

Таймеры. Обзор

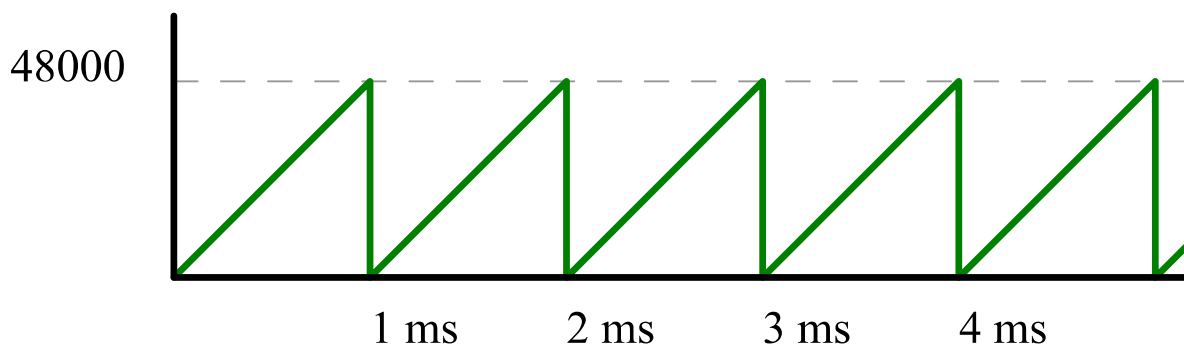
В школе вы наверняка сдавали нормативы на физкультуре, и учитель для замера времени использовал таймер. Но что такое время?

Секунда — время, равное 9 192 631 770 периодам излучения, соответствующего переходу между двумя сверхтонкими уровнями основного состояния атома цезия-133. // [Википедия](#)

Микроконтроллер понятия не имеет, что такое цезий-133 и сколько периодов излучения нужно, чтобы отмерить одну секунду. За то ему понятно, что такое тактовый сигнал. Главная задача таймера -- подсчёт тактов. Условившись, что частота нам известна, можно отмерять время.

$$f = \frac{1}{T}$$

Допустим МК работает на частоте 48 МГц. Это значит что за одну секунду от системы тактирования приходит 48 миллионов тактов. Если нам нужно отмерить одну миллисекунду, то нужно просто досчитать от 0 до $48000000 / 1000 - 1$.



К слову, сигнал от источника тактирования может подаваться не напрямую, а через делитель частоты (т.е. скажем подсчитываться будет только каждый второй пришедший сигнал). Основной характеристикой является его разрядность (битность). Именно от неё зависит максимально число счёта.

Разрядность	Диапазон
12 бит	0 ... 4095
16 бит	0 ... 65535
24 бита	0 ... 16777215

В МК может быть много таймеров, конкретно в `stm32f103c8` их 7 (на самом деле 8).

- **Системный таймер** (`SysTick`). 24-битный таймер, является частью ядра Cortex.
- **Таймеры общего назначения** (`TIM2` , `TIM3` и `TIM4`). 16-битные таймеры, являются периферийными блоками. Могут работать в разных режимах, имеют несколько каналов ¹.
- **Продвинутый таймер** (`TIM1`). Так же как и таймеры общего назначения, является 16-битным, но имеет расширенные возможности.

- **Два сторожевых таймера** (англ. watchdog): независимый (англ. independent, **IWDG**) и оконный (англ. window, **WWDG**). Их главная задача в обеспечении непрерывной работы устройства.
- **Часы реального времени** (англ. real-time clock, **RTC**). Специальный таймер, который может работать от резервного источника питания и подключён к внешнему специальному источнику тактирования (кварцевому резонатору, LSE).

Почти все эти таймеры будут задействованы в нашем устройстве. Системный, два сторожевых таймера и часы реального времени мы рассмотрим позже.

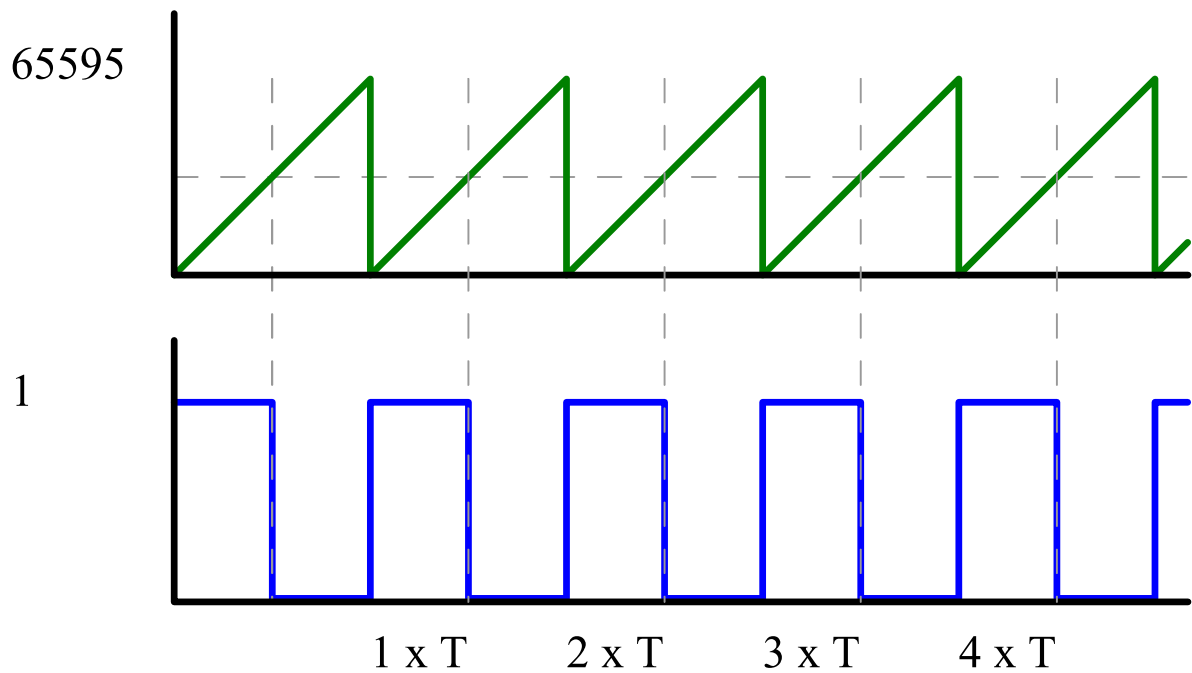
Базовые и продвинутый таймеры умеют работать с инкрементными энкодерами, датчиками Холла, могут генерировать ШИМ ² и многое другое. Кроме того, они могут генерировать прерывания или совершать запрос к другим модулям (например, к DMA — Direct Memory Access) по разным событиям: переполнение (англ. overflow); захват сигнала (англ. input capture); сравнение (англ. output compare); событие-триггер (англ. event trigger). Подробнее рассмотрим эти режимы позже, а пока перейдём к системному таймеру.

Прерывание по переполнению — самый простой способ использования, он уже был показан чуть выше, когда мы говорили про отмер одной миллисекунды. Другие режимы более интересны.

Режим генерации ШИМ

Мы можем настроить таймер на прерывание по переполнению и в обработчике прерывания менять уровень сигнала на порте, таким образом создавая сигнал нужной нам частоты. Если частота не большая — ничего страшного, прерывания будут происходить относительно редко. Повысив частоту, наш МК не будет заниматься ничем другим, кроме как генерацией этого самого сигнала. К тому же нам может понадобиться сигнал, где время высокого логического уровня на ножке больше, чем низкого, а этого заставит нас переконфигурировать таймер каждый раз, когда мы попадаем в прерывание. Не удобно и не эффективно.

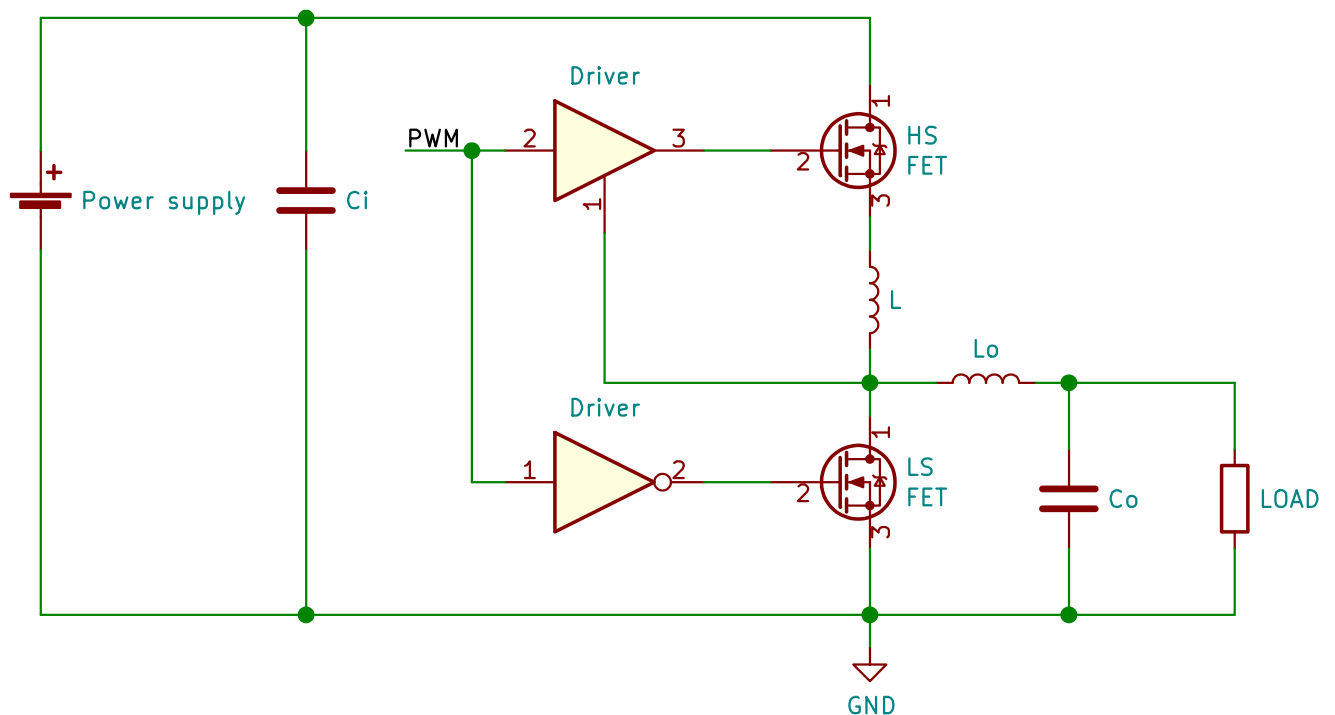
Некоторые таймеры способны генерировать такой сигнал аппаратно. Всё что вам необходимо, это настроить его на нужный режим, задать частоту и заполнение.



При помощи продвинутого таймера можно формировать трехфазный ШИМ, который очень пригодится для управления трехфазным двигателем.

Мёртвая зона

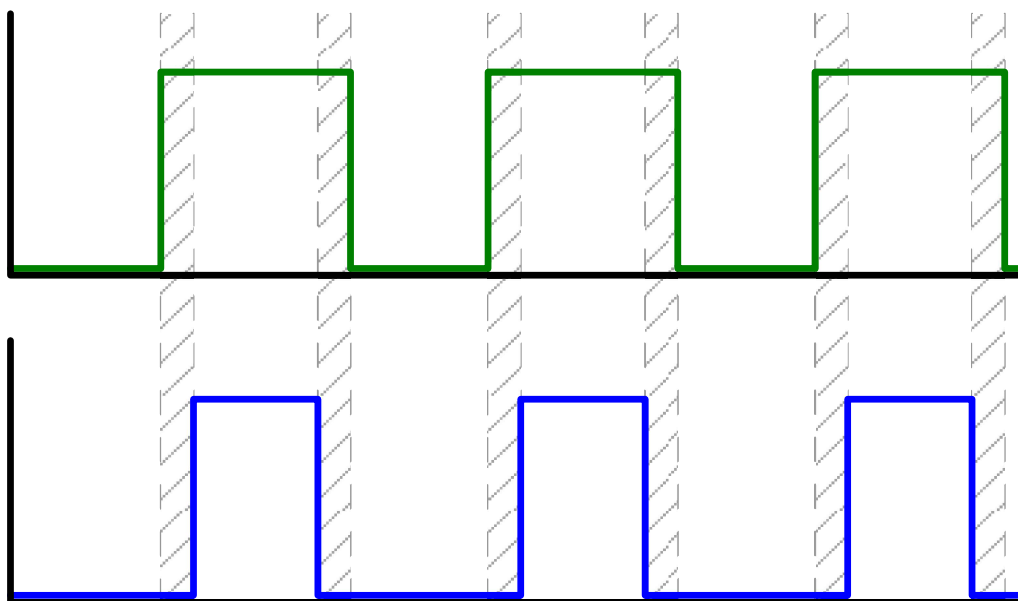
Таймер может генерировать комплементарные ШИМ, т.е. когда один находится в высоком состоянии, второй находится в низком. Такое нужно, например, для реализации синхронного *bus*-конвертера.



В классической схеме никакой проблемы нет, нижнее плечо (транзистор Q_2) состоит из диода, который не пропускает ток на землю. В синхронной версии (более высокая эффективность), нижнее плечо заменено на транзистор, который должен быть закрыт, когда верхний открыт и наоборот. Если они в какой-то момент времени будут открыты оба — произойдёт КЗ.

Таймер может одновременно переключить оба порта ввода-вывода, но вот сами транзисторы не открываются и не закрываются мгновенно. В зависимости от ёмкости затвора это время будет либо маленьким, либо большим. Конечно они не будут полностью открыты одновременно, но этого времени хватит, что бы существенно их нагреть, а значит потерять в эффективности.

Специально для таких случаев в таймере предусмотрена мёртвая зона (англ. *dead-time*), которое автоматически будет вставлять задержки между двумя каналами.



Нам не понадобится такой режим в нашем устройстве.

Захват сигнала

Этот режим хорошо подходит, чтобы измерять период следования импульсов (или их количество, скажем, за секунду). Когда на выход МК приходит импульс, таймер генерирует прерывание, с которого мы можем снять текущее значение счетчика (из регистра `TIM_CCRx`, где `x` — номер канала) и сохранить во внешнюю переменную. Затем придет следующий импульс, и мы простой операцией вычитания получим «время» между двумя импульсами. Отлавливать можно как передний, так и задний фронт импульса, или даже сразу оба. Зачем это нужно? Предположим, у вас есть магнит, который вы приклеили к диску на колесе, а датчик Холла — к вилке велосипеда. Тогда, вращая колесо, вы будете получать импульс каждый раз, когда магнит на колесе будет в той же плоскости, что и датчик Холла. Зная расстояние которое пролетает магнит за оборот и время можно вычислить скорость движения.

Если настроить один из каналов таймера на захват переднего фронта, а другой на захват заднего, то можно будет измерить параметры ШИМ. Первый канал измерит период, а второй заполнение.

Режим сравнения

В таком режиме выбранный канал таймера подключается к соответствующему выводу, и как только таймер достигнет определенного значения, состояние вывода изменится в зависимости от настройки режима (это может быть «1» или «0», или состояние на выходе просто инвертируется).

[Назад](#) | [Оглавление](#) | [Дальше](#)

1. Каналы позволяют использовать один и тот же таймер для разных задач. [↪](#)

2. Широтно-импульсная модуляция, англ. Pulse Width Modulation. [↪](#)