**Блок Init\_PWM\_Timer(1|2|3) инициализации ШИМ**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блоки (1,2,3) реализует настройку широтно-импульсной модуляции с использованием аппаратных таймеров микроконтроллера, тактируемых

импульсами определенной частоты CLK. Таймер подсчитывает количество

тактовых импульсов и в нужные моменты формирует передний и задний

фронты.. Широтно-импульсной модуляцией (ШИМ, или, по-английски, PWM – Power Width Modulation) называют способ регулирования среднего напряжения, подаваемого на нагрузку, за счет изменения скважности импульсов.

В таких микроконтроллере K1986ВЕ92QI ШИМ реализуется на базе аппаратных 16-разрядных таймеров/счетчиков TIMER1, TIMER2 и TIMER3. Каждый из них позволяет организовать до 4 каналов ШИМ (т.е. всего можно сделать 12 каналов ШИМ). Для каждого из трех таймеров можно задать свою частоту ШИМ, а для каждого канала в пределах таймера – и свою длительность импульса. Кроме того, каждый из каналов ШИМ имеет два выхода, подведенных к тому или иному выводу микроконтроллера: прямой (D) и инверсный (N). Прямой выход формирует импульсы без инверсии, а инверсный, соответственно, – с инверсией.

Блок использует функцию Init\_PWM\_Timer(1|2|3) определенную в файле timer.h проекта шаблона сборки кода Test\_Driver. Пользователь может внести изменения в программный код функции.

**Входные порты**

**On** включить таймер

**Init\_port** инициализация каналов таймера (каждый из трех таймеров имеет свои каналы, они представлены в соотв. документации), канал представляет из себя линии ввода-вывода микропроцессора.

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера каналов** | **CHANALS** | Перечислены каналы (линии ввода вывода) используемые при ШИМ | 1 |
| **Период ШИМ** | **PWM\_Period** | Перио ШИМ | 8000 |
| **Заполненин импульса канала (1,2,3 и 4 для TIMER3)** | **CCR1...CCR4** | При работе в режиме ШИМ,  выходной сигнал может формироваться на основании сравнения значения в регистре CCR и  основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. | 4000 |
| **Предделитель частоты** | **PSG** | Значение предварительного делителя счетчика – TIMx\_PSG, при этом основной  счетчик будет считать на частоте CLK= TIMx\_CLK/(PSG+1); | 0 |
| **Включить прерывание** | **Intr** | После истечения периода ШИМ будет возникать прерывание, в обработчике которого будет уменшина на еденицу глобальная переменная fl\_pwm (fl\_pwm = fl\_pwm – 1 ). Это необходимо для того что бы основной поток программы мог выйти из функции ожидания прохождения периода ШИМ DelayTimer\_pwm() | Нет (нет прерывания) |
| **Режим основного счетчика** | **Centr\_PWM** | Центрированный ШИМ или обыкновенный | Нет (обыкновенный) |

**нет**

**Пример timer1.prt**

**Блок Init\_Timer3\_asCapture захват внешних импульсов**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блок реализует подсчет передних и задних фронтов измеряемой последовательности импульсов.

**Входные порты**

**On** включить таймер для начала подсчета внешних импульсов

**Init\_port** инициализация каналов таймера3 канал представляет из себя линии ввода-вывода микропроцессора и должны быть инициализированы с помощью блока настройки порта.

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**Crr1…crr4** массивы содержащие значения основного счетчика (регистр MDR\_TIMERx->CNT) при котором произошел факт захвата по положительноному фронту импульса, для каналов от 1 до 4 соответственно

**Crr11…crr41** массивы содержащие значения основного счетчика (регистр MDR\_TIMERx->CNT) при котором произошел факт захвата по отрицательному фронту, для каналов от 1 до 4 соответственно

**Num1…num4** текущийразмер массивов (**Crr1…crr4, Crr11…crr41**), максимальное значение задается в свойствах блока SIZE

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера каналов** | **CHANALS** | Перечислены каналы (линии ввода вывода) используемые при ШИМ | 1 |
| **Размер массива** | **SIZE** | Максимальный размер кольцевого массива | 1 |
| **Предделитель частоты** | **PSG** | Значение предварительного делителя счетчика – TIMx\_PSG, при этом основной  счетчик будет считать на частоте CLK= TIMx\_CLK/(PSG+1); | 0 |
| **Максимальное значение счетчика** | **ARR** | Максимальное значение счетчика, после достижения максимального значения он сбрасывается в 0 | 8000 |
| **Фильтр** | **Filtr** | Сигнал зафиксирован:  0000 – в 1 триггере на частоте TIM\_CLK;  0001 – в 2 триггерах на частоте TIM\_CLK;  0010 – в 4 триггерах на частоте TIM\_CLK;  0011 – в 8 триггерах на частоте TIM\_CLK;  0100 – в 6 триггерах на частоте FDTS/2;  0101 – в 8 триггерах на частоте FDTS/2;  0110 – в 6 триггерах на частоте FDTS/4;  0111 – в 8 триггерах на частоте FDTS/4;  1000 – в 6 триггерах на частоте FDTS/8;  1001 – в 8 триггерах на частоте FDTS/8;  1010 – в 5 триггерах на частоте FDTS/16;  1011 – в 6 триггерах на частоте FDTS/16;  1100 – в 8 триггерах на частоте FDTS/16;  1101 – в 5 триггерах на частоте FDTS/32;  1110 – в 6 триггерах на частоте FDTS/32;  1111 – в 8 триггерах на частоте FDTS/32 | 0 |
|  |  |  |  |

**Пример pwm\_t1\_t2\_t3.prt**

**Блок START\_TIMER(1|2|3) включить и выключить таймер**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блоки включают и выключают таймеры.

**Входные порты**

**On** включить таймер:

1 – включить таймер

0 – выключить таймер

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Блок START\_TIMERS включить и выключить таймеры**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блок включает и выключает все таймеры (TIMER1, TIMER2, TIMER3) одновременно.

**Входные порты**

**On** включить таймер:

1 – включить таймеры

0 – выключить таймер

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Блок PWM\_Period период ШИМ**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блок определяет период ШИМ инициализирует MDR\_TIMERx->ARR = ARR задается в импульсах.

**Входные порты**

**On** включить таймер:

1 – включить таймеры

0 – выключить таймер

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Таймер** | **TIMER** | Один из трех таймеров (TIMER1, TIMER2, TIMER3) | нет |

**Блок PWM\_State период ШИМ**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блок определяет значение для регистров MDR\_TIMERx->CCRy, изменяя значение хранящееся в данном регистре можно управлять ШИМ.

**Входные порты**

**crr** значение для управления ШИМ , MDR\_TIMERx->CCRy = crr

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Таймер** | **TIMER** | Один из трех таймеров (TIMER1, TIMER2, TIMER3) | нет |
| **Номера каналов** | **CHANALS** | Перечислены каналы (линии ввода вывода) используемые при ШИМ | 1 |

**Блок Init\_PWM\_Timer1\_plus\_ADC реализует ШИМ и опрос каналов АЦП**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да) |

Блоки (1,2,3) реализует настройку широтно-импульсной модуляции с использованием аппаратных таймеров микроконтроллера, тактируемых

импульсами определенной частоты CLK. Таймер подсчитывает количество

тактовых импульсов и в нужные моменты формирует передний и задний

фронты. По истечению периода ШИМ возникает прерывание результатом которого является выход из функции DelayTimer\_pwm() и в зависимости от свойств блока опрос канала выбранного АЦП (1|2). Смысл использования блока заключается в том, что перед самым началом цикла выполнения алгоритма происходит считывание значений каналов АЦП. Это необходимо для того, что бы разнести во времени получение измерений с помощью АЦП и работу других элементов блока управления с целью уменьшения уровня шумов в измерениях (например получаем с помощью АЦП значения токов на фазах двигателя и переключаем ключи на инверторе, два процесса лучше разнести по времени).

**Входные порты**

**On** включить таймер

**Init\_port** инициализация каналов таймера (каждый из трех таймеров имеет свои каналы, они представлены в соотв. документации), канал представляет из себя линии ввода-вывода микропроцессора и контроллера АЦП.

**Sort\_block** порт необходимый для определения последовательности сортировки блока

**Выходные порты**

**ready**  Возвращает 1, это необходимо для определения порядка сортировки последущих блоков

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера каналов** | **CHANALS** | Перечислены каналы (линии ввода вывода) используемые при ШИМ | 1 |
| **Период ШИМ** | **PWM\_Period** | Перио ШИМ | 8000 |
| **Заполненин импульса канала (1,2,3)** | **CCR1...CCR3** | При работе в режиме ШИМ,  выходной сигнал может формироваться на основании сравнения значения в регистре CCR и  основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. | 4000 |
| **Предделитель частоты** | **PSG** | Значение предварительного делителя счетчика – TIMx\_PSG, при этом основной  счетчик будет считать на частоте CLK= TIMx\_CLK/(PSG+1); | 0 |
| **Включить прерывание** | **Intr** | После истечения периода ШИМ будет возникать прерывание, в обработчике которого будет уменшина на еденицу глобальная переменная fl\_pwm (fl\_pwm = fl\_pwm – 1 ). Это необходимо для того что бы основной поток программы мог выйти из функции ожидания прохождения периода ШИМ DelayTimer\_pwm() | Нет (нет прерывания) |
| **Режим основного счетчика** | **Centr\_PWM** | Центрированный ШИМ или обыкновенный | Нет (обыкновенный) |
| **Использовать опрос АЦП каналов** | **ON\_ADC** | В обработчике прерываний использовать ли опрос каналов АЦП или не использовать | ДА |
| **Количество каналов АЦП** | **NUM\_ADC\_CH** | Количество последовательно опрашиваемых АЦП каналов | 1 |
| **Выбор контроллера АЦП** | **ADC** | Можно выбрать АЦП1 или АЦП2 | ADC1 |

**Примечание! Стоит отметить. Что прерывние от UART не работает одновременно с прерыванием от Timer1**

**Пример Timer\_adc.prt**