*Лабораторная работа №2*

**Модуль таймера**

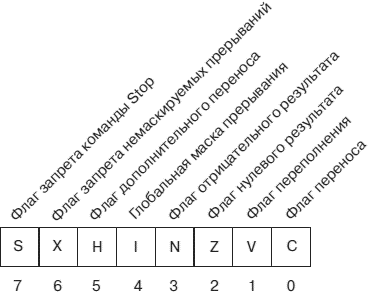
*Цель работы:* Изучение работы счётчика временных интервалов (таймер).

*Приборы и принадлежности:*

1. Плата CSMB12С128;
2. Платформа PBMCUSLK;
3. Персональный компьютер;
4. Проводы;

**Прерывание**

Программно логическая модель центрального процессора содержит регистр признаков CCR, формат которого представлен на рис.1. Особенностью всех семейств МК от компании Freescale Semiconductor является наличие в регистре признаков не только флагов результатов операции, но и дополнительных битов управления подсистемой прерывания. В составе регистра признаков МК 68HC12/HCS12 — два таких бита. Бит I — глобальная маска прерываний — используется для управления маскируемыми прерываниями. Бит X — бит запрета немаскируемых прерываний — управляет немаскируемыми прерываниями. Оба этих бита устанавливаются в 1 в состоянии сброса МК. Обратите внимание, в МК 68HC12/HCS12 установка битов I и X запрещает соответствующие прерывания.



*Рис. 1.Формат регистра состояния CCR*

**Немаскируемые прерывания**

В соответствие со своим названием немаскируемые прерывания не могут быть отключены пользователем. Однако в предыдущем абзаце было упомянуто, что установка бита X в 1 запрещает немаскируемые прерывания. Значение бита X действительно равно 1 в состоянии сброса МК. Однако далее он может быть установлен в 0 под управлением программы инициализации, разрешая тем самым немаскируемые прерывания. Далее этот бит не может быть изменен под управлением программы, и в этом его отличие от бита глобальной маски прерываний I.

Три типа немаскируемых прерываний реализуются в МК 68HC12/HCS12:

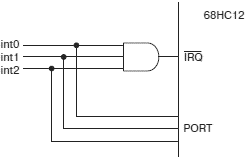
• **Прерывание по внешнему запросу .**Все МК 68HC12/HCS12 имеют вывод внешнего немаскируемого прерывания . Активный уровень сигнала для генерации запроса на прерывание — логический 0.

• **Прерывание по несуществующему коду команды.**Каждая инструкция языка ассемблер МК имеет собственный код. В МК 68HC12/HCS12 коды операций могут быть однобайтовыми и двухбайтовыми. Но не все теоретически возможные коды использованы для кодирования реальных команд процессорного ядра CPU12. Если на этапе выборки кода команды из памяти произошло считывание несуществующего кода команды, то генерируется запрос на немаскируемое прерывание.

• **Программное прерывание — инструкция SWI.**Система команд МК 68HC12/HCS12 имеет инструкцию программного прерывания, которая позволяет перейти к исполнению подпрограммы прерывания из прикладной программы.

***1.3 Маскируемые прерывания***

• **Прерывание по внешнему запросу .**Все МК 68HC12/HCS12 имеют вывод внешнего маскируемого прерывания . Активный уровень сигнала для генерации запроса на прерывание — логический 0. В некоторых приложениях требуется принять запросы от нескольких внешних источников сигналов. Для таких случаев следует использовать дополнительный логический элемент, который объединяет запросы от всех источников по логике ИЛИ (рис. 2). Если запрос на вход   поступил, и МК перешел к выполнению подпрограммы прерывания, то в этой подпрограмме следует опросить линии порта для того, чтобы установить, какой из источников вызвал прерывание. Обработка нескольких объединенных по ИЛИ запросов с программным поиском установившего запрос источника называется поллингом.



*Рис. 2.* *Внешняя цепь для подключения сигналов нескольких внешних запросов на прерывание*

• **Прерывание по таймеру меток реального времени RTI.**Таймер меток реального времени генерирует последовательность равноотстоящих во времени запросов на прерывание. Период повторения запросов настраивается программистом. Эти прерывания могут быть использованы для регулярного выполнения микроконтроллером некоторой задачи. Например, для измерения напряжения аккумуляторной батареи каждые три мин, чтобы сигнализировать о необходимости ее замены. Мы рассмотрим особенности прерываний RTI в главе 7 на примере управления скоростью вращения электрическим двигателем.

• **Прерывание по событию канала захвата/сравнения (IC/OC) таймера.**Восемь одинаковых блоков в составе модуля таймера, которые именуют «каналами», предназначены для контроля за уровнем сигнала на входе канала или для изменения в строго определенный момент времени логического уровня на выходе канала. Заданное программистом изменение входного или выходного сигнала канала рассматривается как событие, которое генерирует запрос на прерывание. Например, если канал настроен на слежение за перепадом входного сигнала из 1 в 0, то когда такое изменение произойдет, будет выставлен запрос на прерывание.

• **Прерывание по переполнению таймера.**Основным блоком модуля таймера является 16 разрядный счетчик временной базы. Этот счетчик невозможно остановить. Также невозможно изменить его коэффициент счета, который составляет 216 = 65536. Поэтому регулярно счетчик временной базы изменяет свой код с $FFFF на $0000. Такое изменение кода называют переполнением счетчика. В момент переполнения по желанию программиста может генерироваться запрос на прерывание, в то время как счетчик продолжает считать дальше. Такие прерывания особенно удобны при необходимости измерения очень больших временных интервалов. Для этого в микроконтроллере производят подсчет, сколько переполнений счетчика произошло за этот временной интервал, и, зная период счета счетчика, определяют длительность исследуемого временного интервала.

• **Прерывание по переполнению счетчика внешних событий.**Когда счетчик внешних событий переполняется, то может генерироваться запрос на прерывание точно так же, как и для счетчика временной базы.

• **Прерывание по событию на входе счетчика внешних событий.** Этот запрос на прерывание формируется, если сигнал на входе счетчика внешних событий изменил свое значение. Характер изменения, т.е. перепад из 0 в 1, или из 1 в 0, или любое изменение логического уровня, определяется программистом.

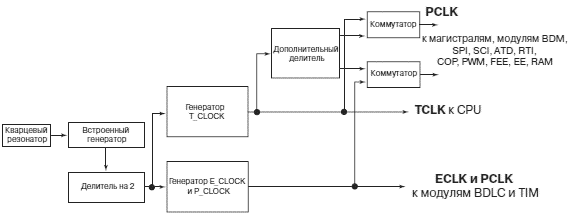
• **Прерывание от модулей контроллеров последовательного ввода/вывода SCI и SPI.**Каждый модуль последовательного ввода/вывода формирует целую группу запросов на прерывание: при завершении передачи слова, при приеме слова, при обнаружении различного рода нарушений в протоколе передачи информации.

• **Прерывание от модуля АЦП.**Модуль аналого-цифрового преобразователя формирует запрос на прерывание, когда процесс оцифровки очередного сигнала завершен, и двоичный код сигнала может быть считан в память МК.

• **Прерывание при выходе МК из энергосберегающих режимов.**Это прерывание позволяет вывести МК из состояния STOP или WAIT, в котором он находился с целью снижения потребляемой энергии. Такие прерывания очень полезны при объединении нескольких МК в информационную сеть. Мы рассмотрим этот тип прерывания более подробно в следующем параграфе.

**Модуль таймера**

Микроконтроллеры семейства 68HC12/HCS12 имеют в своем составе модуль генератора CGM (Clock Generation Module), который генерирует импульсные последовательности для тактирования центрального процессора, межмодульных магистралей, периферийных модулей в составе МК, а также внешние периферийные интегральные схемы. Структурная схема модуля CGM представлена на рис. 3.



*Рис.3.**Структура модуля тактирования CGM*

Микроконтроллеры семейства 68HC12/HCS12 используют три внутренних сигнала тактирования: TCLK, ECLK и PCLK. Эти сигналы образуются путем деления эталонной импульсной последовательности внутреннего генератора с внешним кварцевым резонатором. Сигнал TCLK предназначен для тактирования центрального процессора, импульсные последовательности ECLK и PCLK используются для тактирования межмодульных магистралей и различных периферийных модулей (рис. 3). Модуль тактирования CGM микроконтроллера B32 оснащен также дополнительным делителем, который позволяет существенно снизить частоту одной из импульсных последовательностей тактирования. Низкая частота тактирования таймерных модулей в некоторых применениях позволяет значительно упростить управляющую программу.

Подсистема реального времени МК семейства 68HC12/HCS12 включает основной таймерный модуль, который имеет две модификации — TIM и ECT, и отдельный таймер меток реального времени.

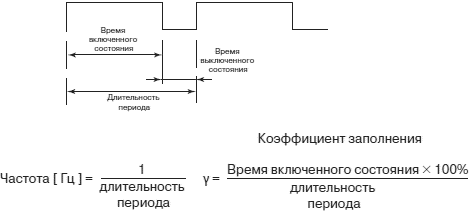
Структура модуля таймера TIM (Timer Interface Module) ориентирована на реализацию трех основных функций:

• **Входного захвата (IC — Input Capture).**Функция входного захвата позволяет производить измерения временных параметров сразу нескольких импульсных сигналов на входах МК. Подсистема входного захвата может быть настроена на измерение длительности единичного или нулевого состояния на входе порта

(рис. 4, а), а также периода, коэффициента заполнения или частоты периодического импульсного сигнала (рис. 4, б).



(а)



(б)

*Рис.4.**Временные характеристики ШИМ сигнала*

• **Выходного сравнения (OC — Output Compare).**Функция выходного сравнения позволяет МК генерировать на нескольких выходах импульсные последовательности с заданными временными характеристиками, такими, как период и коэффициент заполнения для повторяющихся сигналов, длительность единичного или нулевого состояния для неповторяющихся сигналов.

• **Счетчика внешних событий (PA — Pulse Accumulator).**Основная функция этого счетчика — подсчет импульсов (внешних событий) на одном из входов МК. Он также может быть использован для измерения временных параметров внешнего импульсного сигнала большой длительности.

Для реализации функций входного захвата/выходного сравнения (IC/OC) модуль таймера TIM использует восемь идентичных аппаратных блоков, которые принято называть каналами. Каждый из каналов посредством программных установок настраивается на реализацию режима входного захвата или выходного сравнения независимо от режима работы других каналов модуля таймера. Каждый из каналов использует общий счетчик временной базы для фиксации моментов наступления событий. Параллельная работа всех восьми каналов с одним счетчиком временной базы не вносит погрешностей в формируемые или измеряемые временные интервалы, поскольку фиксация этих интервалов реализуется на аппаратном уровне с последующим программным обслуживанием каналов по прерываниям. Каждый канал связан с одной из линий порта PORT Т. Счетчик событий PA в составе модуля TIM также связан с линией 7 порта PORT T. Поэтому линии 0…6 порта PORT T в подсистеме таймера могут использоваться или как входы IC, или как выходы OC, в то время как линия 7 порта PORT T кроме этих двух функций IC/OC может также использоваться как вход тактовых импульсов для счетчика внешних событий.

**Счетчик временной базы**

Основным блоком модуля таймера TIM является 16 разрядный счетчик временной базы TCNT, структурная схема которого представлена на рис.5. Текущий код счетчика используется всеми каналами захвата/сравнения в качестве отсчета момента реального времени. Именно поэтому этот счетчик и называют счетчиком временной базы. Этот счетчик также называют свободно считающим счетчиком. Определение «свободно считающий» отражает следующую особенность работы счетчика. Если работа модуля таймера разрешена, то счетчик временной базы производит непрерывный счет, начиная с минимального кода $0000 до максимального кода $FFFF. При поступлении следующего тактового импульса код счетчика изменяется с $FFFF на $0000. Далее счет продолжается в порядке нарастания кода. Невозможно остановить счетчик под управлением программы, так же как и изменить коэффициент счета счетчика, равный 216. Текущее состояние счетчика отображается в двух 8-разрядных регистрах: TCNTH — старший байт счетчика, TCNTL — младший байт счетчика. В карте памяти МК эти регистры располагаются по следующим адресам: $0084 — TCNTH, $0085 — TCNTL. Вместе оба этих регистра составляют 16-разрядный регистр текущего состояния счетчика временной базы TCNT. Имя TCNT обычно объявляется в заголовочном файле.

**Особенности счетчика временной базы**

Счетчик временной базы тактируется импульсной последовательностью с выхода мультиплексора MUX. Ко входам мультиплексора подключены четыре источника тактирования: основная импульсная последовательность с выхода делителя частоты и дополнительные сигналы PACLK, PACLK/256, PACLK/65536. На вход программируемого делителя частоты подключен сигнал PCLK с выхода подсистемы тактирования. Частота следования импульсов на линии PCLK равна частоте тактирования межмодульных магистралей МК. Коэффициент деления программируемого делителя частоты определяется разрядами PR2…PR0 регистра масок таймера 2 (TMSK2). Формат регистра TMSK2 приведен на рис. 5. Таблица соответствия численного значения коэффициента деления двоичной комбинации разрядов PR2…PR0 представлена на таблице 2.1.

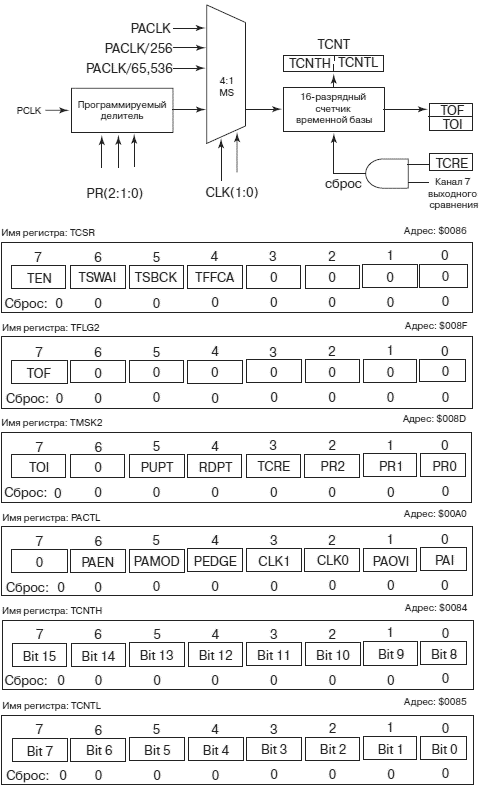
|  |  |
| --- | --- |
| PR[2:1:0 ] | Коэффициент деления |
| 000 | 1 |
| 001 | 2 |
| 010 | 4 |
| 011 | 8 |
| 100 | 16 |
| 101 | 32 |
| 110 | 64 |
| 111 | 128 |

*Таблица 1.**Выбор коэффициента программируемого делителя для счетчика временной базы*

Величина коэффициента деления определяет длительность периода переполнения счетчика временной базы. Поскольку разрядность счетчика равна 16, то коэффициент счета этого двоичного счетчика равен 216 или 65356. Минимальный период переполнения счетчика составляет 8169 мс (216 импульсов × 1/8 МГц), поскольку максимальная частота импульсной последовательности PCLK, равная частоте тактирования межмодульных магистралей, составляет 8 МГц. Однако при задании максимального коэффициента деления, равного 32 (см. таблицу 1), период переполнения счетчика составит уже 262424 мс (216 импульсов × 32/8 МГц).

В некоторых приложениях необходимо формировать временные интервалы, значительно превышающие приведенные расчетные значения. Этого можно достигнуть путем снижения частоты тактирования счетчика временной базы.

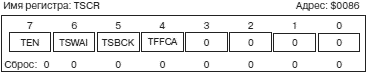
Однако уменьшение частоты импульсной последовательности PCLK связано со снижением производительности процессорного ядра. Поэтому для счетчика временной базы предусмотрены три альтернативных источника тактирования PACLK, PACLK/256, PACLK/65536. Выбор одного из четырех источников тактирования осуществляется двухразрядным кодом CLK1:CLK0 в регистре управления счетчиком внешних событий PACL. Формат регистра PACL представлен на рис.5.



*Рис. 5.**Структура и регистры управления счетчика временной базы модуля таймера*

**Регистр управления модулем таймера**

Регистр управления модулем таймера TSCR (Timer System Control Register) располагается в памяти МК по адресу $0086. Формат регистра представлен на рис. 6. Старший бит регистра TEN разрешает (при TEN=1) или запрещает (при TEN=0) функционирование модуля таймера. Этот бит используется разработчиками для включения или отключения модуля таймера в процессе работы устройства. Отключение модуля полезно с точки зрения снижения потребления энергии, если функции таймера не используются в алгоритме управления. Обратите внимание, что после сброса МК модуль таймера выключен, т.к. бит TEN автоматически устанавливается в 0 в состоянии сброса МК.



*Рис. 6****.****Формат регистра TCSR*

Бит TFFCA управляет механизмом сброса флагов событий модуля таймера. К этим флагам относятся флаг переполнения таймера TOF, флаги событий в каналах захвата/сравнения IC/OC и флаг переполнения счетчика внешних событий. Если бит TFFCA установлен в 0, то для сброса перечисленных флагов следует использовать обычную процедуру, когда в бит установленного флага под управлением программы записывается 1. Попытка записи 0 в бит установленного флага оставит флаг без изменения. Если бит TFFCA установлен в 1, то вводится в действие дополнительный механизм сброса рассматриваемых флагов событий:

• Флаг переполнения счетчика временной базы TOF в регистре TFLG2 будет сброшен автоматически при выполнении операции чтения регистра текущего кода счетчика TCNT;

• Флаг события канала захвата/сравнения CnF (n — номер канала) в регистре TFLG1 будет сброшен автоматически при выполнении операции чтения или записи в регистр данных этого канала;

• Флаги PAONF и PAIF счетчика внешних событий в регистре PAFLG будут сброшены автоматически при чтении или записи в регистр PACNT.

Приведенный ниже программный фрагмент демонстрирует код для разрешения работы модуля таймера.

/\*-----------------------------------------------------------------\*/

/\* MAIN PROGRAMM                                                   \*/

/\*-----------------------------------------------------------------\*/

#include <912b32.h>

void main(void) {

 unsigned char TSCR\_MASK = 0x80; /\*разрешение работы модуля таймера\*/

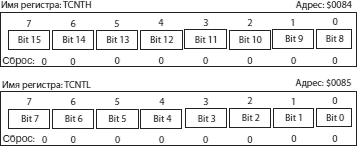
 TSCR = TSCR\_MASK;

}

/\*------------------------------------------------------------------\*/

**Регистр счетчика временной базы**

Если работа модуля таймера разрешена, то счетчик временной базы изменяет свое состояние в порядке естественного счета. Число разрядов счетчика временной базы равно 16, поэтому коэффициент счета счетчика составляет 216. Изменение коэффициента счета не предусмотрено. Счетчик временной базы не возможно остановить, однако его состояние в любой момент времени может быть считано под управлением программы из регистра текущего состояния TCNT (Timer CouN Terregister). Регистр TCNT — 16 разрядный, в памяти МК он занимает две ячейки. По адресу $0084 располагается старший байт регистра TCNTH, по адресу $0085 — младший байт регистра TCNTL. Формат регистра представлен на рис. 7.



*Рис. 7.**Формат регистра TCNT*

Считывание регистра текущего состояния счетчика временной базы следует производить только в 16 разрядном формате, поскольку побайтное чтение может дать неверный результат. Дело в том, что счетчик временной базы может тактироваться максимальной частотой, которая равна *fBUS*. Каждая операция чтения занимает несколько тактов *fBUS*, следовательно, две последовательных операции чтения приведут к тому, что старший и младший байты счетчика будут считаны при разных состояниях этого счетчика и вместе составят неверное его состояние (если режим «чтение на лету» не задействован). Приведенный ниже программный фрагмент демонстрирует, как правильно считать состояние счетчика с использованием двухбайтовых переменных.

/\*-------------------------------------------------\*/

/\* MAIN PROGRAMM                                   \*/

/\*-------------------------------------------------\*/

#include<912b32.h>

void main(void) {

 unsignedintstart\_time; /\*время начала процесса\*/

 unsignedintstop\_time; /\*время окончания процесса\*/

 start\_time=TCNT;

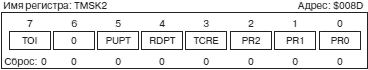
 stop\_time=TCNT;

}

/\*-------------------------------------------------\*/

**Регистр масок таймера 2**

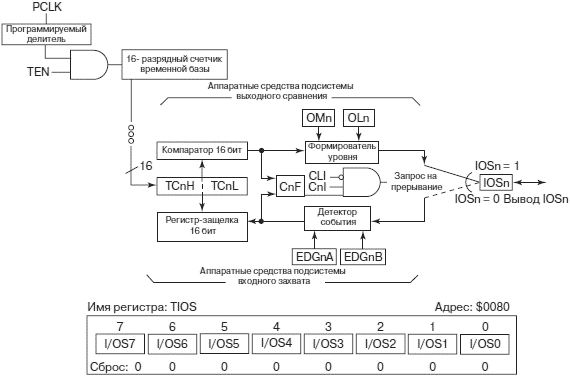
Регистр масок таймера TMSK2 (Timer MaSK register 2) располагается в памяти МК по адресу $008D. Формат регистра представлен на рис. 8. В данном параграфе мы рассмотрим лишь некоторые биты этого регистра. Бит TCRE разрешает сброс счетчика временной базы таймера. Биты PR2:PR1:PR0 устанавливают коэффициент деления программируемого делителя на входе счетчика временной базы в соответствие с таблицом 2.1. Обратите внимание, что минимальный коэффициент деления равен 1, т.е. максимальная частота тактирования счетчика равна *fBUS*. Максимальный коэффициент деления составляет 32. Биты PUPT и TDRB предназначены для управления схемотехникой входных и выходных буферов линий порта PORT T.



*Рис. 8**Формат регистра TMSK2*

**Каналы захвата/сравнения**

Модуль таймера TIM содержит в себе восемь идентичных блоков захвата/сравнения, которые в микропроцессорной технике принято именовать каналами захвата/сравнения. Структурная схема аппаратных средств одного канала захвата/сравнения в составе модуля таймера МК семейства 68HC12/HCS12 представлена на рис. 9.



*Рис.9.**Структура одного канала сравнения/захвата таймера и регистр выбора режима работы каналов TIOS*

Каждый из восьми каналов захвата/сравнения подключен к выводу IOSn, где n — номер канала, n = 0, 1, 2…7. Если канал с номером n конфигурирован как канал захвата, то вывод IOSn автоматически подключается к одноименной линии PTn порта T. Работа в качестве входов подсистемы входного захвата или выходов подсистемы выходного сравнения является альтернативной функцией порта T. Регистр данных порта T расположен по адресу $00AE.

Реализуемая каналом n модуля таймера функция (входной захват или выходное сравнение) определяется битом IOCn регистра TIOC. Регистр расположен по адресу $0080, формат регистра представлен на рис. 9. Если бит IOCn установлен в 1, то канал n работает в режиме выходного сравнения. Если же бит IOCn равен 0, то канал n работает в режиме входного захвата.

Аппаратные средства каждого состоят из 16 разрядного регистра данных канала TCn, 16 разрядных регистра защелки и цифрового компаратора, детектора события, формирователя выходного уровня и триггера события канала (рис. 9). Каждый канал использует в качестве эталона реального времени общий для всех каналов счетчик временной базы.

**Режим входного захвата**

Структура аппаратных средств подсистемы входного захвата IC, которая образуется в результате конфигурирования универсального канала модуля таймера на режим захвата, представлена на рис. 10. Подсистема входного захвата запоминает код счетчика временной базы в момент изменения логического сигнала на входе IOSn (n — номер канала). Изменение логического сигнала распознается детектором события, который может быть программно настроен на один из четырех режимов работы:

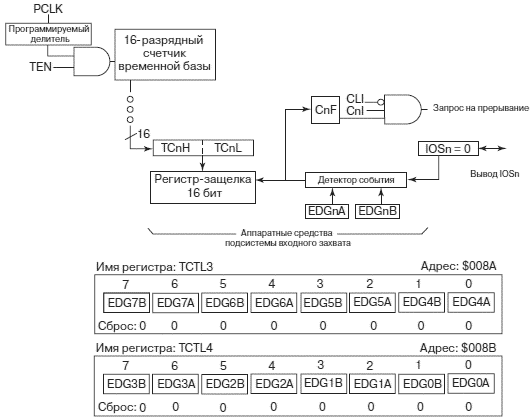
• Распознавание изменения сигнала с 0 на 1 — положительный фронт;

• Распознавание изменения сигнала с 1 на 0 — отрицательный фронт;

• Распознавание любого изменения уровня сигнала;

• Соответствующий вывод МК не подключен к каналу входного захвата и является выводом порта T.

Если детектор определил заданное изменение входного сигнала, то говорят, что наступило событие входного захвата. В момент наступления события код счетчика временной базы запоминается в регистре защелке, одновременно устанавливается триггер события канал CnF (рис. 10). Триггер может быть считан программно, или генерируется запрос на прерывание, если прерывания от канала n модуля таймера разрешены.



*Рис. 10. Структура одного канала таймера в режиме входного захвата и регистры управления каналом TCTL3 и TCTL4*

Для настройки детектора события каждого канала на один из трех перечисленных режимов используются биты EDGnB:EDGnA в регистрах TCTL3 ($008A) и TCTL4 ($008B). Формат этих регистров представлен на рис. 10. Таблица 2.2 устанавливает соответствие между режимом работы детектора события и кодом инициализации в разрядах EDGnB:EDGnA.

|  |  |
| --- | --- |
| **EDGnB: EDGnA** | **Режим детектора событий** |
| 00 | Входной захват не реализуется |
| 01 | Мониторинг нарастающего фронта |
| 10 | Мониторинг спадающего фронта |
| 11 | Мониторинг изменения уровня |

*Таблица 2.2. Выбор режима работы детектора события*

Подсистема входного захвата IC используемся в микропроцессорной технике для измерения различных временных характеристик импульсных сигналов, таких как период следования, коэффициент заполнения, длительность нулевого или единичного состояния. Так для того, чтобы измерить период импульсной последовательности, необходимо запомнить состояние счетчика временной базы в моменты двух соседних изменений сигнала с 0 на 1 (положительный фронт) или с 1 на 0 (отрицательный фронт). Разность этих значений и составит период повторения импульсного сигнала, выраженный в числе периодов частоты тактирования счетчика временной базы. Таким образом будет произведено измерение в относительных единицах конкретной микропроцессорной системы. Если измерение производится с целью управления, то представление временного параметра в относительных единицах обычно является достаточным. Однако, если измеренный параметр должен быть отображен на дисплее, то он должен быть представлен в универсальных единицах измерения, т.е. в микросекундах, миллисекундах и т.д. Для получения численного значения последнего необходимо полученное число относительных единиц умножить на длительность периода частоты тактирования счетчика временной базы. При программировании на Си операция умножения реализуется с использованием стандартной библиотеки. При программировании на ассемблере Вам потребуются дополнительные знания, поскольку операцию умножения необходимо будет исполнять над двухбайтовыми числами.

Если детектор события распознал изменение входного сигнала, которое указано в его текущей конфигурации, то аппаратные средства канала входного захвата автоматически совершают следующие действия:

1. Текущее состояние 16-разрядного счетчика временной базы запоминается в регистре-защелке канала и сразу копируется в 16-разрядный регистр данных канала TCn (n — номер канала). Поскольку данные в регистре TCn не изменяются, то они могут быть считаны побайтно (TCnH — старший байт регистра данных канала, TCnL — младший байт регистра данных), или в двухбайтовом формате. При программировании на Си рекомендуется использовать двухбайтовый формат, используя для этого переменную в формате unsignedinteger.

2. Устанавливается флаг события канала CnF. Этот флаг «сообщает» основной программе о том, что событие произошло, и регистр данных канала TCn должен быть считан программой.

3. Если прерывания по флагу события CnF разрешены (бит CnI установлен), то генерируется запрос на прерывание.

Подсистема входного захвата может быть настроена для реализации разнообразных функций. Рассмотрим пример измерения длительности единичного состояния входного сигнала, т.е. импульса положительной полярности. Предположим, что длительность импульса не превышает периода переполнения счетчика временной базы таймера, исследуемая импульсная последовательность поступает на вход канала 2 модуля таймера, который предварительно настроен в режим входного захвата. Тогда для измерения длительности импульса должна быть реализована следующая последовательность действий:

1. Управляющая программа устанавливает режим работы детектора событий канала 2. Должен быть выбран режим мониторинга нарастающего фронта входного сигнала. Для этого следует установить биты EDG2B:EDG2A в регистрах TCTL3:TCTL4 в состояние 01.

2. Управляющая программа контролирует состояние триггера события C2F.

3. Если триггер C2F установился в 1, то контролируемый сигнал на линии PT2 изменился с 0 на 1. В момент изменения код счетчика временной базы был автоматически переписан в регистр защелку канала 2. Теперь этот код доступен для чтения из регистра данных TC2.

4. Управляющая программа обнаруживает, что триггер C2F установился. Тогда программа считывает двухбайтовый код из регистра данных канала TC2 и записывает его в двухбайтовую беззнаковую переменную rising\_edge.

5. Управляющая программа сбрасывает триггер события C2F посредством записи в бит C2F единицы.

6. Управляющая программа изменяет режим работы детектора событий канала 2. Должен быть выбран режим мониторинга отрицательного фронта входного сигнала. Для этого следует установить биты EDG2B:EDG2A в регистрах TCTL3:TCTL4 в состояние 10.

7. Управляющая программа контролирует состояние триггера события C2F.

8. Если триггер C2F установился в 1, то сигнал на линии PT2 изменился с 1 на 0. В момент изменения код счетчика временной базы был опять автоматически переписан в регистр защелку канала 2.

9. Управляющая программа обнаруживает установленный триггер C2F, считывает двухбайтовый код из регистра данных канала TC2 и записывает его в двухбайтовую беззнаковую переменную falling\_edge.

10. Управляющая программа сбрасывает триггер события C2F посредством записи в бит C2F единицы.

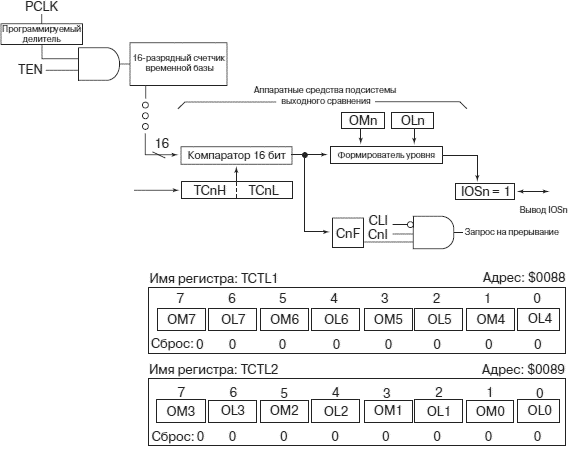
11. Управляющая программа вычисляет число периодов частоты тактирования счетчика временной базы между положительным и отрицательным фронтами исследуемого сигнала: TIME = falling\_edge – rising\_edge. Это число и есть искомая длительность импульса положительной полярности сигнала на входе PT2, выраженная числом периодов частоты тактирования счетчика временной базы.

12. При необходимости длительность импульса может быть представлена в общепринятых единицах измерения времени. Для этого управляющая программа должна выполнить операцию умножения числа TIME на длительность единицы измерения времени таймера, т.е. на длительность периода частоты тактирования счетчика временной базы: tIZM = TIME×1/fBASE.

Регистры специальных функций, связанные с подсистемой входного захвата, мы обсудим позже, после рассмотрения подсистемы выходного сравнения.

**Режим выходного сравнения**

Подсистема выходного сравнения OC используемся в микропроцессорной технике для генерации на выводах МК импульсных сигналов с заданными временными характеристиками. Например, средствами подсистемы выходного сравнения может быть сформирован одиночный импульс предварительно вычисленной длительности, или импульсная последовательность определенной частоты с регулируемым по результатам расчетов в МК коэффициентом заполнения. Структура аппаратных средств подсистемы выходного сравнения OC, которая образуется после конфигурирования универсального канала модуля таймера на режим сравнения, представлена на рис. 11.



*Рис. 11. Структура одного канала таймера в режиме выходного сравнения и регистры управления каналом TCTL1 и TCTL2*

Цифровой компаратор подсистемы выходного сравнения непрерывно сравнивает код счетчика временной базы с 16 разрядным кодом в регистре данных канала TCn (n — номер канала). Момент равенства кодов в микропроцессорной технике называют событием выходного сравнения. Если цифровой компаратор определил равенство кодов, то аппаратные средства канала выходного сравнения автоматически совершают следующие действия:

1. Устанавливается флаг события канала CnF. Обратите внимание, события входного захвата и выходного сравнения отмечаются одним и тем же флагом. Это флаг события канала CnF. Смысловое значение флага (IC или OC) определяется ранее выбранным в процессе инициализации режимом работы канала. Флаг CnF «сообщает» основной программе о том, что событие выходного сравнения произошло, и в регистр данных канала TCn следует записать новое значение кода для сравнения.

2. Если прерывания по флагу события CnF разрешены (бит CnI установлен), то генерируется запрос на прерывание.

3. Формирователь уровня генерирует на выводе канала IOSn предварительно заданный логический уровень. Формирователь уровня в процессе инициализации может быть программно настроен на один из четырех режимов работы:

• Установка вывода в 1;

• Установка вывода в 0;

• Инвертирование уровня сигнала на выводе;

• Соответствующий вывод МК не подключен к каналу выходного сравнения и является выводом порта T.

|  |  |
| --- | --- |
| **OMn: OLn** | **Режим формирователя уровня** |
| 00 | Выход формирователя не соединен с выводом IOSn |
| 01 | Инвертирует сигнал на выходе |
| 10 | Устанавливает выход в 0 |
| 11 | Устанавливает выход в 1 |

*Таблица 2.3. Выбор режима работы формирователя уровня*

Для настройки формирователя уровня каждого канала на один из трех перечисленных режимов работы используются биты OMn:OLn в регистрах TCTL1 ($0088) и TCTL2 ($0089). Формат этих регистров представлен на рис. 11. Таблица 3.3 устанавливает соответствие между режимом работы формирователя уровня и кодом инициализации в разрядах OMn:OLn.

Рассмотрим последовательность действий, которая должна быть реализована для генерации средствами подсистемы выходного сравнения «отрицательного» импульса заданной длительности. Под «отрицательным» импульсом мы будем понимать низкий логический уровень сигнала с последующим его изменением на высокий логический уровень.

1. Управляющая программа считывает текущее состояние счетчика временной базы из двухбайтового регистра TCNT.

2. Управляющая программа устанавливает выход используемого канала в 0.

3. Управляющая программа вычисляет длительность временного интервала, в течение которого на выходе должен удерживаться низкий логический уровень. Обратите внимание, полученная длительность временного интервала должна быть представлена в числе периодов тактирования счетчика временной базы.

4. Управляющая программа производит сложение ранее считанного кода счетчика и полученного кода длительности временного интервала. Полученное значение записывается в регистр данных канала выходного сравнения.

5. Управляющая программа задает режим работы формирователя уровня канала. Целесообразно выбрать режим установки выхода в 1, для чего необходимо записать в разряды OMn:OLn регистров TCTL1: TCTL2 комбинацию 11 (см. таблица 3.3).

6. Когда значение кода регистра данных совпадет с кодом счетчика временной базы, на выходе канала выходного сравнения автоматически, без участия программ установится 1.

**Канал 7 в режиме выходного сравнения**

До настоящего параграфа мы рассматривали все восемь каналов модуля таймера TIM, как полностью идентичные. Однако на самом деле полностью идентичными, работающими или в режиме входного захвата или в режиме выходного сравнения, являются лишь каналы 0…6. Канал 7 в режиме выходного сравнения обладает дополнительными возможностями. При этом рассмотренные режимы захвата/сравнения, совпадающие с режимами каналов 0…6, при соответствующей инициализации им полностью поддерживаются.

Если канал 7 находится в режиме выходного сравнения и его дополнительная функция разрешена, то в момент события в канале 7 устанавливается в назначенное состояние не только выход канала 7, но и любая комбинация выходов каналов 0…6. При этом выбранные для принудительной установки каналы также должны работать в режиме выходного сравнения. Два регистра специальных функций OC7M и OC7D используются для управления режимом принудительной установки по событию в канале 7. Каждый из битов регистра OC7M разрешает (при OC7Mn = 1) режим принудительной установки для одноименного (с номером n) канала. Соответствующий бит в регистре OC7D задает значение, которое должно быть установлено на выходе канала в момент принудительной установки. Например, если в регистре OC7M записан двоичный код 10110001, а в регистре OC7D – 01010101, то с каналом 7 в режиме принудительной установки будут связаны каналы 0, 4, 5. Тогда в момент события выходного сравнения на выходе канала 0 установится 1, на выходе канала 4 установится также 1, на выходе канала 5 установится 0 и на выходе канала 7 — тоже 0. На остальные каналы событие выходного сравнения в канале 7 влияния не окажет.

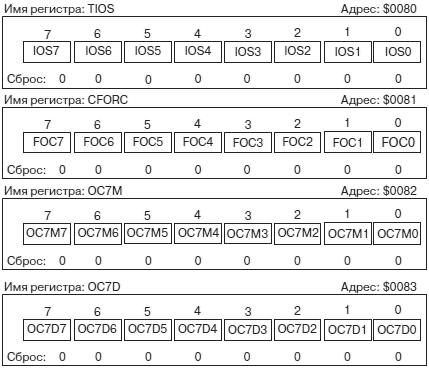
Свойство принудительной установки позволяет два раза за период работы счетчика временной базы изменять состояние на выходах каналов 0…6. Один раз по собственному событию выходного сравнения канала, второй раз — по событию в канале 7.

***Регистры для управления каналами захвата/сравнения***

В данном разделе мы рассмотрим регистры специальных функций, которые используются для управления режимами работы универсальных каналов захвата/сравнения. Формат регистров управления счетчиком временной базы в составе модуля таймера был рассмотрены ранее.

***Регистр режимов каналов захвата/сравнения***

Регистр режимов каналов захвата/сравнения TIOS (Timer Input capture/Output compare Register) располагается в памяти МК по адресу $0080. Формат регистра представлен на рис. 12. Каждый из битов регистра IOSn определяет режим работы канала с номером n. Если бит установлен в 1, то канал работает в режиме выходного сравнения. Если бит равен 0, то канал настроен на режим входного захвата.



*Рис. 12. Формат дополнительных регистров управления таймером: TIOS, CFORC, OC7M, OC7D*

***Регистр принудительного события выходного сравнения***

Регистр принудительного события выходного сравнения CFORC (TimerCompareForceRegister) располагается в памяти МК по адресу $0081. Формат регистра представлен на рис. 12. Установка бита FOCn в 1 немедленно вызывает событие выходного сравнения в канале с номером n, независимо от текущего состояния счетчика временной базы и регистра данных канала. Функция принудительной установки очень удобна для задания начального уровня выхода канала выходного сравнения.

***Регистр разрешения работы под управлением канала 7***

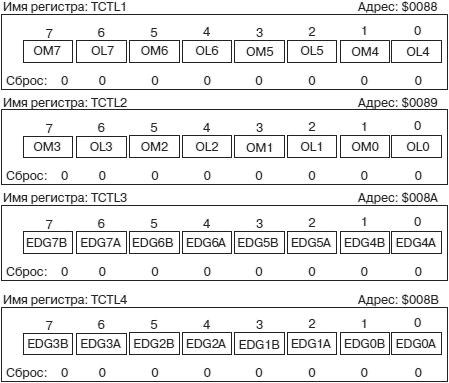
Регистр разрешения работы под управлением канала 7 OC7M (TimerOutputCompare 7 MaskRegister) располагается в памяти МК по адресу $0082. Формат регистра представлен на рис. 12. Каждый из битов регистра OC7M разрешает (при OC7Mn = 1) режим принудительной установки для одноименного (с номером n) канала по событию в канале выходного сравнения 7. Если бит OC7Mn = 0, то канал с номером n работает в автономном режиме.

***Регистр установки данных под управлением канала 7***

Регистр установки данных под управлением канала 7 OC7D (TimerOutputCompare 7 DataRegister) располагается в памяти МК по адресу $0083. Формат регистра представлен на рис. 12. Каждый из битов регистра OC7Dn задает значение, которое должно быть установлено на выходе канала с номером n в момент в момент события выходного сравнения в канале 7. При этом необходимо предварительно разрешить работу желаемых каналов под управлением канала 7 установкой соответствующих битов в регистре OC7M.

***Регистры управления таймером 1 и 2***

Регистры управления таймером TCTL1 и TCTL2 (Timer Control Register) располагаются в памяти МК по адресам $0088 и $0089. Форматы регистров приведены на рис. 13. Каждому каналу таймера поставлены в соответствие два бита OMn:OLn регистров TCTL1 и TCTL2. Биты OMn:OLn определяют один из четырех режимов работы формирователя уровня канала, если этот канал работает в режиме выходного сравнения. Комбинации кодов для битов OMn:OLn приведены на рис. 11.



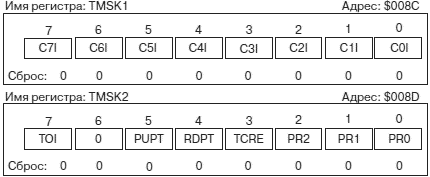
*Рис. 13. Формат основных регистров управления таймером:TCTL1, TCTL2, TCTL3, TCTL4*

**Регистры управления таймером 3 и 4**

Регистры управления таймером TCTL3 и TCTL4 (TimerControlRegister) располагаются в памяти МК по адресам $008A и $008B. Форматы регистров приведены на рис. 13. Каждому каналу таймера поставлены в соответствие два бита EDGnB:EDGnA регистров TCTL3 и TCTL4. Биты EDGnB:EDGnA определяют один из четырех режимов работы детектора события канала, если этот канал работает в режиме входного захвата. Комбинации кодов для битов EDGnB:EDGnAприведены на рис. 2.6.

**Регистр масок таймера 1**

Регистр масок таймера TMSK1 (TimerMaskRegister 1) располагается в памяти МК по адресу $008C. Формат регистра приведен на рис. 2.11. Каждый бит этого регистра CnI разрешает или запрещает прерывания по событию в одноименном (с номером n) канале таймера. Если бит CnI равен 1, то прерывания разрешены. При CnI = 0 прерывания по событию в канале запрещены.



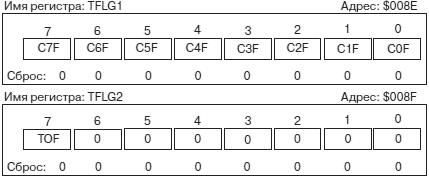
*Рис. 14. Формат регистров масок таймера:TMSK1, TMSK2*

**Регистр масок таймера 2**

Регистр масок таймера TMSK2 (Timer Mask Register 2) располагается в памяти МК по адресу $008D. Формат регистра приведен на рис. 14. Бит TOI разрешает прерывания по флагу переполнения счетчика временной базы TOF. Бит TCRE разрешает сброс счетчика временной базы. Биты PR2:PR1:PR0 устанавливают коэффициент деления программируемого делителя на входе счетчика временной базы в соответствие с таблицой1.

**Регистр флагов таймера 1**

Регистр флагов таймера TFLG1 (Timer Flag Register 1) располагается в памяти МК по адресу $008E. Формат регистра представлен на рис. 15. Каждому каналу таймера поставлен в соответствие флаг события CnF. Флаг CnF устанавливается в 1 автоматически, если в канале произошло событие входного захвата или выходного сравнения, в зависимости от текущего режима работы канала. Установленный бит события CnF вызовет прерывание, если в регистре TMSK1 установлен одноименный бит разрешения прерывания. Флаг события CnF должен быть сброшен под управлением программы, для чего в бит CnF должна быть записана 1. Существует альтернативный способ для сброса флагов события CnF. Если бит TFFCA в регистре TSCR установлен, то чтение или запись в регистр данных канала автоматически сбрасывает бит события этого канала.



*Рис. 15. Формат регистров флагов таймера:TFLG1, TFLG2*

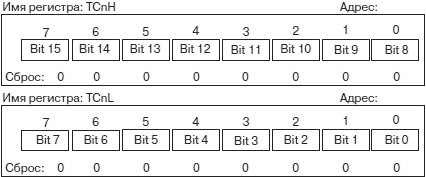
**Регистр флагов таймера 2**

Регистр флагов таймера TFLG2 (TimerFlagRegister 2) располагается в памяти МК по адресу $008F. Формат регистра представлен на рис. 15. В регистре присутствует всего один флаг — флаг переполнения счетчика временной базы TOF. Этот флаг сбрасывается посредством записи 1 в уже установленный бит TOF.

**Регистры данных каналов захвата/сравнения**

Регистры данных каналов захвата/сравнения TCn — 16 разрядные. В памяти каждый регистр представлен двумя 8 разрядными регистрами: TCnH — старший байт регистра данных канала с номером n, TCnL — младший байт регистра данных канала с номером n. Если канал настроен на режим входного захвата, то в регистре данных TCn содержится код счетчика временной базы в момент последнего события входного захвата. Если же канал настроен на режим выходного сравнения, то в регистр данных TCn под управлением программы записывается код момента сравнения. Формат и адреса расположения в памяти восьми регистров данных каналов TC0…TC7 приведены на рис. 16.

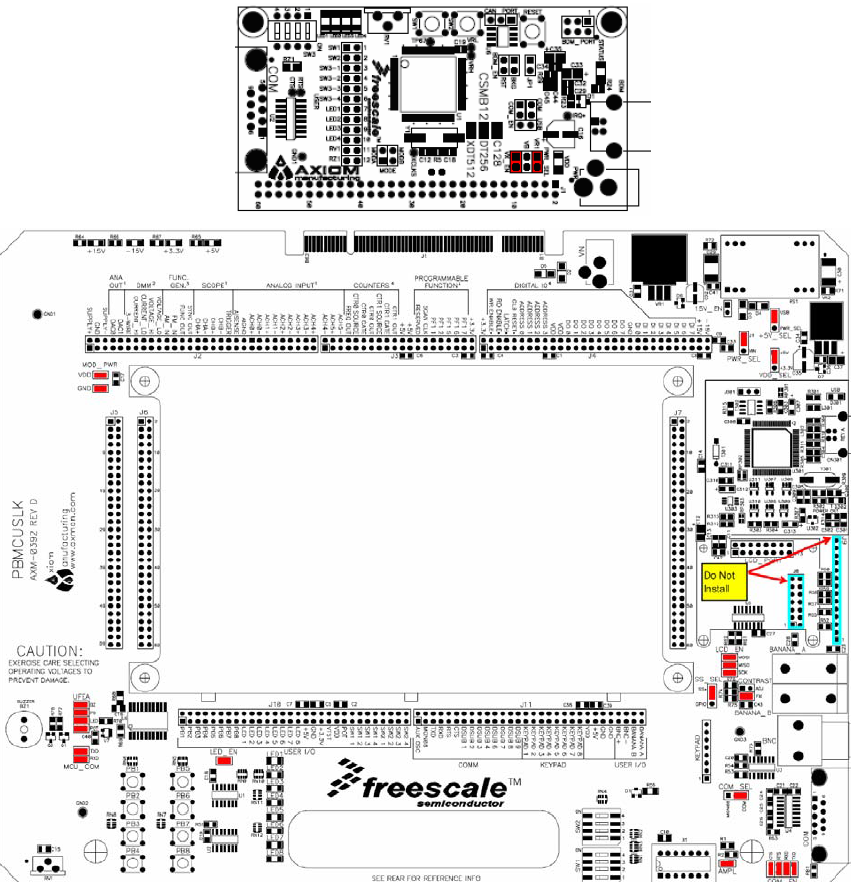
|  |  |
| --- | --- |
| Регистр данных канала 0: TC0H:TC0L | Адрес: $0090–0091 |
| Регистр данных канала 1: TC1H:TC1L | Адрес: $0092–0093 |
| Регистр данных канала 2: TC2H:TC2L | Адрес: $0094–0095 |
| Регистр данных канала 3: TC3H:TC3L | Адрес: $0096–0097 |
| Регистр данных канала 4: TC4H:TC4L | Адрес: $0098–0099 |
| Регистр данных канала 5: TC5H:TC5L | Адрес: $009A–009B |
| Регистр данных канала 6: TC6H:TC6L | Адрес: $009C–009D |
| Регистр данных канала 7: TC7H:TC7L | Адрес: $009E–009FФормат регистров данных таймера:TCnH, TCnL |



*Рис. 16*

# Порядок выполнения работы

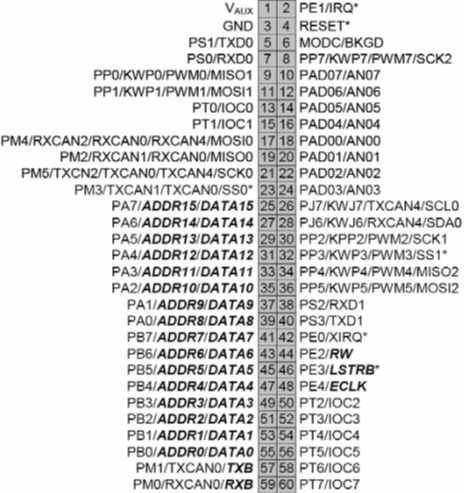
1. *Осторожно вставьте отладочную плату APS12C128 на платформу PBMCUSLK.*
2. *Соотвтственно с рисунком 17 ставим перемычки рабочей платы CSMB12С128 и PBMCUSLK.*

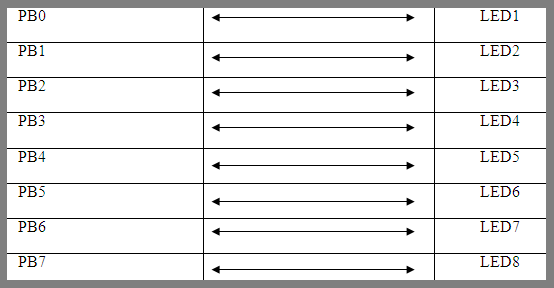
**

*Рис. 17. Конфигурация расположения перемычек на платах CSMB12С128 и PBMCUSL при их*

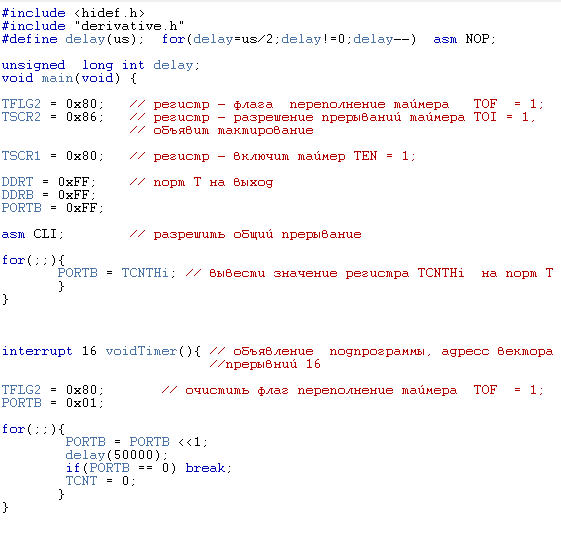
*совместном использовании с компьютером*

1. *Подсоединяем одладочную плату к компьютеру через порт USB*
2. *По ниже веденного схемы соедините порт В микроконтроллера к светодиодам платформы PBMCUSLK.*

**

**

1. *Проверьте переключатели платы APS12C128 платы. Они должны быть в выключенном виде.*
2. *Включите компьютер и дайте ему загрузиться.*
3. *По этому ссылку запускаем CodeWarrior: . Пуск – Все программы – Freescaler Code Warrior - CodeWarrior Development Studio for S12(X) – CodeWarrior IDE.*
4. *Создаем новый проект. В качестве языка программирования выбираем язык Си.*
5. *В окно текстового редактора рабочего среды CodeWarrior IDE, напишем ниже приведенный программный код.*



|  |
| --- |
|  |

1. *С помощью команды “MAKE” проверяем написанный код программы.*
2. *Если проверка прошло успешно, нажимаем “DEBUG”.*
3. *В появившимся новом окне под названием True-Time Simulator & Real-Time Debugge , нажимаем “OK” и после прошивки программы нажимаем кнопку “Start/Continue (F5)”.*
4. *Закрываем окно True-Time Simulator & Real-Time Debugger, и анализируем полученный результат.*

***Выполните следующие задания:***

1. Тактирование счета Таймера поставьте на 32, и проверьте загорание светодиодов. Какие изменения прослеживаются? Объясните.
2. С помощью Таймера выведите прямоугольный сигнал с частотой 15Гц.

***Вопросы для проверки:***

1. Что такое прерывание?
2. Регистры конфигураций таймера.
3. В чем отличие 10 разрядного таймера от 8 разрядного?
4. Обесните алгоритм работы таймера?

***Литература:***

1. [Иди Ф.](http://padabum.com/search.php?author=%D0%98%D0%B4%D0%B8%20%D0%A4.), Сетевой и межсетевой обмен данными с микроконтроллерами - Додэка-XXI,2007

# [Петров И.В.](http://padabum.com/search.php?author=%D0%9F%D0%B5%D1%82%D1%80%D0%BE%D0%B2%20%20%D0%98.%D0%92.), Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования - Солон-Пресс, 2004

# [Шпак Ю.А.](http://padabum.com/search.php?author=%D0%A8%D0%BF%D0%B0%D0%BA%20%D0%AE.%D0%90.), Программирование на языке С для AVR и PIC - МК-Пресс, 2-е издание, 2011

# [Голубцов М.С.](http://padabum.com/search.php?author=%D0%93%D0%BE%D0%BB%D1%83%D0%B1%D1%86%D0%BE%D0%B2%20%D0%9C.%D0%A1.), AVR - от простого к сложному - 2003

# С. Ф. Баррет., Д. Дж. Пак., Встраиваемые системы. Проектирование приложений на микроконтроллерах семейства 68HC12/HCS12 с применением языка С – 2006