

Во первых. Господа, **Capture Unit для таких частот не пригодится.** Capture Unit решает обратную задачу - измерение временных интервалов и чем меньше частота тем лучше. На частоте же 170Khz это будет всего ~100 тактов на период, точность будет мягко сказать хреновая при значительных затратах процессорного времени

Ну тут я полностью с вами не согласен. Именно Capture Unit поможет добиться непревзойдённой точности. Кто нам мешает измерить 170000 периодов в секунду с помощью Capture Unit? Вгрубе так, запустили прерывание Capture Unit, оно сработало, в прерывании записали захваченное время t1 и выключили прерывания Capture Unit. Выждали примерно (!) секунду, включили прерывания Capture Unit опять, сработало, в прерывании записали время t2 и выключили прерывания Capture Unit. Зная количество периодов входной частоты, (t2-t1) и тактовую частоту, вычисляем частоту входного сигнала, точность измерения примерно ±0,02 Гц. Сравните с вашими ±1 Гц. Другим способом на данном железе вы такой точности не добьётесь.

Я ведь потому и спросил о периоде готовности результата. Чем больше период времени выделенный для подсчета частоты, тем точнее можно получить результат за счет матем.обработки. Да, при частоте тактирования таймера 22МГц использование Capture позволяет получить **дискретность** определения периода частоты порядка 45нс. Для частоты 170кГц эта точность хуже 1%. Но если **усреднить** результаты измерения за временной интервал значительно превышающий период частоты сигнала, то мы получим гораздо бОльшее временное разрешение, чем искомые 45нс. За 1 сек можно посчитать частоту 170кГц с точностью почти **на порядок** выше, чем дает единичный отсчет с дискретностью определения периода в 45мс. Впрочем на форуме это уже неоднократно обсуждалось, имеет ли смысл повторяться?

Я ведь потому и спросил о периоде готовности результата. Чем больше период времени выделенный для подсчета частоты, тем точнее можно получить результат за счет матем.обработки. Да, при частоте тактирования таймера 22МГц использование Capture позволяет получить **дискретность** определения периода частоты порядка 45нс. Для частоты 170кГц эта точность хуже 1%. Но если **усреднить** результаты измерения за временной интервал значительно превышающий период частоты сигнала, то мы получим гораздо бОльшее временное разрешение, чем искомые 45нс. За 1 сек можно посчитать частоту 170кГц с точностью почти **на порядок** выше, чем дает единичный отсчет с дискретностью определения периода в 45мс. Впрочем на форуме это уже неоднократно обсуждалось, имеет ли смысл повторяться?

GM за 6 прерываний в секунду решил измерить с точностью + -0.02Гц частоты трех источников. Неужели вы не понимаете всей абсурдности этого?

В AVR при возникновении события захвата в регистры ICRx переписывается текущее значение счетчика TCNT и возбуждается прерывание. GM заявляет, что достаточно всего 6-ти прерываний (событий) в секунду.

Если прерываний всего 6 (тобиш по 2 захвата на каждый входной сигнал), то у нас будет всего лишь однократно отмерянный период сигнала, что и будет соответствовать точности приблизительно + -1% (+ -1.7Khz), как уже отмечали выше. На мой взгляд, ни вы, ни GM до сих пор не понимаете, что для получения точности хотябы близкой к + -1Гц, потребуется несколько тысяч таких выборок (для точности + -0.02Гц - несколько миллионов), соответственно и прерываний будет несколько тысяч/миллионов для каждого входного сигнала, и рассчитывать "дельту" нужно будет успевать на лету, за время пока не произошел следующий захват. А времени то нет, у нас всего ~100 тактов между захватами для одного сигнала. 100/ 3 - по 33 такта на обработку.

ИЗМЕРЕНИЕ ЧАСТОТЫ С ПОМОЩЬЮ AVR МИКРОКОНТРОЛЛЕРА

1) Введём определения

Fo - сигнал опорной частоты,

To - период опорной частоты, $To=1/Fo$,

Fx - входной сигнал (меандр) неизвестной частоты, подлежащей измерению,

Tx - период входной частоты, $Tx=1/Fx$,

Тизм - период измерения входной частоты,

N - количество импульсов опорной частоты за время измерения,

M - количество импульсов входной частоты за время измерения.

2) Формула вычисления частоты $Fx = Fo*M/N$ (поскольку очевидно, что $To*N=Tx*M$ за Тизм).

3) Формула для Fx применима как к варианту с "воротами", так и к варианту со схемой захвата. При использовании схемы захвата M является точным числом, а N "гуляет" в пределах (-1,+1). При использовании "ворот" N является точным числом, а M "гуляет" в пределах (-1,+1).

Вот всё, что нам нужно знать из теории для измерения частоты.

4) Применим наши знания на практике. Выберем период измерения Тизм примерно равным одной секунде. (Примерно потому, что Тизм должен быть ТОЧНО кратен $M \cdot T_x$). Пусть $F_x = 170$ кГц, а $F_o = 16$ МГц. Тогда $M = 170000$, а $N = 16000000$.

Вычислим частоту по нашей формуле $F_{x\text{выч}} = F_o \cdot M / N = 16000000 \cdot 170000 / 16000000 = 170000$ Гц.

Вычислим погрешность метода. Не буду вас мучить выводом соответствующей формулы с помощью частных дифференциалов, кто захочет пусть сам потренируется, скажу только, что относительная погрешность будет суммой модулей относительных погрешностей отдельных составляющих.

Ну и ещё упростим задачу, приняв $\Delta F_o / F_o = 0$ (На самом деле, кратковременная нестабильность кварцевой опоры порядка 10^{-8}) или лучше).

Итак, $F_x = F_o \cdot M / (N \pm 1) = 170000 \pm 0.010625$ Гц. Что нам и требовалось показать. Относительная погрешность составит $6 \cdot 10^{-8}$. Можно добавить 10^{-8} нестабильности опоры, если требуется уточнить погрешность. Кстати, для варианта с "воротами" $F_x = F_o \cdot (M \pm 1) / N = 170000 \pm 1$ Гц, как и было сказано.

5) Перейдём теперь к реализации алгоритма на микроконтроллере. В принципе, всё и так уже ясно, просто поставим точки над i и чёточки на t (:-). Сколько нам нужно захватов таймера, чтобы вычислить частоту? Два - один для захвата начала секундного интервала, и второй для захвата конца того же интервала. Пусть захват работает по прерыванию. При захвате, т.е. при положительном фронте импульса входной частоты, содержимое таймера переписывается в регистр ICR. Прекрасно, в этом же прерывании запомним в регистрах или в памяти два числа – количество импульсов $M_{\text{нач}}$ и $N_{\text{нач}}$, запрещаем прерывание и выходим из него (замечу в скобках, схема захвата "молотит" по-прежнему, но прерывания запрещены), оно нам понадобится только через секунду, займёмся другими делами, скажем, расчетом частоты предыдущего цикла. Через примерно секунду работы (или вынужденного безделья, можно по таймеру) ОПЯТЬ разрешаем прерывания от схемы захвата. Как только оно произойдёт, опять запоминаем два числа – количество импульсов $M_{\text{кон}}$ и $N_{\text{кон}}$, запрещаем прерывание по захвату и выходим из него, вычисляем $M = M_{\text{кон}} - M_{\text{нач}}$, $N = N_{\text{кон}} - N_{\text{нач}}$.

Вот так это всё и работает. Я, конечно многого не упомянул. Все переменные для расчета должны быть соответствующей разрядности, как минимум 32, умножение надо делать до деления, чтобы не потерять точность, вычисление частоты можно вести непрерывно, достаточно просто после вычисления частоты переписать переменные $M_{\text{нач}} = M_{\text{кон}}$, $N_{\text{нач}} = N_{\text{кон}}$. Обязателен учёт переполнения таймеров.

6) По времени выполнения, самые длительные операции – умножение и деление, скажем, по 400 тактов, остальное умещается в 500 или меньше. 1500 тактов из 16 миллионов - процессор практически свободен 99.9% времени.

7) Измерение по похожему алгоритму реализовано мной на МК 8515 и прекрасно работает более 5-ти лет в нескольких тысячах экземпляров (в Германии).

8) Последнее. Надеюсь, что defunct возьмёт назад свои слова о "трепе". Надеюсь также, что сия заметка кому-то реально поможет в понимании, в реализации или ещё в чём-нибудь.

Непонятно откуда взялось и как реализовать это условие? Изначально мы не знаем ни M , ни T_x , нам их нужно измерить. А если мы их знаем, то зачем измерять? 🤔

Ну что ж тут непонятного? Конечно вы ничего не знаете ни о M , ни о T_x . В произвольный (и удобный для вас) момент времени разрешаете прерывание от схемы захвата и ждёте наступления прерывания. Вот это прерывание наступило, записываете свои M и N , запрещаете прерывание. Тупо ждёте в программе (с помощью программной задержки) ОДНУ секунду или 1.2 с или сколько вам надо, затем ОПЯТЬ разрешаете прерывание от схемы захвата и ждёте наступления прерывания. Ну и так далее, как описано мною выше. По M , N и F_o определяете вашу F_x , ну и T_x , если надо. А интервал измерения связан с заданной точностью измерения и наоборот. зависит от решаемой задачи.

Вся прелесть метода в том, что вам НЕ ОБЯЗАТЕЛЬНО ждать точно одну секунду, ждите сколько можно или сколько вам хочется. В результате разного времени ожидания в итоге вы получите совершенно другие M и N , но вычисление по формуле даст тот же самый результат.

Возьмите, да посчитайте для 1с и для 1.3с, практически ничего не изменится.

ЭТОТ АБЗАЦ ПОПРАВЛЕН

Вот, кстати говоря, ещё один метод измерения придумался, и не одной, не трёх, а десятков частот. После получения стартовых $M_{нач}$ и $N_{нач}$ и завершающих пар $M_{кон}$ и $N_{кон}$ для одной частоты подключаете к МК мультиплексором другую частоту и получаете новую пару стартовых M и N . И так для десятка-другого частот, насколько мультиплексора хватит и в зависимости от требуемой точности. В течение одной секунды можно измерить СТО частот, поскольку для достижения точности ± 1 Гц предлагаемым методом требуемое время измерения составляет 10 мс (для максимальной частоты 170 кГц).

Забыл добавить, что можно делать оценку значения входной частоты с помощью метода "временных ворот" что-нибудь порядка 10-20-100 мс, чтобы обеспечить подходящие условия для точного измерения, скажем инфранизких частот, или для определения наличия входной частоты и т.д. Ну это уже примочки для коммерческих продуктов, мы же с вами обсуждаем идею получения максимальной точности на AVR при прочих равных условиях. Согласитесь, что точность в ± 0.01 Гц звучит гораздо лучше, чем точность ± 1 Гц, как тут предлагают некоторые уважаемые, не в обиду им будь сказано.

Насколько я понял, второй случай это тот же метод "временных ворот", но на счетный вход подается не измеряемая частота, а опорная? А временные ворота формируются измеряемой частотой, так? Как это реализуется на реальной периферии с использованием таймера?

Это как бы обобщённый метод "временных ворот", если хотите, M периодов входных и N периодов опорных. Частный случай при $M=1$ - измерение одного периода входной частоты.

Входная частота подается на ножку ICPn, которая является частью схемы захвата (input capture unit). Схема захвата работает следующим образом. По фронту импульса входной частоты содержимое таймера переписывается в специальный регистр (ICRn).

Т.е. получается, что как обычно таймер тактируется опорной частотой, которая является асинхронной по отношению к входной измеряемой. Поэтому захват фронта измеряемой частоты произойдет с точностью до 1 такта опорной.

Можно и по спаду, это настраивается. То есть, если у вас в таймере тикает время, то вы получите точное значение времени прошедшего фронта, скажем, это было начало измерения (значит, число $N_{нач}$). Запомнили его. И т.д. Количество импульсов входной частоты можно считать программно, но можно и другим таймером (счетчик M), я так и делаю. n - номер схемы захвата.

ОК, посчитали мы импульсы и зафиксировали схемой захвата фронт последнего импульса с точностью опять же ± 1 период от опорной частоты.

Немного подправлю. Любой фронт вы фиксируете с точностью в диапазоне (0,1) такт, значит оба фронта фиксируются с точностью ± 1 такт. Это важно осознать, а то вы там ниже уже пишете ± 2 , а это неправильно.

Кстати, а переполнения таймера вы учитываете? При тактовой 16 МГц за 1 сек измерения у 16-битного таймера произойдет 244 переполнения.

А як-жеж(:-)! Специально написал, что все переменные 32-битные.

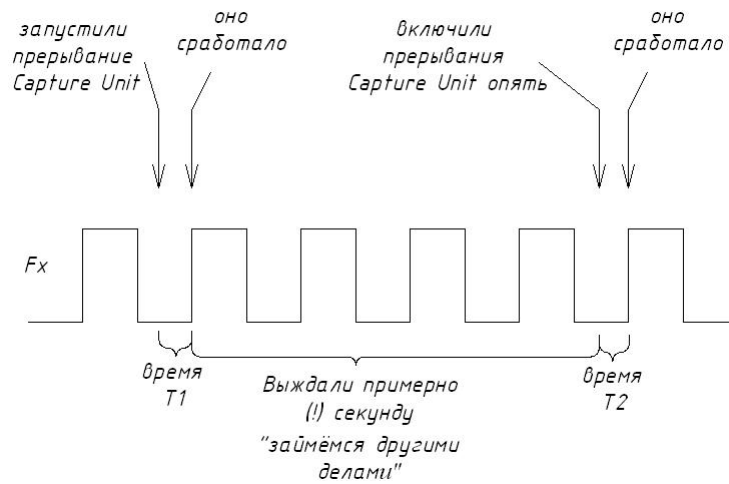
Ладно. Дальше. Пускай измеряемая частота была 169999 Гц. Тогда за время счета 170000 импульсов мы получаем схемой захвата число 16000094.

Вычисляем по вашей формуле $F_x = 16000000 * 170000 / 16000094 (+2) = 169999,001 \text{ Гц} (+0,02 \text{ Гц})$.

Хм. 😊 И вправду получается неплохое разрешение!

Как-то не так вы вычисляете. Для частоты 169999 Гц вы и получите за 1 секунду $M=169999$ импульсов, а импульсов опорной за секунду будет $N=16000000$.

Измеренная частота $F_x = 169999 * 16000000 / (16000000 \pm 1) = 169999 \pm 0.01$. Ну и вот здесь ошибочка, надо писать ± 1 , а не ± 2 , почему, я объяснил выше.



Картинки хорошие, но неполные. Внизу под меандром надо нарисовать много-много импульсов опорной частоты, это будет наше время, заполнение, как угодно назовите.

Вот вы написали "Оно сработало" в конце периода T1 (ваши обозначения на рис.1). Что это означает? Это означает, что содержимое таймера1 от "мелкого" опорного меандра (который вы не нарисовали) переписалось в регистр ICR1, его надо запомнить, а прерывание запретить (в принципе, можно не запрещать, просто не обращать внимания, если время и вычислительная обстановка позволяет, важно, что нужно только первое, которое даёт время начала вашего окна).

Отмечу, что таймер1 продолжает работать, отсчитывает длительность окна. В начале времени T2 на вашем рисунке надо разрешить прерывание опять, всё так, дождаться когда оно сработает (конец периода T2 на вашем рисунке), и запомнить время, сохранённое в ICR1.

Затем нужно вычислить разность этих времен. Что нам даст эта разность? Эта разность N нам даст время (в тиках таймера1) между первым фронтом входной частоты и последним фронтом. Причём это время будет не приблизительным, а ТОЧНЫМ, с точностью ± 1 такт. В то же самое время, таймер0 подсчитает количество импульсов входной частоты M. Теперь вы имеете на руках M, N и F_0 и по формуле можете посчитать F_x .

Скажу ещё раз, для ясности. Сигнал измеряемой частоты надо подать на вход ICP1 и вход timer0/counter0, который работает как счётчик, в то же время timer1 работает как таймер, считает импульсы опорной частоты.

Давайте ляжем по-другому(:-). Запустили таймер1 и прерывание по захвату. Пусть так получилось, что прерывание сработало как раз в то время, когда в таймере оказался 0 (вообще без разницы, что там будет, просто для удобства). Запомнили этот 0 в памяти или регистре, не важно. Сидим ждём, курим, "кирпич ёк"(:-). То есть ничего не делаем, таймер1 подсчитывает время окна, таймер 0 подсчитывает количество входных импульсов. "Ничего не делаем" означает скручиваем 32-битный счетчик, предположим он рассчитан на 800 мс. Затем разрешили прерывание по захвату, оно сработало, у нас в руках число - время последнего фронта входного импульса. Какое оно будет? Зависит от того, сколько импульсов входного сигнала насчиталось. Давайте прикинем.

Если задержка была на 800 мс, то число $M=170000*0.8=136000$, а число $N=16000000*0.8=12800000$. Хорошо, значит $F_x=16000000*136000/12800000=170000\pm0.013$ Гц

Если задержка была на 801 мс, то число $M=170000*0.801=136170$, а число $N=16000000*0.801=12816000$. $F_x=16000000*136170/12816000=170000\pm0.013$ Гц

Если задержка была на 802 мс, то число $M=170000*0.802=136340$, а число $N=16000000*0.802=12832000$. $F_x=16000000*136340/12832000=170000\pm0.013$ Гц

Ну и так далее. Как вы видите размер окна никак не влияет на результат. Но относительная ошибка, естественно, зависит от времени наблюдения. Для упрощения понимания я брал задержку примерно на 1 с.

Подчеркну, окно всё время будет кратным периоду входной частоты, поскольку схема захвата срабатывает от передних фронтов именно входной частоты. Ну а поскольку частота опоры асинхронна по отношению к входной частоте, она будет "гулять" на ± 1 такт.

(rezident @ Apr 10 2007, 09:55) [snapback]234733[/snapback]

Я не знаю синхронный или асинхронный захват в AVR, поэтому насчет +-2 утверждать не буду. В общем (худшем) случае будет +-2 такта.

Захват синхронный, поскольку все входные сигналы тактируются внутренним клоком, т.е. все входные фронты приводятся к единой временной сетке, что есть, не скажу всегда необходимо, но грамотно с точки зрения построения архитектуры МК. Как бы то ни было, фронт нашего сигнала лежит между двумя фронтами тактовой частоты, следовательно максимальная ошибка +1 такт, или, если хотите, ± 0.5 такта. Как кому нравится. Или вот так ещё можно сказать: фронт нашего сигнала лежит в диапазоне от N до N+1 такта, соответственно ошибка лежит в диапазоне от 0 до 1 такта.

(rezident @ Apr 10 2007, 09:55) [snapback]234733[/snapback]

P.S. только для удобства лучше все же иметь в периферии 32-х разрядный таймер, а не 8-ми разрядный 😊

Так ведь нет его! С другой стороны, представьте себе некий 8-10-12-ногий чип, который меряет частоту с относительной точностью 10^{-10} , что могут дать только какие-нибудь монстры с цезиевым или там водородным стандартом, да и то не все. Ну а опору надо брать от ДжиПиЭС(:-).

Ну, если говорить об измерении частоты на голой атмеге128, то как для "метода захвата", так и для "метода ворот" имеется одно ФУНДАМЕНТАЛЬНОЕ ограничение - частота входного сигнала не может быть больше половины тактовой частоты процессора.