

Управление выводами (портами) MSP430F5529

- **Порты P1, P2, P3,...,P8, PJ управляют выводами контроллера. Выводы программируются либо как I/O, либо как вход/выход периферии**
- **Порты объединяются в пары: P1 и P2 = PA, P3 и P4 = PB, P5 и P6 = PC, P7 и P8 = PD.**
- **Для порта могут быть доступны регистры:**
 - **RxIN – чтение данных с вывода**
 - **RxOUT – установка значения выхода**
 - **RxDIR – выбор направления: 0 – вход, 1 – выход**
 - **RxREN – разрешение подтягивающего резистора**
 - **RxDS – выбор допустимой силы вывода**

Управление выводами (портами) MSP430F5529

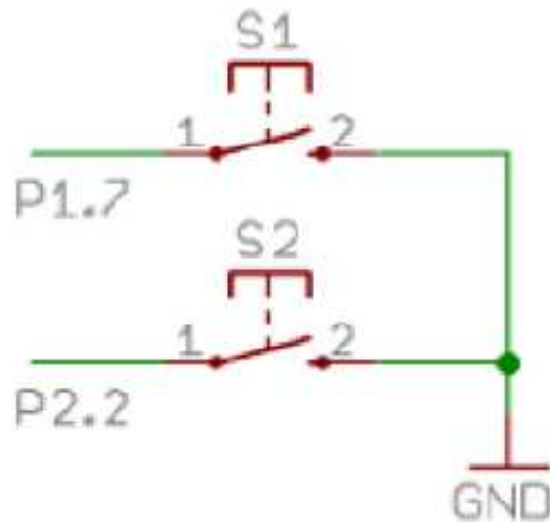
- ***PxSEL*** – выбор функции вывода: 0 – I/O, 1 – периферия
- ***PxIV*** – генерирует число для изменения PC, соответствующее прерыванию с макс. приоритетом
- ***PxIES*** – выбор направления перепада для генерации запроса на прерывание: 0 – по фронту, 1 – по спаду
- ***PxIE*** – разрешение прерывания
- ***PxIFG*** – флаг прерывания
- При работе с прерываниями порты не объединяются в пары

Управление выводами (портами) MSP430F5529

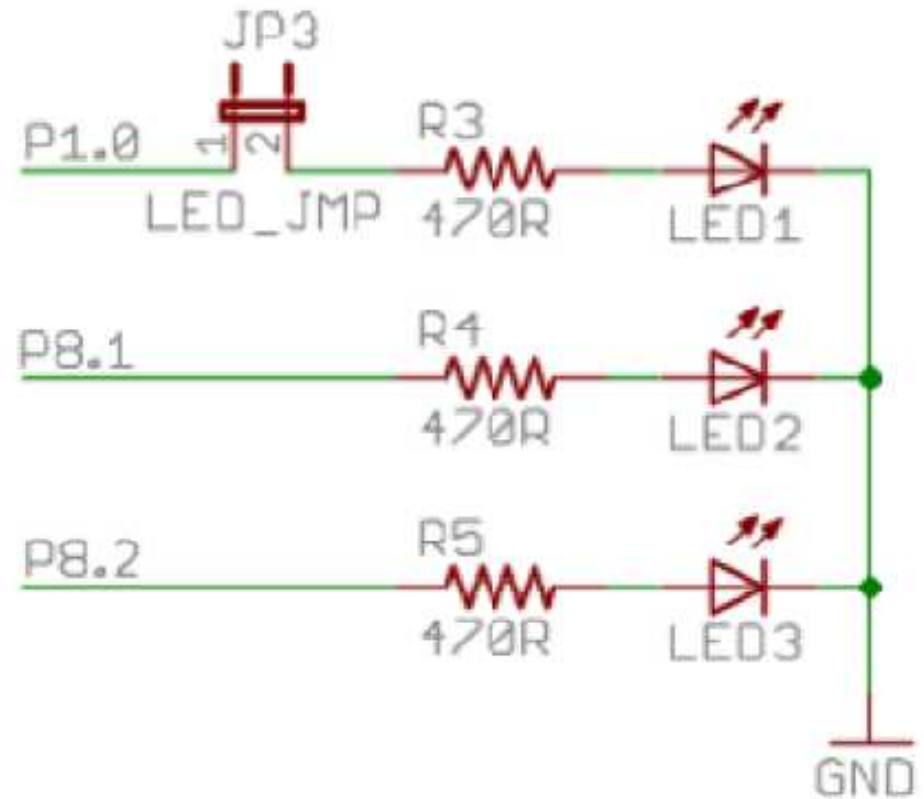
№ порта	1	2	3	4	5	6	7	8	J
База	0200h		0220h		0240h		0260h		0320h
<i>PxIN</i>	0	1	0	1	0	1	0	1	0
<i>PxOUT</i>	2	3	2	3	2	3	2	3	2
<i>PxDIR</i>	4	5	4	5	4	5	4	5	4
<i>PxREN</i>	6	7	6	7	6	7	6	7	6
<i>PxDS</i>	8	9	8	9	8	9	8	9	8
<i>PxSEL</i>	A	B	A	B	A	B	A	B	-
<i>PxIV</i>	E	1E	-	-	-	-	-	-	-
<i>PxIES</i>	18	19	-	-	-	-	-	-	-
<i>PxIE</i>	1A	1B	-	-	-	-	-	-	-
<i>PxIFG</i>	1C	1D	-	-	-	-	-	-	-

Управление выводами (портами) MSP430F5529

User Buttons



User LEDs



Управление выводами (портами) MSP430F5529

■ `msp430f5529.h`

- `#define P1IN (PAIN_L) /* Port 1 Input */`
- `#define P1OUT (PAOUT_L) /* Port 1 Output */`
- `#define P1DIR (PADIR_L) /* Port 1 Direction*/`
- ...
- `#define BIT0 (0x0001)`
- `#define BIT1 (0x0002)`
- `#define BIT2 (0x0004)`
- ...

Управление выводами (портами) MSP430F5529

■ `msp430f5529.h`

Константа Соответствие полю

- `XXX` 1-битовое поле-флаг
- `XXXn` бит n многобитного поля `XXX`
- `XXX_n` значение n многобитного поля `XXX`
- `XXX__n` режим n поля `XXX`, например, при делении на степень 2
- `MOD4` 10000
- `MOD_4` 00100
- `MOD__4` 00010

Управление выводами (портами) MSP430F5529

Theory



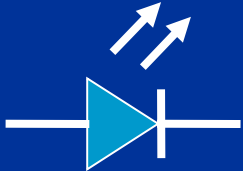
Down



P1.7 (S1)

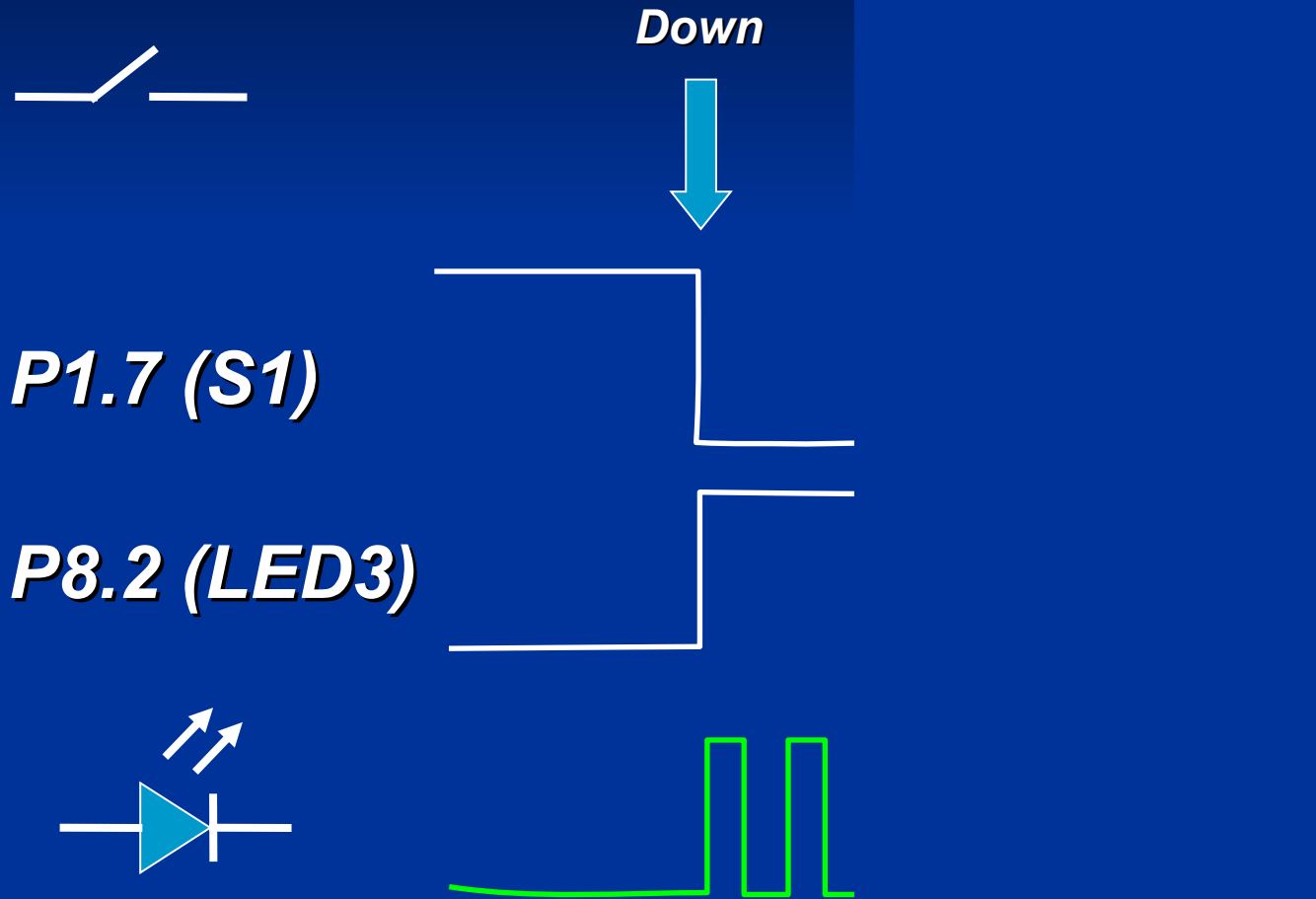


P8.2 (LED3)



Управление выводами (портами) MSP430F5529

Theory



Управление выводами (портами) MSP430F5529

Theory



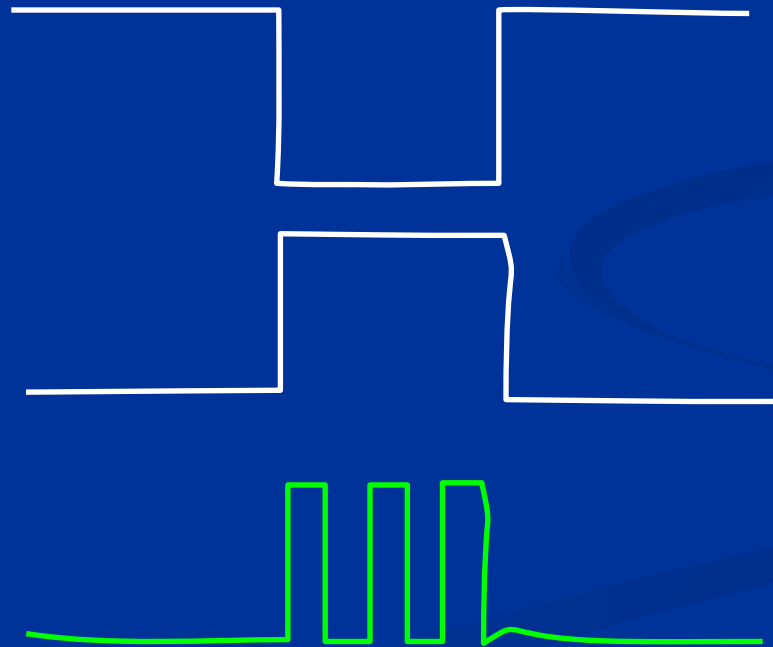
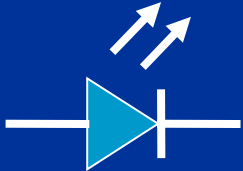
Down

Up



P1.7 (S1)

P8.2 (LED3)



Управление выводами (портами) MSP430F5529

```
#include <msp430.h>
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    volatile unsigned int i;
    P1DIR &= !(BIT7);          // P1.7 (S1) set as input
    P8DIR |= BIT2;              // P8.2 (LED3) set as output
    while(1)
    {
        if (P1IN & BIT7)       // hi level - don't press button
        { P8OUT &= !(BIT2); } // off led
        else P8OUT |= BIT2; // low level - press button - on led
        for(i=25000;i>0;i--);
        P8OUT &= !(BIT2); // off led after small pause
        for(i=25000;i>0;i--); // small pause
    }
    return 0;
}
```

***Видео
01.Digital I/O***

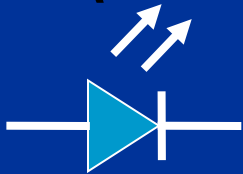
Управление выводами (портами) MSP430F5529

Practice



P1.7 (S1)

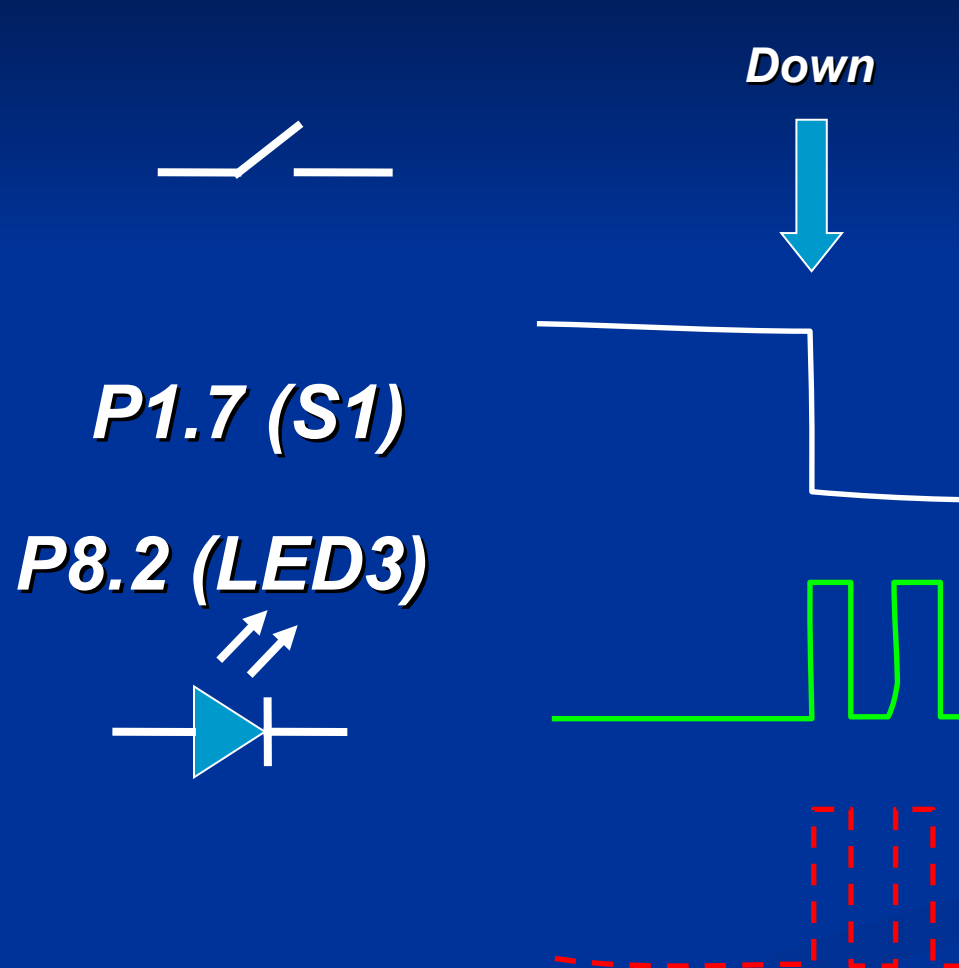
P8.2 (LED3)



Down

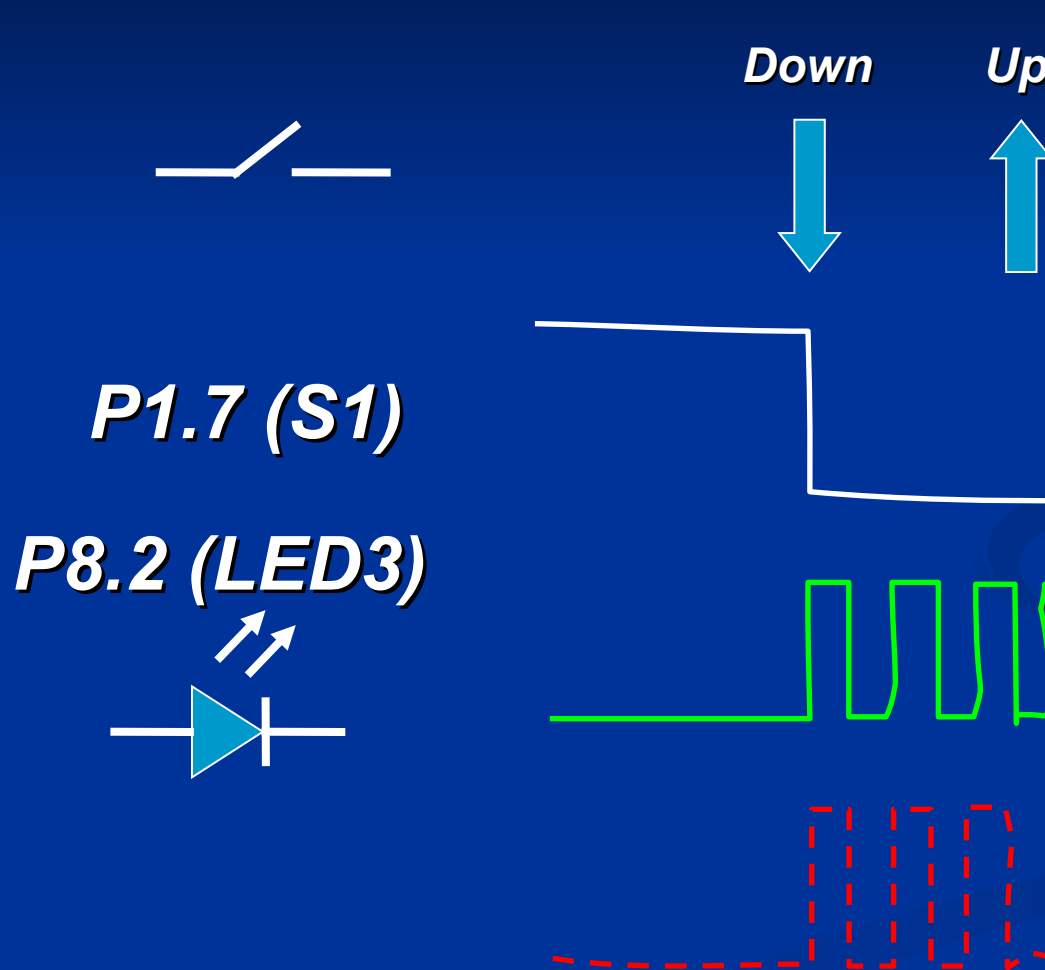


Practice



Управление выводами (портами) MSP430F5529

Practice



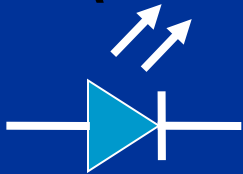
Управление выводами (портами) MSP430F5529

Practice



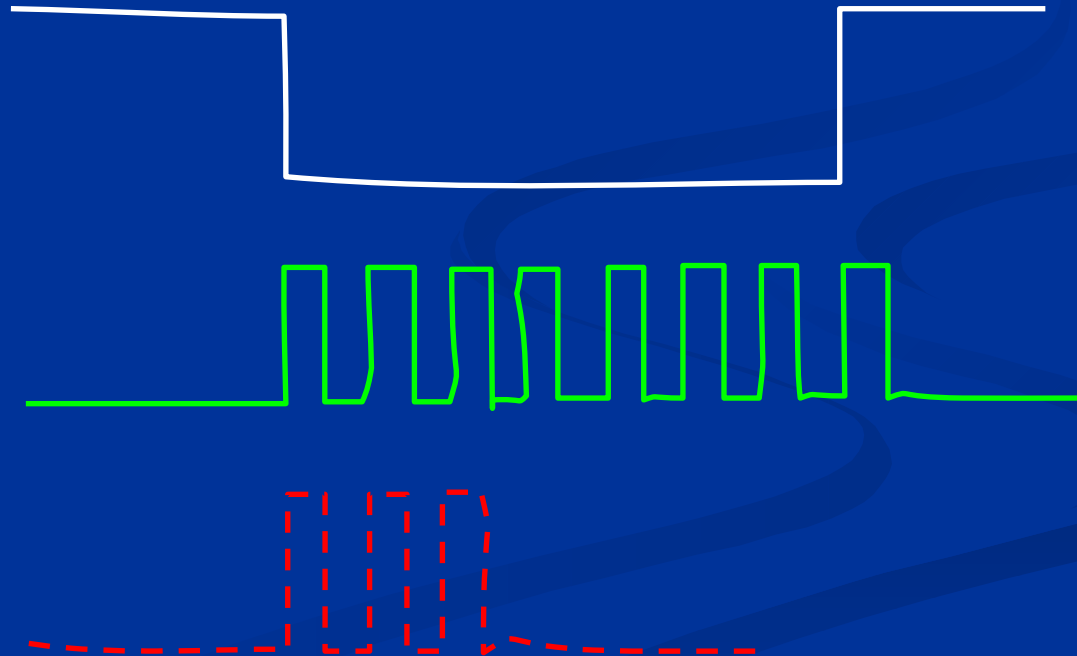
P1.7 (S1)

P8.2 (LED3)



Down

Up



Управление выводами (портами) MSP430F5529

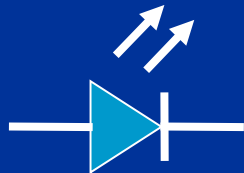
Theory



P1.7 (S1)

Interrupt

P8.2 (LED3)



Down



Управление выводами (портами) MSP430F5529

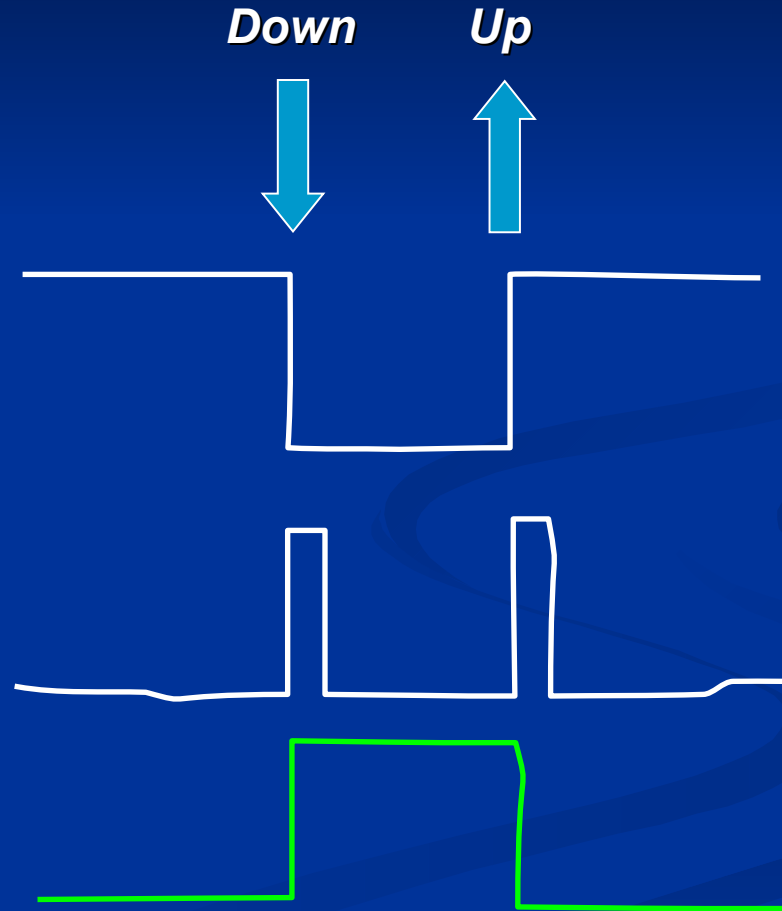
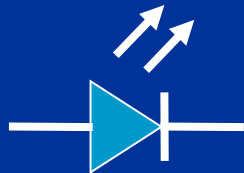
Theory



P1.7 (S1)

Interrupt

P8.2 (LED3)



Управление выводами (портами) MSP430F5529

```
#include <msp430.h>
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    P1DIR &= !(BIT7); // P1.7 (S1) set as input
    P8DIR |= BIT2; // P8.2 (LED3) set as output

    P1OUT |= BIT7; // ????????
    P1IE |= BIT7; // P1.7 (S1) interrupt enable
    P1IES |= BIT7; // P1.7 (S1) edge for interrupt:
                  // high-to-low
    // Enter LPM0, enable interrupts
    __bis_SR_register(LPM0_bits + GIE);
    __no_operation(); // For debugger
    return 0;
}
```

Управление выводами (портами) MSP430F5529

■ *Ink_msp430f5529.cmd*

```
■ ...
■ .int39      : {}                > INT39
■ .int40      : {}                > INT40
■ RTC         : { * ( .int41 ) } > INT41 type = VECT_INIT
■ PORT2       : { * ( .int42 ) } > INT42 type = VECT_INIT
■ TIMER2_A1   : { * ( .int43 ) } > INT43 type = VECT_INIT
■ TIMER2_A0   : { * ( .int44 ) } > INT44 type = VECT_INIT
■ USCI_B1     : { * ( .int45 ) } > INT45 type = VECT_INIT
■ USCI_A1     : { * ( .int46 ) } > INT46 type = VECT_INIT
■ PORT1      : { * ( .int47 ) } > INT47 type = VECT_INIT
■ TIMER1_A1   : { * ( .int48 ) } > INT48 type = VECT_INIT
■ TIMER1_A0   : { * ( .int49 ) } > INT49 type = VECT_INIT
■ DMA         : { * ( .int50 ) } > INT50 type = VECT_INIT
■ ...
```

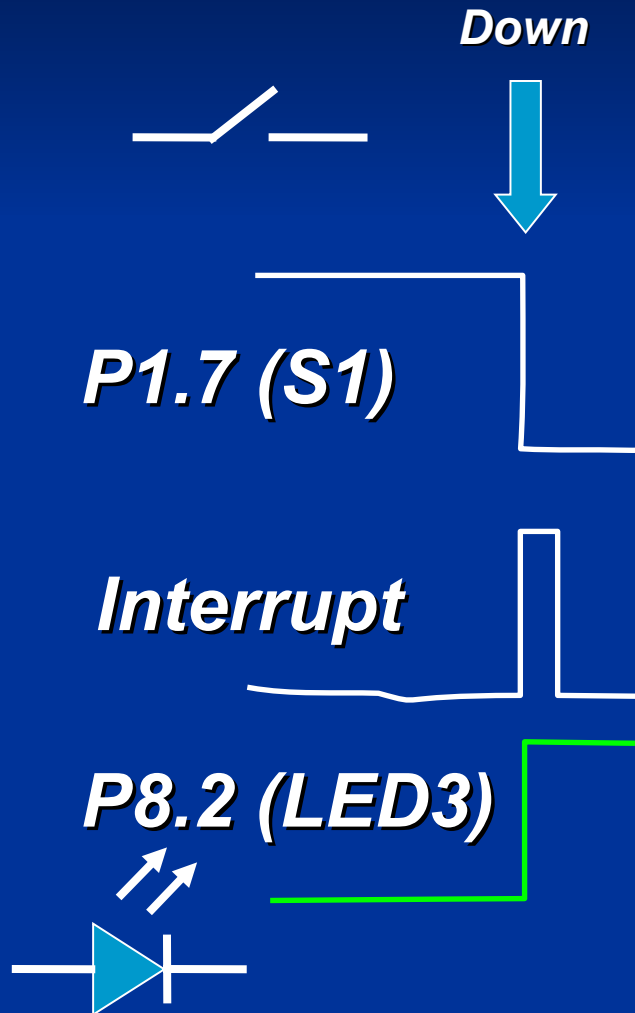
Управление выводами (портами) MSP430F5529

```
#pragma vector=PORT1_VECTOR
__interrupt void PORT1_ISR(void) {
    if (P1IN & BIT7)          // high level - don't press button
        {P8OUT &= !(BIT2);} // off led
    else P8OUT |= BIT2; // low level- press button- on led
    if (!(P1IES & BIT7)) // ?????
        P1REN ^= BIT7; // ?????
    P1IES ^= BIT7; // ?????
    P1IFG = 0; // reset interrupt flag
}
```

***Видео
02.Digital I/O***

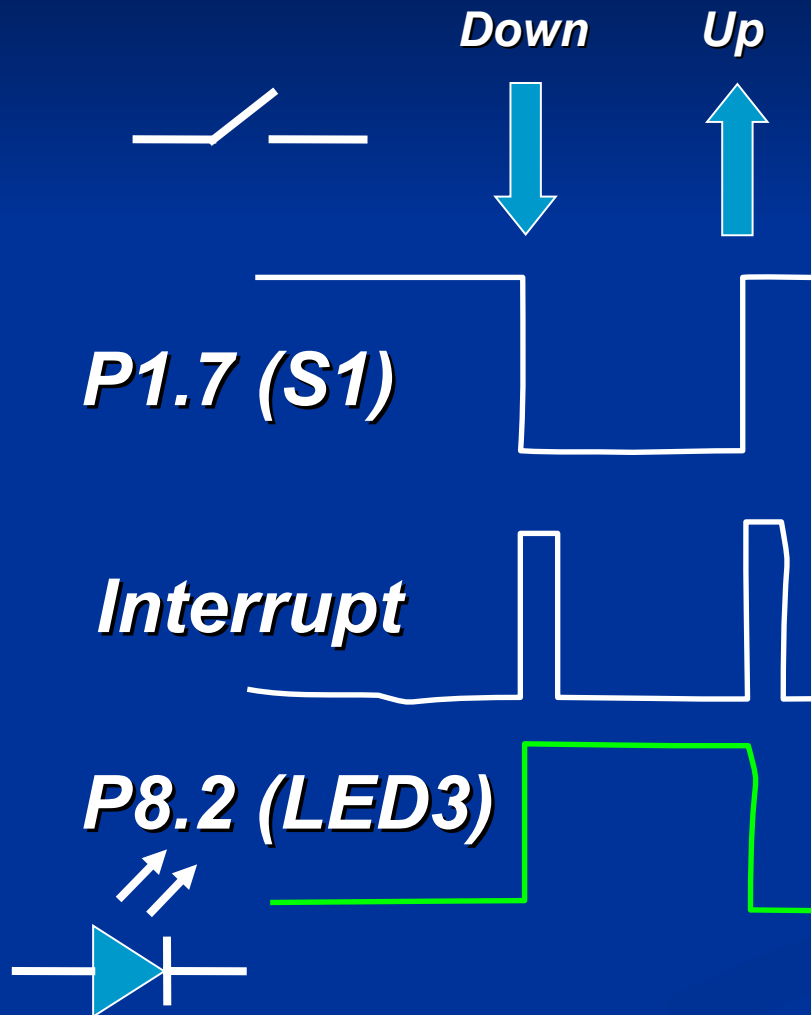
Управление выводами (портами) MSP430F5529

Practice



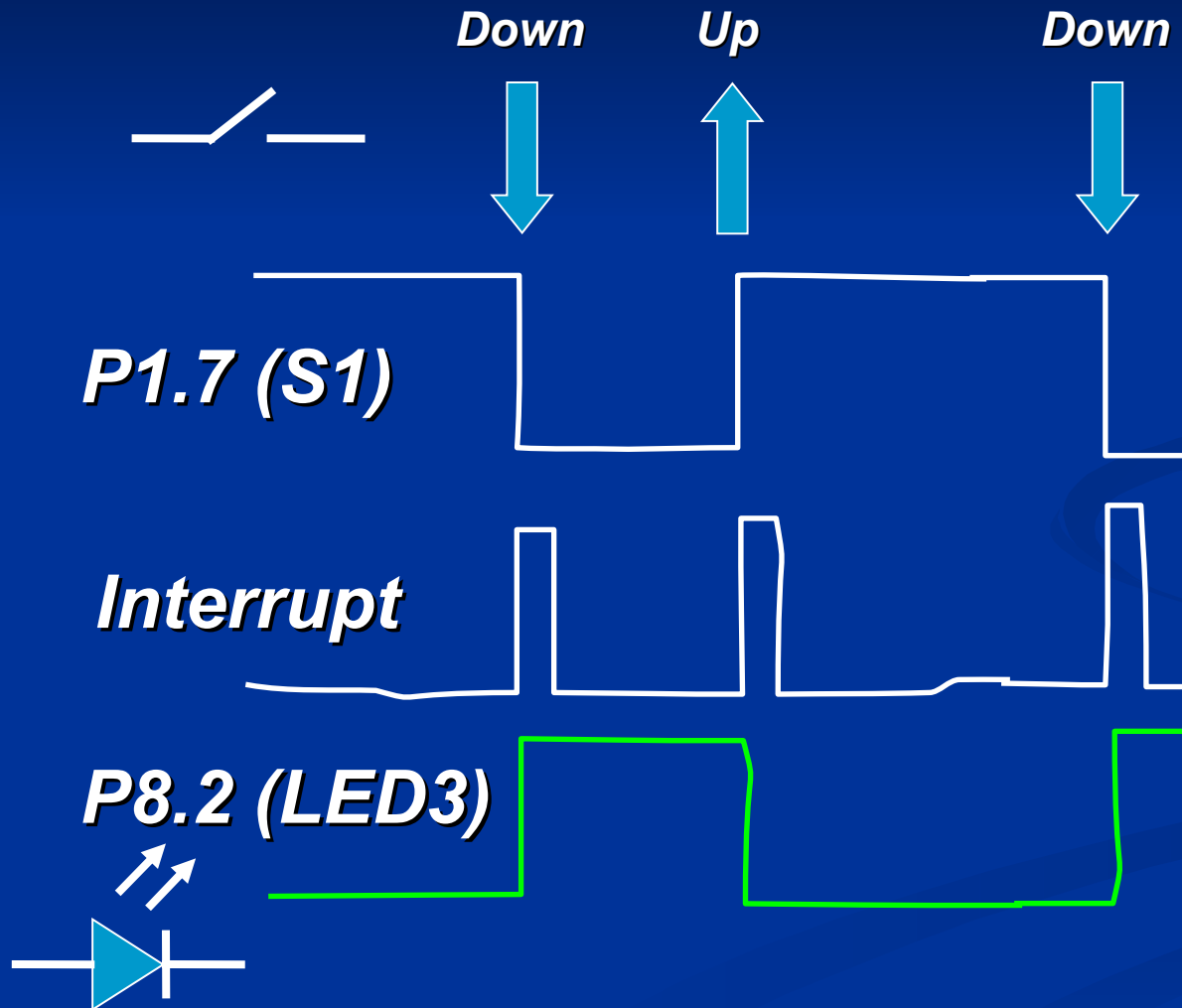
Управление выводами (портами) MSP430F5529

Practice



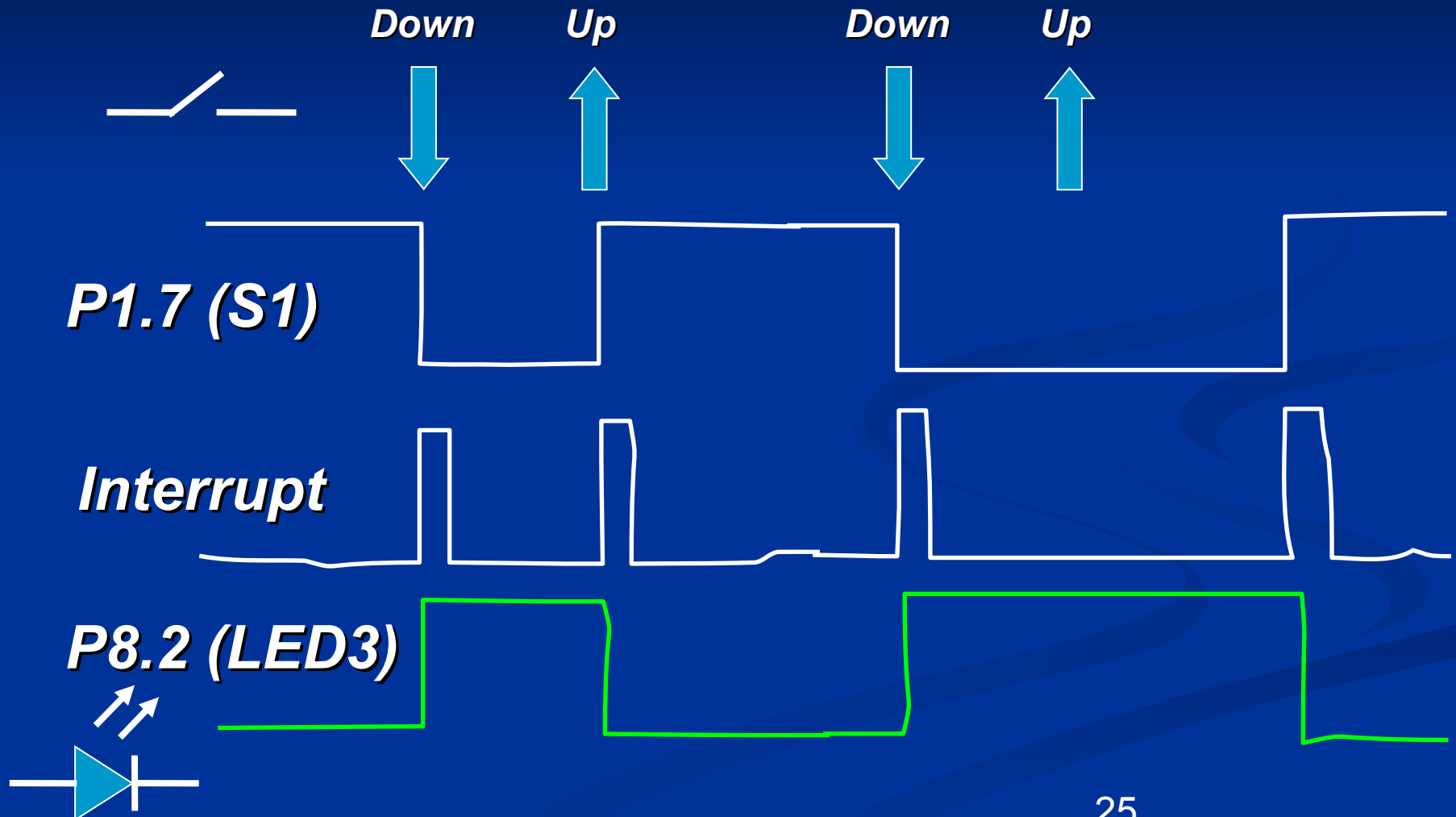
Управление выводами (портами) MSP430F5529

Practice



Управление выводами (портами) MSP430F5529

Practice



Домашнее задание №1

- **E-mail:** *prytkov@bsuir.by*

- **Тема письма:**

МПСИС_<последняя цифра номера группы>_<Фамилия>_дз_<номер задания>

- **Например:** *МПСИС_5_Иванов_дз_1*

1.0. Правильно написать тему письма

1.0. Прислать письмо вовремя

Домашнее задание №1

- 1.1. Объяснить, почему при отпускании кнопки светодиод гаснет не сразу, а спустя достаточно длительный промежуток времени?
- 1.2. Объяснить, что делает код. Почему в режиме работы пина на вход, производится запись в регистр данных?

```
P1OUT |= BIT7;    // ????????
```

- 1.3. Объяснить, что делает код

```
if (!(P1IES & BIT7)) // ?????
```

```
    P1REN ^= BIT7;    // ?????
```

```
P1IES ^= BIT7;      // ?????
```

- 1.4. Почему во второй программе через раз начинает работать, т.е. светодиод гаснет при отпускании кнопки ?
- 1.5. Есть ли в программах ошибки? Если есть, то какие?

Таймеры MSP430F5529

- *WDT – 015Ch. Сторожевой таймер (Watchdog) 32-бит*
- *TAx: TA0 – 0340h, TA1 – 0380h, TA2 – 0400h.*

3 таймера A

(5, 3, 3 регистров захвата соответственно)

- *TBx – 03C0h. 1 таймер B (7 регистров захвата)*
- *RTC_A – 04A0h. Таймер реального времени*

Таймеры MSP430F5529

- **Сторожевой таймер (Watchdog) 32-бит**
- **Основная функция – генерация сигнала сброса при программном сбое, например, зацикливании: если заданный интервал времени истек, генерируется сигнал сброса**
- **Таймер может быть сконфигурирован как интервальный и генерировать сигналы прерываний по истечении заданного промежутка времени**

Таймеры MSP430F5529

- **Свойства сторожевого таймера:**
- **8 программно выбираемых временных интервалов**
- **Сторожевой и интервальный режимы**
- **Защита доступа к управляющему регистру**
- **Может отключаться для экономии энергии**
- **Отказоустойчивый сигнал (источник счетного сигнала не может быть отключен, пока таймер в сторожевом режиме). Это может не позволить перейти в LPM режим**

Таймеры MSP430F5529

- Регистр счетчика непосредственно программно не доступен
- Сигнал на счетный вход может подаваться с SMCLK, ACLK, VLOCLK либо X_CLK от некоторых устройств
- После сброса сторожевой таймер настроен на сторожевой режим, сигнал от SM_CLK. Необходимо остановить, установить либо сбросить таймер до истечения установленного интервала, иначе будет сгенерирован сигнал сброса PUC
- Флаг запроса на прерывание сбрасывается автоматически после обслуживания либо может быть сброшен программно
- Адреса обработчиков в сторожевом и интервальном режиме различны

Таймеры MSP430F5529

Регистр	Биты	Поле	Назначение
WDTCTL	15-8	WDTPW	Пароль на доступ к регистру
	7	WDTHOLD	Остановка таймера (=1)
	6-5	WDTSEL	Источник счетного сигнала
	4	WDTTMSEL	Режим: 0 –сторожевой, 1 - интервальный
	3	WDTCNTCL	Очистка регистра счетчика.
	2-0	WDTIS	Выбор интервала (входная частота делится на константу)

Таймеры MSP430F5529

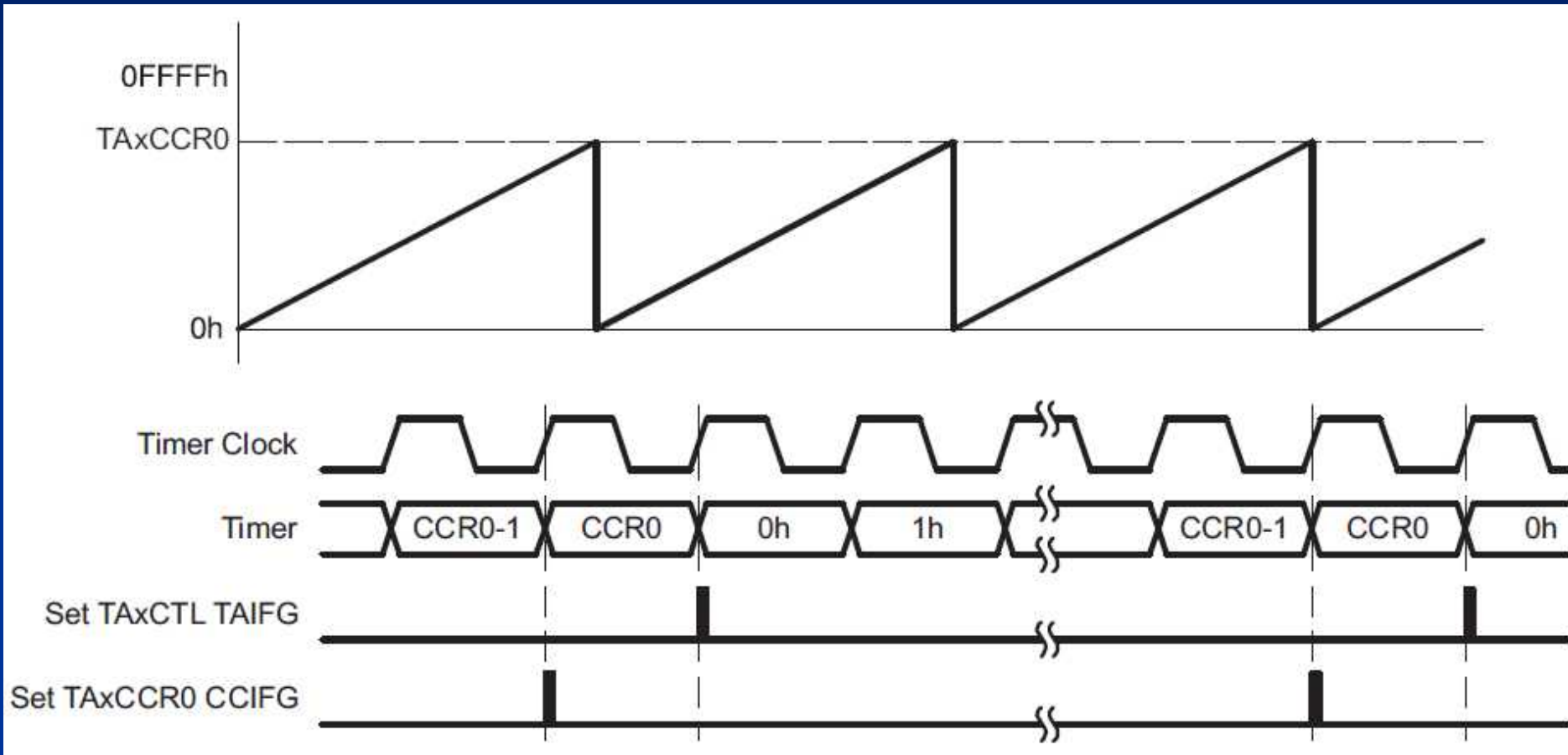
- **Таймер А – 16 бит, 3 таймера (ТА0, ТА1, ТА2)**
- **4 режима**
- **Выбор источников счетного импульса**
- **5 (3, 3) регистров захвата/сравнения**
- **Поддержка множественного захвата/сравнения, управления выходами с возможностью широтно-импульсной модуляции и интервальное время**
- **Асинхронный вход и защелкивание выхода**
- **Регистр счетчика доступен программно**
- **Счет по фронту тактового импульса**
- **Возможность генерации прерывания при переполнении**

Таймеры MSP430F5529

- **Источники входного импульса: ACLK, SMCLK, внешние CAXCLK, INCLK**
- **Деление входной частоты на 2,3,4,5,6,7,8**
- **Режимы:**
- **Остановка**
- **Прямой счет (до уровня TAxSSR0) (Up Mode)**
- **Непрерывный режим (Continuous Mode)**
- **Реверсивный счет (Up/Down mode)**

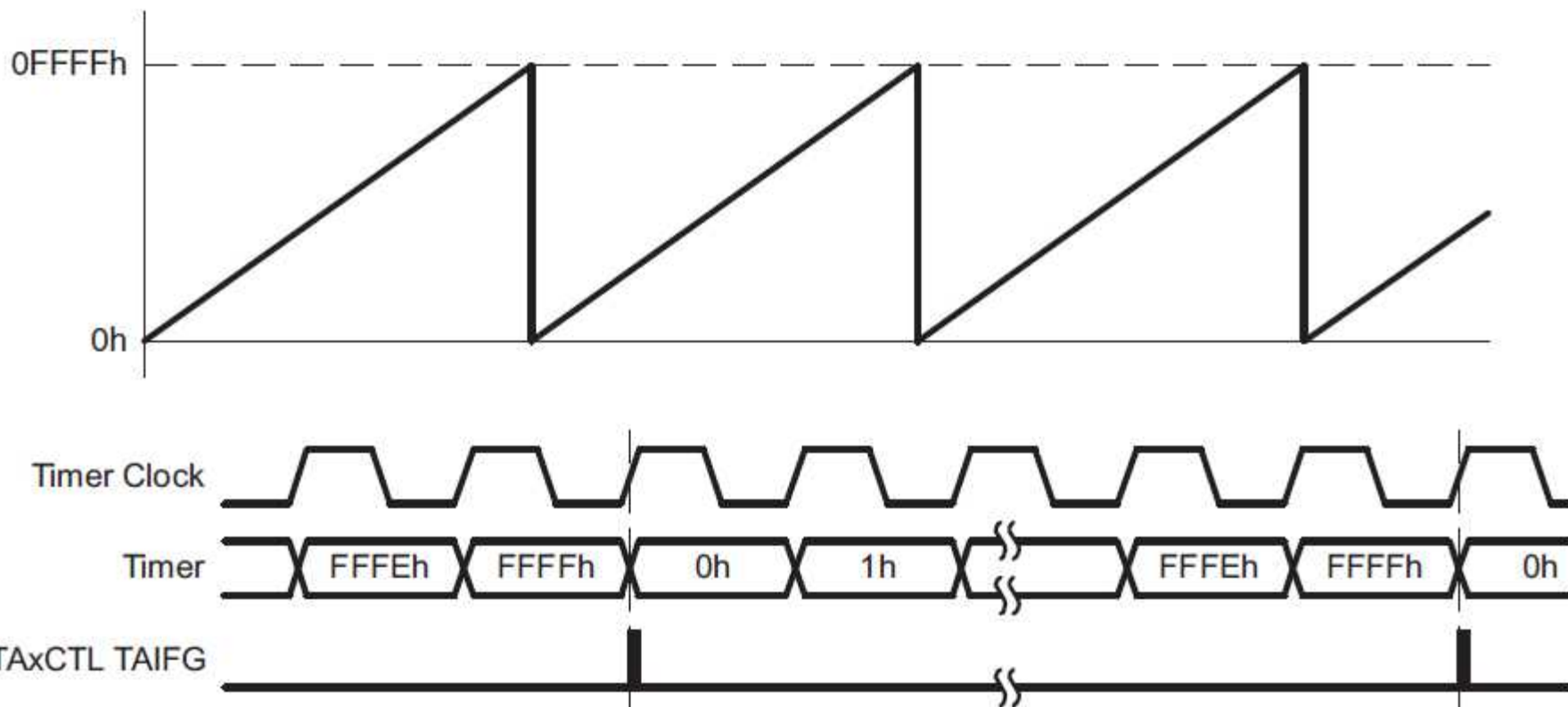
Таймеры MSP430F5529

■ Режим прямого счета (до уровня $TAxCCR0$) (Up Mode)



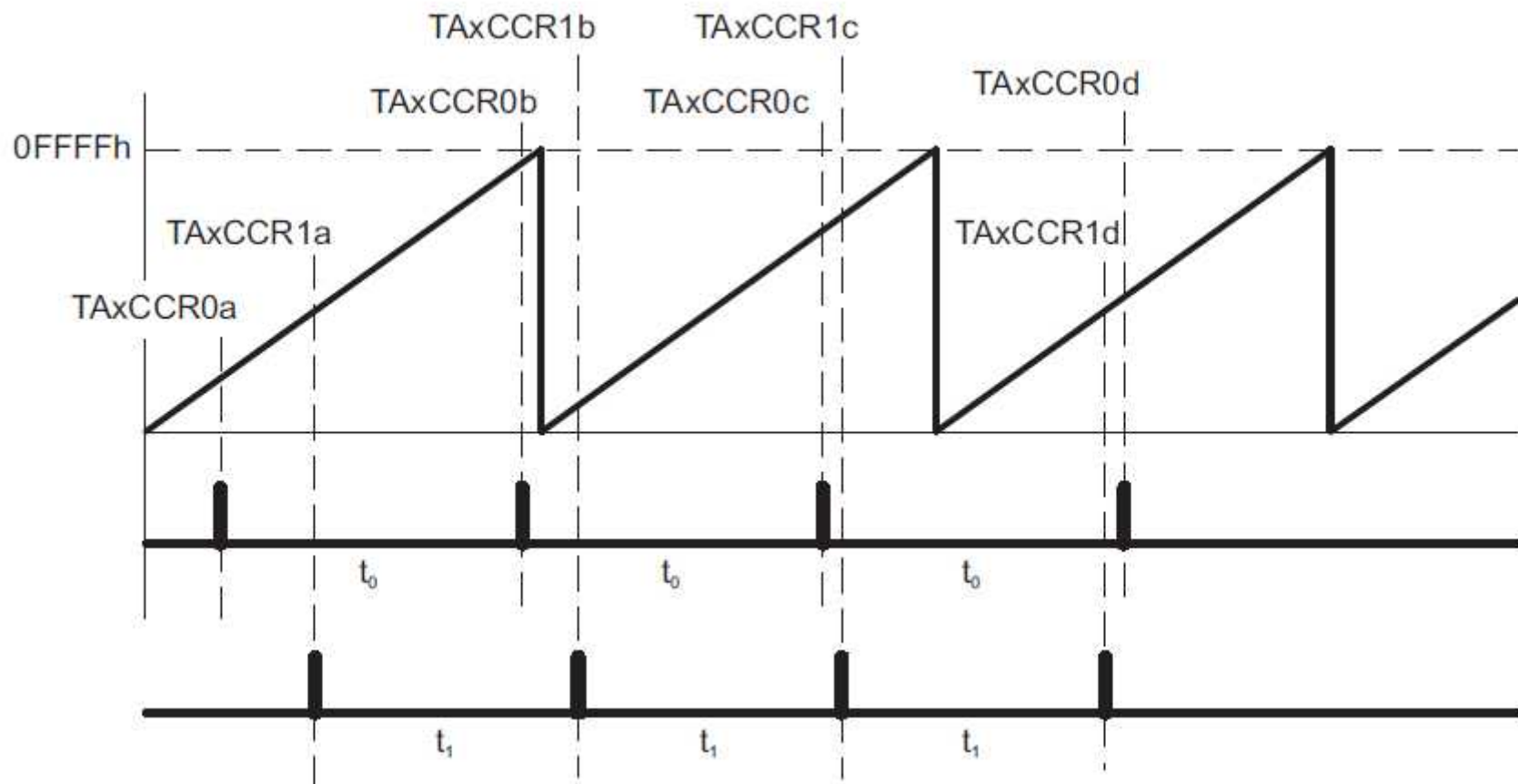
Таймеры MSP430F5529

■ Непрерывный режим (Continuous Mode)



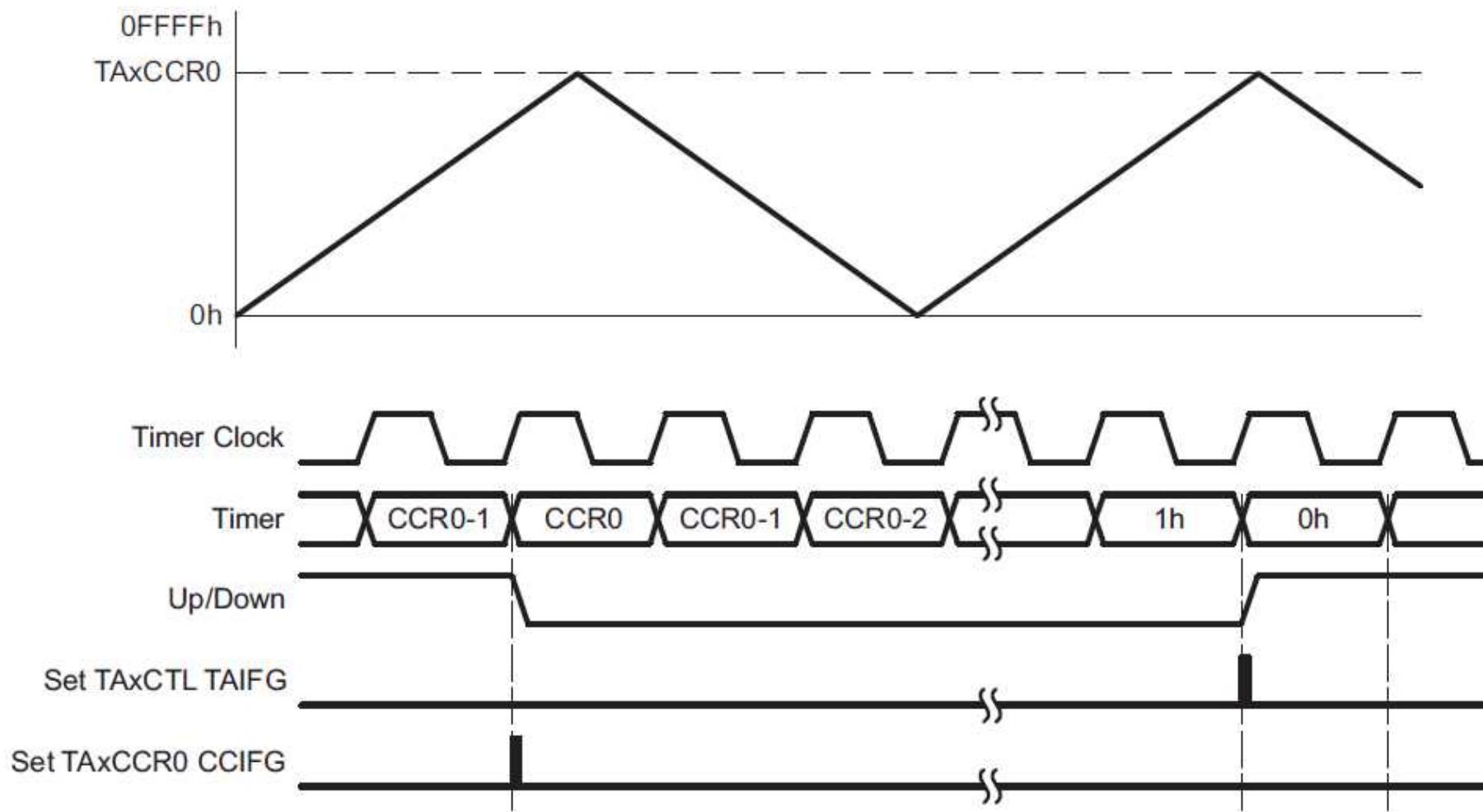
Таймеры MSP430F5529

■ Непрерывный режим (Continuous Mode)



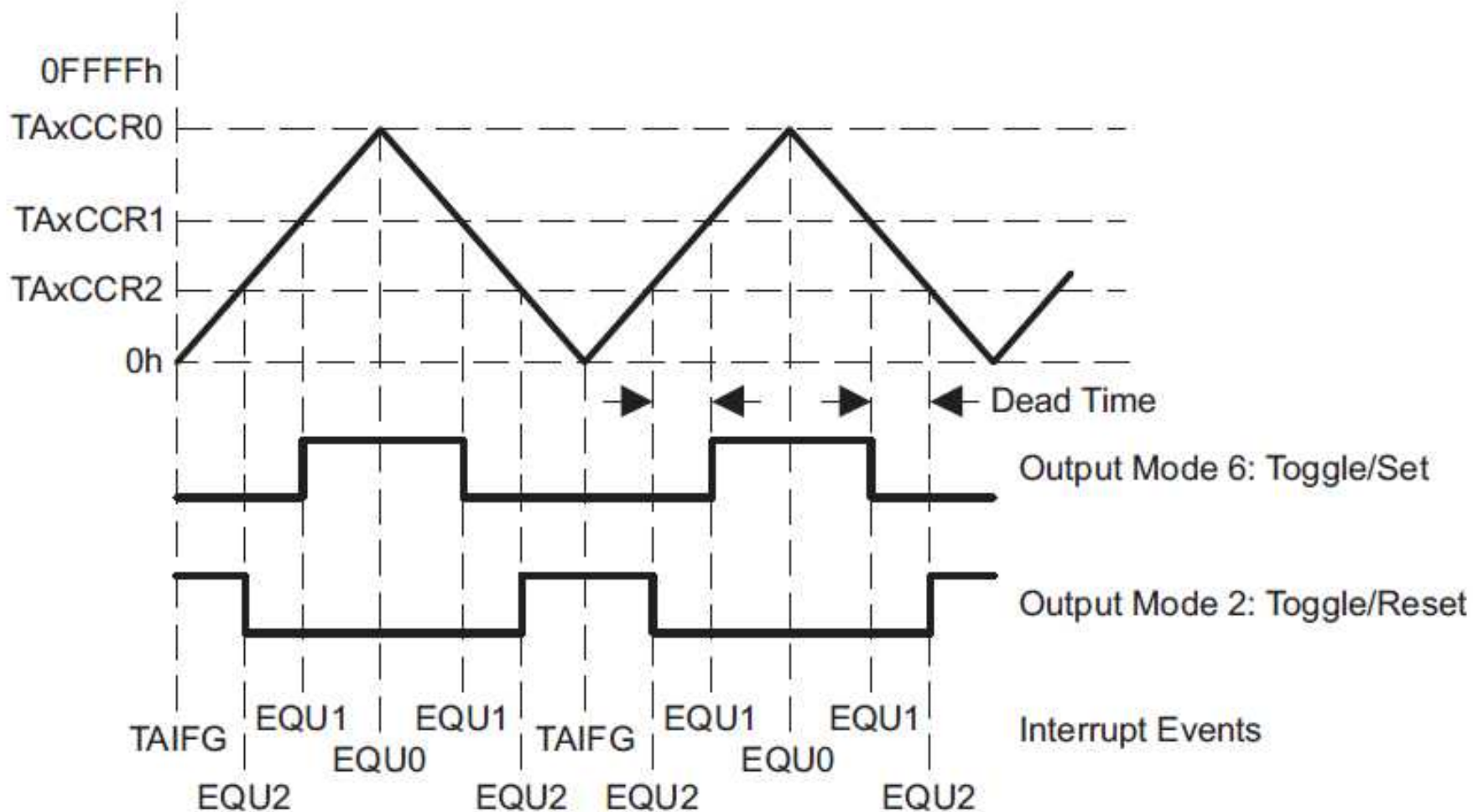
Таймеры MSP430F5529

■ Режим реверсивного счета (Up/Down Mode)



Таймеры MSP430F5529

■ Режим реверсивного счета (Up/Down Mode)



Таймеры MSP430F5529

- **Режим захвата:**
- **Позволяет сохранить время события**
- **Входы захвата CCIxA, CCIxB соединены со внешними выводами или внутренними сигналами, выбор линии задается**
- **Захват происходит по фронту, спаду, либо и по тому и по другому**
- **Сигнал захвата асинхронный, что может привести к гонкам. Для синхронизации сигнала захвата с тактовым импульсом, рекомендуется устанавливать бит SCS**

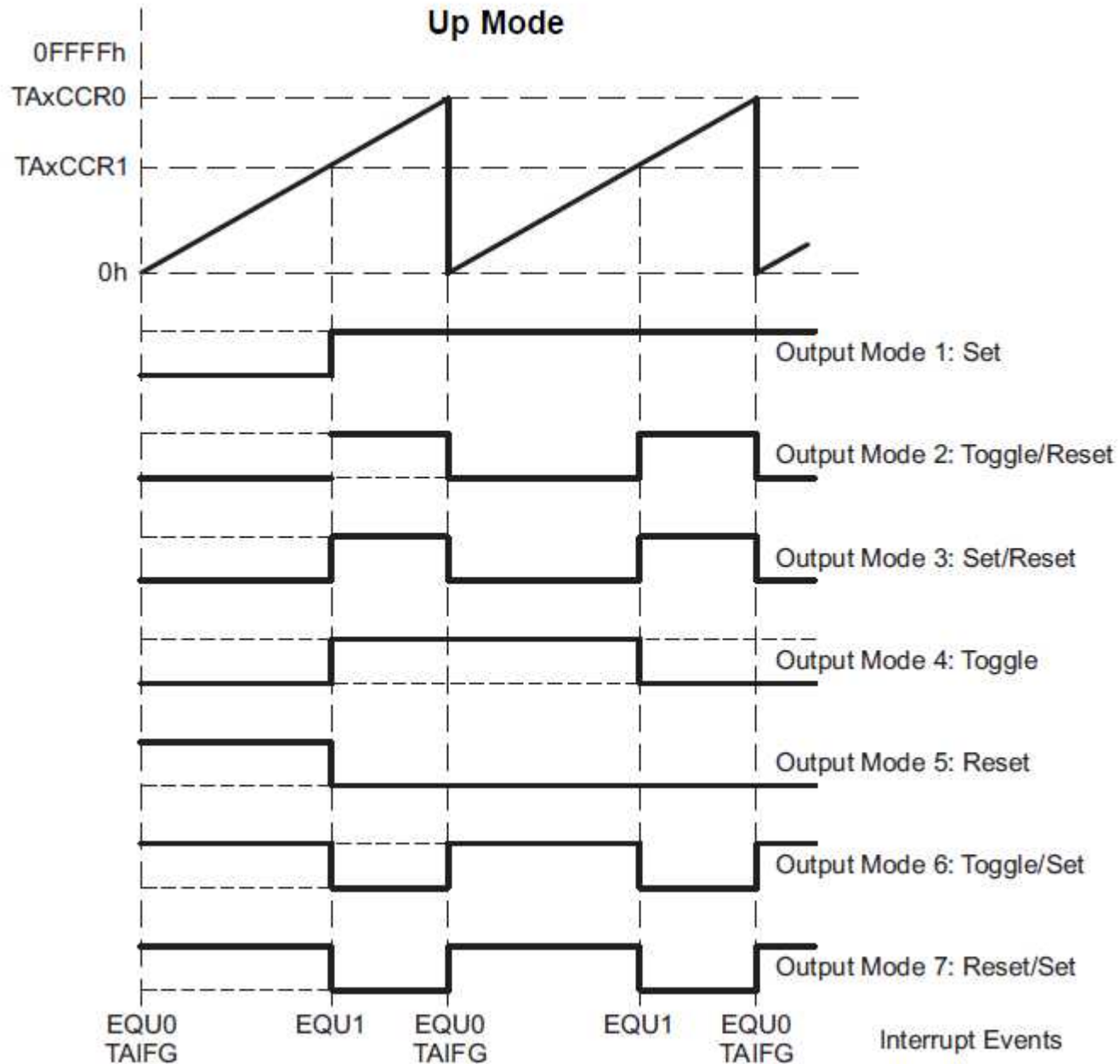
Таймеры MSP430F5529

- **Режим сравнения:**
- **Позволяет формировать ШИМ-сигнал или прерывание по заданному интервалу**
- **Режимы выхода:**
- **Output. На выходе постоянный уровень**
- **Set. Однократная установка при достижении заданного значения TAxCCRn**
- **Toggle/Reset. Выход меняется при достижении значения TAxCCRn, сбрасывается при достижении TAxCCR0**

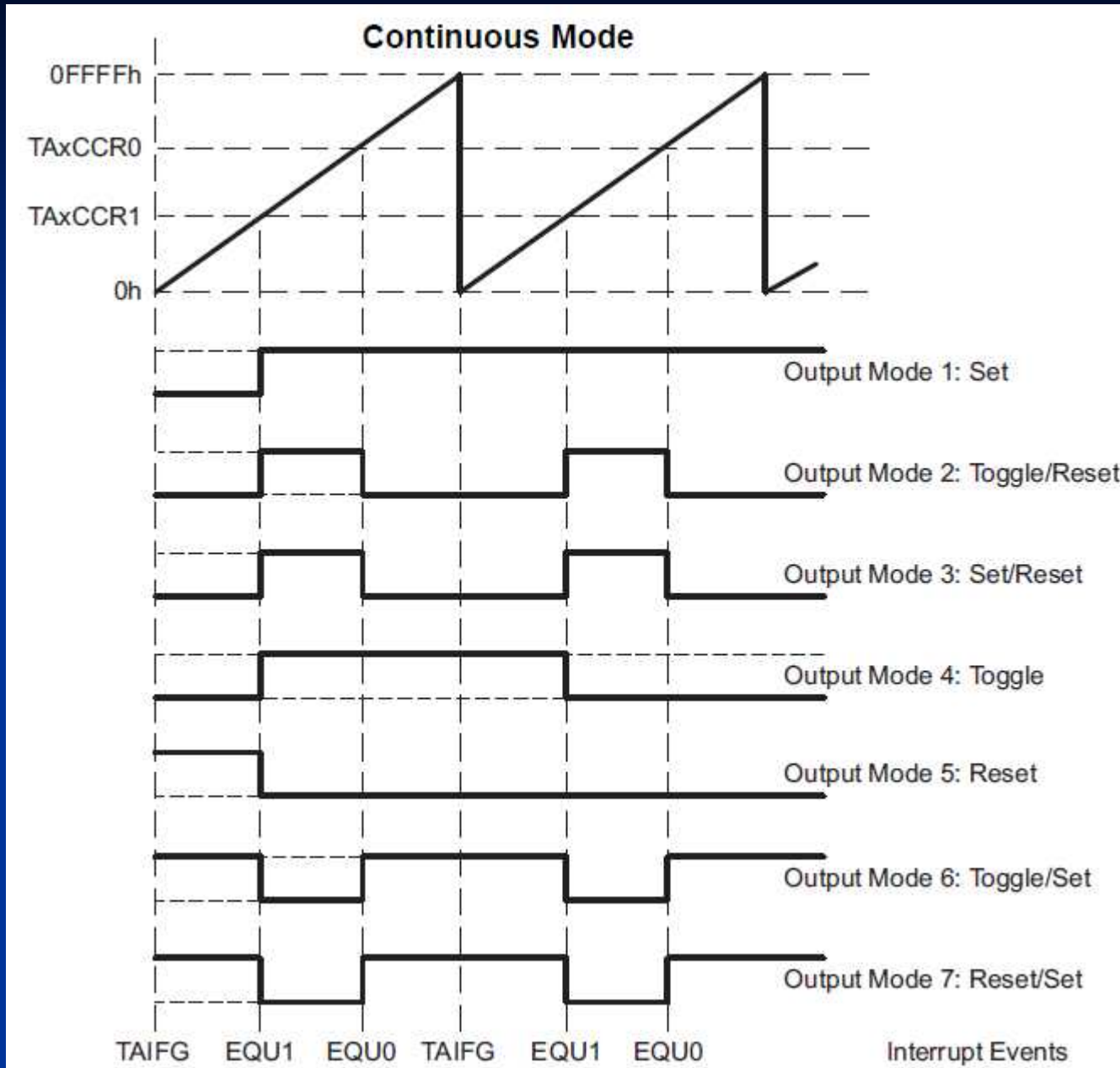
Таймеры MSP430F5529

- **Set/Reset.** Выход устанавливается при достижении значения $TAxCCRn$, сбрасывается при достижении $TAxCCR0$
- **Toggle.** Выход меняется при достижении значения $TAxCCRn$
- **Reset.** Однократный сброс при достижении заданного значения $TAxCCRn$
- **Toggle/Set.** Выход меняется при достижении значения $TAxCCRn$, устанавливается при достижении $TAxCCR0$
- **Reset/Set.** Выход сбрасывается при достижении значения $TAxCCRn$, устанавливается при достижении $TAxCCR0$

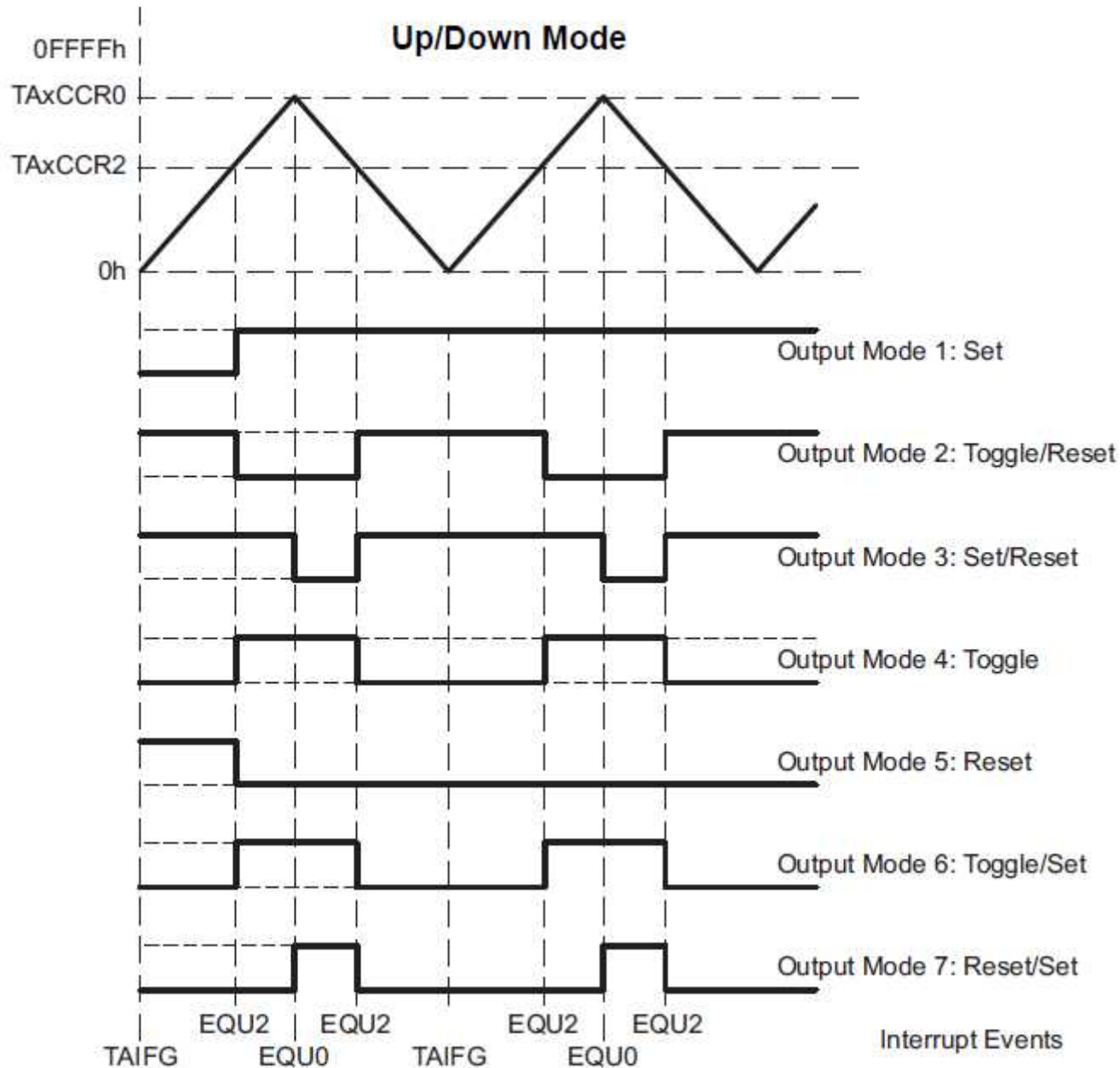
Таймеры MSP430F5529



Таймеры MSP430F5529



Таймеры MSP430F5529



Таймеры MSP430F5529

Разобраться с работой в разных режимах выхода

- Два вектора прерываний для таймера:
- Для *TAxSSCR0* *CCIFG*
- Для всех остальных *CCIFG* и для *TAIFG*

Таймеры MSP430F5529

■ Регистры таймера A

Регистр	Адрес	Назначение
TAxCTL	0340h	Регистр управления
TAxCCTL0 - 6	0342h- 034Eh	Управление захватом/сравнением
TAxR	0350h	Счетчик
TAxCCR0 - 6	0352h- 035Eh	Захват-сравнение
TAxIV	036Eh	Вектор прерывания
TAxEX0	0360h	Расширение 0

Таймеры MSP430F5529

Регистр	Биты	Поле	Назначение
TAxCTL	8-9	TASSEL	Выбор источника синхроимпульса
	6-7	ID	Делитель на входе
	4-5	MC	Режим
	2	TACLR	Очистка счетчика
	1	TAIE	Разрешение прерываний
	0	TAIFG	Флаг прерывания
TAxR	0-15	TAxR	Значение счетчика
TAxCCTLn	14-15	CM	Режим захвата (фронт, спад)
	12-13	CCIS	Выбор входа захвата/сравнения

Таймеры MSP430F5529

Регистр	Биты	Поле	Назначение
TAxCSCTLn	11	SCS	Синхронизация источника захвата
	10	SCCI	Синхронизация входа захвата/сравнения (защелкивание)
	8	CAP	Режим:0-сравнение, 1-захват
	5-7	OUTMOD	Режим выхода
	4	CCIE	Разрешение прерывания захвата/сравнения
	3	CCI	Вход сравнения/захвата
	2	OUT	Значение выхода в постоянном режиме

Таймеры MSP430F5529

Регистр	Биты	Поле	Назначение
TAxCCTLn	1	COV	Переполнение захвата
	0	CCIFG	Флаг прерывания п сравнению/захвату
TAxCCR0	0-15	TAxCCR0	Сравниваемое значение / сохраненное захваченное значение
TAxIV	0-15	TAxIV	Регистр вектора прерывания
TAxEX0	0-2	TAIDEX	Расширенный делитель входа

Таймеры MSP430F5529

- Таймер В. Отличия от таймера А:
- 7 регистров захвата/сравнения
- Разрядность счетчика
программируется 8, 10, 12, 16 бит
- Регистр TVXSSCRn с двойной
буферизацией и может быть
сгруппирован
- Все выходы имеют
высокоимпедансное состояние
- Не поддерживается бит SCS1

Таймеры MSP430F5529

- *Таймер RTC_A (Real Time Clock) :*
- *Конфигурируемые часы реального времени с функцией календаря и счетчика общего назначения*
- *Возможность прерываний*
- *Выбор формата BCD или двоичный в режиме часов реального времени*
- *Программируемый будильник*
- *Подстройка коррекции времени*

Таймеры MSP430F5529

```
#include <msp430.h>
```

```
int main(void) {  
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer  
    P8DIR |= BIT1;           // P8.1 (LED2) set as output  
    P8DIR |= BIT2;           // P8.2 (LED3) set as output  
    P8OUT |= BIT1;           // P8.1 (LED2) on  
    P8OUT &= ~(BIT2);        // P8.2 (LED3) off  
  
    TA0CCTL0 = CCIE;         // CCR0 interrupt enabled  
    // SMCLK, divide by 4, contmode, clear TAR  
    TA0CTL = TASSEL_2 | ID_2 | MC_2 | TACLRL ;  
  
    // Enter LPM0, enable interrupts  
    __bis_SR_register(LPM0_bits + GIE);  
    __no_operation();        // For debugger  
    return 0;  
}
```

Таймеры MSP430F5529

```
#pragma vector = TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void)
{
    // P8.1 & P8.2 (LED2 & LED3) switch
    P8OUT ^= (BIT1 | BIT2);
}
```

Рассчитать частоту на входе таймера

Рассчитать частоту мигания светодиодов

Домашнее задание №2

2.1. Разобрать режимы выходов таймера. Если обнаружены ошибки и неточности, пояснить, что не так

2.2. Рассчитать частоту на входе таймера для приведенного фрагмента кода. Подробно объяснить, откуда и как сформировался этот сигнал

2.3. Рассчитать частоту мигания светодиодов для приведенного фрагмента кода