НАВИГАЦИЯ

Глава VII

Широтно-импульсная модуляция

Цель работы:

– получение представлений о процессе широтно-импульсной модуляции;

– получение навыков регулирования мощности с использованием цифровых сигналов;

– реализация широтно-импульсной модуляции на базе микроконтроллера.

Оборудование:

– отладочный комплект для микроконтроллера 1986ВЕ92У;

– программатор-отладчик J-LINK (или аналог);

– модуль расширения для работы с микроконтроллерной техникой;

– цифровой мультимет р;

– персональный компьютер.

Программное обеспечение:

– операционная система Windows 7 / 8 / 10;

– среда программирования Keil µVision MDK-ARM 5.24;

– драйвер программатора J-LINK;

– примеры кода программ.

7.1. Понятие широтно-импульсной модуляции

При проектировании систем управления нередко появляется задача регулирования мощности, подводимой к некоторой нагрузке. Это позволяет, например, изменить яркость свечения светодиода или лампы накаливания, скорость вращения вала электродвигателя, температуру электронагревателя.

На рисунке 7.1 приведена упрощенная функциональная схема управления мощностью, подводимой к нагрузке.

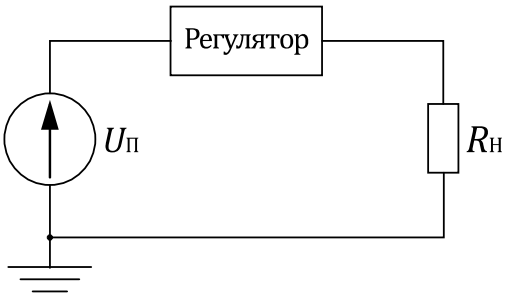

Рисунок 7.1 – Схема регулирования мощности, подводимой к нагрузке

Схема состоит из трех элементов: источника питания постоянного тока $E_П$, сопротивления нагрузки $R_Н$ (эквивалент управляемого устройства) и регулятора. Т.е. источник питания вырабатывает для нагрузки некоторую мощность, а регулятор позволяет ее изменять.

Регулятор может быть реализован по-разному. Самым очевидным вариантом является использование полевого транзистора с управляющим p-n переходом. В этом случае мощность, подводимая к нагрузке, будет определяться напряжением на затворе

транзистора; этим напряжением можно управлять, например, с помощью цифро-аналогового преобразователя.

Но недостатком такой схемы является низкий КПД, особенно при малой относительной мощности. Дело в том, что мощность, невостребованная нагрузкой, выделяется на транзисторе в виде тепла. Поэтому если мощность на нагрузке должна быть примерно равна мощности источника, то рассеивания на транзисторе практически не происходит, и схема работает с высоким КПД. Если же мощность на нагрузке должна быть значительно меньше мощности источника, то разность этих мощностей будет рассеиваться на транзисторе, и КПД схемы будет низким.

Альтернативным способом регулирования мощности является **широтно-импульсная модуляция** (**ШИМ**, англ. Pulse-Width Modulation, **PWM**). Идея способа заключается в подаче на нагрузку $R_Н$ прямоугольных импульсов длиной τ и периодом T (рисунок 7.2).

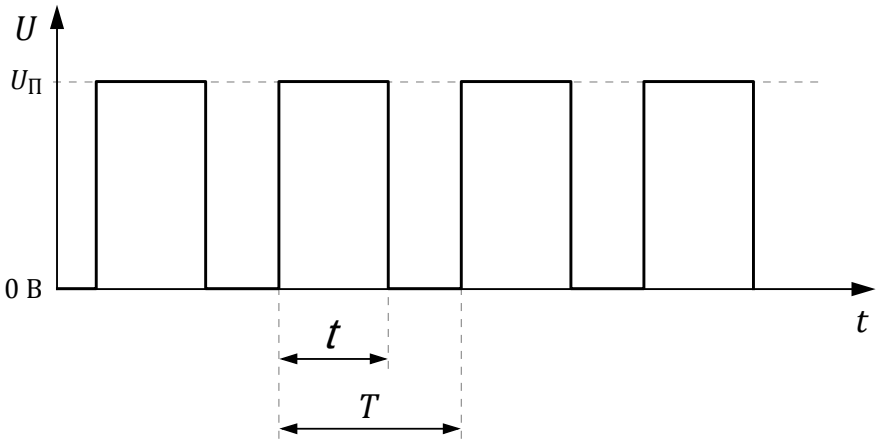

Рисунок 7.2 – Прямоугольные импульсы

Для этого используют транзистор, работающий в **ключевом режиме** и обеспечивающий только два устойчивых состояния: открытое и закрытое. При этом нагрузка будет как бы периодически подключаться к источнику питания $U_П$ на время τ, а затем отключаться от него на оставшуюся часть периода ($T - \tau$). Тогда средняя мощность $P_Н$, подводимая к нагрузке, будет определяться формулой:

$$\bar{P}_Н = \frac{U_П^2}{R} \times \frac{\tau}{T}. \qquad (7.1)$$

В таком случае получается, что, **изменяя длительность (ширину) импульсов, можно регулировать среднюю мощность, потребляемую нагрузкой**: чем выше длительность импульсов τ, тем выше мощность $P_Н$.

Отношение длительности импульсов τ к их периоду T называют **коэффициентом заполнения** (англ. Duty Cycle); его обычно выражают в процентах:

$$D = \frac{\tau}{T} \times 100\%. \qquad (7.2)$$

Коэффициент заполнения показывает, какой процент максимальной мощности потребляется нагрузкой.

Существует также сходный по смыслу термин «скважность». **Скважность** – это отношение периода следования импульсов T к длительности импульсов τ, т.е. величина, обратно-пропорциональная коэффициенту заполнения:

$$S = \frac{T}{\tau} = \frac{1}{D}. \tag{7.3}$$

Скважность характеризует расстояние между импульсами, как бы величину скважин сигнала.

В большинстве случаев использование коэффициента заполнения практически более удобно, чем скважности, поскольку его относительно изменение происходит в диапазоне от 0 до 1, тогда как соответствующая скважность изменяется от бесконечности до 1.

На рисунках 7.3а и 7.3б для примера показаны осциллограммы импульсов с разными коэффициентами заполнения.

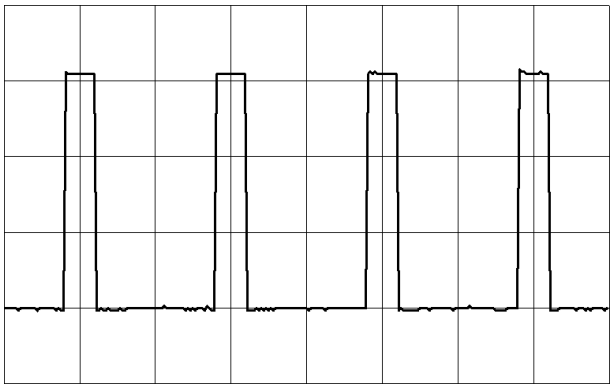

Рисунок 7.3а – Осциллограмма импульсов с низким коэффициентом заполнения (D = 20%, S = 5.00)

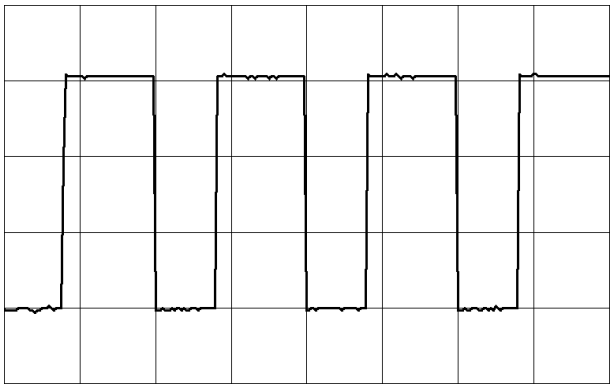

Рисунок 7.3б – Осциллограмма импульсов с высоким коэффициентом заполнения (D = 70%, S = 1.43)

Широтно-импульсная модуляция обеспечивает **высокий КПД** за счет того, что транзистор, используемый для коммутации, работает в режиме ключа:

– когда он закрыт (т.е. имеет очень высокое внутренне сопротивление), ток через него практически не течет и мощность, соответственно, не потребляется;

– когда он открыт (т.е. имеет очень малое внутреннее сопротивление), ток течет через него беспрепятственно, и мощность на нем не рассеивается в виде тепла.

Для примера выполним расчет КПД системы для элементов модуля расширения: транзистора КП505А ($R_{\text{откр.}}$ = 0.5 Ом) и лампы накаливания ($U_{\text{лампы}}$ = 3.5 В, I = 0.2 А).

Сразу нужно отметить, что подключать лампу (или иную нагрузку) напрямую к микроконтроллеру нельзя, т.к. цифровые линии микроконтроллеров серии 1986ВЕ9х способны вырабатывать ток до 6 мА, т.е. многократно меньше требуемого значения. Поэтому подключать нагрузку следует через N-канальный полевой транзистор с общим стоком, затвор которого управляется микроконтроллером (рисунок 7.4).

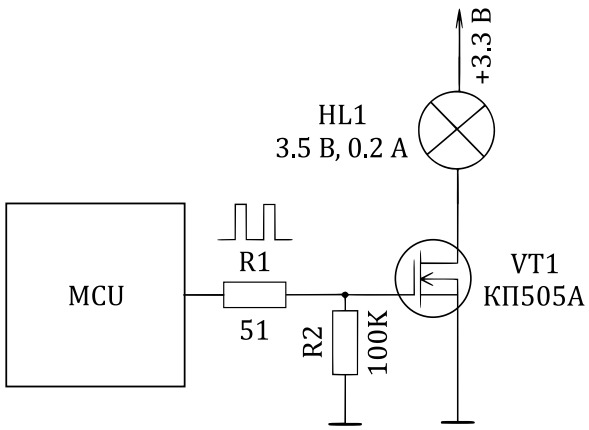

Рисунок 7.4 – Схема подключения лампы накаливания к микроконтроллеру

КПД (η) определяется, как отношение полезной мощности системы к затраченной, т.е. мощности, потребляемой лампой ($P_{\text{лампы}}$), к сумме этой мощности и мощности, рассеиваемой на транзисторе ($P_{\text{транз.}}$).

Мощность, потребляемая лампой при напряжении питания $U_{\text{П}}$ = 3.3 В, будет равна:

$$P_{\text{лампы}} = I \times \left(U_{\text{П}} - U_{\text{транз.}}\right) = I \times \left(U_{\text{П}} - I \cdot R_{\text{откр.}}\right) = 0.2\text{А} \times (3.3\text{В} - 0.2\text{А} \times 0.5\text{Ом}) = 0.64\text{Вт}.$$

Мощность, рассеиваемая на транзисторе, составит:

$$P_{\text{транз.}} = I \times U_{\text{транз.}} = I^2 \times R_{\text{откр.}} = (0.2\ \text{А})^2 \cdot 0.5\ \text{Ом} = 0.02\ \text{Вт}.$$

Тогда КПД системы будет равен:

$$\eta = \frac{P_{\text{лампы}}}{P_{\text{лампы}} + P_{\text{транз.}}} \times 100\% = \frac{0.64\ \text{Вт}}{0.64\ \text{Вт} + 0.02\ \text{Вт}} \times 100\% \approx 97\%.$$

Следует отметить, что такой КПД будет сохраняться при любой скважности импульсов, и, следовательно, при любой требуемой средней мощности, подводимой к нагрузке. Но с ростом частоты следования импульсов КПД будет снижаться.

7.2. Выбор частоты импульсов

Выбор частоты следования импульсов – важный аспект реализация широтно-импульсной модуляции. Её следует выбирать из определенного диапазона.

С одной стороны, **слишком низкая частота** может стать причиной следующих проблем:

1. **Пульсация тока.** Если рассмотреть предельный случай, когда частота следования импульсов будет равна 1 Гц, то невооруженным глазом будут заметны колебания мощности на нагрузке: лампа или светодиод, к примеру, будут мигать, вал двигателя – дёргаться; нагревательные элементы будут быстро выходить из строя из-за частых температурных колебаний. Все это, очевидно, неприемлемо.

Если касаться освещения, то интересно отметить, что человеческий глаз способен замечать колебания до 50 Гц прямым зрением (если рассматривать объект в упор) и до 72 Гц – периферийным (если смотреть вдоль объекта). В экспериментах, однако, надо

иметь в виду, что лампа накаливания **инертна**: нити накала требуется время на нагрев и остывание, поэтому колебания будут незаметны и на меньших частотах. При этом нужно понимать, что даже неразличимые глазом световые пульсации, тем не менее, регистрируются сетчаткой и мозгом, и могут приводить к усталости глаз и повышенной утомляемости человека в целом. Согласно последним исследованиям, негативное влияние световых пульсаций полностью исчезает на частотах более 300 Гц. [x]

2. **Переполнение аппаратных таймеров.** Забегая вперед необходимо сказать, что широтно-импульсная модуляция на базе микроконтроллеров реализуется с помощью аппаратных таймеров, имеющих ограниченную разрядность. Период модуляции должен укладываться в период перезагрузки таймера. В принципе, за счет предделителя период перезагрузки задается достаточно гибко, но он определяет и верхнюю границу допустимой частоты, поэтому при разработке программы это нужно обязательно иметь в виду. Данный аспект будет подробно рассмотрен в разделе 7.3.1.

3. **Шумы вибраций.** При управлении двигателями с использованием широтно-импульсной модуляции на частоте, соответствующей слышимому частотному диапазону человека (до ~20 КГц), могут возникать шумы, похожие на свист; они обусловлены вибрацией обмоток. Такие шумы являются сильным раздражителем слуха, поэтому если конечное устройство планируется использовать в постоянной близости с человеком, то лучше бы использовать бóльшую частоту импульсов.

С другой стороны, **слишком высокая частота** может привести к ряду других проблем:

1. Дополнительные потери мощности на транзисторе. Между затвором и истоком транзистора всегда имеется паразитная емкость, не позволяющая транзистору мгновенно открыться или закрыться: для этого емкость сначала должна зарядиться или разрядиться. Это означает, что прямоугольные импульсы на практике всегда являются **трапециевидными**. Процесс перезаряда емкости называют **переходным**, а режим работы транзистора в это время – **активным**. За время переходного процесса на транзисторе падает в среднем до половины напряжения питания, а, следовательно, рассеивается и до половины мощности.

Чем ниже частота импульсов, тем незначительнее часть периода, которую занимает переходной процесс. Но с ростом частоты время переходного процесса может стать соизмеримо с шириной импульсов; это приведет к большим потерям мощности на транзисторе: КПД упадет, а транзистор будет сильно греться.

Так, например, для транзистора КП505А при $U_{ЗИ}$ = 10 В, $U_{СИ}$ = 30 В время открытия составляет 33 наносекунды, время закрытия – 180 наносекунд. С учетом специфики схемы модуля расширения (наличие пассивных элементов, напряжение питания 3.3 В) эти значения будут ниже – примерно 21 и 113 наносекунд соответственно. Эти данные позволяют сделать вывод, что если при коэффициенте заполнения D = 20% период сигнала составит $T = (21\text{ нс} + 113\text{ нс}) \cdot 100\%/D = 0.67$ мкс, т.е. если частота импульсов будет равна $F \approx 1.5$ МГц, то фронты завалятся настолько, что импульсы станут треугольными (рисунок 7.5б); КПД в этом случае снизится до ~50%. Если использовать еще более высокую частоту или снизить коэффициент заполнения, то транзистор вовсе перестанет открываться: напряжение на затворе не будет успевать подниматься до порога открытия.

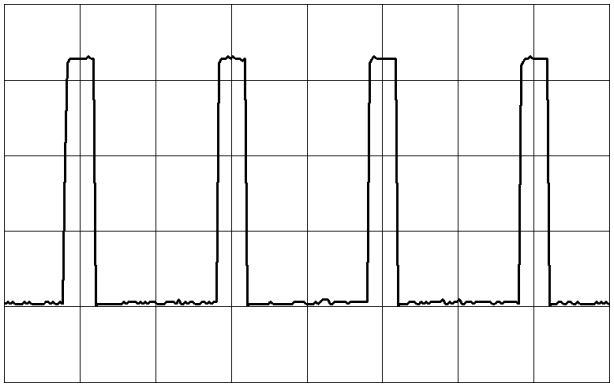

Рисунок 7.5а – Осциллограмма импульсов на истоке транзистора (*F* = 10 КГц, *D* = 20%)

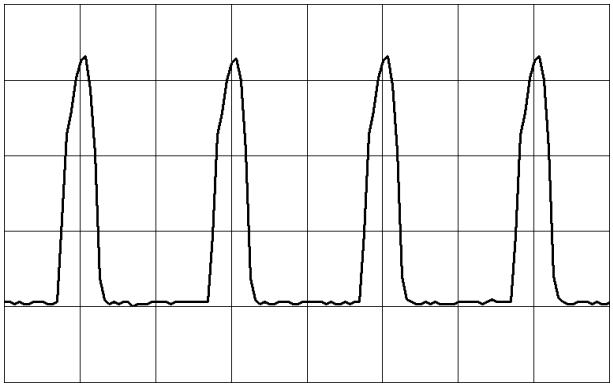

Рисунок 7.5б – Осциллограмма импульсов на истоке транзистора (*F* = 1.5 МГц, *D* = 20%)

2. Снижение вариативности в выборе скважности. Эта проблема также обусловлена возможностями аппаратных таймеров. Идея в том, что таймер подсчитывает тактовые импульсы, и в нужные моменты формирует фронты модулированного сигнала. Чем больше тактов приходится на период сигнала, тем больше вариантов для положения фронта и, соответственно, выбора скважности. Этот момент будет рассмотрен в разделе 7.3.1, а сейчас можно лишь отметить, что максимальная частота широтно-импульсной модуляции при тактовой частоте ядра 80 МГц и сохранении возможности задания коэффициента заполнения с шагом 1% составляет ~792 КГц.

3. Создание электрических помех в системе. Широтно-импульсная модуляция высокой частоты может стать источником сильных электрических помех в системе.

Помехи – это отдельная тема, которой посвящен не один десяток книг. Сейчас же следует просто иметь в виду, что они – это одна из причин, по которой не следует использовать слишком высокую частоту импульсов.

Все перечисленные соображения необходимо учитывать при разработке системы с широтно-импульсной модуляцией. На практике же ее частоту выбирают из диапазона от 10 до 200 КГц.

7.3. Реализация широтно-импульсной модуляции на базе микроконтроллеров серии 1986ВЕ9х

Существует два способа реализация широтно-импульсной модуляции на базе микроконтроллеров: программный и аппаратный.

Программный способ сводится к тривиальному циклическому управлению линией ввода-вывода: для создания переднего фронта импульса на линии создается логическая единица, заднего – логический ноль. Длительность импульса и период сигнала могут быть организованы различными средствами, например, системным таймером.

Программная реализация ШИМ требует много ресурсов вычислительного ядра и обладает низкой точностью; она используется только при работе с устройствами, не имеющими встроенного ШИМ-контроллера. Данный способ не будет подробно рассматриваться по причине его нерациональности.

Аппаратный способ задействует для формирования модулированного сигналы таймера общего назначения. Таймеры позволяют генерировать импульсы с высокой точностью, и после инициализации работают полностью автономно, без участия ядра.

В микроконтроллеры серии 1986ВЕ9х встроено **три таймера** общего назначения; каждый таймер имеет **четыре канала** для генерации модулированного сигнала (т.е. суммарно можно организовать до 12 каналов). **Для каждого из трех таймеров можно задать собственную частоту импульсов, а для каждого канала в пределах таймера – собственную скважность.** Кроме того, каждый канал имеет **два вывода**: прямой (*D*) и инверсный (*N*).

Идея заключается в том, после требуемой инициализации уровень напряжения на канале начинает определяться состоянием таймера. Для задания нужных параметров используются три регистра: уже знакомые нам *CNT* и *ARR*, а также регистр сравнения *CCR*. В простейшем случае реализации широтно-импульсной модуляции таймер инкрементируется от 0 до значения *ARR*, а для канала выбирается такой режим работы:

– U_{CH} = 3.3 В при $CNT < CCR$;

– U_{CH} = 0 В при $CCR \geq CNT \geq ARR$.

Графическое представление этого процесса изображено на рисунке 7.6.

Из рисунка 7.5 должно стать очевидно, что период импульсов определяется значением регистра *ARR*, а скважность – значением регистра *CCR*. О том, как связать эти значения с конкретной частотой речь пойдет в следующем разделе.

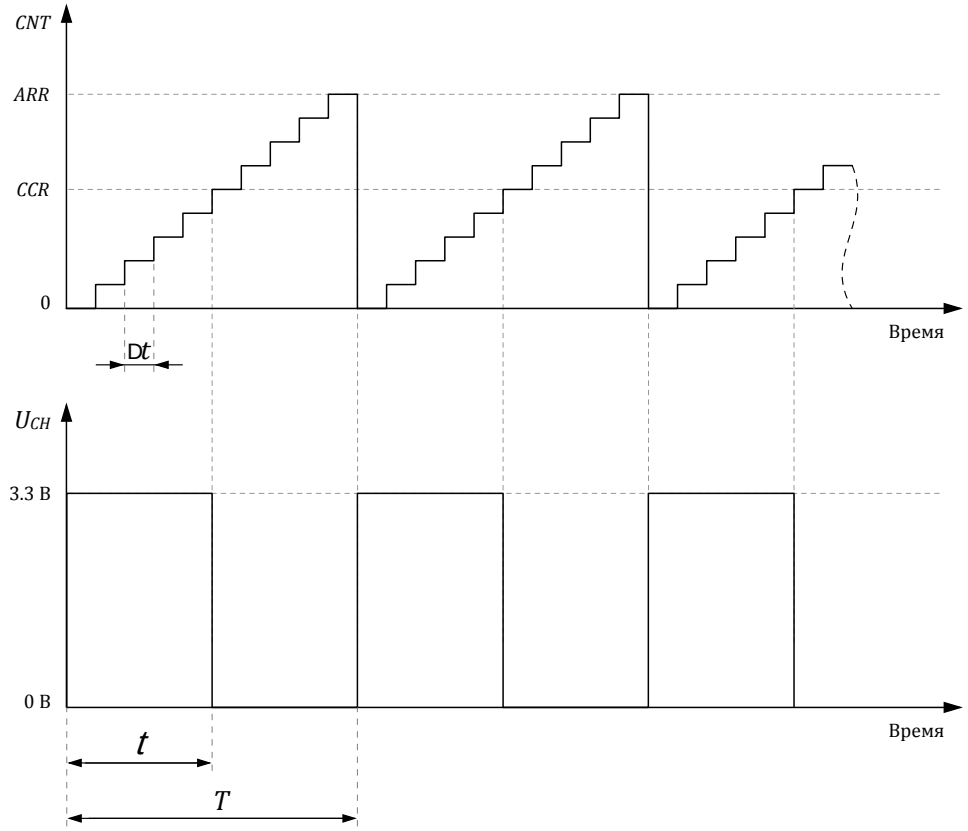

Рисунок 7.6 – Изменение напряжения на канале
в зависимости от состоянии таймера

7.3.1. Задание требуемой частоты импульсов

Частота импульсов определяется значениями двух регистров: предделителя тактовой частоты *PSG* и автоматической перезагрузки *ARR*. Метод расчета здесь может быть следующим.

С одной стороны, частота импульсов тривиально выражается через их период:

$$F = \frac{1}{T}. \tag{7.4}$$

С другой стороны, период импульсов T можно определить, как сумму отрезков времени Δt, требуемых на каждое изменение счетчика:

$$T = \Delta t \times (ARR + 1). \tag{7.5}$$

Обратите внимание, что количество временных отрезков на единицу больше, чем значение *ARR*, т.к. отсчет ведется с нуля.

Время Δt зависит от тактовой частоты ядра *SystemCoreClock* и выбранного предделителя *PSG*:

$$\Delta t = \frac{PSG + 1}{SystemCoreClock}. \tag{7.6}$$

Тут также следует помнить, что фактический предделитель тактовой частоты будет на единицу больше заданного.

Комбинируя формулы 7.4 – 7.6 можно получить такую зависимость:

$$F = \frac{SystemCoreClock}{(PSG + 1) \times (ARR + 1)}. \tag{7.7}$$

Используя формулу 7.7 можно найти требуемые значения *ARR* и *PSG* для получения произвольной частоты. Однако при этом получается уравнение с двумя неизвестными. К его решению можно подойти по-разному; один из способов – выразить из формулы 7.7 значение *ARR*:

$$ARR = \frac{SystemCoreClock}{F \times (PSG + 1)} - 1, \tag{7.8}$$

и рассчитать его для некоторого значения частоты при *PSG* = 0. Значение *ARR* ограничено разрядностью таймера – $(2^{16} - 1) = 65\,535$, поэтому если результат вычислений окажется больше указанного значения, то следует увеличить предделитель и выполнить повторный расчет. Например, если требуется обеспечить частоту импульсов *F* = 1 КГц, то:

$$ARR = \frac{80 \times 10^6}{10^3 \times (0 + 1)} - 1 = 79\,999.$$

Полученное значение больше 65 535, поэтому необходимо увеличить **фактическое** значение предделителя по крайней мере в два раза (PSG = 1), тогда

$$ARR = \frac{80 \times 10^6}{10^3 \times (1 + 1)} - 1 = 39\,999.$$

По формуле 7.7 также можно определить минимальную и максимальную частоту при заданном предделителе. **Минимальная частота** будет достигаться при максимальном значении периода, т.е. при ARR = 65 535:

$$F_{min} = \frac{SystemCoreClock}{65\,536 \times (PSG + 1)}. \quad (7.9)$$

Максимальная частота, напротив, будет достигаться при минимальном значении периода, но здесь есть один нюанс: если установить значение ARR = 1, то для выбора коэффициента заполнения импульсов не останется вариантов: он сможет быть либо 0%, либо 100%. По этой причине при расчете максимальной частоты следует выбирать значение ARR так, чтобы обеспечить возможность задания заполнения с шагом хотя бы в 1%, т.е. ARR = 100:

$$F_{max} = \frac{SystemCoreClock}{101 \times (PSG + 1)}. \quad (7.10)$$

Ключевую формулу 7.8 нетрудно программно реализовать в виде макроса:

```
// Фактический предделитель тактовой частоты таймера TIMER1
#define TIMER1_PRESCALER (1 + 1)

// Требуемая частота широтно-импульсной модуляции (в Гц)
#define PULSE_FREQUENCY 1000

// Макрос для расчета периода импульсов в тактах таймера
#define PULSE_PERIOD(F) (uint16_t)(SystemCoreClock /           \
                        ((uint32_t)F * TIMER1_PRESCALER) - 1)
```

7.3.2. Инициализация таймера в режиме широтно-импульсной модуляции

Рассмотрим процесс инициализации **первого канала первого таймера**.

Инициализация таймера в режиме широтно-импульсной модуляции состоит из следующих шагов:

1. Включение тактирования.
2. Деинициализация.
3. Конфигурация таймера.
4. Конфигурация порта ввода-вывода.
5. Конфигурация каналов таймера.
6. Конфигурация линий каналов.
7. Конфигурация регистров сравнения.

1, 2, 3. Включение тактирования, **деинициализация** и **конфигурация таймера** – стандартные процедуры; они полностью идентичны приведенным в разделе 3.3.2. Особого внимания заслуживают лишь регистры предделителя тактовой частоты *PSG* и автоматической перезагрузки *ARR*, т.к. они непосредственно определяют частоту генерируемых импульсов:

```
// Включение тактирования таймера TIMER1
MDR_RST_CLK->PER_CLOCK |= (1 << RST_CLK_PCLK_TIMER1_Pos);
MDR_RST_CLK->TIM_CLOCK |= (1 << RST_CLK_TIM_CLOCK_TIM1_CLK_EN_Pos);

// Деинициализация таймера TIMER1
// (тело процедуры приведено в разделе 3.3.2)
TIMER_Reset(MDR_TIMER1);

// Предделитель тактовой частоты таймера TIMER1
MDR_TIMER1->PSG = TIMER1_PRESCALER - 1;

// Значение автоматической перезагрузки таймера
MDR_TIMER1->ARR = PULSE_PERIOD(PULSE_FREQUENCY);

// Общая конфигурация таймера
MDR_TIMER1->CNTRL =
  (0 << TIMER_CNTRL_CNT_EN_Pos)      // Работа таймера (пока отключен)
| (1 << TIMER_CNTRL_ARRB_EN_Pos)     // Режим обновления регистра ARR (при перезагрузке)
| (0 << TIMER_CNTRL_DIR_Pos)         // Направление счета (прямой счет)
| (0 << TIMER_CNTRL_FDTS_Pos)        // Частота выборки (не используется)
| (0 << TIMER_CNTRL_CNT_MODE_Pos)    // Режим счета (такт. импульсы с фикс. напр-ем)
| (0 << TIMER_CNTRL_EVENT_SEL_Pos);  // Триггер счета (тактовые импульсы)
```

4. Конфигурация порта ввода-вывода необходима, чтобы превратить линию в канал таймера. В соответствии с таблицей П.1, канал *TMR1_CH1* соединен с линиями *PA1*, *PD1* и *PF6*. Можно выбрать любую из них, однако линии *PA1* и *PD6* использовать нежелательно, т.к. они выполняют функции дисплея и JTAG-интерфейса отладочной платы соответственно. Поэтому следует настроить линию PF6 на работу в режиме альтернативной функции:

```
// Включение тактирования порта F
MDR_RST_CLK->PER_CLOCK |= (1 << RST_CLK_PCLK_PORTF_Pos);

// Конфигурация линии PF6 для работы в качестве канала таймера
MDR_PORTF->OE     |=  (1 << 6);   // Направление линии (вывод)
MDR_PORTF->FUNC   &= ~(3 << 12);  // Сброс битов регистра FUNC
MDR_PORTF->FUNC   |=  (2 << 12);  // Функция линии (альтернативная)
MDR_PORTF->ANALOG |=  (1 << 6);   // Режим работы линии (цифровой)
MDR_PORTF->PULL   &= ~(1 << 6);   // Подтяжка к цепи питания (отключена)
MDR_PORTF->PULL   &= ~(1 << 22);  // Подтяжка к земле (отключена)
MDR_PORTF->PD     &= ~(1 << 6);   // Управление линией (драйвер)
MDR_PORTF->PD     &= ~(1 << 22);  // Триггер Шмитта (отключен)
MDR_PORTF->PWR    |=  (3 << 12);  // Крутизна импульсов (высокая)
MDR_PORTF->GFEN   &= ~(1 << 6);   // Цифровой фильтр (отключен)
```

5. Конфигурация каналов таймеров выполняется через набор регистров `MDR_TIMERx->CHy_CNTRL`, где `x` – номер таймера, `y` – номер канала (таблица 7.1).

Таблица 7.1 – Описание регистров MDR_TIMERx->CHy_CNTRL

№ битов	Функциональное имя битов	Описание
31...16	–	–
15	CAP_nPWM	Режим работы канала: 0 – широтно-импульсная модуляция; 1 – захват.
14	WR_CMPL	Флаг процесса записи данных в регистр *CCR*: 0 – запись данных завершена; 1 – запись данных в процессе.
13	ETREN	Остановка импульсов по сигналу на линии *ETR*: 0 – остановка запрещена; 1 – остановка разрешена.
12	BRKEN	Остановка импульсов по сигналу на линии *BRK*: 0 – остановка запрещена; 1 – остановка разрешена.
11...9	OCCM[2:0]	Формат выходного сигнала *REF* в режиме широтно-импульсной модуляции (таблица 7.2).
8	OCCE	Работа линии *ETR*: 0 – работа запрещена; 1 – работа разрешена.
7...6	CHPSC[1:0]	Предделитель частоты входного сигнала: 00 – предделитель отключен; 01 – / 2; 10 – / 4; 11 – / 8.
5...4	CHSEL[1:0]	Выбор события для фиксации счетчика в регистр CCR: 00 – передний фронт на конфигурируемом канале; 01 – задний фронт на конфигурируемом канале; 10 – передний фронт на других каналах (вариант 1); 11 – передний фронт на других каналах (вариант 2).
3...0	CHFLTR[3...0]	Фильтрация входного сигнала. Функция будет рассмотрена в главе VIII.

Из всех параметров регистра для реализации широтно-импульсной модуляции необходимо настроить лишь два: режим работы и формат сигнала *REF*:

```
// Конфигурация канала CH1 таймера TIMER1
MDR_TIMER1->CH1_CNTRL =
  (0 << TIMER_CH_CNTRL_CHFLTR_Pos)     // Фильтрация входного сигнала (-)
| (0 << TIMER_CH_CNTRL_CHSEL_Pos)      // Событие для фиксации счетчика в регистр CCR (-)
| (0 << TIMER_CH_CNTRL_CHPSC_Pos)      // Предделитель частоты входного сигнала (-)
| (0 << TIMER_CH_CNTRL_OCCE_Pos)       // Использование сигнала ETR (-)
| (6 << TIMER_CH_CNTRL_OCCM_Pos)       // Формат выходного сигнала REF (6)
| (0 << TIMER_CH_CNTRL_BRKEN_Pos)      // Сброс по сигналу BRK (-)
| (0 << TIMER_CH_CNTRL_ETREN_Pos)      // Сброс по сигналу ETR (-)
| (0 << TIMER_CH_CNTRL_CAP_NPWM_Pos);  // Режим работы канала (ШИМ)
```

Формат выходного сигнала *REF* определяет алгоритм работы канала. Все возможные форматы приведены в таблице 7.2.

Таблица 7.2 – Форматы сигнала *REF*

Номер формата	**Значение выходного сигнала**	
	При *CCR1_EN* = 0	**При *CCR1_EN* = 1**
000 [0]	Всегда «0»	Всегда «0»
001 [1]	«1» при *CNT* = *CCR*	«1» при *CNT* = *CCR* или *CNT* = *CCR1*
010 [2]	«0» при *CNT* = *CCR*	«0» при *CNT* = *CCR* или *CNT* = *CCR1*
011 [3]	Инверсия при *CNT* = *CCR*	Инверсия при *CNT* = *CCR* или *CNT* = *CCR1*
100 [4]	Всегда «0»	Всегда «0»
101 [5]	Всегда «1»	Всегда «1»
110 [6]	Прямой счет (*DIR* = 0): «1» при *CNT* < *CCR*; «0» при *CNT* ≥ *CCR*. Обратный счет (*DIR* = 1): «1» при *CNT* ≤ *CCR*; «0» при *CNT* > *CCR*.	Прямой счет (*DIR* = 0): «0» при *CNT* ≤ *CCR* ∨ *CNT* ≥ *CCR1*; «1» при *CCR* < *CNT* < *CCR1*. Обратный счет (*DIR* = 1): «1» при *CNT* ≤ *CCR* ∨ *CNT* ≥ *CCR1*; «0» при *CCR* < *CNT* < *CCR1*.
111 [7]	Прямой счет (*DIR* = 0): «0» при *CNT* < *CCR*; «1» при *CNT* ≥ *CCR*. Обратный счет (*DIR* = 1): «0» при *CNT* ≤ *CCR*; «1» при *CNT* > *CCR*.	Прямой счет (*DIR* = 0): «1» при *CNT* ≤ *CCR* ∨ *CNT* ≥ *CCR1*; «0» при *CCR* < *CNT* < *CCR1*. Обратный счет (*DIR* = 1): «0» при *CNT* ≤ *CCR* ∨ *CNT* ≥ *CCR1*; «1» при *CCR* < *CNT* < *CCR1*.

Для формирования прямоугольных импульсов достаточно использовать только один регистр сравнения (т.е. *CCR1_EN* = 0). При этом **формат 6** выбирают при **пилообразном опорном сигнале** с таймера (однонаправленный счет со сбросом, рисунок 7.5), а **формат 7** – при **треугольном опорном сигнале** (двунаправленный счет). Первый вариант используется гораздо чаще; он также проще для понимания, поэтому далее речь пойдет именно о нем.

6. Конфигурация линий каналов также выполняется через набор регистров `MDR_TIMERx->CHy_CNTRL1`, где `x` – номер таймера, `y` – номер канала (таблица 7.3).

Таблица 7.3 – Описание регистров `MDR_TIMERx->CHy_CNTRL1`

№ битов	**Функциональное имя битов**	**Описание**
31…13	–	–
12	NINV	Инверсия инверсной линии: 0 – линия не инвертируется; 1 – линия инвертируется.
11…10	NSELO[1:0]	Режим работы инверсной линии: 00 – на линии всегда «0»; 01 – на линии всегда «1»; 10 – на линии сигнал *REF*; 11 – на линии сигнал *DTG*.

Продолжение таблицы 7.3

9...8	NSELOE[1:0]	Разрешение генерации выходного сигнала на инверсной линии: 00 – запрещена; 01 – разрешена; 10 – определяется сигналом *REF*; 11 – определяется сигналом *DTG*.
7...5	–	–
4	INV	Инверсия прямой линии: 0 – линия не инвертируется; 1 – линия инвертируется.
3...2	SELO[1:0]	Режим работы прямой линии: 00 – на линии всегда «0»; 01 – на линии всегда «1»; 10 – на линии сигнал *REF*; 11 – на линии сигнал *DTG*.
1...0	SELOE[1:0]	Разрешение генерации выходного сигнала на прямой линии: 00 – запрещена; 01 – разрешена; 10 – определяется сигналом *REF*; 11 – определяется сигналом *DTG*.

Данный регистр позволяет настроить обе линии каждого канала: **прямую** и **инверсную**. Для управления большинством устройств вполне **достаточно одной линии**; две линии с противофазными сигналами требуются лишь для управления двигателями.

Для каждой линии может быть настроено три параметра: **разрешение генерации выходного сигнала**, **выбор сигнала** и **инверсия сигнала**.

Разрешение генерации выходного сигнала, как видно из таблицы 7.3, может быть статическим (разрешена или запрещена), а может быть динамическим – зависеть от сигналов *REF* или *DTG*. В большинстве случаев применимо статическое разрешение.

В качестве режима работы линии для генерации прямоугольных импульсов следует выбрать настроенный прежде сигнал *REF* формата 6.

В инверсии сигнала обычно нет потребности; целесообразность ее использования определяется конкретикой управляемой системы.

Из этих соображений конфигурация линий канала может выглядеть так:

```
// Конфигурация линий канала таймера
// Прямая линия
MDR_TIMER1->CH1_CNTRL1 =
  (1 << TIMER_CH_CNTRL1_SELOE_Pos)   // Выходной сигнал на прямой линии (разрешен)
| (2 << TIMER_CH_CNTRL1_SELO_Pos)    // Режим работы прямой линии (REF)
| (0 << TIMER_CH_CNTRL1_INV_Pos)     // Инверсия прямого линии (отсутствует)

// ...Инверсная линия
| (0 << TIMER_CH_CNTRL1_NSELOE_Pos)  // Выходной сигнал на инверсной линии (запрещен)
| (0 << TIMER_CH_CNTRL1_NSELO_Pos)   // Режим работы инверсной линии (отключена)
| (0 << TIMER_CH_CNTRL1_NINV_Pos);   // Инверсия инверсной линии (отсутствует)
```

7. Конфигурация регистров сравнения производится через набор регистров MDR_TIMERx->CHy_CNTRL2, где x – номер таймера, y – номер канала (таблица 7.4).

Таблица 7.4 – Описание регистров MDR_TIMERx->CHy_CNTRL2

№ битов	Функциональное имя битов	Описание
31...4	–	–
3	CCRRLD	Режим обновления регистров *CCR* и *CCR1*: 0 – мгновенное обновление; 1 – обновление после перезагрузки.
2	CCR1_EN	Разрешение работы регистра *CCR1*: 0 – запрещена; 1 – разрешена.
1...0	CHSEL1[1:0]	Выбор события для фиксации значения основного счетчика *CNT* в регистр *CCR1*: 00 – передний фронт на конфигурируемом канале; 01 – задний фронт на конфигурируемом канале; 10 – передний фронт на других каналах (вариант 1); 11 – передний фронт на других каналах (вариант 2).

Регистр сравнения *CCR1*, как уже говорилось, излишен при формировании импульсов. Обновление регистров сравнения лучше осуществлять при очередной перезагрузке таймера, чтобы не было ситуации, при которой в последовательность попадают импульсы случайной ширины:

```
// Конфигурация регистров сравнения
MDR_TIMER1->CH1_CNTRL2 =
  (0 << TIMER_CH_CNTRL2_CHSEL1_Pos)   // Событие для фиксации значения регистра CCR1 (-)
| (0 << TIMER_CH_CNTRL2_CCR1_EN_Pos)  // Использование регистра CCR1 (не используется)
| (1 << TIMER_CH_CNTRL2_CCRRLD_Pos);  // Обновление регистров сравнения (при перезагрузке)
```

Затем следует записать соответствующее требуемому коэффициенту заполнения значение в регистр *CCR*. Это значение должно составлять какую-то часть периода перезагрузки таймера, поэтому для его вычисления целесообразно использовать макрос, описанный в конце раздела 7.3.1:

```
// Коэффициент заполнения (в процентах)
#define DUTY_CYCLE 70

// Частота широтно-импульсной модуляции (в Гц)
#define PULSE_FREQUENCY 10000

// Вычисление ширины импульсов (в отсчетах таймера),
// и запись вычисленного значения в регистр сравнения CCR
MDR_TIMER1->CCR1 = PULSE_PERIOD(PULSE_FREQUENCY) * DUTY_CYCLE / 100;
```

Здесь необходимо отметить, что MDR_TIMER1->CCR1 – это регистр *CCR* первого канала; для второго канала такой регистр именуется MDR_TIMER1->CCR2 и т.д.

Также следует иметь в виду один нюанс: если требуется получить коэффициент заполнения равный 100%, то такой расчет приведет к тому, что значения *CCR* и *ARR*

будут равны. При этом, согласно таблице 7.2, в момент равенства этих значений на линии канала будет логический ноль. Из-за этого фактический сигнал будет не постоянным высоким, как ожидается, а иметь однотактные провалы (рисунок 7.7).

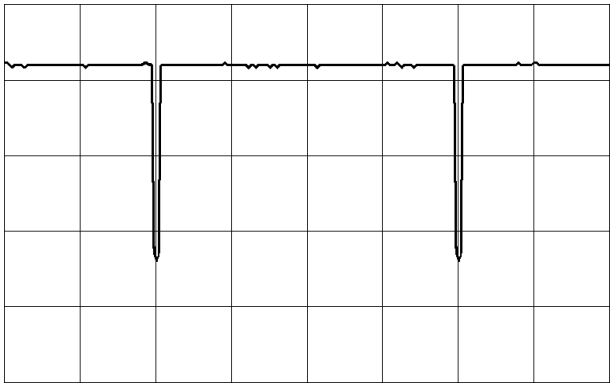

Рисунок 7.7 – Осциллограмма импульсов при *CCR* = *ARR*

Чтобы такого не происходило и заполнение сигнала было полным следует задать значение *CCR* хотя бы на единицу больше, чем *ARR*:

```
// Исключение однотактных провалов
if (DUTY_CYCLE == 100)
    MDR_TIMER1->CCR1 = PULSE_PERIOD + 1;
```

В завершение инициализации остается лишь запустить работу таймера:

```
// Запуск таймера
MDR_TIMER1->CNTRL |= (1 << TIMER_CNTRL_CNT_EN_Pos);
```

7.3.3. Мертвая зона

Как уже было отмечено, инверсные линии каналов обычно используются для управления двигателями. При этом прямая и инверсная линии должны работать в противофазе, т.е. в то время, когда на прямой линии высокий уровень напряжения на инверсной линии должен быть низкий, и наоборот (рисунок 7.8а).

Однако из-за различных задержек и помех фронты одной из линий могут сместиться, что приведет к появлению временных интервалов, в течение которых на обоих линиях будет одинаковый уровень сигнала, в частности – высокий. В этом случае через двигатель будет протекать так называемый *сквозной ток* или *ток утечки*.

Чтобы этого не происходило создается специальная задержка между фронтами импульсов на прямой и инверсной линиях (рисунок 7.8б). Такая задержка называется **мертвой зоной** или **мертвым временем**, а узел, ее создающий, традиционно называется **генератором мертвого времени** (англ. Dead-Time Generator, **DTG**).

Длительность мертвой зоны зависит от частоты импульсов и особенностей системы; она должна выбираться так, чтобы гарантировать отсутствие перекрытия высоких уровней на линиях.

Чтобы использовать DTG в микроконтроллерах серии 1986BE9x следует, во-первых, настроить регистр `MDR_TIMERx->CHy_DTG` выбранного канала, где `x` – номер таймера, `y` – номер канала (таблица 7.5).

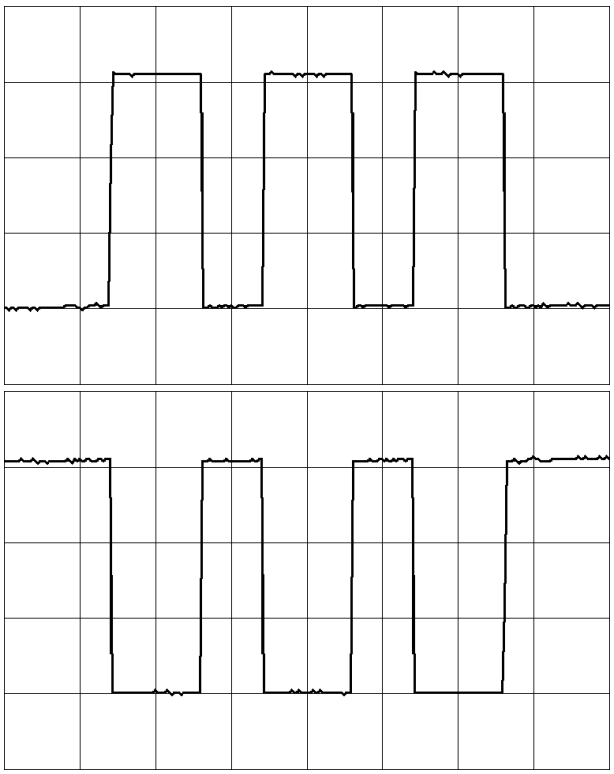

Рисунок 7.8а – Осциллограммы импульсов на прямой и инверсной линиях без мертвой зоны

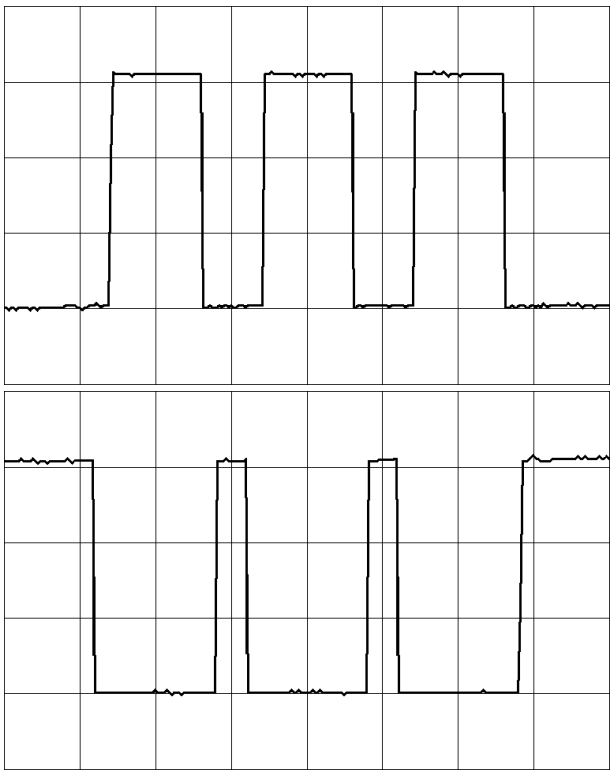

Рисунок 7.8б – Осциллограмма импульсов на прямой и инверсной линиях с мертвой зоной

Таблица 7.5 – Описание регистров `MDR_TIMERx->CHy_DTG`

№ битов	**Функциональное имя битов**	**Описание**
31…16	–	–
15…8	DTG[7:0]	Первый делитель частоты (0…255).
7…5	–	–
4	EDTS	Выбор такового сигнала: 0 – *TIM_CLK*; 1 – *FDTS*.
3…0	DTGx[3:0]	Второй делитель частоты (0…16).

Длительность мертвой зоны в тактах определяется по формуле:

$$DTG_{delay} = DTG \times (DTGx + 1). \quad (7.11)$$

В качестве тактового сигнала при этом может быть использован либо общий сигнал для таймера *TIM_CLK*, либо специально сформированный сигнал *FDTS*.

Для получения мертвой зоны определенной величины можно вывести такую расчетную формулу:

$$DTG \times (DTGx + 1) = \frac{TIM\_CLK \times P_{DTG}}{2 \times 100 \times F_{PWM}}, \quad (7.12)$$

где F_{PWM} – частота широтно-импульсной модуляции,

P – величина мертвой зоны (в процентах от периода модуляции),

двойка в знаменателе учитывает, что расчет ведется для одной части мертвой зоны.

Таким образом, для получения 5-процентной мертвой зоны при частоте широтно-импульсной модуляции 10 КГц и тактовой частоте 80 МГц конфигурация регистра может выглядеть так:

```
// Конфигурация мертвой зоны
MDR_TIMER1->CH1_DTG =
  (0 << TIMER_CH_DTGX_Pos)      // Второй делитель частоты
| (0 << TIMER_CH_DTG_EDTS_Pos)  // Выбор тактового сигнала (TIM_CLK)
| (200 << TIMER_CH_DTG_Pos);    // Первый делитель частоты
```

И также в конфигурации линий канала нужно указать использование сигнала *DTG*:

```
// Конфигурация линий канала таймера
// Прямая линия
MDR_TIMER1->CH1_CNTRL1 =
  (1 << TIMER_CH_CNTRL1_SELOE_Pos)   // Выходной сигнал на прямой линии (разрешен)
| (3 << TIMER_CH_CNTRL1_SELO_Pos)    // Режим работы прямой линии (DTG)
| (0 << TIMER_CH_CNTRL1_INV_Pos)     // Инверсия прямого линии (отсутствует)

// ...Инверсная линия
| (1 << TIMER_CH_CNTRL1_NSELOE_Pos)  // Выходной сигнал на инверсной линии (разрешен)
| (3 << TIMER_CH_CNTRL1_NSELO_Pos)   // Режим работы инверсной линии (DTG)
| (0 << TIMER_CH_CNTRL1_NINV_Pos);   // Инверсия инверсной линии (отсутствует)
```

Сигнал *DTG* в данном случае – это тот же сигнал *REF*, но имеющий мертвую зону.

Следует иметь в виду, что сигнал *REF* после прохождения блока DTG по некоторым причинам инвертируется; при этом программно инвертировать линии обратно не следует, т.к. в этом случае мертвая зона будет представлять собой области логической единицы, а не нуля, т.е. весь смысл мертвого времени будет утрачен.

Есть еще две линии, которые уместно упомянуть в данном разделе: *BRK* и *ETR*. Эти линии не имеют непосредственного отношения к мертвой зоне, но могут быть задействованы при работе с двигателями.

Линии *BRK* и *ETR* используются для аварийной остановки двигателей; таймер при этом прекращает генерацию ШИМ, а прямая и инверсная линии переводятся в исходное состояние.

Линия *BRK* (англ. Brake) – это специальная аварийная линия, останавливающая работу двигателя **асинхронно** с тактовой частотой системы.

Линия *ETR* (англ. External Timer Reference) – позволяет останавливать двигатель **синхронно** с тактовой частотой, но помимо аварийной функции используется для работы таймера с внешней частотой.

Для остановки двигателя может быть использована любая из этих линий. Для этого соответствующей ей порт должен быть правильно сконфигурирован, а в регистре MDR_TIMERx->CHy_CNTRL (таблица 7.1) должна быть разрешена обработка сигнала с линии. В дополнение может быть настроен регистр MDR_TIMERx->BRKETR_CNTRL (стр. 301 Спецификации), но это не критично.

Описание программных проектов

Для корректного исполнения программных проектов требуется подключить модуль расширения к отладочной плате согласно таблице 7.6.

Таблица 7.6 – Подключение модуля расширения к отладочной плате

№ п/п	Модуль расширения		Отладочная плата	
	Имя контакта	Имя разъема	Имя контакта	Имя разъема
1	GND	XP3	1	X27
2	3V3		3	
3	VT_GATE		25	
4	SHAFT		10	X26

Проект **Sample 7.1** производит настройку таймера для генерации широтно-импульсной модуляции; импульсы подаются лампу накаливания или вентилятор. Заполнение импульсов изменяется с помощью кнопок *UP* и *DOWN*.

Проект **Sample 7.2** также производит настройку таймера, но в комбинации с аналого-цифровым преобразователем для изменения коэффициента заполнения с использованием потенциометра.

Задачи для самостоятельной работы

[*проект Sample 7.1*]

1. Найдите минимальную частоту широтно-импульсной модуляции, при которой мигание лампы не будет заметно ни прямым, ни периферийным зрением.

2. Найдите максимальную частоту широтно-импульсной модуляции, при которой форма импульсов становится треугольной при коэффициенте заполнения D = 20%, но сохраняется полная амплитуда – 3.3 В.

Сигнал следует измерять на истоке транзистора (средняя ножка), учитывая при этом, что он будет инвертирован. Объясните причину инверсии.

[*проект Sample 7.2*]

3. Модифицируйте проект таким образом, чтобы при повороте ручки потенциометра изменялась частота импульсов по экспоненциальному закону в диапазоне от 20 Гц до 20 КГц.

Контрольные вопросы

1. Что такое широтно-импульсная модуляция?
2. Как выбрать частоту широтно-импульсной модуляции?
3. Как скважность импульсов влияет на мощность, подводимую к нагрузке?
4. Чем различаются понятия скважности и коэффициента заполнения?
5. Какими средствами реализуется широтно-импульсная модуляция с применением микроконтроллеров?
6. Сколько каналов широтно-импульсной модуляции можно реализовать на базе микросхем серии 1986ВЕ9х?
7. Что такое мертвая зона?