

Применение встроенных в микроконтроллер EFM8LB12 конфигурируемых логических элементов для прецизионного измерения временных интервалов

Алексей Кузьминов (compmicrosys@mail.ru)

В статье описан пример применения встроенных в МК EFM8LB12 конфигурируемых логических элементов – Configurable Logic Unit (CLU) совместно с таймерами для измерения временных интервалов с погрешностью, не превышающей 0,2 мд. Сконфигурированная на CLU антидребезговая схема позволила подключить кнопку, задающую временной интервал, непосредственно к МК, а сконфигурированный Т-триггер – аппаратно соединить три таймера (Т3, Т4 и РСА-таймер) для получения 48-разрядного таймера, способного измерять время между двумя событиями в широком временном диапазоне и с высокой точностью. Приводится схема тестовой платы и программные средства для её работы.

Введение

При проведении различного рода измерений часто требуется определение времени, прошедшего между двумя событиями, с относительно высокой точностью. Вот простой пример: требуется измерить объём V газа или воды, прошедший через какой-либо измерительный прибор при известном (или измеренном) расходе Q за определённый интервал времени T . Объём V , как известно, вычисляется как $V = Q \times T$. Естественно, от точности измерения интервала времени T зависит точность определения объёма V . Для формирования интервала времени T могут использоваться различного типа коммутационные устройства (контактные или бесконтактные). При ручном управлении часто применяют кнопки, при некоторой автоматизации процесса в качестве коммутационных

устройств могут использоваться герконы. Например, в приборах, измеряющих объём газа (газовых счётчиках), при прохождении определённого объёма газа (к примеру, 1 м^3) замыкается геркон. В любом случае контактные коммутационные устройства обладают одним хорошо известным и неприятным свойством: дребезгом контактов. Если в измерительной системе используется относительно высокоскоростное устройство, например, микроконтроллер (МК), который должен сформировать определённое действие, скажем, по фронту сигнала, подключённого к какому-либо его порту, то прямое подключение кнопки к порту может привести к ложным срабатываниям, поскольку дребезжащие контакты сформируют не один фронт, а целый пакет импульсов, длящийся от долей мс до нескольких мс, в зависимости от свойств кнопки. Для исключения дребезга контактов применяют как программные, так и аппаратные средства.

Программные средства используют различного рода задержки с длительностью, заведомо превышающей длительность процесса дребезга контактов. Для формирования временных задержек часто используют встроенные в МК таймеры. Однако программные средства обладают существенным недостатком, так как требуют дополнительных ресурсов МК: необходимую работу процессора для выполнения подобной про-

граммы, увеличение памяти для её хранения, таймеров и т.п.

Что касается аппаратных средств, то для устранения дребезга контактов используют различного рода антидребезговые схемы. Они бывают двух типов. Схемы первого типа основаны на различного рода RC-цепочках, которые затягивают фронт сигнала на время, заведомо превышающее время дребезга (это в некотором смысле сродни программному способу). Обычно схемы такого типа используют микросхемы ждущих мультивибраторов. Схемы второго типа используют триггеры. Вот пример (рис. 1) старой, как мир, антидребезговой схемы на основе RS-триггера, построенного на двух логических элементах 2И-НЕ. При нажатии кнопки и первом же касании контакта триггер перебрасывается в противоположное состояние, и дальнейший дребезг контактов не приведёт к изменению состояния триггера. При этом на выходах триггера Q и \bar{Q} сформируются единственные фронт и срез соответственно. Если кнопку отпустить (или отжать, если она с фиксацией), то при размыкании контакта дребезг будет отсутствовать, а при первом же касании противоположного контакта триггер установится в исходное состояние, и на выходах Q и \bar{Q} сформируются единственные срез и фронт соответственно. Такая схема имеет существенное преимущество по сравнению со схемами первого типа, так как у таких схем для надёжности значительно удлиняют время задержки, что не всегда приемлемо. Недостаток аппаратных средств – необходимость использования самих этих средств (например, для схемы рис. 1 требуется микросхема 561ЛА7, подача на неё питания с положенными по штату конденсаторами и два резистора). Хотя стоимость этих средств и невелика, однако место, занимаемое ими на плате, на порядок больше места, занимаемого корпусом МК ($4 \times 4 \text{ мм}$ – QFN32, и ещё меньше ($3 \times 3 \text{ мм}$) – QFN24

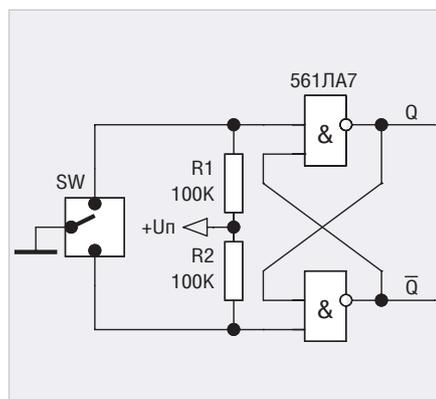


Рис. 1. Антидребезговая схема на RS-триггере

у МК EFM8LB12). А увеличение площади платы на порядок приводит к такому же повышению её стоимости; кроме того, больший размер платы требует соответствующего увеличения размера корпуса под неё, что также увеличивает стоимость устройства.

Избежать подобных аппаратных и вышеупомянутых программных средств позволяет одно интересное новшество, которое в последнее время стали вводить в состав МК. Речь идёт о встроенных в МК конфигурируемых логических элементах – Configurable Logic Unit, сокращённо CLU. Такие CLU работают абсолютно автономно, совершенно не требуют участия процессора МК и, кроме того, могут даже обходиться вообще без тактирования, т.е. работать в асинхронном режиме.

Помимо CLU, в состав МК включают таймеры, обладающие новыми свойствами. Например, в МК EFM8LB12 имеются два таймера (Т3 и Т4), которые можно включить один за другим и получить из двух 16-разрядных таймеров один 32-разрядный; его с успехом можно применить для измерения временных интервалов (см. выше).

Информация об особенностях CLU крайне скудна. Что в руководстве пользователя МК EFM8LB12 [1], что в примере применения CLU [2] не описаны даже все функции CLU, не говоря уже о способах их применения и особенностях. Кроме того, путём экспериментов автором обнаружено несколько багов при совместной работе CLU и таймеров. Поэтому для того, чтобы было понятно дальнейшее изложение, ниже в кратком виде приводятся все функции CLU и особенности их применения при совместной работе с таймерами.

Итак, в МК EFM8LB12 имеются четыре CLU: CLU0, CLU1, CLU2 и CLU3. Каждый CLU имеет 3 входа: А, В и С – и один выход. Входы А и В – стандартные логические входы (как, например, два стандартных входа у 561ЛА7 – рис. 1), а вход С – дополнительный вход переноса (carry) с выхода предыдущего CLU. Например, вход С CLU1 – это выход CLU0, вход С CLU0 – выход CLU3 и т.п., т.е. они как бы расположены по кругу. Функции каждого CLU – основные функции логических элементов: И (AND), ИЛИ (OR), исключающее ИЛИ (XOR) и НЕ (NOT). Записываются функции следующими значками: «&» – И, «|» – ИЛИ, «^» – исключающее ИЛИ, «~» – НЕ. Кроме того, используются значки скобок «(», «)» и символы «А», «В» и «С».

Например, чтобы получить один логический элемент 2И-НЕ, как, например, в 561ЛА7 (рис. 1), необходимо записать: «~(А & В)». При этом соответствующий CLU будет выполнять эту функцию, а перенос С учитываться не будет (поскольку С в выражение не входит). Входы каждого CLU можно подключить: к внешним выводам (портам) МК (их, правда, ограниченное количество), к выходу этого же либо другого CLU, к выходам таймеров (например, Т3, Т4). Выходы каждого CLU можно вывести на внешние выводы МК (порты) напрямую (автономный или асинхронный режим работы), причём для каждого выхода CLU существует один-единственный порт, изменить который нельзя. Например, для выхода CLU2 предусмотрен порт P2.2, а для выхода CLU3 – порт P2.5 (для EFM8LB12-QFN32). Кроме того, выходы каждого CLU можно подключить ко входу данных (D) встроенного D-триггера, который может тактироваться по входу С либо системной тактовой частотой (SYSCLK), либо таймером (ALTCLK), а выход Q этого триггера уже подключить к внешнему порту. Этот режим называется (в описании МК) синхронным.

По поводу таймеров. Входы (тактирование) таймеров (Т2...Т5) можно подключить (в том числе) либо к системной тактовой частоте, делённой на 12 (SYSCLK/12), либо к выходу внешнего генератора и тактировать с частотой этого генератора, делённой на 8 (EXTCLK/8). Таймер Т4 можно тактировать (в том числе) таймером Т3.

Теперь о нескольких багах, обнаруженных автором:

- если таймер Т3 тактируется от внешнего тактового генератора (EXTCLK/8), а Т4 – от Т3, то Т4 не запускается и не работает. Работа Т4 при тактировании от Т3 возможна только при тактировании Т3 системной тактовой частотой, делённой на 12 (SYSCLK/12);
- если ко входу CLU подключён выход таймера Т3 или Т4, а выход CLU – к внешнему порту, то при переполнении этих таймеров на выходе CLU (и соответственно на выходе порта) появляется очень короткий импульс, который непригоден, например, для тактирования PCA таймера по его входу ECI;
- как следует из описания МК EFM8LB12 [1], он может тактироваться (в том числе) от внутреннего генератора № 1 с системной такто-

вой частотой SYSCLK = 72 МГц либо от внешнего генератора (EXTCLK) с максимальной частотой до 50 МГц. Однако, как показала практика, МК прекрасно работает от внешнего генератора с частотой EXTCLK = 72 МГц. Теперь, после вышеупомянутых пояснений и багов, обнаруженных автором, несложно уже будет понять функциональные схемы тестируемого устройства, предназначенного для точного измерения временных интервалов. Эти функциональные схемы приведены далее.

Дальнейшее изложение будет построено следующим образом.

Вначале будут приведены функциональные схемы устройства, затем – его принципиальные схемы. Далее будет приведена разводка платы внешнего генератора и внешний вид платы устройства. После этого будут описаны программные средства (настройка всех устройств МК, описание основной программы и программы для калибровки частоты внешнего тактового генератора). В конце статьи будут приведены результаты работы устройства.

Функциональные схемы

Основная идея измерения временного интервала с помощью 3 таймеров – Т3, Т4 и PCA-таймера, включённых один за другим, – между нажатием кнопки с фиксацией и её отжатием заключается в следующем (рис. 2). После первого нажатия кнопки по срезу сигнала, поданного с выхода антидребезговой схемы на вход В частотомера, измеряющего длительность (время), запускаются таймеры, которые считывают входные импульсы, поступающие с внешнего генератора фиксированной и известной частоты (72 МГц). После того как кнопка отжата, на частотомере отражается время (длительность нажатого состояния кнопки), а таймеры останавливаются. Далее программа на МК снимает содержимое всех трёх таймеров и через определённый порт (P1.6) выводит количество импульсов, равное количеству импульсов, поступивших в таймеры, на второй вход частотомера (А), предназначенный для счёта импульсов. По второму нажатию кнопки выводится количество импульсов, подсчитанное Т3. Далее, когда закончен вывод импульсов с Т3, кнопка отжимается, а показания частотомера записываются. По третьему нажатию кнопки на частотомер выводится количество импульсов, рав-

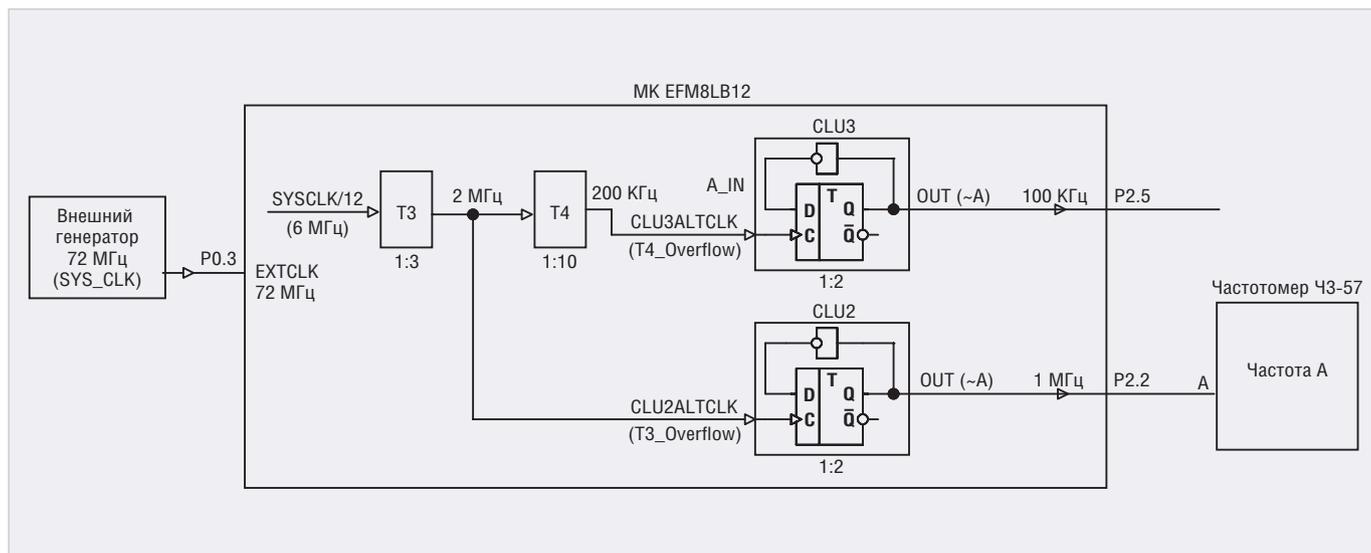


Рис. 3. Функциональная схема измерения частоты внешнего генератора

дём битовую переменную, например, EC, которую приравняем порту P2.5 (в программе это выглядит так: $EC = P2 \wedge 5$). Обозначим содержимое PCA-таймера как [PCA] и введём двухбайтную переменную N, которая будет определять количество импульсов, поступивших с T4 на CLU3. Для расчёта N можно применить следующую формулу: $N = 2 \times [PCA] + EC$. В этом случае при поступлении чётного количества импульсов (например, 20) содержимое PCA будет равно 10 ($[PCA] = 10$), $EC = 0$, и $N = 2 \times 10 + 0 = 20$. При поступлении 21 импульса $[PCA] = 10$, $EC = 1$, и $N = 2 \times 10 + 1 = 21$, что и требуется. Вот таким способом и была снята вышеописанная неопределённость. Хотя вышеописанные рассуждения больше относятся к программным средствам, о которых речь впереди, на взгляд автора, для полного понимания здесь нелишне их привести.

Если содержимое PCA-таймера строго равно количеству импульсов, поступивших на его вход EC1, поскольку начальное содержимое PCA-таймера нулевое и он просто работает в режиме счёта поступающих импульсов, то содержимое T3 и T4 отнюдь не равно количеству пришедших на их входы импульсов. Дело в том, что, помимо счёта импульсов, T3 и T4 должны ещё обеспечивать деление входной частоты на определённый коэффициент деления. В частности, T3 должен делить входную частоту в 60 000 раз, чтобы выходная частота соответствовала 100 Гц ($6 \text{ МГц} = 6\,000\,000 \text{ Гц} / 60\,000 = 100 \text{ Гц}$), а T4 – в 5 раз ($100 \text{ Гц} / 5 = 20 \text{ Гц}$). Для этого они работают в режиме автоперезагрузки. Это означает, что, когда содержимое таймера равно максимальному

значению ($0x\text{FFFF}_{16} = 6\,553\,5_{10}$), с приходом следующего импульса (65 536-го) таймер переполнится, и в его содержимое из регистров перезагрузки перепишется число, равное разнице между максимальным значением 65 536 и числом, равным коэффициенту деления. Для T3 это число равно $65\,536 - 60\,000 = 5536$ ($0x15A0$), а для T4 число перезагрузки равно $65\,536 - 5 = 65\,531$ ($0x\text{FFFB}$). В этом случае, когда в T3 придет 60 000 импульсов, а в T4 – 5 импульсов, таймеры переполнятся, и произойдёт автоматическая загрузка чисел 5536 и 65 531 в T3 и T4 соответственно. Поэтому для того, чтобы узнать количество импульсов, пришедших в таймеры (NT3 и NT4), необходимо из их содержимого вычесть число перезагрузки. Для T3 $NT3 = [T3] - 5536$, а для T4 $NT4 = [T4] - 65\,531$, где [T3] и [T4] – содержимое таймеров T3 и T4 соответственно.

Для того чтобы определить время счёта каждого таймера, необходимо количество импульсов, пришедших в каждый таймер, умножить на период T следования импульсов или, что то же самое, разделить на входную частоту F (так как $T = 1/F$). Для T3 время $TT3[c] = NT3/FT3 = NT3/6000000 \text{ Гц}$, для T4 $TT4[c] = NT4/FT4 = NT4/100 \text{ Гц}$, для PCA-таймера $TPCA[c] = NPCA/FPKA = NPCA/20 \text{ Гц}$. Полное время T будет равно сумме времён каждого из таймеров: $T = TT3 + TT4 + TPCA$. Но это ещё не всё. Дело в том, что вышеописанные рассуждения базировались на том основании, что частота внешнего тактового генератора строго равна 72 МГц. На самом деле, как следует из описания внешнего тактового генератора (в данном случае это Si546CAA001075BBG – см. далее), частота этого генера-

тора указана с погрешностью в ± 20 мд, т.е. $\pm 20 / 1\,000\,000$ от заявленного значения 72 МГц, что составляет $\pm 72\,000\,000 \text{ Гц} \times 20 / 1\,000\,000 = \pm 1440 \text{ Гц}$. Другими словами, частота генератора находится в интервале от $71\,988\,560 \text{ Гц}$ ($72\,000\,000 \text{ Гц} - 1440 \text{ Гц}$) до $72\,001\,440 \text{ Гц}$ ($72\,000\,000 \text{ Гц} + 1440 \text{ Гц}$). Для того чтобы узнать, какую же реальную частоту F выдаёт внешний генератор, её (или её, делённую на какой-либо коэффициент K) можно просто измерить частотомером и внести необходимую поправку на время T. Об этом – см. далее.

Здесь же вернёмся к функциональной схеме рис. 2. В нижней её части показана антидребезговая схема на основе CLU1 и CLU2, устраняющая дребезг контактов кнопки. Как видно из этой схемы, она представляет собой простейший RS-триггер, где A_IN CLU1 и A_IN CLU2 – его входы, выходы OUT CLU1 и OUT CLU2 – это выходы триггера Q и \bar{Q} соответственно, причём выход CLU1 соединён со входом B CLU2, а выход CLU2 соединён со входом B CLU1. Если записать логическое выражение для каждого CLU как $\sim(A \& B)$, то функция каждого CLU будет представлять собой 2И-НЕ. Если сравнить этот RS-триггер с рис. 1, то получим почти полную идентичность. Разница лишь в том, что в этой схеме (рис. 2) отсутствуют резисторы R1 и R2 (рис. 1). Но если в портах P1.2 и P1.1 разрешить слаботоковые подтяжки к питанию (Weak PullUp), т.е. установить опцию Pull-ups Enabled (см. далее), то получим полную идентичность, а значит, эта схема будет подавлять дребезг контактов кнопки так же, как это описано для схемы рис. 1.

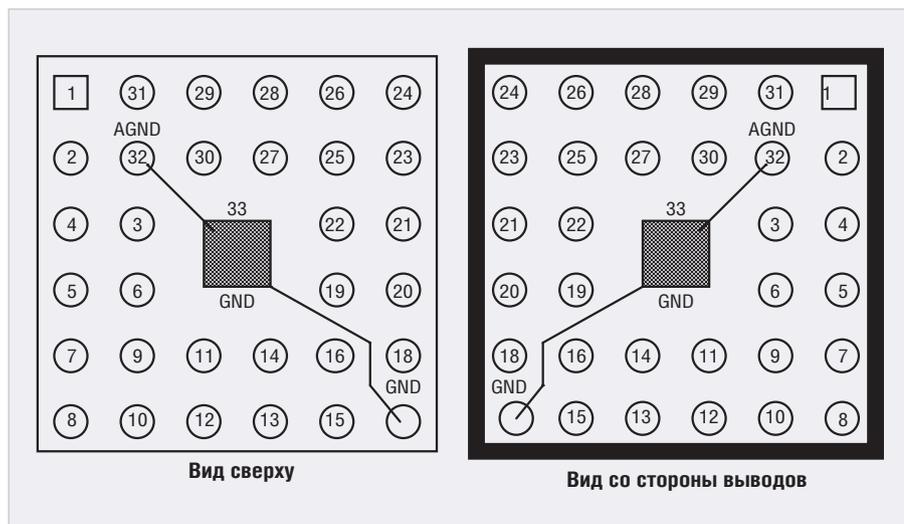


Рис. 7. Контакты переходника QFN32 – PGA32

ром используется кабель (рис. 6а), который своим одним концом подключается к выходному разъёму адаптера (X1, рис. 6а), а на втором его конце расположен разъём HU-03 (X2, рис. 6а), который подключается к разъёму X1 платы. RC-цепочка R1-R2-C1 предназначена для штатной работы интерфейса C2, по которому программируется МК и, кроме того, при включении питания запускает МК (Power-On-Reset – POR).

Если скупость не позволяет приобрести достаточно дорогой USB DEBUG адаптер (сейчас его можно купить за 3000–5000 руб.), для программирования МК можно использовать COM-порт компьютера, для чего достаточно изготовить одноканальный преобразователь уровней интерфейса RS-232 в уровни TTL (и обратно), например, с помощью известного двухканального преобразователя уровней MAX232 или используемого автором одноканального ADM3101, о котором говорится в статье автора [5]. В этой статье приведены аппаратные средства, разработанные автором, и программные средства, бесплатно предоставляемые компанией Silicon Laboratories для программирования МК с COM-порта компьютера.

К БП (+5 В) плата подключается кабелем (рис. 6б), на одном конце которого расположен разъём РС4ТВ(Р), подключаемый к БП, а на втором – разъём HU-02 (X2, рис. 6б), который подключается к разъёму питания платы (X2, рис. 4).

Кнопка (PSM2-1) представляет собой двухпозиционный переключатель с двумя группами контактов SW1 и SW2 и встроенным светодиодом LD2, торец которого выведен на лицевую поверхность кнопки. Кнопка устрое-

на так, что если нажать на неё и отпустить, это нажатое состояние фиксируется, и для того, чтобы вернуть в первоначальное состояние, на неё надо ещё раз нажать и отпустить (назовём это «отжать»). При нажатом состоянии кнопки загорается светодиод, отражающий это состояние. Резистор R4, расположенный на плате, – токоограничивающий. Группа контактов SW1 служит для подключения входов A CLU1 и CLU2 МК DD1 в соответствии с функциональной схемой рис. 2, а группа контактов SW2 – для подключения питания +5 В к светодиоду LD2.

Для подключения к частотомеру в соответствии с функциональной схемой рис. 2 использован 4-контактный цанговый штыревой разъём PSLM4 (X4), на который выведены сигналы CLU1_OUT (порт P1.0, 25-й вывод DD1), N (T3, T4, PCA-Timer) (порт P1.6, 19-й вывод DD1) и «земля». Выход CLU3_OUT (P2.5, 12-й вывод DD1) соединён со входом PCA0_ECI (P2.3, 14-й вывод DD1). На двухконтактный разъём WF-02 (X5) выведен выход CLU2_OUT (P2.2, 15-й вывод DD1) и «земля». К этому разъёму одним концом подключается кабель, второй конец которого подключается ко входу А частотомера для измерения частоты в соответствии с функциональной схемой рис. 3.

Схема рис. 4 скоммутирована вручную на макетной плате проводом МГТФ-0,03. Для МК использован переходник QFN32-PGA32 (его разводка приведена в [4]), контакты (соответствующие номерам выводов МК) которого указаны на рис. 7. Разводка платы генератора по схеме рис. 5 приведена на рис. 8 и в файле разводки (*.лауб) в дополнительных материалах на сайте

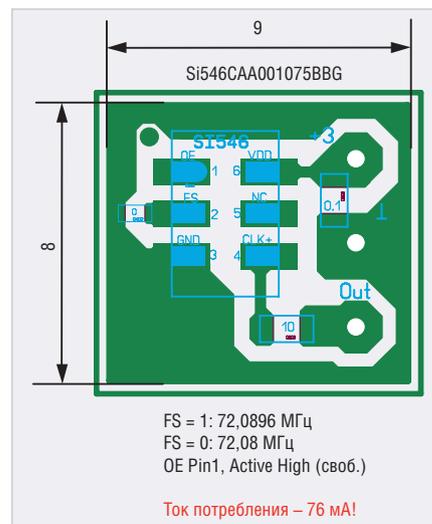


Рис. 8. Разводка платы генератора

FS = 1: 72,0896 МГц
FS = 0: 72,08 МГц
OE Pin1, Active High (своб.)

Ток потребления – 76 мА!

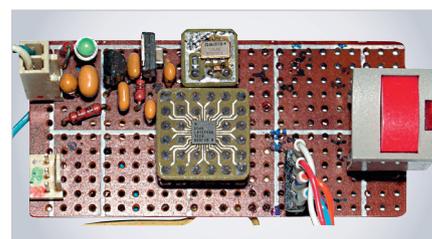


Рис. 9. Общий вид тестовой платы

журнала. Сама плата (рис. 9) небольшого размера (6,5×3 см).

Разводка платы генератора и общий вид тестовой платы. Программные средства

Программные средства состоят из двух программ. Первая, основная, предназначена для измерения длительности по времени нажатого состояния кнопки с помощью CLU и таймеров и вывода информации на частотомер в соответствии с функциональной схемой рис. 2. Вторая – для измерения частоты в соответствии с функциональной схемой рис. 3. В свою очередь, каждая из программ состоит из двух частей. Первая часть – программа инициализации устройства (InitDevice.c), вторая – основная (main). В качестве исходной автор выбрал готовую программу для CLU (EFM8LB1_Configurable_Logic_AND.c), приведённую в качестве примера в среде Simplicity Studio, бесплатно предоставляемой компанией Silicon Laboratories. Эта программа была модифицирована в соответствии с функциональной схемой рис. 2. Программа инициализации устройства InitDevice.c не требует написания её текста, поскольку он генерируется автоматически средой Simplicity Studio в соответствии с настройкой всех опций МК. Основная

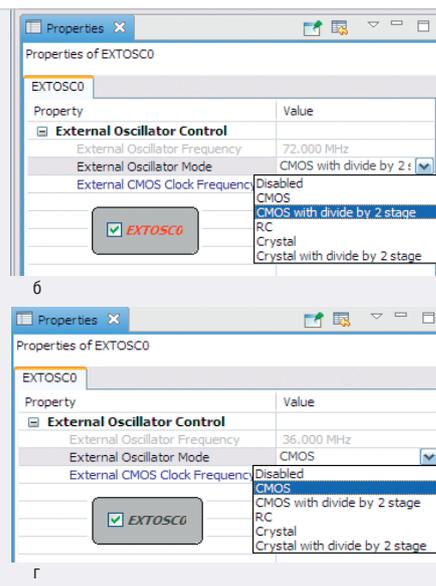
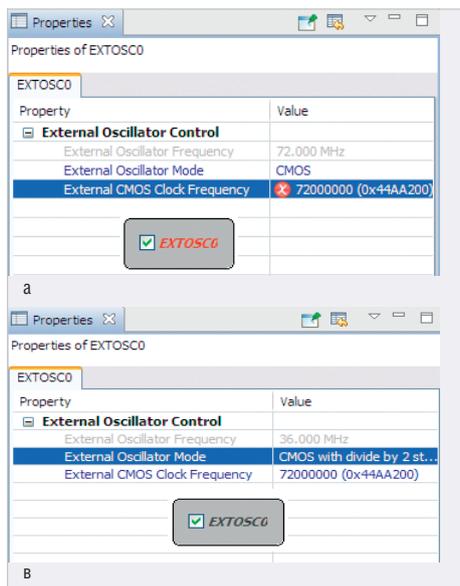


Рис. 10. Общее меню настроек МК EFM8LB12

программа написана автором на C51; её текст приведён в дополнительных материалах на сайте журнала. Словесное описание функций этой программы было уже приведено ранее в разделе статьи о функциональных схемах и, чтобы не повторяться, здесь не приводится. Поэтому ниже приводится описание настройки всех опций МК для получения программы InitDevice.c.

В основном меню среды Simplisity Studio (рис. 10) красными овалами обозначены опции, которые необходимо настроить, а синими – только разрешить (Reset Sources, Supply Monitor, Voltage Regulator) и запретить (Watchdog Timer).

Вначале необходимо настроить опцию внешнего генератора (EXTOSC0). Нажав на эту кнопку, получим окно (рис. 11а), в котором необходимо выбрать частоту 72 МГц (72 000 000) и режим CMOS. В результате такого выбора получим ошибку с красным кружком, а на кнопке надпись EXTOSC0 будет красного цвета.



Далее выберем вместо CMOS опцию CMOS with divide by 2 stage, т.е. снизим тактовую частоту в 2 раза (рис. 11б). В этом случае ошибка пропадёт, а частота установится равной 36 МГц (рис. 11в). После этого вновь выберем опцию CMOS (рис. 11г). В этом случае частота установится равной 72 МГц, но ошибки уже не будет (рис. 11д).

Далее необходимо нажать на кнопку Clock Control и в открывшемся окне выбрать опции External Oscillator и SYSCLK/1, тогда системная частота МК SYSCLK установится равной 72 МГц.

Далее необходимо нажать кнопку Core и в открывшемся окне выбрать опцию SYSCLK is below 75 MHz (это требуется для установки соответствующей скорости чтения памяти МК).

Далее необходимо настроить все необходимые CLU, для чего нажать кнопку Configurable Logic.

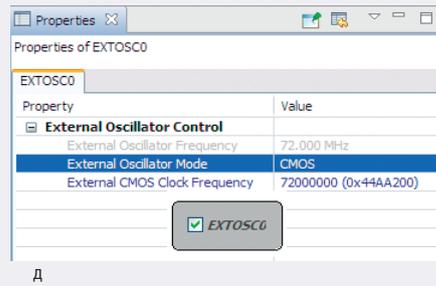


Рис. 11. Последовательность настройки внешнего генератора частотой 72 МГц (а, б, в, г, д)

Вначале необходимо настроить CLU1 и CLU2. В открывшихся окнах (рис. 12а, рис. 12б) необходимо разрешить оба CLU, разрешить вывод в порт CLU1 и запретить вывод в порт CLU2. Далее выбрать опцию LUT (Look Up Table – таблица истинности), т.е. указать, что требуется непосредственный (асинхронный) выход CLU (а не через D-триггер). Далее для CLU1 (рис. 12а) необходимо подключить вход А CLU1

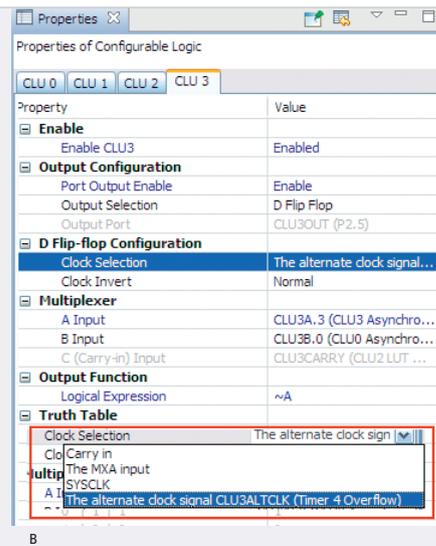
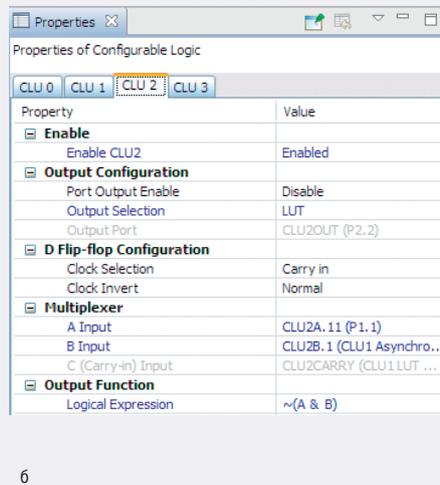
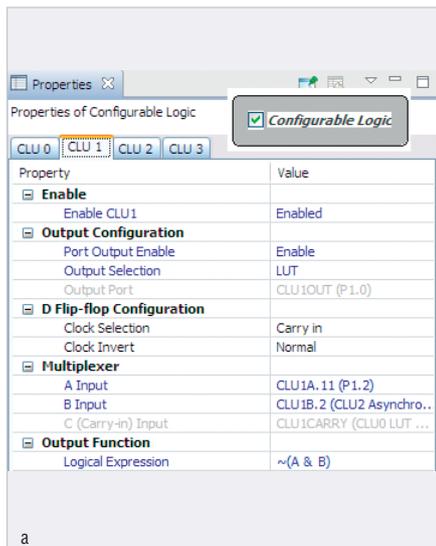


Рис. 12. Настройка конфигурируемой логики: а) CLU1; б) CLU2; в) CLU3

к порту P1.2, а вход В – к выходу CLU2. Для CLU2 (рис. 12б) подключить вход А к порту P1.1, а вход В – к выходу CLU1. Далее для обоих CLU записать логическое выражение $\sim(A \& B)$. После выбора этих опций для CLU1 и CLU2 получим RS-триггер в соответствии с рис. 2.

Теперь необходимо настроить CLU3 (рис. 12в). Здесь необходимо разрешить работу CLU, разрешить вывод его выхода в порт и указать, что вывод будет производиться через D-триггер (D Flip Flop). В качестве тактирования D-триггера выбрать таймер T4 (в красном прямоугольнике внизу рис. 12в). Далее соединить вход А CLU3 с его выходом и записать логическое выражение $\sim A$. Выбрав эти опции для CLU3, получим T-триггер, который тактируется с частотой переполнения T4 (Timer 4 overflow).

Далее настраиваем таймеры T3, T4 (рис. 13) и PCA-таймер (рис. 14). Подробно описывать окна настройки таймеров, на взгляд автора, не имеет смысла – из окон настройки и так всё понятно. Здесь только необходимо обратить внимание, что все три таймера остановлены (Stop), а для того, чтобы соединить вход ECI PCA с портом (P2.3), необходимо установить галочку в окне Port I/O Mapping у PCA ECI (рис. 14б).

Далее необходимо настроить порты МК. Для этого из режима DefaultMode Peripherals нужно перейти в режим DefaultMode Port I/O. В этом случае выведется окно, в котором будет отражён корпус МК со всеми входными и выходными сигналами (рис. 15а), а справа – окно свойств портов (рис. 15б). В этом окне свойств необходимо обратить внимание на первую строчку, где по умолчанию выбрано свойство Pull-Ups Enabled, означающую, что ко всем портам, настроенным как входы с открытым стоком (Digital OpenDrain I/O), будут подключены слаботочковые подтяжки к питанию (Weak Pullup). Как было упомянуто выше, эти слаботочковые подтяжки выполняют роль резисторов R1 и R2 (рис. 1) и таким образом дополняют схему RS-триггера на CLU1 и CLU2 (рис. 2).

Прежде всего, командами Skip необходимо «передвинуть» вход PCA0_ECI, начальное положение которого P0.0, поближе к выходу CLU3 (CLU3_ASYNC_OUT, P2.5), чтобы перемычка между этими двумя портами была покороче

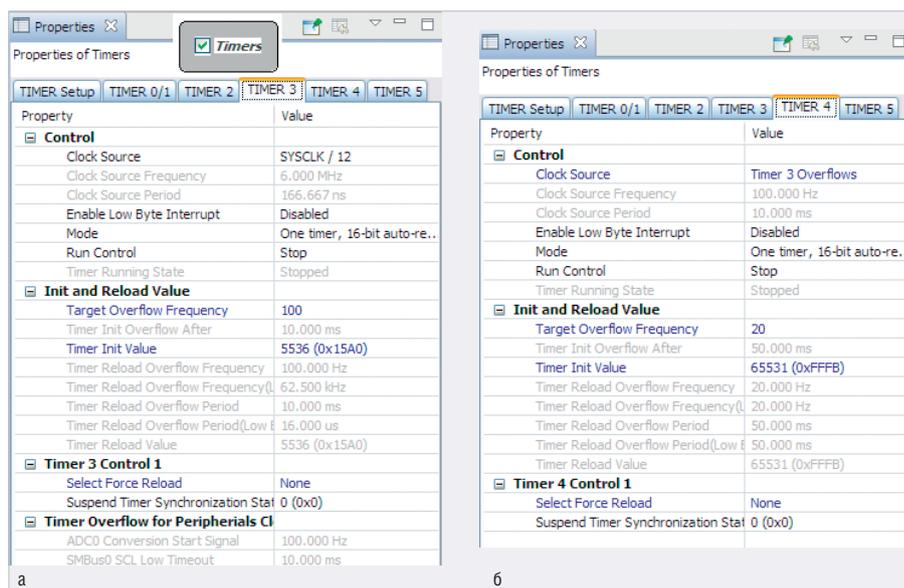


Рис. 13. Настройка таймеров: а) T3; б) T4

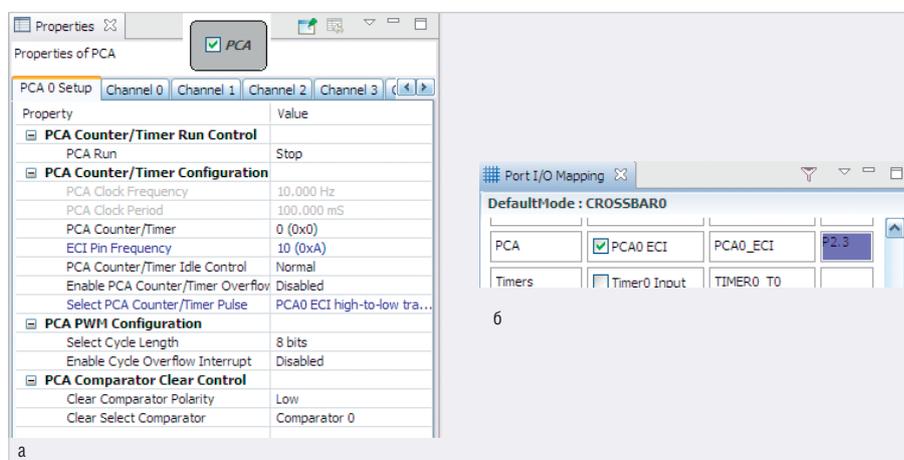


Рис. 14. Настройка PCA-таймера: а) свойства; б) подключение входа ECI к порту P2.3

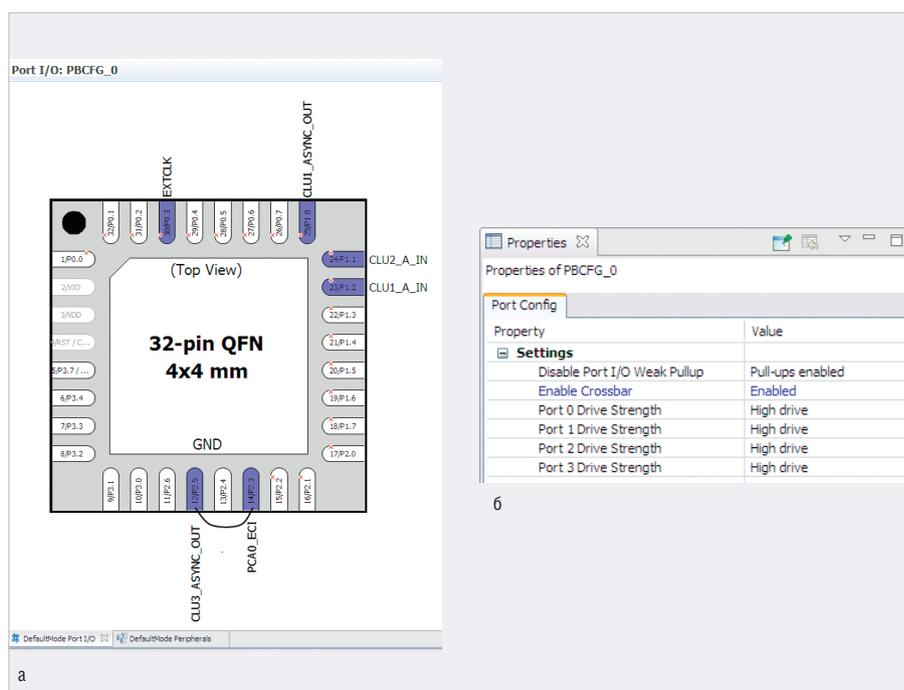


Рис. 15. Порты МК EFM8L12: а) общий вид корпуса МК с входными и выходными сигналами; б) электрические свойства портов МК

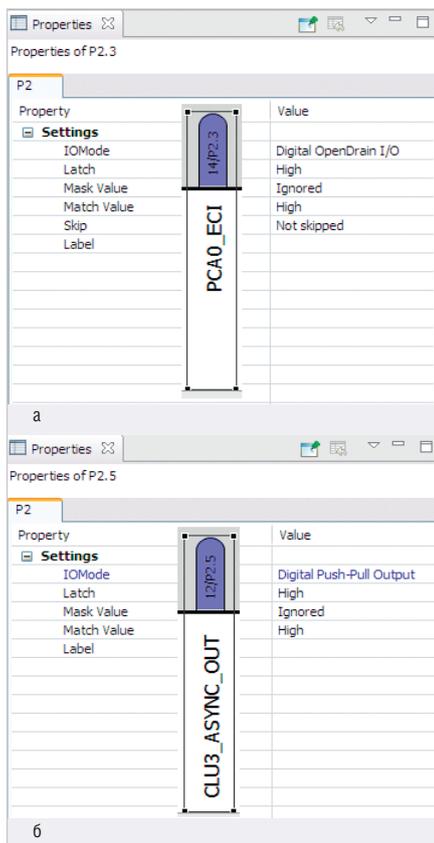


Рис. 16. Примеры установки входных и выходных свойств портов: а) вход с открытым стоком (Digital OpenDrain I/O); б) пушпультный выход (Digital Push-Pull Output)

че (на рис. 15а она показана условно). После этого входы EXTCLK, CLU2_A_IN, CLU1_A_IN и PCA0_ECI настроить как цифровые входы (Digital OpenDrain I/O), а выходы CLU1_ASINC_OUT, CLU3_ASINC_OUT и порт P1.6, по которому осуществляется вывод количества импульсов N в частотомер, настроить как цифровые выходы (Digital Push-Pull Output). Пример – на рис. 16.

На этом настройка всех устройств закончена.

Далее необходимо все настройки записать, для чего нажать кнопку с двойной дискетой в левом верхнем углу меню. При этом все настройки запишутся, и автоматически сгенерируется файл InitDevice.c уже на языке C51. Далее в верхнем меню необходимо выбрать опцию Project, а в нём – подопцию Build Project и нажать на неё. Тогда произойдёт трансляция всей программы и сгенерируется файл загрузочного формата EFM8LB1_Configurable_Logic_AND_3.hex, который уже можно запрограммировать в МК.

Текст основной программы EFM8LB1_Configurable_Logic_AND.c и оттранслированная программа

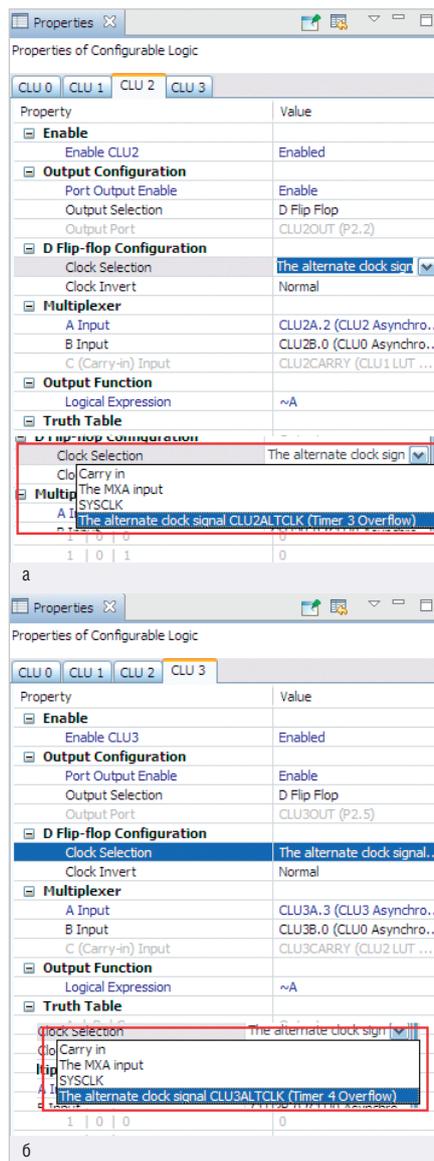


Рис. 17. Настройка CLU2 (а) и CLU3 (б)

EFM8LB1_Configurable_Logic_AND_3.hex приведены в дополнительных материалах к статье на сайте журнала.

Настройка устройств для второй программы, предназначенной для измерения частоты, более простая. Она отличается от первой только настройкой CLU2 и CLU3 (рис. 17), таймеров T3 и T4 (рис. 18) и портов (рис. 19). Подробно описывать эти настройки, на взгляд автора, нет смысла, поскольку на рис. 17–19 всё уже указано.

Необходимо обратить внимание в настройке таймеров (рис. 18) на то, что таймеры запускаются уже в режиме инициализации (Run Control – Start). Это означает, что при включении питания программа сразу начинает работать и выдавать сигналы с частотами, которые требуется измерить (1 МГц и 100 КГц).

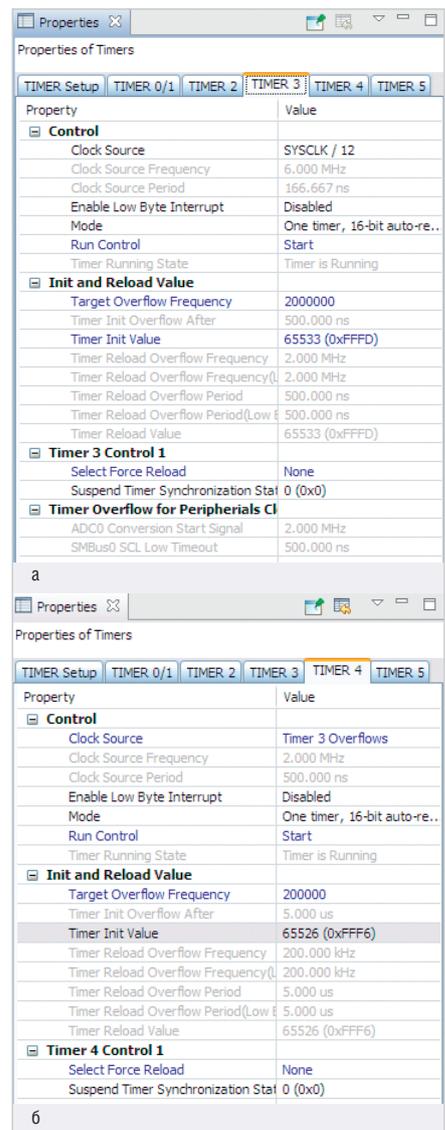


Рис. 18. Настройка таймеров T3 (а) и T4 (б)

В настройке портов (рис. 19) выходы CLU (CLU2_ASINC_OUT, CLU3_ASINC_OUT) необходимо настроить как цифровые выходы (Digital Push-Pull Output), а вход внешнего генератора EXTCLK – как цифровой вход (Digital OpenDrain I/O).

Основная программа для этого случая – вообще «никакая»:

```
void main (void)
{
    enter_DefaultMode_from_RESET();

    while (1) {}
}
```

Первым оператором она вызывает программу инициализации устройства InitDevice.c (enter_DefaultMode_from_RESET());, а второй оператор (while (1) {}) бесконечно обращается к самому себе. Это означает, что процессор МК участие в программе не принимает,

Таблица 1. Измерение времени в интервале 5...5,5 с

Таймеры	Количество импульсов, N	N/Частота [Гц]	Время T [с]
T3	56 047	56 047/6 000 000	0,00934117
T4	2	2/100	0,02
PCA	106	106/20	5,3

Таблица 2. Измерение времени в интервале 5,5...6 с

Таймеры	Количество импульсов, N	N/Частота [Гц]	Время T [с]
T3	41 690	41 690/6 000 000	0,00694833
T4	1	1/100	0,01
PCA	116	116/20	5,8

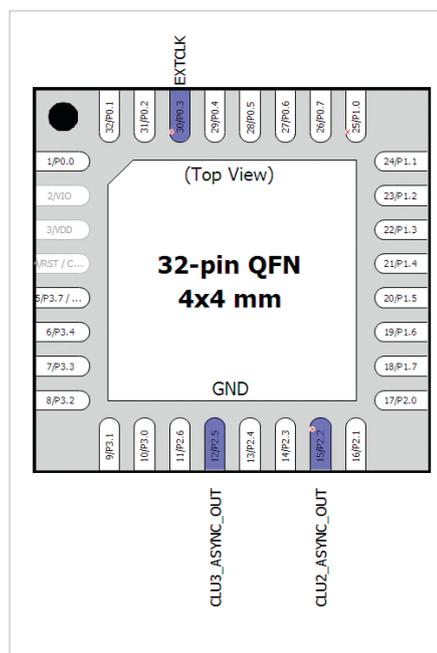


Рис. 19. Общий вид корпуса МК с входными и выходными сигналами или, другими словами, устройства (таймеры и CLU) работают автономно и не требуют участия процессора.

После всех настроек конфигурацию также необходимо записать, сгенерировать программу инициализации устройств и оттранслировать всю программу. Оттранслированный вариант этой программы (EFM8LB1_Configurable_Logic_AND_2.hex) приведён в дополнительных материалах к статье на сайте журнала.

На этом описание программных средств закончено.

Результаты

Частота, полученная на выходе CLU2 после таймера T3 (1 МГц) и измеренная частотомером, составила 999,9887 кГц, а на выходе CLU3 после таймера T4 (100 кГц) – 99,9989 кГц (см. рис. 3).

Коэффициент поправки, КЧ, для частоты 999,9887 кГц будет равен: $KЧ = 999,9887/1000 = 0,9999887$. Значение первой частоты более точное, так как у него (значения) больше цифр, поэтому она была взята за основу. Но если частота меньше, значит, период (1/F) и вообще все временные параметры, в частности, время измерения, больше, поэтому поправочный коэффициент, КТ, на который необходимо умножить время измерения, будет равен: $КТ = 1/0,9999887 = 1,0000113$.

Для того чтобы проверить правильность формулы расчёта количества импульсов, поступающих на PCA-таймер ($N = 2 \times [PCA] + EC$ – см. выше), или, другими словами, работу поправки EC, было проведено два эксперимента. В первом время измерения было чуть меньше 5,5 секунд (когда $EC = 0$), во втором – чуть больше ($EC = 1$). Частоты, поступающие на входы T3, T4 и PCA таймера, составляют соответственно $FT3 = 6 \text{ МГц} = 6\,000\,000 \text{ Гц}$, $FT4 = 100 \text{ Гц}$, $FPCA = 20 \text{ Гц}$ (рис. 2), поэтому для получения времени, измеренного каждым таймером, необхо-



Компактные высоковольтные преобразователи напряжения



ПРОПОРЦИОНАЛЬНЫЕ И СТАБИЛИЗИРОВАННЫЕ МОДЕЛИ

Технические параметры

- Входное напряжение 5, 12, 24 В
- Выходные напряжения от 2 до 10 кВ
- Мощность от 2 мВт до 15 Вт
- Диапазон температур от -55 до +70°C
- Длительный ресурс

Применение

- Медицинская диагностика
- Научное оборудование
- Авиационно-космическая техника

ProCHIP
POWERED BY ProSOFT



ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

(495) 232-2522 • INFO@PROCHIP.RU • WWW.PROCHIP.RU

димом соответствующее количество импульсов N разделить на соответствующую частоту (или, что то же самое, умножить на период), а для получения общего времени T все времена сложить. Результаты измерений приведены в табл. 1 и табл. 2. Итого:

- по табл. 1: $T = 5,32934117$ с; с учётом поправки ТП = $T \times \text{КТ} = T \times 1,0000113 = 5,3294013$ с; время ТЧ [мкс], измеренное частотомером, составило $5\,329\,402$ мкс = $5,329402$ с;
- по табл. 2: $T = 5,81694833$ с; с учётом поправки ТП = $T \times 1,0000113 = 5,8170141$ с; время ТЧ [мкс], измеренное частотомером, составило $5\,817\,015$ мкс = $5,817015$ с.

Как видно из сравнения времени, измеренного частотомером (ТЧ), и времени, измеренного МК (ТП), разница между ними не превышает 1 мкс. Если подсчитать относительную погрешность измерения времени МК, то она будет составлять $(1/1\,000\,000\text{ с})/5\text{ с} = 0,0000002$, что составляет 0,00002%, или 0,2 мд.

Автор, признаться, предполагал получить абсолютную погрешность

в районе десятков мкс, в крайнем случае, единиц мкс. Но на то, что погрешность составит всего 1 мкс, автор даже не рассчитывал и был несколько шокирован таким результатом. Было проведено с десяток экспериментов с временем менее 5,5 с и с временем более 5,5 с. Один раз погрешность составила 2 мкс, два раза ТЧ и ТП вообще полностью совпали. В остальных случаях погрешность составила 1 мкс.

Заключение

Использование конфигурируемых логических элементов (CLU), встроенных в МК EFM8LB12, совместно с таймерами позволяет конструировать устройства, измеряющие временные интервалы с достаточно высокой точностью. Однако это лишь один из примеров использования CLU. Область их применения существенно шире. Они дают возможность создавать внутри МК несложные логические устройства с такими функциональными свойствами (например, антидребезговые схемы, Т-триггеры и т.п.), которые ранее требовали дополнительных аппа-

ратных средств (микросхемы и другие электронные компоненты), и тем самым позволяют от них избавиться. А это, в свою очередь, существенно упрощает «обвязку» МК, что приводит к снижению сложности схемы всего устройства, уменьшению размеров его платы и корпуса для него и, в конечном счете, к значительной экономии средств на производство самого устройства.

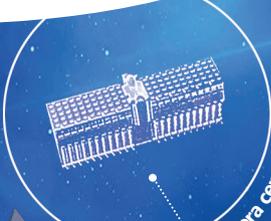
Литература

1. EFM8 Laser Bee Family. EFM8LB1 Reference Manual // URL: efm8lb1-rm.pdf. Silabs.com.
2. AN921: Configurable Logic Unit // URL: AN921_CLU.pdf. Silabs.com.
3. Кузьминов А.Ю. Электронные устройства. Современные аппаратные и программные средства. М.: Перо, 2021.
4. Кузьминов А.Ю. Связь между компьютером и микроконтроллером. Современные аппаратные и программные средства. М.: Перо, 2018.
5. Кузьминов А. Программирование микроконтроллеров EFM8 с помощью встроенного загрузчика программ // Радио. 2018. № 12.

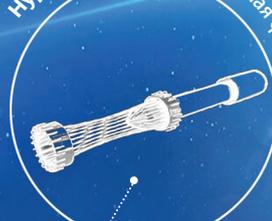


smiths interconnect

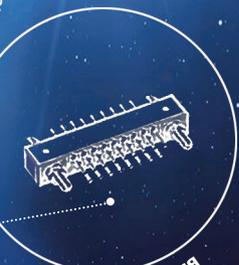
ВАША БЕЗОПАСНОСТЬ — НАША ОТВЕТСТВЕННОСТЬ



SpaceNXT™ Aurora серия



Hypertac® — гиперболическая форма контакта



KA серия

Разъемы для космической, авиационной, медицинской техники и железнодорожного транспорта

<p>Высокоскоростные разъемы QuadraX/Twinax Разъемы на печатную плату Кабельные сборки</p>	<p>Высокочастотные разъемы Оптические соединители Соединители с подпружиненными контактами</p>
---	--



ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

(495) 232-2522 ■ INFO@PROCHIP.RU ■ WWW.PROCHIP.RU

Реклама



ЧИТАЙТЕ, КАК ВАМ УДОБНО



**ПЕЧАТНАЯ И ЦИФРОВАЯ
ВЕРСИИ ЖУРНАЛА**

ЖУРНАЛ С ГАРАНТИРОВАННОЙ ДОСТАВКОЙ

**СОВРЕМЕННАЯ
ЭЛЕКТРОНИКА**

**ПОДПИСКА
НА ЖУРНАЛ**

онлайн: www.soel.ru • +7 495 232-0087 • info@soel.ru
на почте: по каталогу «Урал-пресс» (индекс для печатной версии 36280)