

## Основные сведения о правильной оцифровке реальных сигналов.

Для сохранения интересующей нас информации частота дискретизации выбирается не менее чем в 2 раза выше наивысшей (реально необходимой) гармоники сигнала.

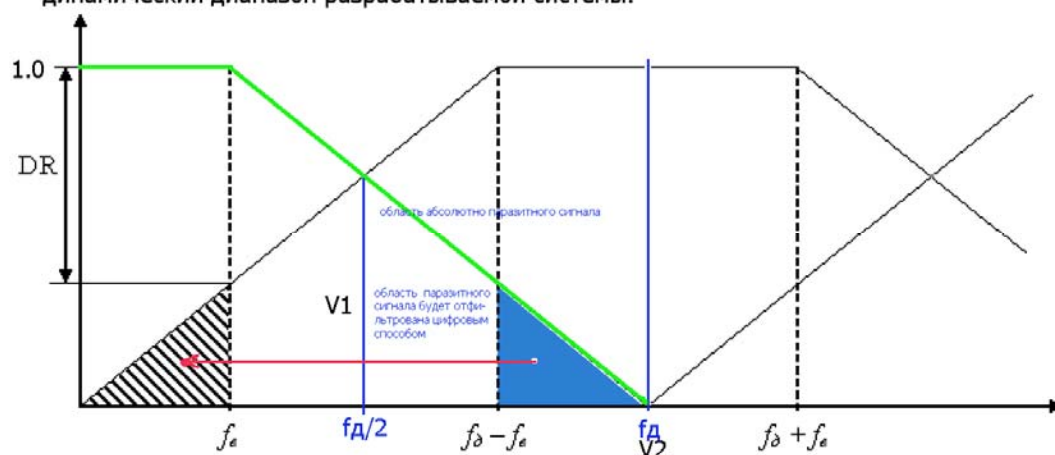
Для предотвращения зашумления результата внеполосным шумом (находящимся выше наивысшей гармоники сигнала, в том числе и выше частоты дискретизации) необходимо соблюдение пропорций как на рис.1.

DR, динамический диапазон цифрового устройства

$f_v$  - верхняя граничная частота пропускания аналогового фильтра перед АЦП.

$f_d$  - частота дискретизации АЦП.

В приведённом примере составляющие спектра, которые попадают в диапазон частот от  $f_v$  до  $f_d/2$ , не представляют интереса, так как они будут в дальнейшем отфильтрованы цифровым фильтром. Поэтому они не ограничивают динамический диапазон разрабатываемой системы.



**Рисунок 1** Влияние частоты дискретизации на требования к характеристикам аналогового фильтра.

Все шумы и сигналы из синей области окажутся приплюсованными к отсчетам АЦП зепкально перенесенными по частоте в нужную нам область частоты от 0 до  $f_v$ . Здесь бороться с шумами и помехами способен только аналоговый фильтр перед АЦП. Желательно иметь максимально возможное и нарастающее подавление в полосе частот от  $f_v$  до  $f_d$ .

Зеленым нарисована АЧХ фильтра перед АЦП. Уровень 1,0 – максимальное входное напряжение АЦП.  $V_1$  – остаточный уровень от входного сигнала на половинной от частоты дискретизации; он в идеале не должен превышать 1...2 разрядов АЦП.  $V_2$  – остаточный уровень от входного сигнала на частоте дискретизации; он в идеале должен быть заметно меньше цены (в вольтах или процентах от максимального входного напряжения АЦП) одного разряда АЦП. Из представленного следует, что надо повышать частоту дискретизации в соответствии с реализуемым аналоговым фильтром перед АЦП. Если в сигнале присутствуют близкие по частотам помехи (в нашем случае 50 Гц, 100 Гц, 150 Гц, част. мерцания ламп или мониторов, наводки от сети и др.), то для хорошей цифровой фильтрации помехи также должны быть качественно оцифрованы. В нашем случае получается  $f_d = \sim 330...450$  Гц.

Здесь заложен главный компромисс между ценой-реализуемостью аналогового входного фильтра и ценой-скоростью АЦП, а также вычислительными ресурсами для цифровой фильтрации.

Есть очень плохой эффект при выборке и дискретизации. Это алиасинг (Алиасинг, наложение — в статистике, обработке сигналов и смежных дисциплинах эффект, приводящий к наложению, до неразличимости различных непрерывных сигналов при их дискретизации. Алиасинг — одна из главных проблем при аналого-цифровом преобразовании видео- и аудиосигналов. Неправильная дискретизация аналогового сигнала приводит к тому, что высокочастотные его составляющие накладываются на низкочастотные, в результате чего восстановление сигнала во времени приводит к его искажениям. Для предотвращения этого эффекта частота дискретизации должна быть достаточно высокой и сигнал должен быть надлежащим образом отфильтрован перед оцифровкой.) – из википедии. Физически это нарезка всего спектра входного сигнала на

кусочки  $0 \dots F_d, F_d \dots F_d * 2, F_d * 2 \dots F_d * 3, \dots \infty$  и помещение стопкой этих кусочков в область  $0 \dots F_d$  с их суммированием. При этом получается жуткая каша, в случае невыполнения требований как на рис.1.

Для примера, в хороших звуковых картах Creative Fd в 256 раз выше макс. частоты звукового диапазона.

## Что мы имеем.

В нашем случае недорогого массового изделия выбор АЦП сделан разработчиками микропроцессора – 10 бит – это 1023 уровня от 0 до опорного напряжения (у нас это напряжение питания). Формально с учетом погрешности  $\pm 1$  (в условиях отсутствия помех) разряд АЦП это будет динамический диапазон 54 дБ. Помехи и наводки суммируясь с сигналом отбирают часть дин. диапазона, в случае вялых анемичных девушек с плохим периферическим кровообращением, львиную его долю.

В связи с этим предстоит искусная борьба за расширение динамического диапазона, выделение сигнала из помех цифровой фильтрацией, введением эффективной цифровой АРУ для поддержания постоянного размера сигнала на экране и для качественной стат. обработки.

## Что мы пробовали.

В RB16 были: аналоговый входной НЧ фильтр 2 порядка, аналоговая (ВЧ фильтр 3-го порядка в обр. связи 1 го каскада усилителя) схема стабилизации постоянного уровня входного сигнала в пределах диапазона входных сигналов АЦП (около половины напряжения питания), цифровая простая (из-за очень небольших вычислительных ресурсов микроконтроллера) АРУ (автом. регул. усиления). Вся эта музыка требовала 5 операционных усилителей и  $\sim 20$  резисторов и конденсаторов.

В RB18 было: аналоговый входной НЧ фильтр 1 порядка (2 детали дополнительно), цифровая схема стабилизации постоянного уровня входного сигнала в пределах диапазона входных сигналов АЦП посредством ШИМ микроконтроллера (она стремится путем подачи вычисленного напряжения на вход единственного оп. усилителя (суммирование с входным сигналом) поддерживать на его выходе уровень равный 510 отсчетам АЦП), простой цифровой НЧ фильтр для прореживания отсчетов АЦП в связи с низкой пропускной способностью вычислительного USB. Основная фильтрация и цифровая АРУ (можно ее назвать и нормализацией уровня) выполняются в PC приложении. Здесь была обнаружена засада в малом отношении сигнал/шум при обследовании вялых анемичных девушек с плохим периферическим кровообращением.

## Что мы поняли.

Надо соединить достоинства RB16 и RB18 и тем устранить имеющиеся недостатки.

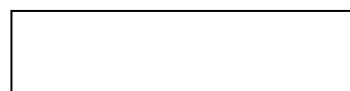
От RB16 берем фотодиод (большой динамический диапазон по освещенности), виртуальный COM порт (достаточная пропускная способность, драйвера для всех Windows), входной НЧ фильтр 2 порядка.

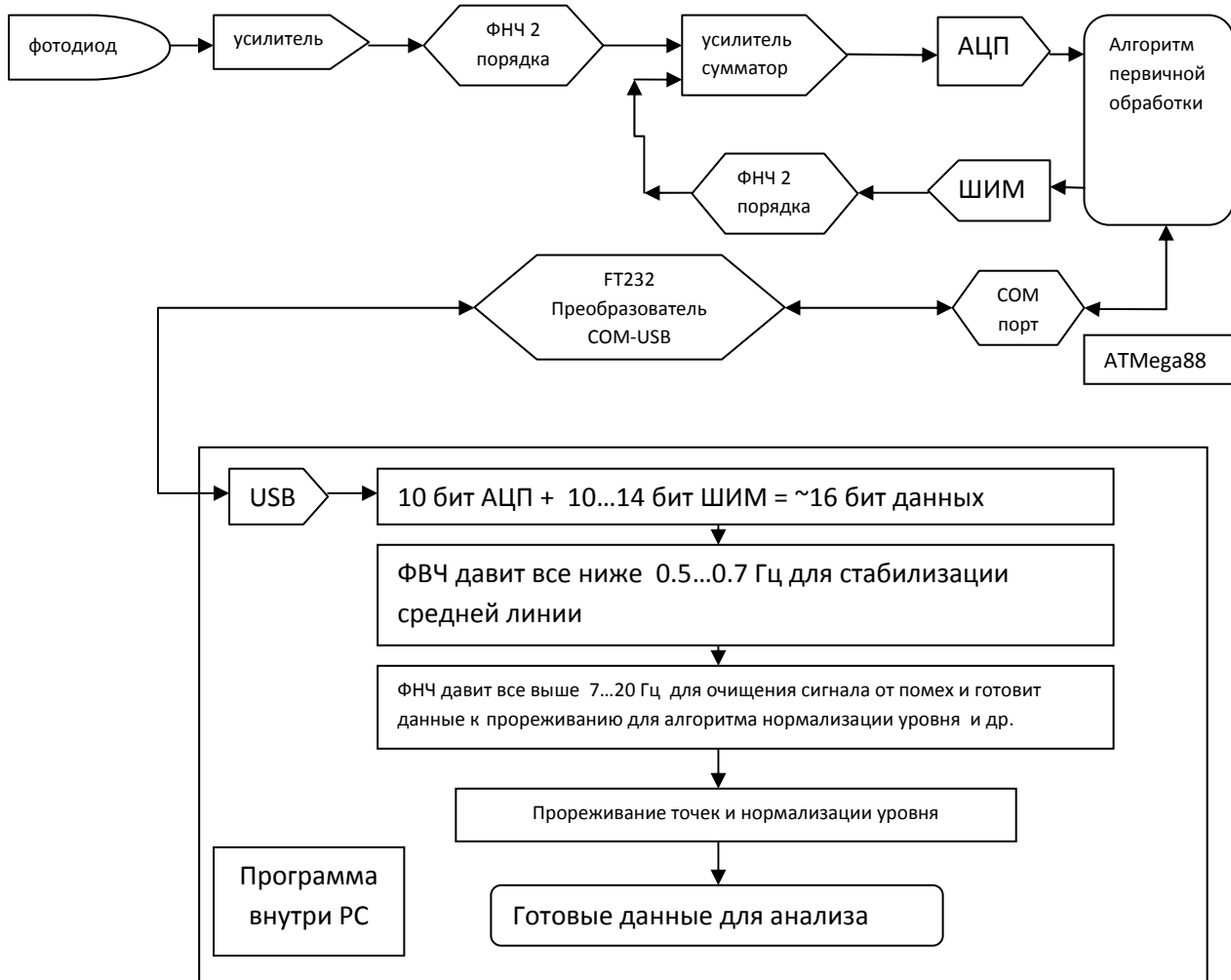
От RB18 берем цифровую схема стабилизации постоянного уровня входного сигнала в пределах диапазона входных сигналов АЦП посредством ШИМ микроконтроллера, которая дает возможность вычислить дополнительные биты (было опробовано АП в одном приборе специально сделанным для ВН) к 10 полученным от АЦП и тем самым расширить динамический диапазон цифрового представления сигнала, что необходимо для качественной фильтрации и нормализации уровня.

Все это должно превратится в вариант RB20 – прибор сбалансированный по потребительским и технологическим свойствам.

## Функциональная схема RB20.

Вот такой представляется схема во втором приближении.





**Алгоритм первичной обработки** – получает 10 бит из АЦП и на их основе получает число для загрузки в регистр ШИМ контроллера, как это сейчас сделано в RB18. Частота оцифровки жестко определяется тремя вещами:

1. Аппаратурой ШИМ контроллера
2. Кварцевым резонатором
3. Допустимыми частотами работы COM порта.

Основное - частота оцифровки равна частоте ШИМ, и не противоречить двум другим пунктам.

В PC надо посылать 10 бит из АЦП и несколько бит из ШИМ. Сколько бит из ШИМ посылать надо решить практически т.к. он 14 разрядный и посылать так много наверняка излишне и непрактично.

В первой пробе частота оцифровки была около 388 Гц.

**Программа внутри PC** – получает пары отсчетов АЦП и ШИМ и по алгоритму, учитывающему передаточную функцию усилителя сумматора с вычислительной обратной связью получает отсчет с разрядностью не менее 16 бит. Сколько из этих бит несет информацию о сигнале, станет ясно после фильтрации.

Далее ФВЧ давит все ниже 0.5...0.7 Гц для стабилизации средней линии, что должно заметно подавить артефакты движения испытуемого.

Далее ФНЧ давит все выше 7...20 Гц (здесь возможна регулировка в зависимости от качества исходного сигнала, чем хуже сигнал, тем меньше значение) для очищения сигнала от помех. Алгоритм нормализации уровня должен иметь буфер отсчетов на 3...4 периода пульса. Работа алгоритма нормализации уровня сводится к вычислению дисперсии отсчетов сигнала в буфере и вычислению коэффициента, на который надо умножить отсчет сигнала для того, чтобы дисперсия отсчетов сигнала в буфере находилась в заданном диапазоне, а значит и амплитуда сигнала стабилизируется на заданном уровне. Сейчас этот алгоритм реализован в персональной версии для прибора RB18.

В результате этой обработки сигнал хорошего уровня будет иметь 7...11 значащих разрядов, а сигнал малого уровня будет иметь 5...8 значащих разрядов, что является достаточным для представления на экране и последующей мат. обработки.

## Порядок действий.

Разработка и опробование вычислительных блоков может производиться независимо друг от друга на начальном этапе.

Все неясности и недосказанности выясняются в рабочем порядке по мере необходимости.

Первая проба у АП слеплена им из различных ранее существовавших неоптимальных кусков различных тестовых программ. Фильтры были рассчитаны на САПР (случайно доступной на короткое время в одном ящике) в виде объектников. В таком виде она явно не может являться источником для нормального проекта RB20.

Фильтры надо проектировать сразу оптимальными по требованиям к выч. ресурсам.

Здесь надо иметь ввиду, что частоты и длины фильтров сильно влияют на разные отклонения от желаемого результата. Часто изменение какого либо из параметров фильтра на 1...3% приводит к замечательному результату.

Тузов В. В.

Конкретные дополнения от А.Кириллова.

### О фильтре ВЧ

Частота среза – первая гармоника самой низкой частоты пульса по уровню 0.7 / ~ 0.76 Hz /

/\*\*\*\*\*\*/

/\* 889 Tap High Pass Hamming \*/

/\* Finite Impulse Response \*/

/\* Sample Frequency = 260.0 Hz \*/

/\* Standard Form \*/

/\* Arithmetic Precision = 4 Digits \*/

/\* Pass Band Frequency = 750.0 mHz \*/

/\*\*\*\*\*\*/

100 mHz ~ - 40 dB.(миллигерцы)

Т.к. происходит сглаживание, можем увеличить число градаций на 3 ... 4 бита.

### О фильтре НЧ

Частота среза – третья гармоника самой высокой частоты пульса по уровню 0.7 / ~ 7.5 Hz /

/\*\*\*\*\*\*/

/\* 91 Tap Low Pass Blackman \*/

/\* Finite Impulse Response \*/

/\* Sample Frequency = 260.0 Hz \*/

/\* Standard Form \*/

/\* Arithmetic Precision = 5 Digits \*/

/\* Pass Band Frequency = 7.5 Hz \*/

/\*\*\*\*\*\*  
50 Hz ~ - 60 dB.

Т.к. происходит сглаживание, можем увеличить число градаций на 1 ... 2 бита.

## О нормализации амплитуды

Стабилизация заданной дисперсии в окне длиной 1.5...2 периода самого медленного пульса. Я

довольствовался прямоугольным окном и стабилизировал амплитуду FPG на уровне 9 ... 10 бит.

Крайне не рекомендую менять последовательность фильтров и прореживать, надеюсь, понятно почему.

Все данные относятся к RB-18, кварц - 12.0 MHz и Sample Frequency = 260.0 Hz - 52 Hz \* 5.

Напомню 52

Hz при 11 битах получилась от узкой полосы USB канала, пришлось фильтровать Fadc = 366 с

прореживанием на 7, из-за сглаживания повысить разрешение  $\sqrt{7} \sim 2.6$ , т.е. ~1 бит, и передавать

приращения. Но эти извращения, я надеюсь, Вы повторять не будете, а в RB-16 остановитесь на Fadc

~ 450 Hz, уточни для кварца 7.3728 Hz и PWM – 14 bit. От точно вычисленной частоты пляши и

проектируй все протоколы передачи, фильтры, интервалы и т.д. Удачи.

А.Кириллов 20.03.2010

## О работе PWM / отработано на RB-18 /

```
// =====  
// ***** A D C F P G *****  
// =====
```

adc:

```
if( Flag_A != 0 ) goto end;
```

```
sadcl = ADCL; / -младший байт результата АЦП
```

```
sadch = ADCH; / -старший байт результата АЦП
```

```
Flag_A = 1;
```

```
Flag_M = 0;
```

```
DatADC = (((long)(sadch))<<8 ) + (long)(sadcl) - 511;
```

```
AD = DatADC; // отклонение от 511
```

```
if(AD >= 0) { AD = AD * AD *(-1);}
```

```
else { AD = AD * AD *(1);}
```

```
Sum += AD ; // колим квадраты отклонений, но без потери знака. Квадраты – чтобы ОС была нелинейная, более частая сетка при низкой амплитуде.
```

```
if(Sum > 536000000) Sum = 536000000; // чтобы не переполнить PWM ~ 2**29
```

```
if(Sum < 3) Sum = 3;
PW_H = GET_PWM_H(Sum); // Леша сделал подпрограмму, где из Sum выкусывает PWM_H
PW_L = GET_PWM_L(Sum); // ... PWM_L – по сути сдвиг и маска. Кажется >>23 и >>15 .
// 29 – 23 → 6 старших PWM_H , 29 – 15 → 8 младших PWM_L
OCR1BH = PW_H; // ***** Load PWM *****
OCR1BL = PW_L;
...
```