

ПЕРЕДЕЛЫВАЕМ DSO138

или

как получить горсть пятаков на грош

или

сделать что-то похожее на инструмент из игрушки

Предыстория и введение

На Али продается за очень недорого интересная игрушка – осциллограф под названием DSO138. Он снискал уже довольно большую популярность среди любителей электроники, но параметры этого приборчика, увы, позволяют его более-менее полноценно использовать только для отладки очень низкочастотных схем. Собственно, он и не позиционируется как инструмент, а скорее, как DIY-kit для начинающих электронщиков.

Собран этот «игрушечный» осциллограф на микроконтроллере STM32F103, и при достаточно грамотном схемотехническом решении цифровой части, наличии довольно приличного цветного дисплея 320X240 точек, и не самом поганом аналоговом тракте, все, увы, гробится очень слабыми АЦП на борту 32F103. Заявленная полоса в 200 кГц может быть признана таковой только с очень большой натяжкой. Да, наличие или отсутствие сигнала с такой частотой он покажет, но вот реально посмотреть что-то сверх этого не получится.

Есть и еще несколько, так скажем, странных недоделок. Например, USB порт в наличии имеется, но использовать его можно только для подачи питания, на связь с компьютером он не работает. Причина? Скорее всего выбранная тактовая частота ядра не бьется с 48 МГц, необходимыми для корректной работы USB. Сервисные функции, например, та же калибровка нуля, без которой жить просто невозможно, реализованы очень неудобно, да и хотелось бы, чтобы сервиса было побольше, например, была бы возможность измерения параметров сигнала с использованием курсоров.

При этом есть у 103-й серии чуть более старший брат – STM32F303, по ножкам совместим практически полностью, но по интересующим нас параметрам существенно лучше, на борту 4 АЦП с частотой преобразования 5 МГц (6 МГц с 10-ти битным разрешением). При таком раскладе, если использовать все 4 АЦП параллельно с 10-ти битным разрешением, можно получить временное разрешение до честных 24 MSPS (миллионов отсчетов в секунду). Стоит микроконтроллер тоже недорого, на том же Али можно легко найти за опять же весьма умеренные деньги. Понятно, что идея поменять микроконтроллер возникла практически сразу после того, как я этот самый DSO138 и попробовал.

При этом из игрушки может получиться вполне полноценный инструмент, которым уже могли бы пользоваться даже и профессионалы, а не только начинающие любители. Вот с такими мыслями я и решил попробовать сделать что-нибудь с китайской игрушкой в свободное от работы время.

Постановка задачи, или, что хотелось бы получить в результате

Задачу я себе поставил следующую – получить в результате минимальной переделки железа на основе DSO138 и новой программы инструмент, годный для отладки цифровых схем с микроконтроллерами, что подразумевало бы:

- Полосу хотя бы $0 \div 2$ МГц при возможности видеть наличие/отсутствие сигнала с частотой до 8 МГц (популярная частота кварцевых резонаторов, применяемых с микроконтроллерами);
- Возможность использования USB порта для связи с компьютером (сохранения осциллограмм);
- Нормальное сервисное меню вместо скудного набора комбинаций кнопок;

У DSO138 продается два варианта, просто DSO138 и DSO138-mini. Второй (мини) показался мне более распространенным и подходящим для переделки. Его основное отличие в том, что есть четкое деление на цифровую и аналоговую части (платы). Вот его-то (DSO138 mini) я взял и начал переделывать. Под раздачу попали версия I цифровой, или как в оригинале основной (Main Board) платы и версия J аналоговой (Analog Board).

В честь того, что используется 303-й микроконтроллер, проект получил название DSO303.

Цифровая часть (плата)

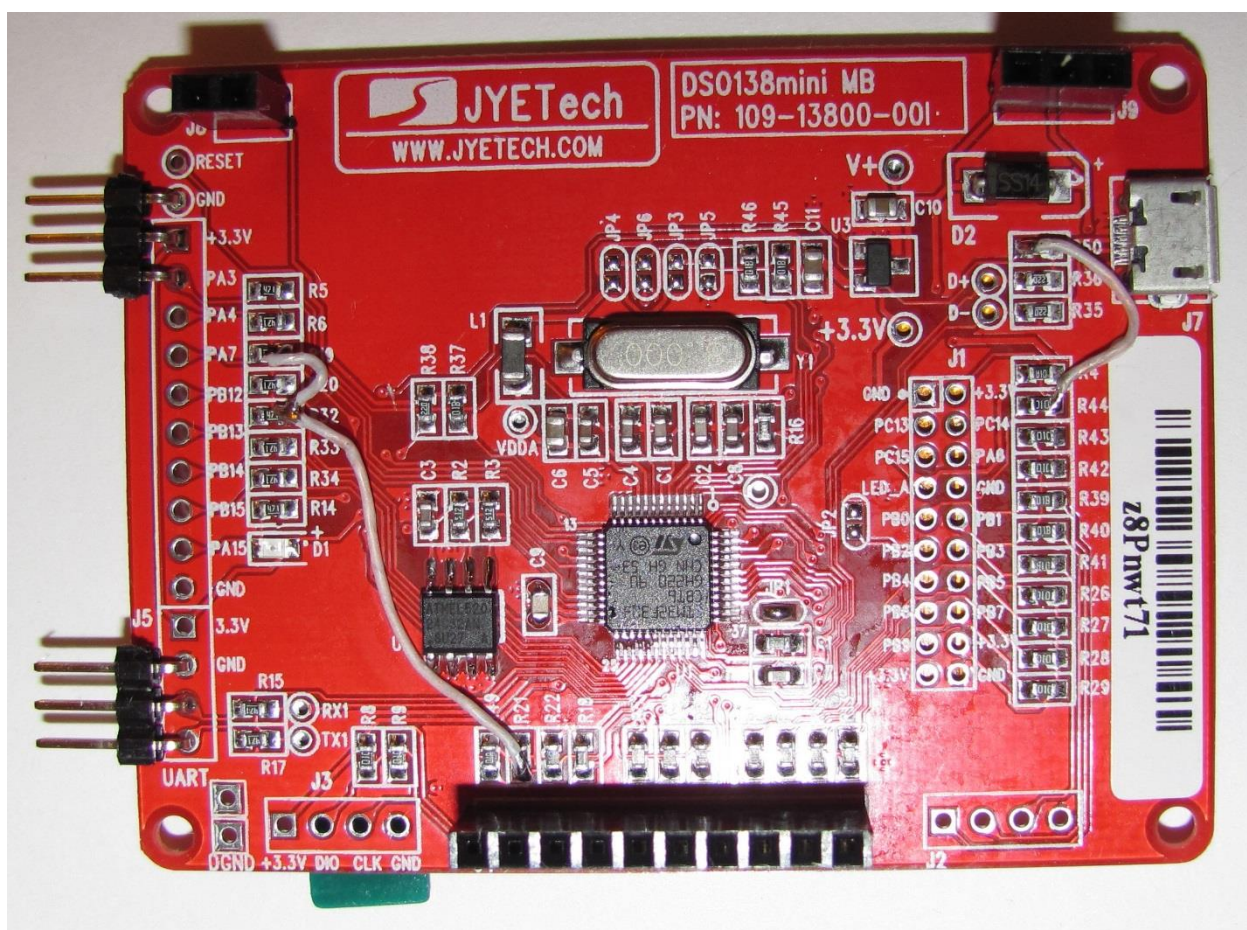


Рис.1. Переделки цифровой части (общий вид).

Переделки цифровой части (Рис.1) минимальны и сводятся к следующему:

- 1) Режем одну дорожку, соединяющую R50 и PA7 микроконтроллера, и соединяем эту ножку R50 с питанием +3,3В (Рис.2 и Рис.3). Это необходимо по двум причинам: порт PA7 нужен для одного из входов АЦП, подача постоянного питания на R50 позволит использовать USB и в режиме загрузчика прошивки.



Рис.2. Режем дорожку.

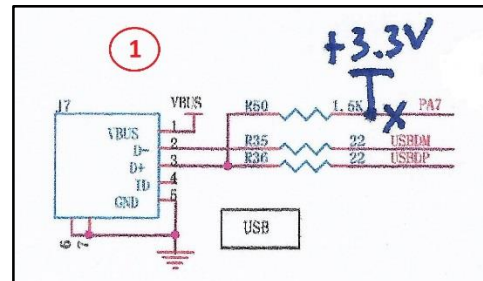
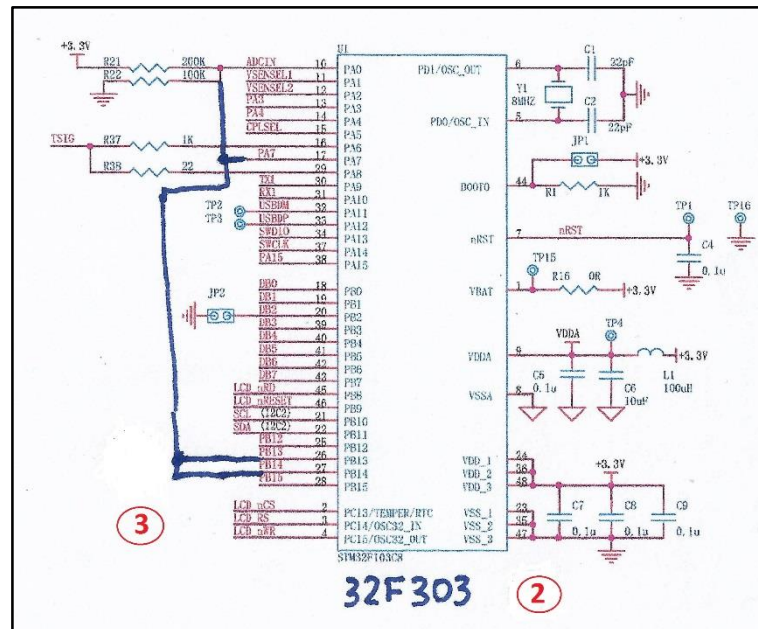


Рис.3. Переделки USB порта.

- 2) Заменяем микроконтроллер на STM32F303CB или STM32F303CC. Для работы программы необходимы минимально 128 кБ флэш-памяти и 40 кБ ОЗУ.
- 3) Соединяем вместе четыре входа АЦП: ножки PA0 (10), PA7(17), PB13(26), PB14(27). Рисунок 4.



оказался сильно не похожим на какой-либо стандартный. А дальше по ходу работы над программой выяснилось, что необходимо минимум 4 кБ энергонезависимой памяти для хранения калибровок и прочих данных. Поэтому, ту микросхему, что есть на плате, и которая имеет емкость по некоторым данным только 2 кБ, все равно менять бы пришлось.

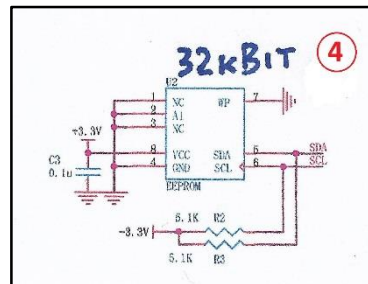


Рис.5. Замена EEPROM.

- 5) Запайваем штырьковые разъемы на 3 контакта на последовательный порт (низ платы на Рис.1) и на порт PA3 + GND (верх платы на Рис.1). Последовательный порт можно использовать одновременно и наравне с USB для соединения с компьютером и передачи данных. Порт PA3 используется как вход (аналоговый!) для подачи сигнала внешней синхронизации (триггера). На этот порт заведен вход компаратора внутри микроконтроллера. Уровень (порог) срабатывания триггера задается программно в меню осциллографа.

В оригинале также была возможность подачи внешней синхронизации, но там требовался цифровой логический сигнал уровня 3,3В.

ВНИМАНИЕ: Сигнал синхронизации подается на вход микроконтроллера напрямую. Максимально допустимое напряжение при этом не должно превышать 3,3В. Отрицательное относительно земли (GND) платы напряжение на этом входе соответственно также крайне нежелательно.

Аналоговая часть (плата)

В аналоговой части в оригинале используется операционный усилитель TL084 с паспортной полосой пропускания где-то в районе 4 МГц. Понятно, что, если мы хотим хотя бы видеть сигнал частотой 8 МГц, то эту микросхему также придется поменять на более широкополосную. Переделки аналоговой части (Рис.6) собственно и сводятся к такой замене и сопутствующим изменениям в номиналах элементов входного тракта.

Итак, меняем TL084 на любой счетверенный усилитель с полосой от 100 МГц в подходящем корпусе. В моем случае я остановился на AD8054 (другой вариант, например, TL974). Полоса пропускания у этих микросхем по паспорту – 150 МГц, хорошо, и для наших целей более чем достаточно.

Далее, сразу же выяснилось, что смещение нуля в общем случае для AD8054 существенно больше, и, соответственно, оставлять вход усилителя совсем открытым без подтяжки резистором к земле, как в оригинальной схеме, не получится, что собственно и

привело к изменениям схемотехники входной части. На вход операционного усилителя был добавлен резистор 110 кОм.

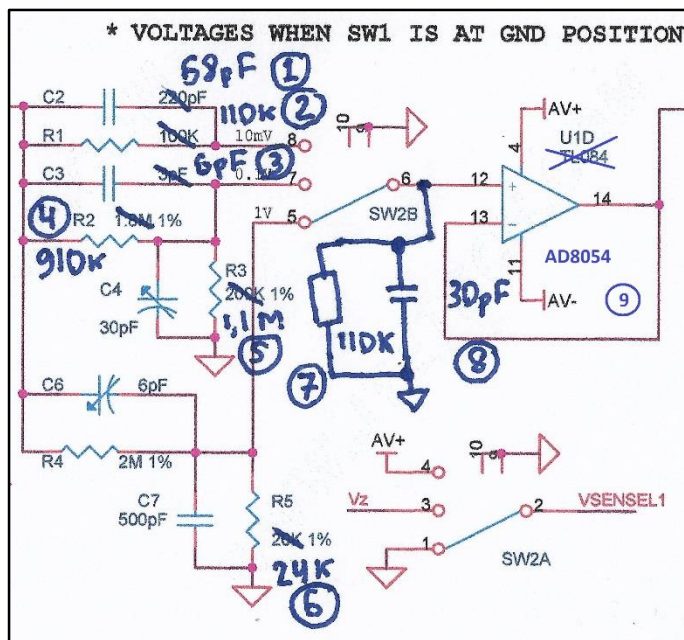


Рис.6. Изменения входной части аналогового тракта.

Дальнейшие изменения – это в некоторой степени компромисс между желанием оставить как можно более нетронутой печатную плату (резать ничего не надо), и способности на приемлемом уровне входные параметры. Понятно, что в нашей ситуации оставить большое входное сопротивление на диапазоне 10 мВ – это несбыточная мечта. В оригинальной схеме на этом диапазоне на вход осциллографа подключался вход ОУ без делителя, и входное сопротивление определялось просто входным сопротивлением ОУ.

В нашем случае ставить большой делитель на этом диапазоне 10 мВ равносильно неприемлемо большой потере в точности и качестве обработки этого сигнала АЦП, либо необходимости введения каким-то образом дополнительного усиления в тракте, что приводит к уже существенным переделкам платы. Поэтому в предположении, что основной фронт работ – это цифровая схемотехника, для которой этот диапазон не является основным рабочим, я ограничился делением на 2, что далее компенсируется в программе, и входным сопротивлением 200 кОм, что конечно плохо, но приемлемо.

Еще одно компромиссное решение – сделать входное сопротивление 1 МОм именно для диапазона 1В, что позволяло бы использовать на этом диапазоне щуп с делителем 1:10, для диапазона 0,1В я счел это не столь актуальным. При этом для диапазона 0,1В входное сопротивление получается 660 кОм, тоже плохо, но приемлемо.

В любом случае, я полагаю, что эту часть каждый имеет право и волен делать по своему усмотрению, предлагаемое решение только один из возможных вариантов.

В зависимости от того, какой усилитель будет использован, возможно, будет необходимо добавить в схему питания ОУ еще один преобразователь для получения отрицательного напряжения параллельно уже существующему. В схеме в оригинале используется ICL7660 для получения отрицательного напряжения. Данная микросхема допускает параллельное соединение неограниченного числа таких преобразователей для

увеличения выходного тока. Если окажется, что одного преобразователя ICL7660 недостаточно для питания выбранного ОУ, то на свободное место на плате надо будет установить еще одну микросхему ICL7660 с двумя электролитическими конденсаторами 100 мкФ соединенными аналогично U2, C12, C13, и подключить ее параллельно существующей.

Проконтролировать необходимость такой установки просто по значению напряжения «V-» в контрольной точке TP8, в идеале это напряжение должно быть по значению не меньше -3В, еще лучше - -3,5В. Если измеренное напряжение существенно меньше, то поставить еще один ICL7660 необходимо.

Программа

Ну а теперь переходим к самому вкусному для меня – к прошивке...

Основная идея была в том, чтобы реализовать как можно больше действий по обработке входного сигнала «железным» образом и не нагружать этим само ядро микроконтроллера, что гарантировало бы, например, то, что все оцифрованные данные будут консистентны во времени. Таким образом схема сбора данных выглядит так:

1. По сигналам от блоков сравнения таймера (TIM1 CC1...4) с фиксированными сдвигами по времени запускаются АЦП 1...4, каждый от своего сигнала (в матрице внутренних соединений микроконтроллера такая возможность есть);
2. Данные после оцифровки по DMA записываются в буфера;
3. Об окончании набора буфера программа узнает по прерыванию от контроллера DMA;
4. При внутренней синхронизации триггер вырабатывается блоком компаратора аналогового watch-dog АЦП2;
5. При внешней синхронизации триггер дает компаратор COMP2, на один вход которого подается сигнал синхронизации, а на другой напряжение с одного из каналов ЦАП, что позволяет задавать уровень срабатывания триггера.

Программа при этом крутится в бесконечном цикле:

1. Опросили управление (сигналы от кнопок и переключателей);
2. Если что-то поменялось – обновили экран;
3. Если есть готовность к новому циклу - запустили сбор данных;
4. Если есть готовность данных – обработали данные, обновили экран, поставили флаг готовности нового цикла;

Так как для синхронизации нужен фронт, который может быть как нарастающим, так и спадающим, а сигнал с аналогового watch-dog АЦП при внутренней синхронизации только потенциальный, то для выделения фронтов его пришлось еще завести на вход таймера ETR1 TIM8. И уже по прерыванию таймера отслеживать триггер.

Далее, я решил для упрощения программы, что все 4 АЦП будут работать всегда, даже на длинных временах развертки, когда и одного АЦП было бы вполне достаточно. Это может оказаться полезным, так как таким образом можно учетверить эффективную скорость выборки при медленных развертках, что может помочь, например, при поиске «иголок в стогу», или для выделения огибающей при шумном сигнале.

В принципе на бумаге все выглядело достаточно гладко... При 10 битной оцифровке получалось 12 тактов на отсчет при частоте 72 МГц, что давало при 4-х работающих со сдвигом АЦП $72 \text{ МГц} / 12 * 4 = 24 \text{ МГц}$ (24MSPS), или 24 точки за 1 микросекунду.

На медленных развертках все прекрасно и заработало, но вот при попытке получить максимальную скорость начались овраги... Для начала выяснилось, что смещение АЦП меняется при увеличении частоты выборки, и его нужно калибровать для каждой частоты оцифровки отдельно и, вдобавок, довольно часто. Но главная беда была в том, что явно были видны пропуски в оцифровке, что приводило при стыковке данных от разных АЦП к полной мешанине. Причем, чем выше была частота входного сигнала и чаще прерывания от синхронизации (триггеры), тем непригляднее выглядела картинка.

Казалось бы, благодаря матричной структуре (Рис.7) и разделению ОЗУ на два блока, процессы вычитывания данных из АЦП по DMA (используются S3, S4, M2, M5), и работа основной программы, которая при этом использует все остальное, но «практически» не пересекается с ресурсами, которые используются в данный момент DMA, должны вполне спокойно сосуществовать одновременно.

Figure 1. STM32F303xB/C and STM32F358xC system architecture

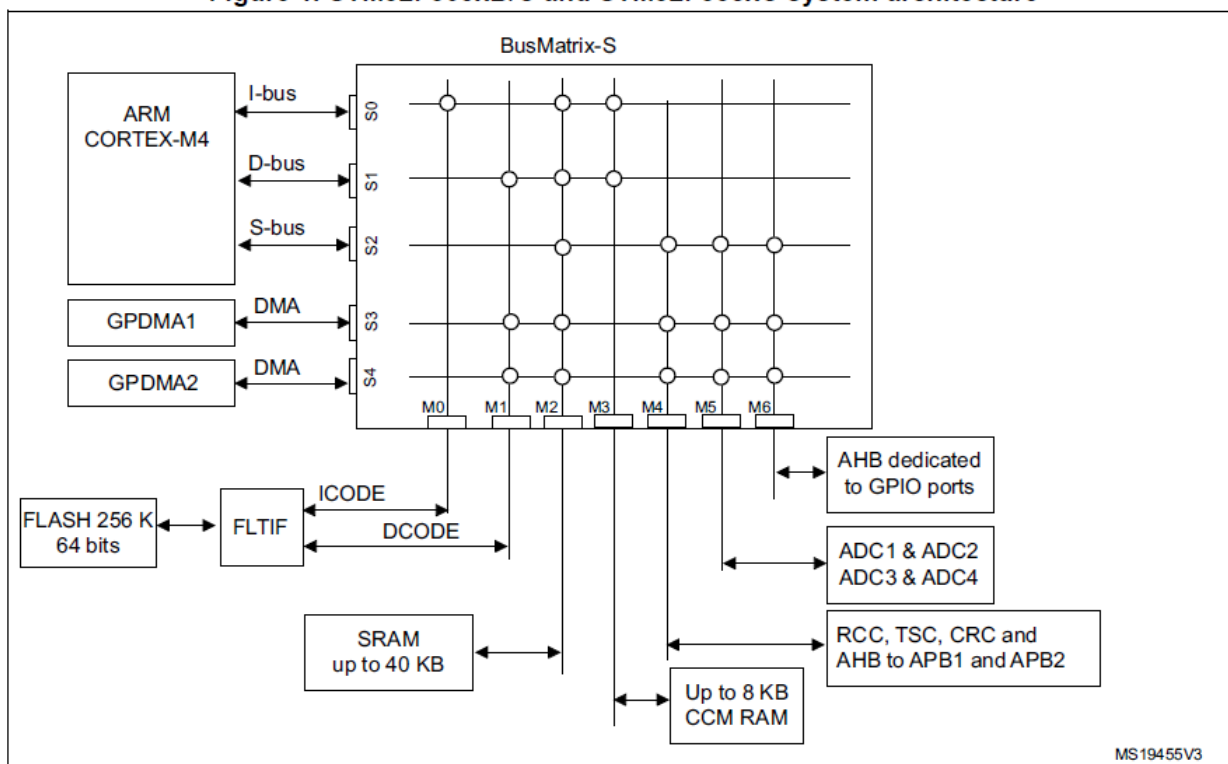


Рис.7. Архитектура системы STM32F303 из Reference Manual (для справки).

Но, увы, фактически тех операций, которые происходят при обработке триггеров синхронизации данных, а по прерыванию надо слезить и в блок АЦП, и в контроллере DMA прочитать текущий счетчик, с учетом того, что шины DMA загружены толи 8 тактов из 12, толи 12 из 12, сколько точно я так и не смог выяснить) оказалось достаточно, чтобы испортить картинку.

У АЦП есть еще второй режим работы, когда они работают попарно с фиксированным сдвигом. Вычитывать пару, при этом, можно только одной операцией, что должно резко облегчить жизнь и доступ к шине. Сказано – сделано. Для режимов с быстрой разверткой (быстрее 5 мксек. на клетку) сделал вариант с попарной работой. Не помогло... Картинка

стала лучше, в том смысле, что сбои теперь стали идти как бы волнами, но все равно ничего хорошего. Посмотрел Errata, оп-па, как раз прямо специально мой случай, попарная работа АЦП с DMA для больших частот дискретизации **не гарантируется!** Предлагается использовать АЦП в одиночном режиме. Ситуация оказалась практически патовой.

После долгих размышлений и попыток что-то сделать пришла в голову мысль, что, поскольку сбои теперь возникают не всегда, а доступ к контроллеру DMA теперь не приводит к фатальным последствиям (шина-то все ж разгрузилась!), то можно попытаться просто контролировать возникновение сбоя, сравнивая счетчики разных каналов постоянно, и просто повторять цикл набора данных в случае, если в процессе был этот сбой обнаружен. Такая тактика оказалась приемлемой. Сбои все равно остались, но стало хотя бы что-то более-менее видно.

Итоги

Итог и результат трудов - на Рис.8. На вход подан меандр частотой 1 МГц.

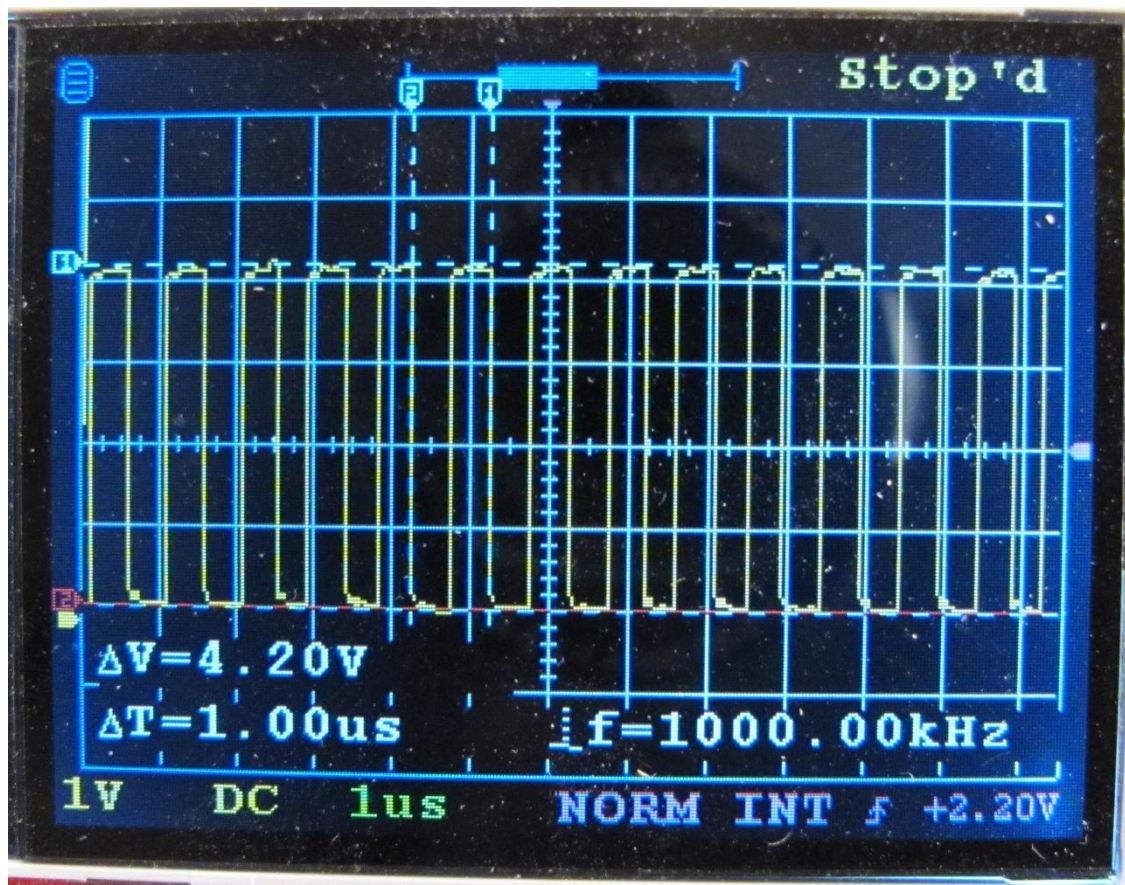


Рис.8. Меандр частотой 1 МГц.

Сигнал 2 МГц тоже вполне четко видно, но разглядеть подробности уже совсем не получится. Наличие/отсутствие сигнала можно спокойно увидеть для достаточно высоких частот, по крайней мере 25 МГц на ножках кварца микроконтроллера видно четко (Рис.9). При этом наблюдаем весьма коварный эффект биений между частотой дискретизации (в данном случае 12 МГц) и частотой сигнала – 25 МГц, который обязательно возникает, если частота дискретизации кратно меньше частоты сигнала. Сигнал выглядит как будто он

имеет частоту 1 МГц, но это как раз разность между удвоенной частотой дискретизации и сигналом.

Также отметим, что щуп пришлось поставить в положение 1:10, так как входное сопротивление в 220 кОм, подключенное к ножке кварца напрямую, естественно, срывает генерацию. Коэффициент деления получается при этом

$$9,1 \text{ МОм} + 220 \text{ кОм} / 220 \text{ кОм} \approx 42,$$

т.е. в клетке примерно 420 мВ.

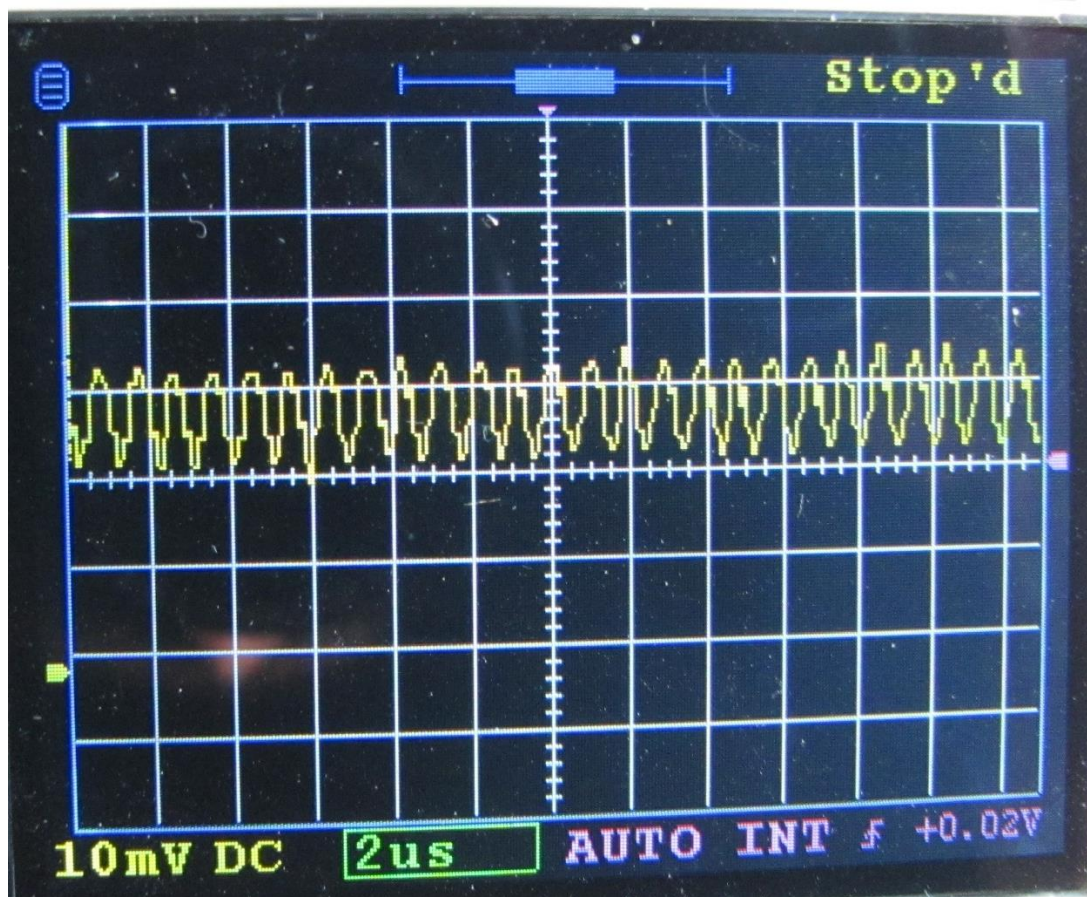


Рис.9. Сигнал с кварцевого резонатора 25 МГц.

Русской душе хочется всегда чего-то большего, чем то, что достижимо. Прикинув теоретическую максимально достижимую скорость дискретизации с точки зрения входа АЦП, а это определяется минимальным временем открытия УВХ, которое в нашем случае $1,5 \text{ такта} \times (1 / 72 \text{ МГц}) \approx 20,8 \text{ наносекунд}$, что дает 48 MSPS, стал думать, а можно ли как-то этого достичь при имеющемся железе?

И вот, в итоге все же удалось посредством некоторого, но вполне законного фокуса, перепрыгнуть через голову и получить для некоторых, не для всех конечно, вариаций входного сигнала, эффективные 48 MSPS и время развертки 500 и 200 наносекунд на клетку, не без искусственных артефактов (Рис.10), но иногда и вообще абсолютно чисто (Рис.11)! Но, фокус работает только для правильно-периодических сигналов, и

разглядывать так сигналы частотой выше 1 МГц все равно, увы, не получится. Про все это далее в отдельной публикации...

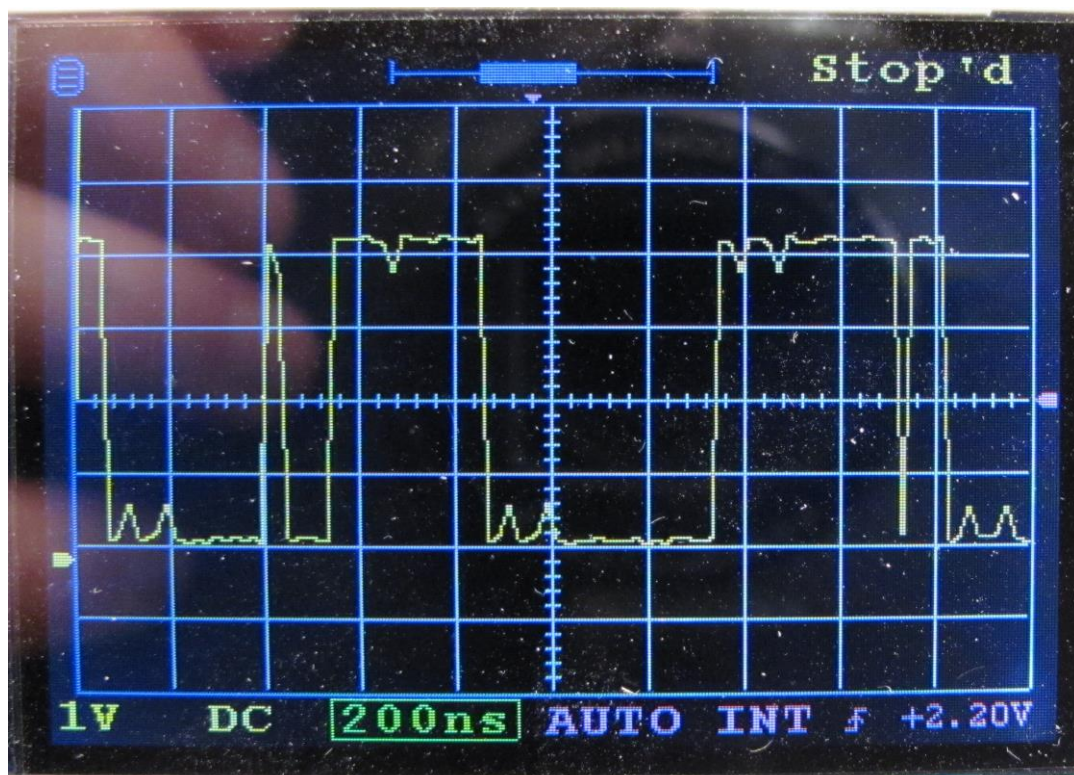


Рис.10. 48 MSPS. Меандр 1 МГц. С артефактами.

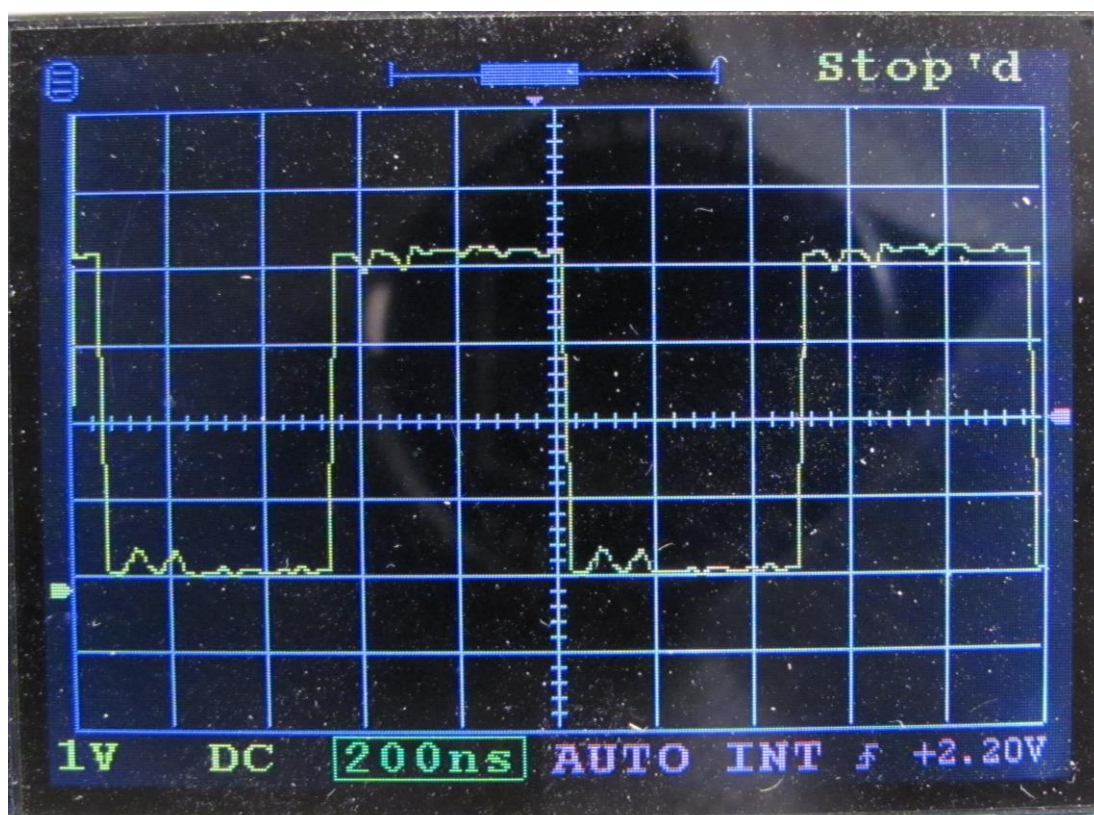


Рис.11. 48MSPS. Меандр 1 МГц.

Заключение

Результат для работы с цифровыми микроконтроллерными устройствами хочу признать достаточно хорошим. Программой и инструкцией к ней, само собой, готов поделиться со всеми желающими (пока бесплатно). Несколько слов об экономике:

- 1) DSO138 mini от 1000 руб. на Али (в комплекте со щупом и корпусом где-то в районе 1500 руб.);
- 2) STM32F303CBT6 от 300 руб. (тоже на Али);
- 3) AD8054ARZ от 30 руб. (там же);
- 4) ICL7660 от 40 руб. за партию в 5 шт. (там же);
- 5) Остальная рассыпуха не более 50 – 100 руб;
- 6) Программа – пока бесплатно, если будет коммерческий интерес, т.е. большое число желающих, то буду брать за лицензию деньги, но не дороже 1000 рублей.

Таким образом можно уложиться в 2000, максимум 3000 рублей. Что-то подобное по характеристикам и возможностью применения для отладки микроконтроллерных устройств скорее всего будет USB-приставкой к компьютеру, а не полноценным автономным прибором, и с ценником в районе 7 тысяч рублей.

Если заинтересовало, пишите, готов ответить на все интересующие вопросы и поделиться программой...