczba wyprowadzeń	8	16	16	16, 20	20	20	20	32
Rozdzielczość przetwornika A/C	10 bitów	8 bitów	8 bitów	10 bitów	10 bitów	10 bitów	10 bitów	10 bitów

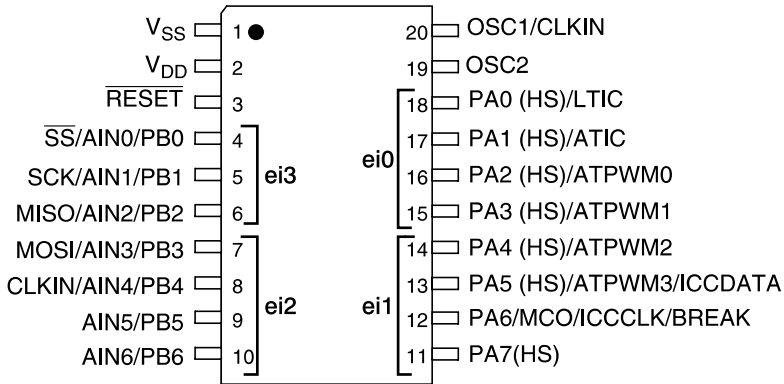
Zestawienie dostępnych wersji mikrokontrolerów ST7LITE i ich najważniejszych cech przedstawiono w **tabeli 1.1**. Jednostka centralna we wszystkich mikrokontrolerach jest 8-bitowa. Przestrzeń adresowa ma rozmiar 64 kB i obejmuje wszystkie rodzaje pamięci (Flash, SRAM, EEPROM), co jest zgodne z architekturą von Neumanna. Dzięki temu dostęp do każdego typu pamięci odbywa się przy użyciu tych samych instrukcji, co znacznie upraszcza dostęp do przechowywanych w pamięci programu stałych oraz umożliwia wykonanie programu umieszczonego zarówno w pamięci Flash, jak i RAM.

1.1. Mikrokontroler ST7FLITE19

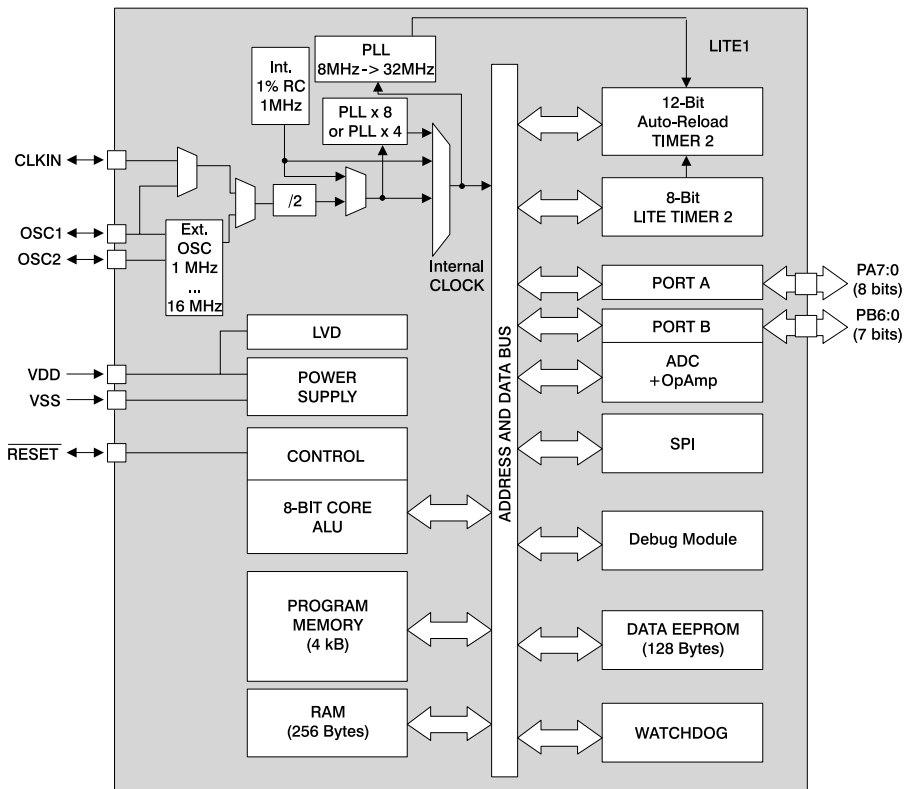
Wszystkie programy przykładowe, które zostaną przedstawione w dalszej części książki, napisano dla mikrokontrolera ST7FLITE19. Rozmieszczenie jego wyprowadzeń pokazano na **rysunku 1.1**. Schemat blokowy mikrokontrolera ST7FLITE19 widzimy na **rysunku 1.2**.

1.1.1. Mapa pamięci mikrokontrolera ST7FLITE19

Mapę pamięci mikrokontrolera ST7FLITE19 zamieszczono na **rysunku 1.3**. Jak widać, wszystkie dostępne obszary pamięci są zlokalizowane w jednej, ciągłej

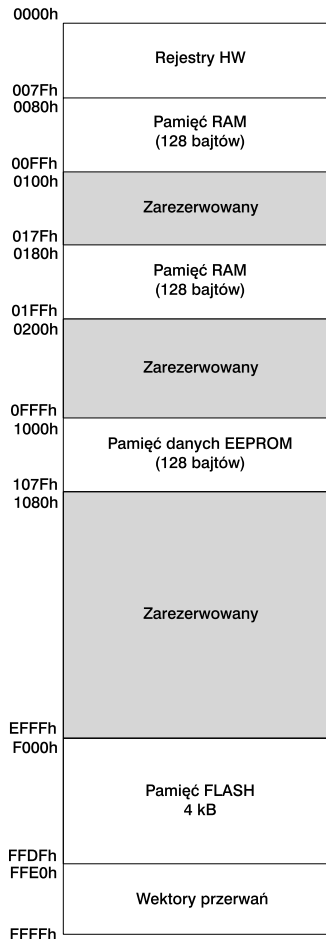


Rys. 1.1. Rozmieszczenie wyprowadzeń mikrokontrolera ST7FLITE19



Rys. 1.2. Schemat blokowy mikrokontrolera ST7FLITE19

przestrzeni adresowej. Na początku tej przestrzeni jest zlokalizowany obszar rejestrów sprzętowych. Ich wykaz zawarto w **tabeli 1.2**. Do dyspozycji programisty jest dostępnych 256 bajtów pamięci RAM, z czego 128 bajtów, znajdujących się w przestrzeni o adresach 80h...FFh, tworzy tzw. stronę zerową, do której dostęp jest możliwy przy zastosowaniu adresowania 8-bitowego, co skraca czas wykony-



Rys. 1.3. Mapa pamięci mikrokontrolera ST7FLITE19

wania programu. Obszar ten jest wykorzystywany jako pamięć danych programu, natomiast druga część pamięci RAM, o adresach 180h...1FFh, jest przeznaczona na stos. Mikrokontroler ST7FLITE19 wyposażono również w 128 bajtów pamięci EEPROM ulokowanej pod adresem 1000h. Pamięć ta cechuje się trwałością 300 tysięcy cykli zapisu/kasowania przy gwarantowanym okresie zachowania danych przez 20 lat w temperaturze 55°C.

Pamięć programu Flash jest ulokowana od adresu F000h i zajmuje 4 kB. Obszar ten podzielono na dwa sektory: 0 i 1, przy czym rozmiar sektorów jest konfigurowany za pomocą bajtów konfiguracyjnych *Option Bytes* (omówimy je w dalszej części rozdziału). Podstawową różnicą pomiędzy sektorami 0 a 1 jest fakt, że sektor 0 można programować wyłącznie programatorem zewnętrznym ICP (ang. *In Circuit Programming*), natomiast sektor 1 może być programowany z poziomu uruchomionej aplikacji IAP (ang. *In Application Programming*). Dzięki temu możliwe jest modyfikowanie zawartości sektora 1 pamięci programu przez uruchomiony pro-

Tab. 1.2. Rejestry mikrokontrolera ST7FLITE19

Adres	Układ	Etykieta	Nazwa	Stan po zerowaniu	Uwagi	
0000h	Port A	PADR	Port A Data Register	FFh	R/W	
0001h		PADDR	Port A Data Direction Register	00h	R/W	
0002h		PAOR	Port A Option Register	40h	R/W	
0003h	Port B	PBDR	Port B Data Register	FFh	R/W	
0004h		PBDDR	Port B Data Direction Register	00h	R/W	
0005h		PBOR	Port B Option Register	00h	R/W	
0006h	Obszar zarezerwowany (2 bajty)					
0007h						
0008h	Lite Timer 2	LTCISR2	Lite Timer Control/Status Register 2	0Fh	R/W	
0009h		LTARR	Lite Timer Auto-reload Register	00h	R/W	
000Ah		LTCNTR	Lite Timer Counter Register	00h	R	
000Bh		LTCISR1	Lite Timer Control/Status Register 1	0x00 0000b	R/W	
000Ch		LTICR	Lite Timer Input Capture Register	xxh	R	
000Dh	Auto-Reload Timer 2	ATCSR	Timer Control/Status Register	0x00 0000b	R/W	
000Eh		CNTRH	Counter Register High	00h	R	
000Fh		CNTRL	Counter Register Low	00h	R	
0010h		ATRH	Auto-Reload Register High	00h	R/W	
0011h		ATRL	Auto-Reload Register Low	00h	R/W	
0012h		PWMCR	PWM Output Control Register	00h	R/W	
0013h		PWM0CSR	PWM0 Control/Status Register	00h	R/W	
0014h		PWM1CSR	PWM1 Control/Status Register	00h	R/W	
0015h		PWM2CSR	PWM2 Control/Status Register	00h	R/W	
0016h		PWM3CSR	PWM3 Control/Status Register	00h	R/W	
0017h		DCR0H	PWM0 Duty Cycle Register High	00h	R/W	
0018h		DCR0L	PWM0 Duty Cycle Register Low	00h	R/W	
0019h		DCR1H	PWM1 Duty Cycle Register High	00h	R/W	
001Ah		DCR1L	PWM1 Duty Cycle Register Low	00h	R/W	
001Bh		DCR2H	PWM2 Duty Cycle Register High	00h	R/W	
001Ch		DCR2L	PWM2 Duty Cycle Register Low	00h	R/W	
001Dh		DCR3H	PWM3 Duty Cycle Register High	00h	R/W	
001Eh		DCR3L	PWM3 Duty Cycle Register Low	00h	R/W	
001Fh		ATICRH	Input Capture Register High	00h	R	
0020h		ATICRL	Input Capture Register Low	00h	R	
0021h		TRANCR	Transfer Control Register	01h	R/W	
0022h		BREAKCR	Break Control Register	00h	R/W	
0023h		Obszar zarezerwowany (11 bajtów)				
-						
002Dh						
002Eh		Watchdog	WDGCR	Watchdog Control Register	7Fh	R/W
002Fh		Flash	FCSR	Flash Control/Status Register	00h	R/W
0030h		EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
0031h		SPI	SPIDR	SPI Data I/O Register	xxh	R/W
0032h			SPICR	SPI Control Register	0xh	R/W
0033h		SPICSR	SPI Control Status Register	00h	R/W	
0034h	ADC	ADCCSR	A/D Control Status Register	00h	R/W	
0035h		ADCDRH	A/D Data Register High	xxh	R	

Tab. 1.2. Rejestry mikrokontrolera ST7FLITE19 (cd.)

0036h		ADC DRL	A/D Amplifier Control/Data Low Register	0xh	R/W
0037h	ITC	EICR	External Interrupt Control Register	00h	R/W
0038h	MCC	MCCSR	Main Clock Control/Status Register	00h	R/W
0039h	Clock and Reset	RCCR	RC oscillator Control Register	FFh	R/W
003Ah		SICSR	System Integrity Control/Status Register	0000 0xx0b	R/W
003Bh	Obszar zarezerwowany (1 bajt)				
003Ch	ITC	EISR	External Interrupt Selection Register	0Ch	R/W
003Dh	Obszar zarezerwowany (12 bajtów)				
–					
0048h					
0049h	AWU	AWUPR	AWU Prescaler Register	FFh	R/W
004Ah		AWUCSR	AWU Control/Status Register	00h	R/W
004Bh	Debug Module	DMCSR	DM Control Register	00h	R/W
004Ch		DMSR	DM Status Register	00h	R/W
004Dh		DMBK1H	DM Breakpoint Register 1 High	00h	R/W
004Eh		DMBK1L	DM Breakpoint Register 1 Low	00h	R/W
004Fh		DMBK2H	DM Breakpoint Register 2 High	00h	R/W
0050h		DMBK2L	DM Breakpoint Register 2 Low	00h	R/W
0051h	Obszar zarezerwowany (47 bajtów)				
–					
007Fh					

gram. Właściwość tę można wykorzystać do uaktualnienia oprogramowania bądź też do ładowania kodu programu inną drogą niż za pomocą programatora ICP.

Ostatnie 32 bajty obszaru pamięci Flash zajmuje tablica wektorów przerwań. W tablicy tej należy umieścić adresy procedur obsługi poszczególnych przerwań. W obszarze tym nie wolno umieszczać żadnych instrukcji, co często czynią początkujący użytkownicy mikrokontrolerów ST7, a co jest poważnym błędem. Wykaz poszczególnych prze-

Tab. 1.3. Przerwania oraz adresy wektorów przerwań w mikrokontrolerze ST7FLITE19

Numer	Źródło	Opis	Rejestr	Priorytet	Adres wektora
–	RESET	Reset	–	Najwyższy	FFFEh - FFFFh
–	TRAP	Przerwanie programowe			FFCCh - FFFDh
0	AWU	Przerwanie od układu Auto Wake-Up	AWUCSR		FFFAh - FFFBh
1	ei0	Przerwanie zewnętrzne EI0	–		FFF8h - FFF9h
2	ei1	Przerwanie zewnętrzne EI1			FFF6h - FFF7h
3	ei2	Przerwanie zewnętrzne EI2			FFF4h - FFF5h
4	ei3	Przerwanie zewnętrzne EI3			FFF2h - FFF3h
5	Lite Timer	Przerwanie od Lite Timer RCT2	LTCSR		FFF0h - FFF1h
6	–	Nie używane	–		FFEEh - FFEFh
7	SI	Przerwanie od układu AVD	SICSR		FFCCh - FFEDh
8	AT Timer	Przerwanie od Output Compare lub Input Capture	PWMxCSR/ATCSR		FFFAh - FFFBh
9	AT Timer	Przerwanie od przepelnienia timera	ATCSR		FFE8h - FFE9h
10	Lite Timer	Przerwanie od Input Capture	LTCSR		FFE6h - FFE7h
11	Lite Timer	Przerwanie od Lite Timer RTC1	LTCSR		FFE4h - FFE5h
12	SPI	Przerwanie od układu SPI	SPICSR		FFE2h - FFE3h
13	–		–	Najniższy	FFE0h - FFE1h

rwań wraz z adresami poszczególnych wektorów przerwań przedstawiono w **tabeli 1.3**. Komórki pamięci o adresach FFFEh...FFFFh są przeznaczone na wektor zerowania i koniecznie należy umieścić tam adres pierwszej instrukcji wykonywanego programu. W przeciwnym razie program nie zostanie prawidłowo uruchomiony.

Dwie pierwsze komórki pamięci EEPROM oraz dwie ostatnie komórki obszaru pamięci programu Flash (tuż przed tablicą wektorów przerwań) są fabrycznie zaprogramowane danymi kalibracji generatora RC wbudowanego w mikrokontroler. Przy taktowaniu mikrokontrolera z wewnętrznego generatora RC, jeśli zależy nam na uzyskaniu wysokiej dokładności generowanej częstotliwości, dane kalibracyjne należy zapisać do rejestru RCCR. Dane kalibracyjne są przygotowane dla dwóch wartości napięcia zasilającego: 5 V (bajt przechowywany pod adresem 1000h i FFDEh) oraz 3 V (1001h i FDFh), w obu przypadkach przy temperaturze otoczenia 25°C. Dla napięcia 5 V częstotliwość generatora RC wynosi 1 MHz, natomiast dla napięcia 3 V – 700 kHz.

1.1.2. Lista instrukcji mikrokontrolera ST7FLITE19

Mikrokontroler ST7FLITE19, podobnie jak każdy inny mikrokontroler z rodziny ST7, obsługuje 63 instrukcje. Ich zestawienie i opis znajduje się w Dodatku. Szczegółowy opis wszystkich instrukcji oraz dostępnych trybów adresowania wraz z przykładami można znaleźć w dokumencie „ST7 Programming Manual” dostępnym na stronie www.stmcu.com.

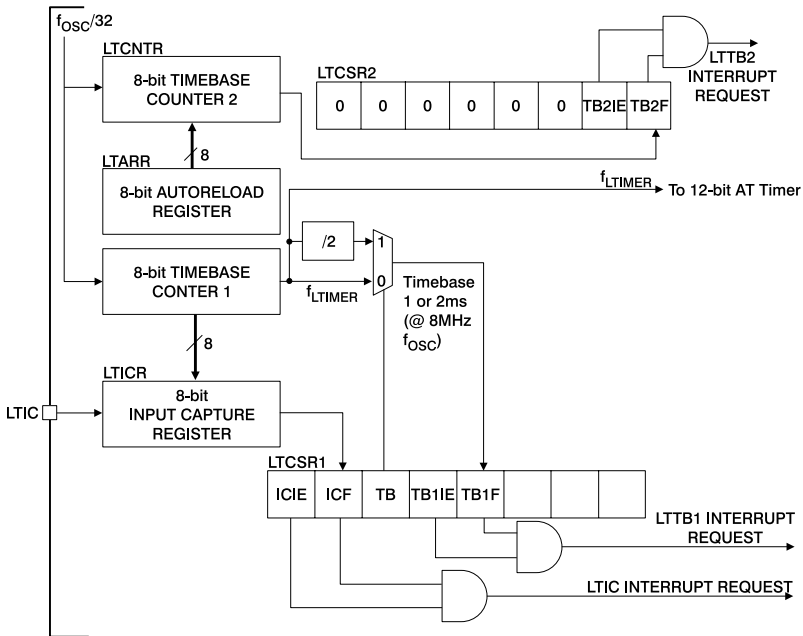
1.1.3. Układy peryferyjne mikrokontrolera ST7FLITE19

Najważniejszą częścią każdego mikrokontrolera, poza jednostką centralną (CPU), są wbudowane układy peryferyjne. Ich liczba oraz stopień skomplikowania zależy od typu mikrokontrolera, w którym występują. Podstawową cechą wszystkich układów peryferyjnych spotykanych w mikrokontrolerach ST7LITE jest ich prosta budowa i stosunkowo łatwa obsługa. W tym podrozdziale skrótowo zostaną omówione najważniejsze układy peryferyjne mikrokontrolera ST7FLITE19. Ich szczegółowy opis znajduje się w dokumentacji mikrokontrolera, która powinna być obowiązkową lekturą podczas pisania programów. Przykłady obrazujące korzystanie z większości opisanych tutaj układów są zawarte w rozdziałach 3 i 4.

Lite Timer

Budowę Lite Timera pokazano na **rysunku 1.4**. Składa się on z dwóch 8-bitowych liczników liczących w górę. Są one taktowane przebiegiem o stałej częstotliwości, uzyskiwanej w wyniku podziału przez 32 częstotliwości sygnału taktującego CPU. Licznik *Timebase Counter 1* liczy od 0 do 249. W momencie przejścia z wartości 249 do 0 występuje zdarzenie przepełnienia licznika. W przypadku taktowania jednostki centralnej sygnałem o częstotliwości 8 MHz zdarzenie to występuje dokładnie co 1 ms. Czas ten może zostać dwukrotnie wydłużony poprzez ustawienie bitu TB znajdującego się w rejestrze LTCSR1.

Przepełnienie licznika powoduje ustawienie flagi TB1F. W przypadku, gdy bit TB1IE znajdujący się w rejestrze LTCSR1 będzie ustawiony (oraz flaga globalnego zezwolenia na przerwanie będzie wyzerowana), wystąpi wywołanie procedury obsługi przerwania. Flaga TB1F musi zostać wyzerowana programowo po-

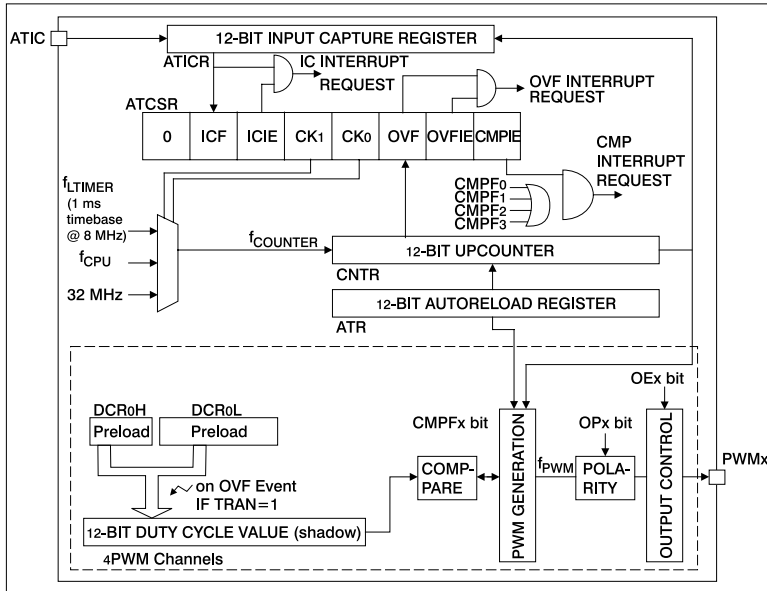


Rys. 1.4. Schemat blokowy Lite Timera

przez odczyt wartości rejestru LTCR1, w przeciwnym razie przerwanie zostanie potraktowane jako nieobsłużone i po opuszczeniu procedury obsługi przerwania nastąpi natychmiastowe ponowne jej wywołanie. Licznik *Timebase Counter 1* ma możliwość przechwycenia jego zawartości w momencie wystąpienia na wejściu LTIC opadającego lub narastającego zbocza. Zdarzenie przechwycenia zawartości licznika powoduje ustawienie flagi ICF znajdującej się w rejestrze LTCR1. Jeśli bit ICIE, znajdujący się również w rejestrze LTCR1, jest ustawiony, to nastąpi wywołanie procedury obsługi przerwania od zdarzenia przechwycenia zawartości licznika. Działanie licznika *Timebase Counter 2* jest podobne do działania licznika *Timebase Counter 1* z tym wyjątkiem, że istnieje możliwość ustalenia wartości, od której licznik rozpoczyna zliczanie, co jest równoznaczne z regulacją częstotliwości występowania zdarzenia przepełnienia licznika. Można tego dokonać poprzez zapis wartości początkowej do rejestru LTARR. Zawartość tego rejestru jest automatycznie ładowana do licznika przy każdym jego przepełnieniu, które następuje w momencie przejścia z wartości 255 do wartości określonej zawartością rejestru LTARR. Zdarzenie przepełnienia licznika *Timebase Counter 2* powoduje ustawienie flagi TB2F, która jest w rejestrze LTCR2. W rejestrze tym znajduje się również bit zezwolenia na przerwanie od przepełnienia licznika TB2IE.

Timer AT2

Dwunastobitowy licznik AT2 jest najbardziej rozbudowanym układem peryferyjnym mikrokontrolera ST7FLITE19. Jego budowę pokazano na **rysunku 1.5**. Głównym elementem timera jest 12-bitowy licznik zliczający w górę.

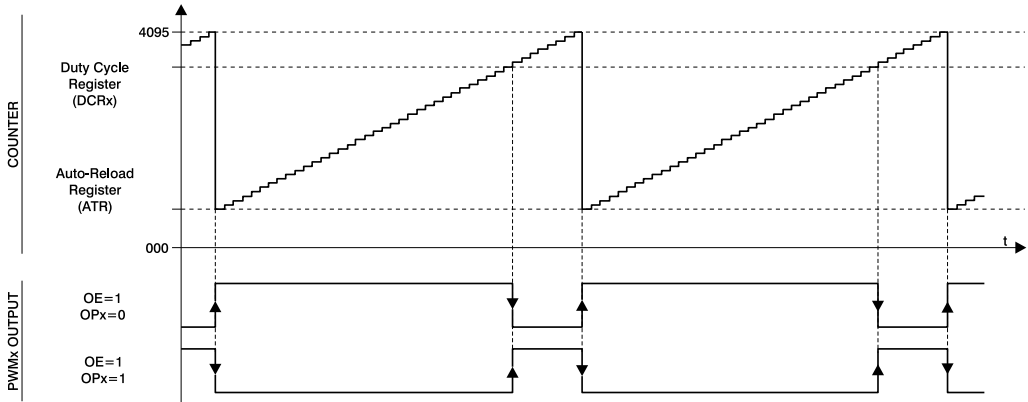


Rys. 1.5. Schemat blokowy timera AT2

Dostępne są trzy źródła taktowania licznika: przebieg taktujący jednostkę centralną, przebieg podstawy czasu licznika Lite Timer oraz przebieg o częstotliwości maksymalnej 32 MHz uzyskiwany przez pomnożenie przez 4 częstotliwości taktującej jednostkę centralną. Wynika z tego, iż częstotliwość 32 MHz jest możliwa do uzyskania tylko przy taktowaniu jednostki centralnej sygnałem o częstotliwości 8 MHz. Należy pamiętać, iż każdy sygnał taktujący podawany z zewnątrz mikrokontrolera jest zawsze dzielony przez 2 przed podaniem go do jednostki centralnej i układów peryferyjnych.

W wyniku przepełnienia licznika AT2 jest ustawiana flaga OVF, która znajduje się w rejestrze ATCSR. W przypadku, gdy bit OVFIE, który również znajduje się w rejestrze ATCSR, jest ustawiony, nastąpi wywołanie procedury obsługi przerwania od przepełnienia licznika. Licznik AT2 ma funkcję automatycznego ładowania wartości początkowej. Przeładowanie następuje zawsze po przepełnieniu licznika. Zawartość rejestrów licznika można tylko odczytać, nie ma możliwości bezpośredniego zapisania do licznika żadnej wartości. W związku z tym po wyzerowaniu mikrokontrolera licznik zawsze musi odliczyć 4096 impulsów sygnału taktującego, po których nastąpi załadowanie nowej wartości początkowej znajdującej się w rejestrach ATRH i ATRL. Licznik AT2 ma funkcję przechwytywania jego zawartości w odpowiedzi na sygnał podany z zewnątrz. Opadające lub narastające zbocze na wyprowadzeniu ATIC powoduje przepisanie stanu licznika do rejestrów ATICRH i ATICRL oraz ustawienie flagi ICF znajdującej się w rejestrze ATCSR. W przypadku, gdy bit ICIE jest ustawiony, nastąpi zgłoszenie przerwania. Układ AT2 może pełnić funkcję czterokanałowego generatora PWM, którego zasadę działania przedstawiono na **rysunku 1.6**.

W trybie PWM licznik zlicza od wartości określonej zawartością rejestru ATR do wartości maksymalnej, przy czym w momencie przepełnienia licznika następuje



Rys. 1.6. Przebiegi ilustrujące działanie generatora PWM w timerze AT2

ustawienie stanu wysokiego na wyprowadzeniu PWMx, natomiast w momencie zrównania się wartości licznika z wartością rejestru DCRx zachodzi wyzerowanie wyprowadzenia PWMx, przy czym licznik liczy dalej. Tak więc zawartość rejestru DCRx określa współczynnik wypełnienia sygnału generowanego na wyprowadzeniu PWMx. Wszystkie cztery kanały PWM pracują z tą samą częstotliwością f_{PWM} określoną zawartością rejestru ATR zgodnie ze wzorem:

$$f_{PWM} = \frac{f_{COUNTER}}{4096 - ATR}$$

Maksymalną wartością, jaka może zostać zapisana do rejestru ATR, jest 4094, gdyż wartość ta musi być mniejsza od wartości w rejestrze DCR, która z kolei może przyjąć maksymalną wartość 4095. Maksymalna rozdzielczość regulacji współczynnika wypełnienia jest określona wzorem:

$$R = \frac{1}{4096 - ATR}$$

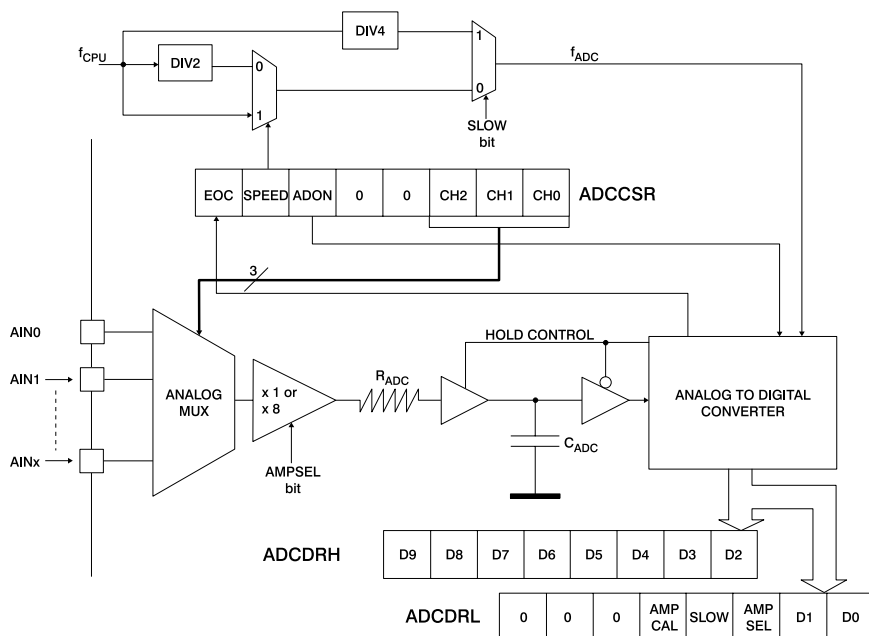
Z powyższych równań wynika, iż wraz ze wzrostem częstotliwości pracy układu PWM spada rozdzielczość regulacji współczynnika wypełnienia sygnału generowanego na wyjściach PWMx.

Przetwornik analogowo-cyfrowy

Mikrokontroler ST7FLITE19 wyposażono w 10-bitowy przetwornik analogowo-cyfrowy o siedmiu wejściach analogowych, które są podłączone do wyprowadzeń portu B. Budowę przetwornika ilustruje **rysunek 1.7**.

Częstotliwość taktowania przetwornika może zostać wybrana z trzech dostępnych wartości: F_{CPU} , $F_{CPU}/2$ i $F_{CPU}/4$. Za wybór odpowiada bit SPEED, który znajduje się w rejestrze ADCCSR oraz bit SLOW z rejestru ADCDRL.

Wyboru aktywnego wejścia multiplexera wejściowego dokonuje się za pomocą bitów CH0:CH2 znajdujących się w rejestrze ADCCSR. Bit ADON służy do włączenia przetwornika, natomiast bit EOC sygnalizuje zakończenie procesu przetwarzania napięcia na wartość cyfrową. Rejestr ADCDRL pełni podwójną funkcję: przechowuje dwa najmłodsze bity wyniku pomiaru oraz zawiera bity sterujące



Rys. 1.7. Schemat blokowy przetwornika A/C wbudowanego w mikrokontrolery ST7FLITE19

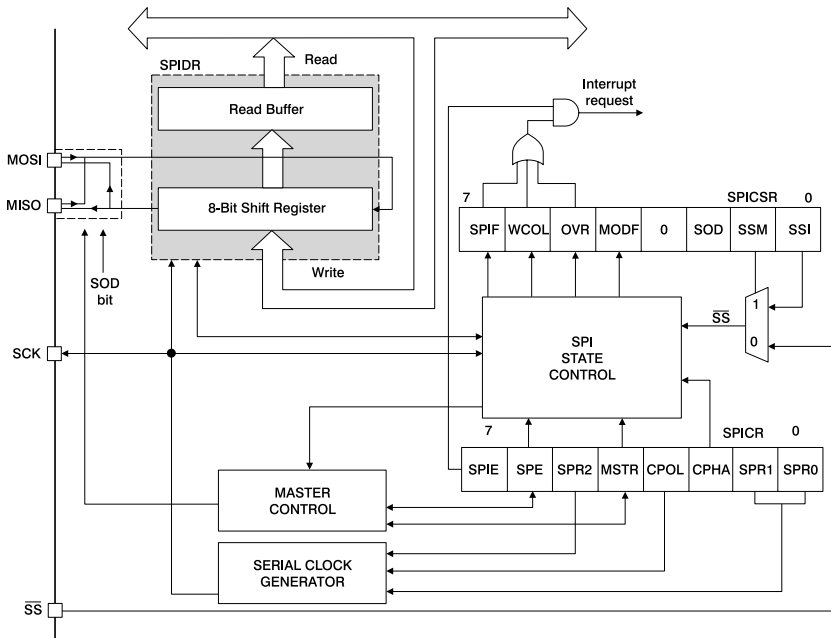
wzmacniaczem wejściowym. Wzmacniacz wejściowy jest uaktywniany ustawieniem bitu AMPSEL i umożliwia ośmiokrotne wzmocnienie sygnału wejściowego, który w tym przypadku nie może przekraczać wartości 430 mV. Ustawienie bitu AMPCAL spowoduje zwarcie jego wejścia do masy, co umożliwi odczytanie offsetu wnoszonego przez wzmacniacz i jego późniejszą programową korekcję.

Interfejs SPI

Interfejs SPI jest w większości mikrokontrolerów ST7LITE jedynym sprzętowym interfejsem komunikacyjnym. Budowę układu SPI zastosowanego w mikrokontrolerze ST7FLITE19 przedstawiono na **rysunku 1.8**.

Z interfejsem SPI są powiązane następujące rejestry:

- SPIDR – rejestr danych używany do nadawania i odbioru danych transmitowanych interfejsem SPI. Jeśli układ pracuje w trybie master, to zapis do tego rejestru rozpoczyna proces transmisji danych.
- SPICR – rejestr sterujący. Rejestr składa się z następujących bitów:
 - SPIE (bit 7). Bit zezwolenia na przerwanie od interfejsu SPI.
 - SPE (bit 6). Bit uaktywniający wyjście układu SPI na powiązanim z nim wyprowadzeniu mikrokontrolera.
 - SPR2 (bit 5). Bit dezaktywujący dzielnik częstotliwości przez 2. Wykorzystywany razem z bitami SPR0:SPR1 do określenia prędkości transmisji danych.
 - MSTR (bit 4). Bit wyboru trybu pracy interfejsu SPI: 0 – *slave*, 1 – *master*.
 - CPOL (bit 3). Bit wyboru polaryzacji sygnału zegarowego: 0 – w stanie nieaktywnym linia SCK przyjmuje stan niski; 1 – w stanie nieaktywnym linia SCK przyjmuje stan wysoki.



Rys. 1.8. Schemat blokowy interfejsu SPI w mikrokontrolerze ST7FLITE19

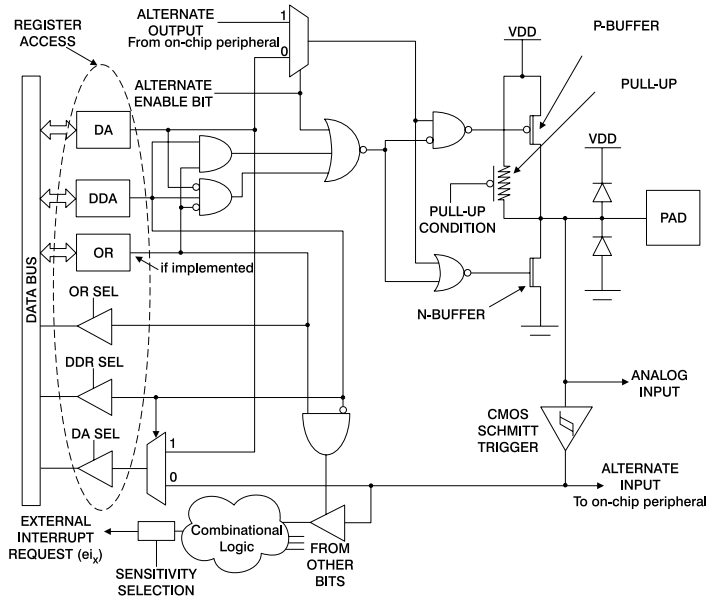
- CPHA (bit 2). Bit wyboru fazy sygnału zegarowego: 0 – dane są ważne przy pierwszej zmianie stanu na linii SCK: 1 – dane są ważne przy drugiej zmianie stanu na linii SCK.
 - SPR1:SPR0 (bit 1:0). Bity wyboru częstotliwości na linii SCK. Używane razem z bitem SPR2.
- SPICSR – rejestr statusowo-kontrolny. Składa się z następujących bitów:
- SPIF (bit 7). Flaga sygnalizująca zakończenie transmisji danych. Flaga jest zerowana sekwencją programową poprzez dostęp do rejestru SPICSR, a następnie odczyt bądź zapis rejestru SPIDR.
 - WCOL (bit 6). Flaga sygnalizująca kolizję danych. Jest ustawiana w przypadku zapisu do rejestru danych przed zakończeniem poprzedniej transmisji.
 - OVR (bit 5). Flaga sygnalizująca nadpisanie odebranych danych. Jest ustawiana w momencie, gdy odebrane dane są gotowe do przetransferowania z bufora do rejestru SPIDR, a poprzednio odebrany bajt nie został odczytany z rejestru SPIDR.
 - MODF (bit 4). Flaga sygnalizująca niezgodność trybu pracy. Jest ustawiana, gdy wyprowadzenie \overline{SS} znajduje się w stanie niskim podczas pracy układu w trybie master.
 - SOD (bit 2). Bit dezaktywujący wyjście układu SPI. Jeśli jest ustawiony, alternatywna funkcja wyprowadzenia mikrokontrolera jest dezaktywowana.
 - SSM (bit 1). Bit zarządzający wyprowadzeniem \overline{SS} . Jeśli SSM jest ustawiony, tryb pracy układu SPI jest konfigurowalny programowo, a wyprowadzenie mikrokontrolera, do którego przypisana jest alternatywna funkcja \overline{SS} , może pracować jako zwykle wyprowadzenie I/O.

- SSI (bit 0). Przy ustawionym bicie SSM bit SSI pełni rolę aktywacji funkcji *slave* układu SPI.

1.1.4. Porty I/O mikrokontrolera ST7FLITE19

Mikrokontroler ST7FLITE19 wyposażono w 15 wyprowadzeń mogących pełnić funkcję portów wejścia-wyjścia (*Input/Output*) ogólnego zastosowania. Wyprowadzenia portu B mogą alternatywnie pełnić funkcję wejść przetwornika analogowo-cyfrowego, natomiast wyprowadzenia portu A (oznaczone jako HS) umożliwiająysterowanie obciążenia włączanego w stanie niskim prądem o natężeniu do 20 mA (czyli umożliwiają bezpośrednie sterowanie np. diodami LED). Schemat ilustrujący budowę portu zamieszczono na **rysunku 1.9**.

Z każdym portem I/O są powiązane trzy rejestry: rejestr danych PxDR, rejestr wyboru kierunku PxDDR oraz rejestr opcji PxOR. Rejestr danych służy do zapisu bądź odczytu stanu wyprowadzeń portu, natomiast rejestry wyboru kierunku i opcji służą do konfiguracji portu. Zestawienie możliwych kombinacji i ich znaczenie dla standardowych wyprowadzeń I/O zebrano w **tabeli 1.4**.



Rys. 1.9. Budowa portu I/O w mikrokontrolerze ST7FLITE19

Tab. 1.4. Zestawienie możliwych kombinacji bitów DDR i OR oraz funkcje spełniane przez linię I/O w zależności od ich stanów

Konfiguracja	DDR	OR
Wejście „plywające”	0	0
Wejście z rezystorem podciągającym	0	1
Wyjście z otwartym drenem	1	0
Wyjście przeciwsoodne (push-pull)	1	1

Tab. 1.5. Zestawienie możliwych kombinacji bitów DDR i OR oraz funkcje spełniane przez linię I/O w przypadku uaktywnienia przerwania na linii wejściowej

Konfiguracja	DDR	OR
Wejście „plywające”	0	0
Wejście generujące przerwanie z rezystorem podciągającym	0	1
Wyjście z otwartym drenem	1	0
Wyjście przeciwsoodne	1	1

Jeżeli wyprowadzenie wybrano jako źródło przerwania zewnętrznego, to jego konfiguracja jest zgodna z przedstawioną w **tabeli 1.5**.

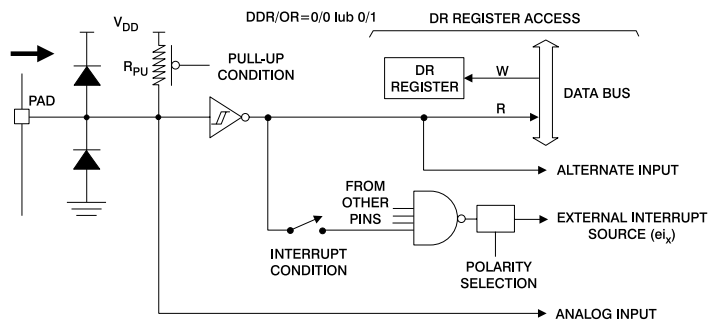
Z informacji zamieszczonych w tabeli wynika, że wyprowadzenie może być źródłem przerwania zewnętrznego wyłącznie wtedy, gdy jest skonfigurowane jako wejście z rezystorem podciągającym. Odpowiedni bit w rejestrze opcji może w tej sytuacji pełnić rolę bitu maskującego wybrane przerwianie.

Z portu B wyprowadzono na zewnątrz obudowy tylko 7 linii (PB0...PB6). Bit 7 rejestru PBDR po odczycie będzie zawsze miał wartość jedynki logicznej. Należy mieć ten fakt na uwadze przy wykonywaniu na wartości portu operacji logicznych bądź arytmetycznych.

Zarówno zapis, jak i odczyt stanu portu odbywa się poprzez ten sam rejestr danych. Wynika z tego pewna niedogodność – istnieje ryzyko niezamierzonej modyfikacji zawartości rejestru danych wyprowadzeń skonfigurowanych jako wejściowe w wyniku wykonania na rejestrze danych operacji typu odczyt-modyfikacja-zapis.

Na **rysunku 1.10** pokazano budowę portu skonfigurowanego jako wejście. Wynika z niego, że odczyt rejestru danych powoduje w rzeczywistości odczytanie stanu wyprowadzeń, a nie stanu bitów w rejestrze danych. Wykonanie teraz typowej operacji odczyt-modyfikacja-zapis, mającej na celu np. ustawienie wyprowadzenia wyjściowego portu, spowoduje przepisanie stanu linii wejściowej do rejestru danych wyprowadzenia wejściowego. W większości przypadków nie stanowi to żadnego problemu, gdyż zawartość rejestru danych nie wpływa na pracę wyprowadzenia wejściowego. Problemy mogą się natomiast pojawić w sytuacji, gdy kierunek pracy wyprowadzenia jest zmieniany podczas działania programu. Należy wówczas zadbać, aby w rejestrze danych znajdował się zawsze stan wymagany dla trybu wyjściowego. Można to uzyskać na dwa sposoby: pierwszym z nich jest unikanie typowych operacji typu odczyt-modyfikacja-zapis, jak np. korzystanie z instrukcji BRES/BSET oraz stosowanie w celu modyfikacji stanu wyprowadzeń wyjściowych odpowiednio rozszerzonych operacji odczyt-modyfikacja-zapis. Rozszerzenie to musi za każdym razem wpisywać wymaganą wartość do odpowiedniego bitu rejestru danych. Drugim sposobem jest swobodne korzystanie z instrukcji BSET/BRES i ustawienie wymaganej wartości bitu w rejestrze danych dopiero przed zmianą kierunku pracy z wejścia na wyjście.

Co prawda firma STMicroelectronics nie zaleca stosowania instrukcji BSET/BRES do modyfikacji stanu rejestru danych, ale ze względu na wygodę używania tych



Rys. 1.10. Budowa portu I/O skonfigurowanego jako wejście

instrukcji uważam, iż rozsądne ich wykorzystanie nie powinno sprawiać żadnych nieoczekiwanych problemów.

1.1.5. Bajty konfiguracyjne (Option Bytes)

Mikrokontroler ST7FLITE19, podobnie jak większość współczesnych mikrokontrolerów, ma specjalne bity służące do sprzętowej konfiguracji mikrokontrolera. Za ich pomocą można wybrać źródło i zakres częstotliwości sygnału taktującego, włączyć wbudowany oscylator RC czy też układ watchdog. Bajty konfiguracyjne są dostępne wyłącznie w trybie programowania mikrokontrolera za pomocą typowych narzędzi ICP (np. programatora ST7-STICK). Nie jest natomiast możliwy dostęp do bajtów konfiguracyjnych z poziomu uruchomionego programu.

OPTION BYTE 0

Pierwszy z bajtów konfiguracyjnych składa się z następujących bitów:

OPT7 = zarezerwowany, zawsze musi mieć wartość 1.

OPT[6:4] = OSCRANGE[2:0]

Bity te określają źródło i zakres (w przypadku zewnętrznego oscylatora kwarcowego) sygnału taktującego mikrokontroler.

OSCRANGE			Źródło i zakres sygnału taktującego
2	1	0	
0	0	0	Zewnętrzny kwarc 1...2 MHz
0	0	1	Zewnętrzny kwarc 2...4 MHz
0	1	0	Zewnętrzny kwarc 4...8 MHz
0	1	1	Zewnętrzny kwarc 8...16 MHz
1	0	0	Zewnętrzny kwarc 32768 Hz
1	0	1	Sygnal zegarowy na wyprowadzeniu OSC1
1	1	1	Sygnal zegarowy na wyprowadzeniu PB4
1	1	0	Zarezerwowane

OPT[3:2] = SEC[1:0]

Bity określają rozmiar sektora 0 pamięci Flash.

SEC1	SEC0	Rozmiar sektora 0
0	0	512 B
0	1	1 kB
1	0	2 kB
1	1	4 kB

OPT1 = FMP_R

Bit blokujący możliwość odczytu zawartości pamięci programu Flash. Skasowanie tego bitu spowoduje wcześniejsze skasowanie całej pamięci programu Flash.

OPT0 = FMP_W

Bit blokujący możliwość zapisu pamięci programu Flash.



Stanu tego bitu po zaprogramowaniu nie można zmienić. Po jego zaprogramowaniu zawartość pamięci Flash nigdy więcej nie będzie mogła być zmieniana, również bajt konfiguracyjny nie będzie mógł być modyfikowany.

OPTION BYTE 1

Składa się on z następujących bitów:

OPT7 = PLLx4x8

Bit odpowiedzialny za wybór mnożnika układu PLL

0: PLLx4

1: PLLx8

OPT6 = PLL0FF

Bit uaktywniający układ PLL

0: układ PLL aktywny

1: układ PLL nieaktywny

OPT5 = PLL32OFF

Bit uaktywniający układ PLL 32 MHz

0: układ PLL32 aktywny

1: układ PLL32 nieaktywny

OPT4 = OSC

Bit uaktywniający generator RC

0: generator RC włączony

1: generator RC wyłączony



Generator RC o dokładności generowanej częstotliwości 1% jest dostępny wyłącznie w mikrokontrolerach ST7FLITE15 i ST7FLITE19.

OPT3:2 = LVD[1:0]

Bity odpowiedzialne za uaktywnienie układu LVD i wybór poziomu jego zadziałania.

LVD1	LVD0	Konfiguracja
1	1	Układ LVD wyłączony
1	0	Najwyższy poziom (ok. 4,1 V)
0	1	Średni poziom (ok. 3,5 V)
0	0	Najniższy poziom (ok. 2,8 V)

OPT1 = WDG SW

Bit określający sposób aktywacji układu watchdog

0: sprzętowy (układ watchdog stale aktywny)

1: programowy (układ watchdog aktywowany programowo)

OPT0 = WDG HALT

Bit określający reakcję na sygnał zerujący od układu watchdog w trybie Halt

0: sygnał zerujący nie jest generowany w trybie Halt

1: sygnał zerujący jest generowany w trybie Halt

Domyślne wartości bajtów konfiguracyjnych przedstawiono na **rysunku 1.11**.

Wartość domyślna	OPTION BYTE 0								OPTION BYTE 1							
	7				0				7				0			
	Res.	OSCRANGE 2:0			SEC1	SEC0	FMP R	FMP W	PLL x4x8	PLL OFF	PLL32 OFF	OSC	LVD1	LVD0	WDG SW	WDG HALT
	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1

Rys. 1.11. Domyślne stany bitów w bajtach konfiguracyjnych

1.2. Moduł LITEcomp – podstawowe informacje

Bohater tej książki – moduł LITEcomp – jest prostym komputerkiem zbudowanym na bazie mikrokontrolera ST7FLITE19. Oprócz mikrokontrolera na płycie drukowanej modułu umieszczono podstawowe elementy zewnętrzne umożliwiające uruchomienie urządzenia.

Standardowe wyposażenie modułu jest następujące:

- mikrokontroler ST7FLITE19;
- rezonator kwarcowy 16 MHz;
- złącze programatora ICSP (np. ZL17PRG);
- złącze wyświetlacza LCD ze sterownikiem HD44780;
- złącze USB umożliwiające zasilanie modułu;
- dwa przyciski;
- dioda LED sygnalizująca obecność napięcia zasilającego;
- złącza szpilkowe z wyprowadzonymi wszystkimi liniami I/O mikrokontrolera;
- złącza szpilkowe z wyprowadzonymi liniami zasilania;
- zworki konfiguracyjne.

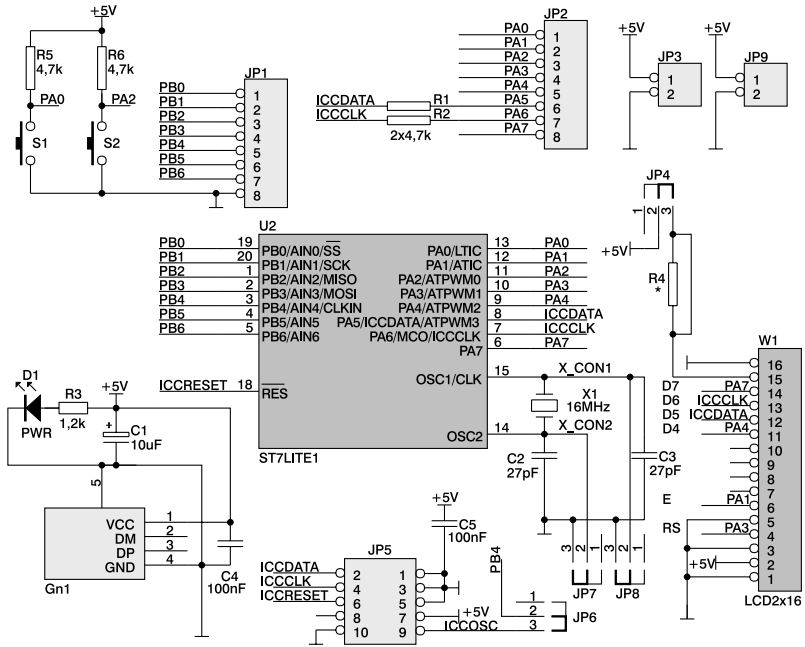
Schemat LITEcompa jest widoczny na **rysunku 1.12**. Rozmieszczenie elementów systemu na płycie drukowanej przedstawiono na **rysunku 1.13**.

Złącze USB jest wykorzystywane wyłącznie do dostarczenia napięcia zasilającego, nie jest więc możliwa wymiana danych poprzez interfejs USB. W przypadku zasilania modułu ze złącza USB należy zachować szczególną ostrożność przy korzystaniu z wyświetlacza LCD. Niektóre modele wyświetlaczy mają odsłonięte punkty lutownicze, na których występuje napięcie zasilania, dokładnie nad metalową obudową złącza USB, połączoną z masą układu. Przypadkowe dociśnięcie takiego wyświetlacza do metalowej obudowy złącza USB może spowodować zwarcie, które może zakończyć się uszkodzeniem płyty głównej komputera bądź wyzerowaniem komputera. Aby się przed tym ustrzec, można okleić obudowę złącza USB taśmą izolacyjną. Do zasilania modułu można wykorzystać również złącza JP3 i JP9 (**rysunek 1.14**). Złącza te mogą również dostarczać zasilanie do zewnętrznych układów dołączanych do modułu LITEcomp (przy zasilaniu modułu z USB).

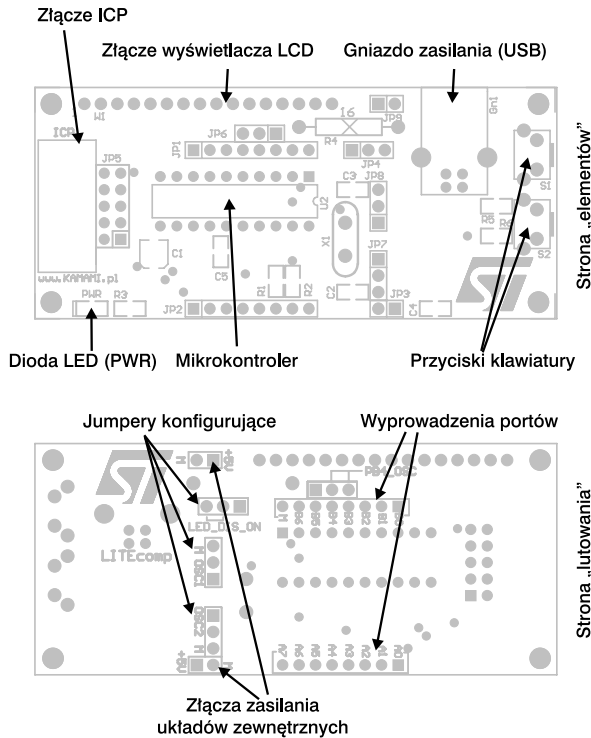
Zaprogramowanie pamięci mikrokontrolera umieszczonego w podstawie modułu LITEcomp jest możliwe dzięki złączu ICP (**rysunek 1.15**).

UWAGA

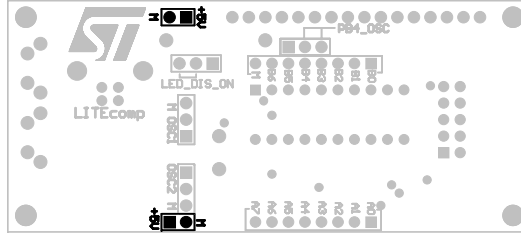
W przypadku programowania pamięci Flash mikrokontrolera zastosowanego w komputerku LITEcomp za pomocą programatora ST7-STICK (lub jego odpowiednika) uruchomienie programu jest możliwe wyłącznie po odłączeniu wtyczki programatora od gniazda ICP.



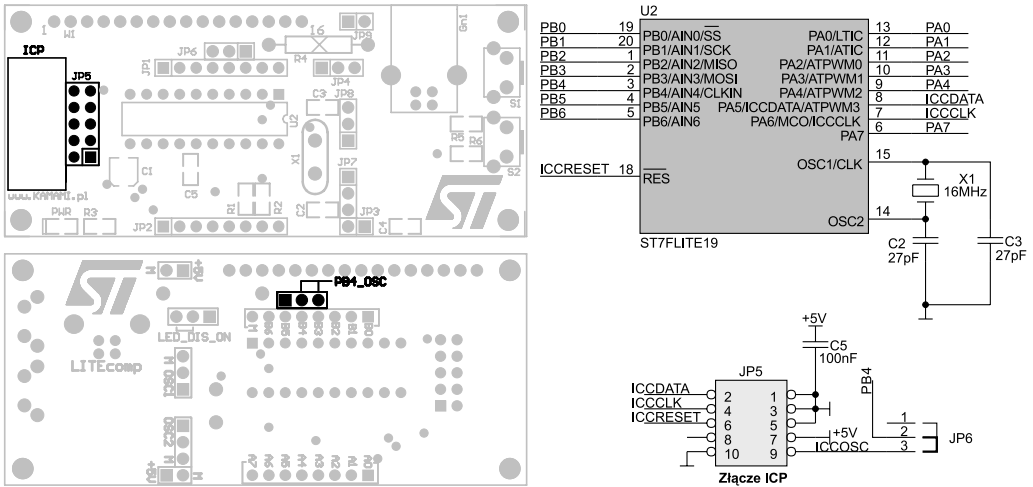
Rys. 1.12. Schemat elektryczny LITEcompa



Rys. 1.13. Rozmieszczenie najważniejszych elementów na płycie LITEcompa



Rys. 1.14. Rozmieszczenie dodatkowych złączy zasilających na płytce LITEcompa

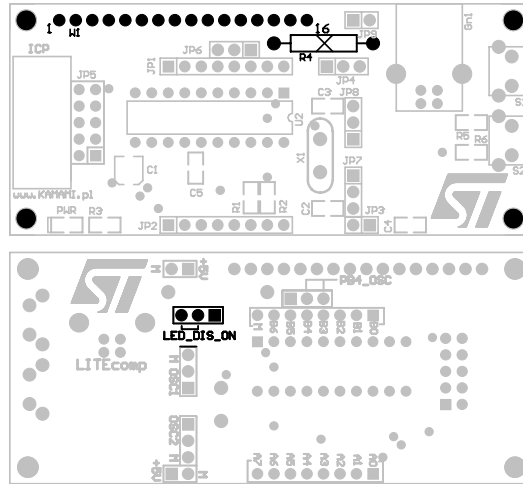


Rys. 1.15. Umieszczenie na płytce LITEcompa złącza ICP oraz zworki umożliwiającej dołączenie do mikrokontrolera zewnętrznego sygnału zegarowego

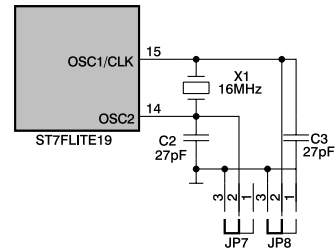
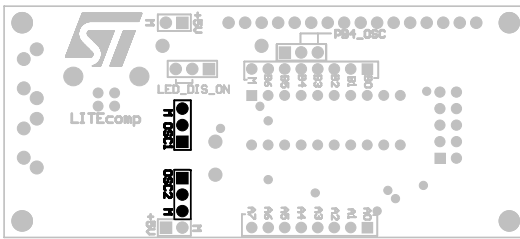
Na płytce drukowanej LITEcompa znajduje się również złącze alfanumerycznego wyświetlacza LCD 2×16 znaków (**rysunek 1.16**) umożliwiający podłączenie wyświetlacza za pomocą typowej listwy kołkowej (goldpinów). Wyświetlacz nie wchodzi w skład zestawu i należy go kupić oddzielnie. LITEcomp nie został wyposażony w potencjometr regulacji kontrastu wyświetlacza LCD – wyprowadzenie regulacji kontrastu jest na stałe podłączone do masy zasilania. Może to spowodować, że w przypadku niektórych modułów LCD nie będzie możliwe uzyskanie dobrej widoczności wyświetlanych znaków.

Zworki konfiguracyjne

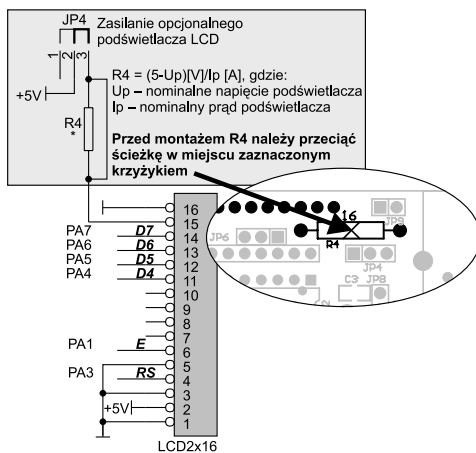
Moduł LITEcomp wyposażono w kilka zwerek umożliwiających konfigurację modułu. Zworka JP6 (rysunek 1.14) służy do dołączenia do wyprowadzenia PB4 sygnału taktującego wytwarzanego przez programator (istotne wyłącznie podczas programowania pamięci Flash w systemie). Jest to przydatne w sytuacji, gdy omyłkowo skonfigurujemy mikrokontroler do pracy ze źródłem sygnału zegarowego, które nie jest dostępne lub nie pracuje poprawnie. W tej sytuacji należy skorzystać z trybu programowania *ICP OPT Disable*.



Rys. 1.16. Umieszczenie na płycie LITEcompa złącza modułu LCD oraz zworki dołączającej napięcie zasilające do podświetlacza modułu



Rys. 1.17. Umieszczenie na płycie LITEcompa zwerek JP7 i JP8, odpowiadających za konfigurację wyprowadzeń generatora taktującego mikrokontroler



Rys. 1.18. Schemat elektryczny ilustrujący sposób dołączenia opcjonalnego rezystora ograniczającego prąd zasilający podświetlacz modułu LCD

Zworki JP7 i JP8 (**rysunek 1.17**) służą do dołączenia wyprowadzeń OSC1 i OSC2 do masy w przypadku korzystania z wewnętrznego generatora RC.

Zwórka JP4 (**rysunek 1.16**) służy do włączenia podświetlenia wyświetlacza LCD. W zależności od konstrukcji układu podświetlania wyświetlacza LCD może zaistnieć konieczność zastosowania rezystora ograniczającego prąd podświetlacza. Na płycie drukowanej modułu przewidziano miejsce na rezystor R4 ograniczający prąd podświetlacza. Miejsce pod ten rezystor zlokalizowane jest tuż przy złączu wyświetlacza LCD. Przed montażem tego rezystora konieczne jest przecięcie ścieżki w miejscu oznaczonym krzyżykiem (**rysunek 1.18**).