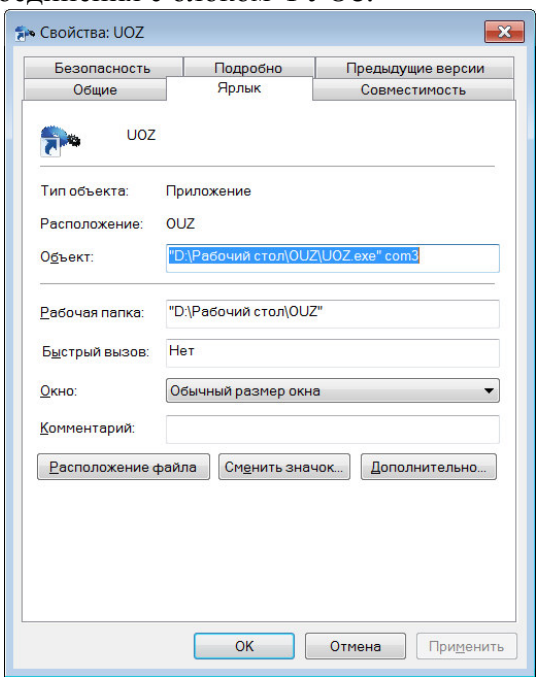
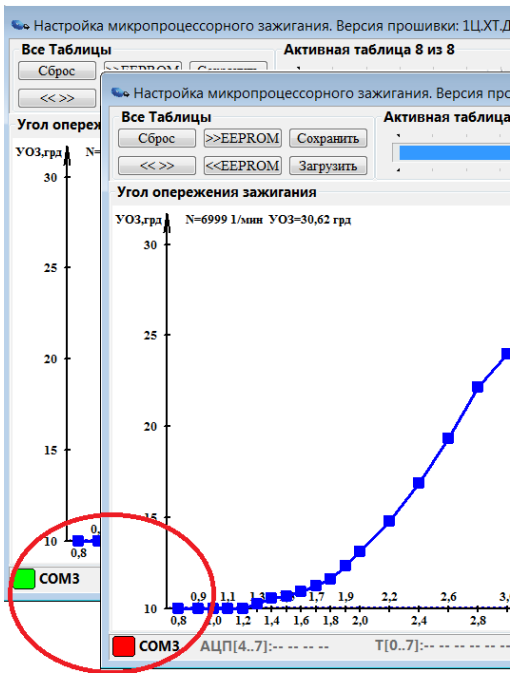
Версия от 01.10.2021
(с учетом изменений от 28.08.2021)

1. В программу UOZ добавлена поддержка номера COM порта, указанного как параметр при запуске. Ускоряет запуск при больших номерах COM портов. Поддержка – до COM15. Добавлен индикатор соединения с блоком ФУОЗ.





2. Настройка Ядра. Для начальной установки параметров Ядра используется порт CoreSetup, для внесения изменений в него используется макрос **SetCoreSetup Value.** Назначение бит параметра Value

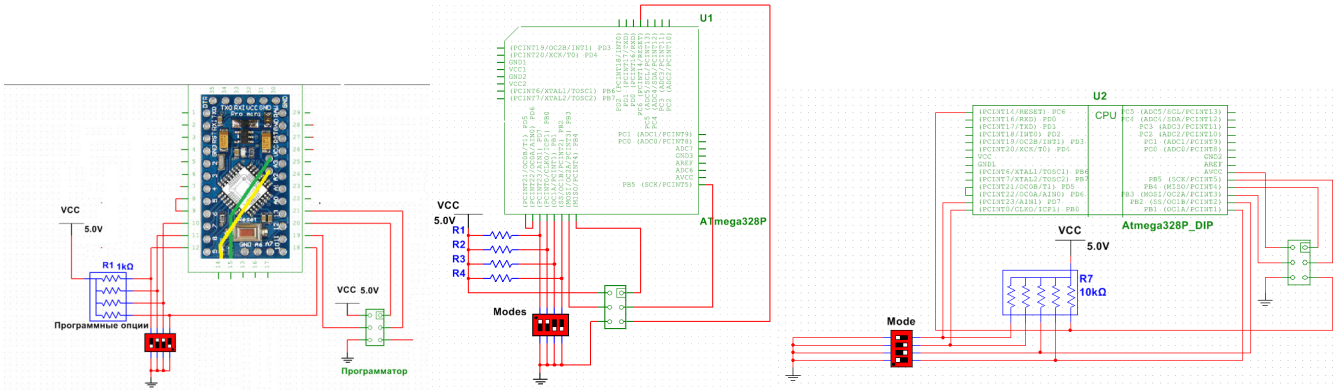
7-4	Служебный флаг - при настройке может иметь произвольное значение
3	Режим работы корректора нагрузки на оборотах выше табличных
2	Корректор нагрузки (0 - включен, 1 – отключен)
1	АЦП (0 - включен, 1 – отключен)
0	UART (0 - включен, 1 – отключен)

Начальное значение порта CoreSetup - **все включено** кроме бита №1, который автоматически сбрасывается Ядром, если пользователь активировал любой канал АЦП (_ADCCanCorSelectr, _ADCCanTblSelectr, _ADCCanA, _ADCCanB) и наоборот устанавливается в противном случае.. Пример:

SetCoreSetup 0b00000110 ; Отключен корректор нагрузки и отключен АЦП

В Ядро добавлена возможность **отключения отдельных параметров Ядра, не отключенных** при запуске макросом **SetCoreSetup**. Корректировки параметров системы происходит без перепрошивки микроконтроллера с помощью DIP переключателей на плате или изменения ячейки памяти EEPROM с заранее известным адресом (**EEPROM аналог DIP переключателей**). Следует отметить что, использование ячеек памяти возможно лишь в случае, когда количество используемых таблиц не превышает 16 или используется так называемая «желтая» перемычка - п.4. Еще раз уточню, что все манипуляции с DIP переключателеми, в том числе с их EEPROM аналогами, позволяют отключать **только включенную ранее макросом SetCoreSetup функцию!**

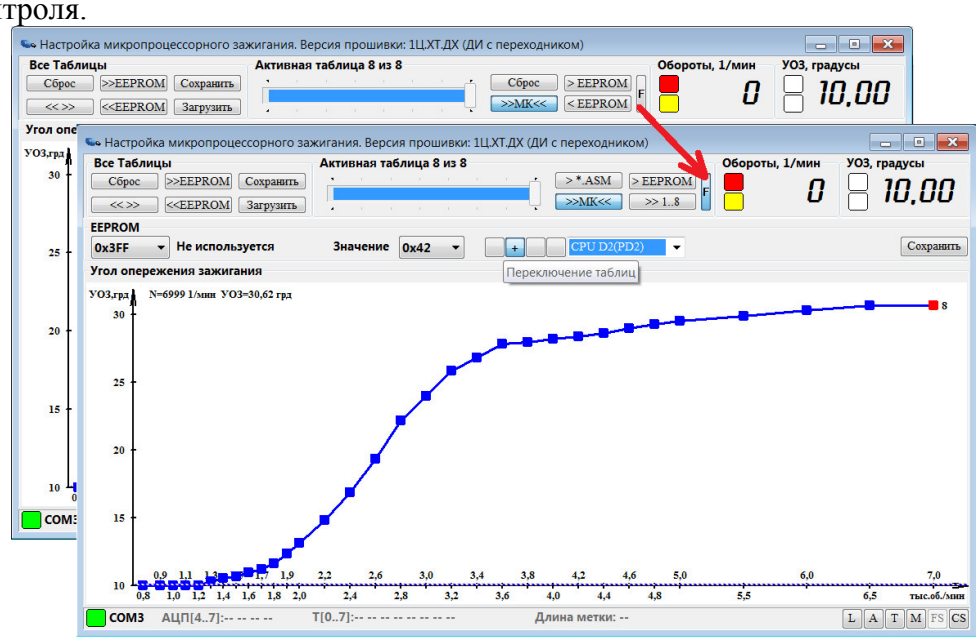
DIP переключатели (перемычки). Анализ схем коллег дал возможность постулировать общий подход к построению вариантов управления с помощью DIP перемычек на свободных входах микроконтроллера. Так DIP переключатели при желании следуют подключить следующим образом:



Назначение DIP переключателей следующее:

PD7(ain1)	Если уровень на входе равен 0, то ручной корректор отключен
PB0(icp)	Если уровень на входе равен 0, то переключатель таблиц отключен.
PB1(osc1)	Если уровень на входе равен 0, то корректор нагрузки отключен.
PB2(ss)	Резерв

EEPROM аналог DIP переключателей. Внесены изменения и в программу OUZ. Она теперь различает Ядра с поддержкой Flash команд и при включении дополнительного функционала (кнопка F), дает пользователю доступ к содержанию всего адресного пространства памяти EEPROM. Пользователю остается выбрать ячейку энергонезависимой памяти с неиспользуемым Ядром адресом и пользоваться её битами для управления параметрами Ядра (**EEPROM аналог DIP переключателей**). Четыре младших бита предназначены для управления светодиода на плате и для удобства объедены в один орган контроля.



Назначение бит ячейки памяти EEPROM следующее:

7	Если бит равен 1 (+ в программе), то ручной корректор включен
6	Если бит равен 1 (+ в программе), то переключатель таблиц включен.
5	Если бит равен 1 (+ в программе), то корректор нагрузки включен.
4	Если бит равен 1 (+ в программе), то корректор нагрузки будет продолжать работу на оборотах, выше табличных.
3-0	Выбор режима работы установленного на плате Arduino светодиода.

3. Для использования DIP переключателей или их EEPROM аналогов создан макрос SetOptions Mode,Value. Параметр Mode определяет режим работы макроса:

0 или LED - Только установка режима работы светодиода. Параметр Value в этом случае режим работы установленного на плате Arduino светодиода;

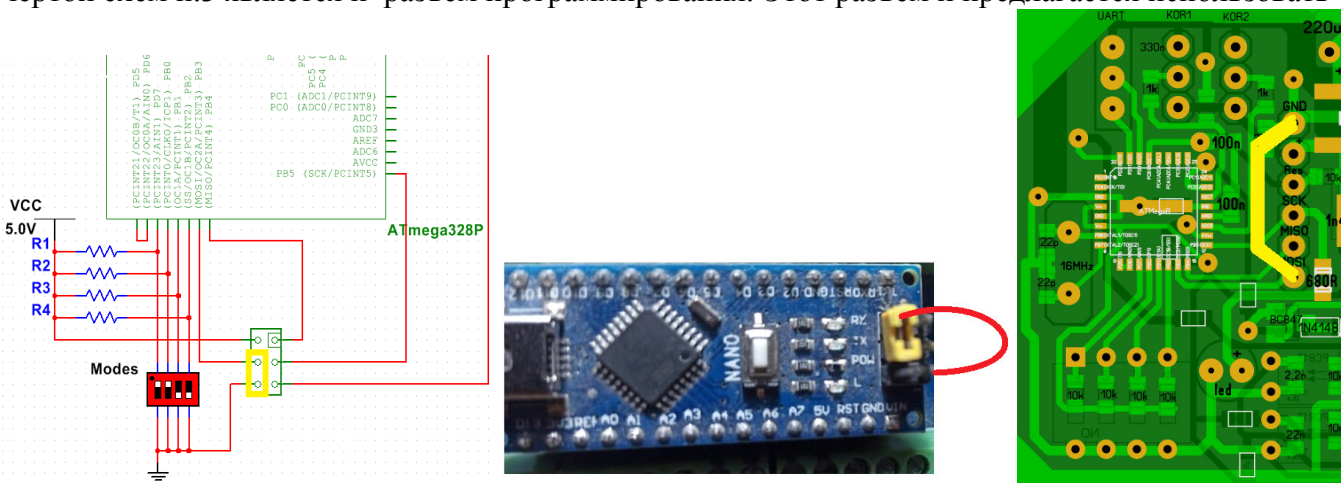
1 или DIP – Применение DIP переключателей. В этом случае параметр Value также определяет режим работы установленного на плате Arduino светодиода;

2 или EEPROM – Применение EEPROM аналогов DIP переключателей. В этом случае параметр Value задает адрес ячейки памяти EEPROM. Режим работы установленного на плате Arduino светодиода содержится уже в упомянутой ячейке.

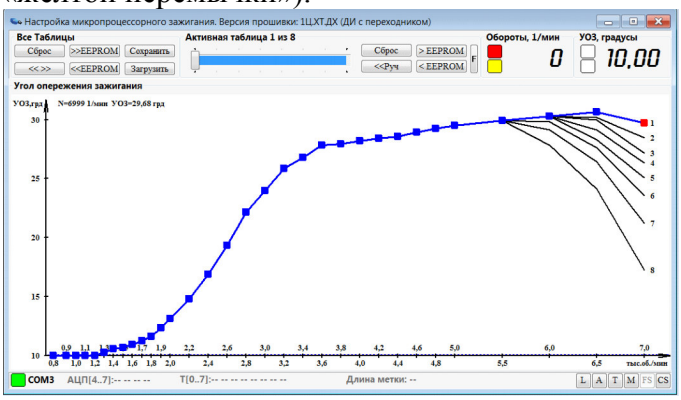
Пример (одна из нижеприведенных строк):

SetOptions LED, 2
SetOptions DIP,2
SetOptions EEPROM,0x3FF

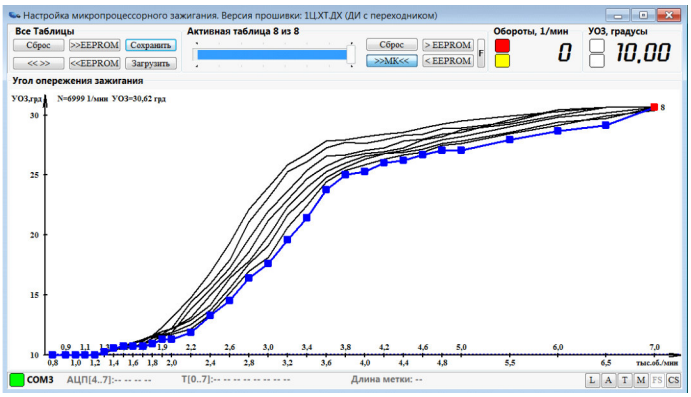
4. В Ядро добавлена функция «**желтая перемычка**» - при наличии этой перемычки набор таблиц будет считан из области программы, а не из области EEPROM как обычно. Данную функцию можно использовать или как выбор альтернативного набора таблиц или выбор источника таблиц при аварии внутренней EERPOM (аварийный режим восстановления работы без компьютера). Видно, что общей чертой схем п.3 является и разъем программирования. Этот разъем и предлагается использовать



При включении «желтой перемычки» Ядро сверит контрольную сумму сохраненного набора таблиц в области программ. В случае ошибки таблицы будут загружены как обычно из EEPROM (как без «желтой перемычки»).



Загрузка из EEPROM



Загрузка из области программы («желтая» перемычка)

Для обеспечения возможности такого режима расширил действие программы UOZ при нажатии на кнопку “ > *.ASM ”. После отладки набора таблиц следует нажать на эту кнопку и произвести повторное программирование МК измененной прошивкой. В конец листинга исходного файла теперь сохраняется и весь набор таблиц (с контрольной суммой) для восстановления без компьютера.

5. Применена преобразованная функция округления при задании констант:
$round\left(\frac{A}{B}\right) = int\left(\frac{A}{B} + 0{,}5\right) = int\left(2\frac{A}{B} + 1\right) \gg 1$, где “>>1“ – логический сдвиг в право (деление на 2). В противном случае иногда имеет место потери в младшем разряде из-за «глюков» avrmasm2.

6. Программное деление столбиком на константу условную К заменено на многоразрядное умножение с помощью встроенного умножителя mul с учетом сдвига результата:

$$int\left(\frac{A}{\text{К}}\right) = \text{int}\left(\frac{A}{2^P} * \frac{2^P}{K}\right) = int\left(\frac{A * C}{2^P}\right) = int(A * C) \gg p$$

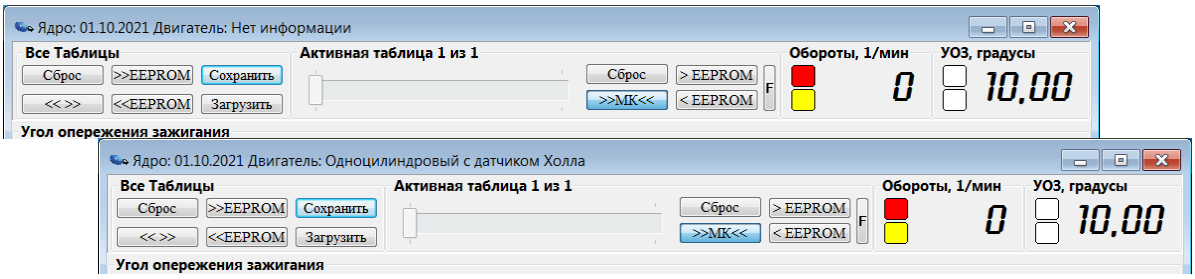
Сдвиг вправо на p разрядов осуществляется автоматически при суммировании после умножения. Константа $C = int\left(\frac{2^P}{K}\right)$ при необходимости предварительно размещается в ОЗУ. Степень p выбирается исходя из требуемой точности.

7. Применен механизм условной компиляции по средствам директив avrasm2 (.if, #ifdef, #define ....). **Цель - сокращение объемов ОЗУ, ускорение работы прошивки при не использовании отдельных узлов за счет их автоматического изъятия из кода при компиляции**. Минимизировано количество блоков, которых требуется включать или отключать по флагам. Это введение позволило **кардинально изменить подход к заданию параметров прошивки**.

8. Длина строки версии прошивки увеличена до 127 символов. Строка "изъята" из ОЗУ. Освобожденное ОЗУ используется для хранения чисел механизма п.6. Сама строка версии разделена на две части – Дата Ядра и Модель или описание двигателя. Первая часть сохранена в Ядре. Вторая часть **может быть добавлена** с помощью механизма п.7. пользователем в моторной части строкой:

#define _EngineInfo "Одноцилиндровый с датчиком Холла ".

Ключевые слова - «**может быть добавлена**». Иными словами, такой строки может не быть совсем. На рисунке приведено отличие в отображении программой UOZ в этих случаях.



9. Откорректирована (оптимизирована) функция поиска номера диапазона Selectr - номер ячейки в таблице.

10. Исправлен математический аппарат по проведению линейной интерполяции и вычисления кода для таймера «задержки» между узлами. Математический аппарат приведен в соответствие формулам в описании предыдущей версии:

$$\text{OCR1A} = \frac{N \cdot \left[fmul\left\{B_k^{fp8} \cdot A_{k+1}^{fp8}\right\} + fmul\left\{128 - B_k^{fp8},\ A_k^{fp8}\right\} + C^{fp16}\right]}{\text{Ka} \cdot 32768} - \Delta\text{T}$$

$$A_k^{fp8} = int\left[\frac{\Delta - \alpha(M, \ldots)}{Base - \Delta} \cdot Ka \cdot 128\right],\ B_k^{fp8} = int\left[\frac{N - n_k}{n_{k+1} - n_k} \cdot 128\right],\qquad k = Selectr,$$

$$C^{fp16} = -round\left[\frac{\text{Апертура} \cdot Ka \cdot 32768}{Base - \Delta}\right] + 2 \cdot round\left[\frac{\text{Апертура} \cdot Ka \cdot 32768}{Base - \Delta} \cdot \frac{ADC_j}{256}\right] \qquad (1)$$

$$\Delta T = round\left[(ProtectCount \cdot\ ProtectClk + 6)\frac{8}{1 + Fn}\right]$$

$$TblSelectr = int\left[ADC_i \cdot \frac{TblCount}{256}\right]$$

i – номер канала АЦП для выбор таблицы, j – номер канала АЦП ручного корректора УОЗ
Ранее в прошивке были технической ошибке использованы эквивалентные на 100% формы.

11. С помощью механизма п.6 применена альтернативная функция деления произведения $\frac{N\cdot\left[fmul\left\{B_k^{fp8}\cdot A_{k+1}^{fp8}\right\}+fmul\left\{128-B_k^{fp8},\ A_k^{fp8}\right\}+C^{fp16}\right]}{32768}$ из (1) на константу Ka через многоразрядное умножение - $C = int\left(\frac{2^{16}}{Ka}\right)$. Для учета частных случаев, когда Ka есть 2 в степени создан макрос DivKa.

12. Деление в выражении $B_k^{fp8} = int\left[\frac{N-n_k}{n_{k+1}-n_k}\cdot 128\right]$ (1) также заменено на многоразрядное умножение на заранее рассчитанные константы $C_k = int\left(\frac{2^{24}}{n_{k+1}-n_k}\right)$

$$B_k^{fp8} = (N-n_k)\cdot C_k \gg (24-7)$$

Вычисление трехбайтовых констант C_k происходит в Главной разметочной таблице. Макрос FTW в Ядре откорректирован для заполнения ОЗУ этими константами FTW $n_k,\ C_k$ -

FTW 0x1BDC, 0x007760

Все вышеперечисленное позволило значительно снизить время вычисления. Следует отметить что, из-за особенности функции поиска номера диапазона Selectr имеет место быть полезная особенность - чем выше обороты, тем меньше время выполнения математических расчетов. В таблице сведены результаты сравнения скорости работы предыдущей и новой версии прошивки.

Версия прошивки	Затраты на вычисления, мкс/град		
	Минимальные обороты по таблице	Максимальные обороты по таблице	Обороты выше табличных
Предыдущая	76/0,4	56/2,4	30/1,3
Новая	41/0,2	21/0,9	16/0,7

В знаменателе указаны потери времени в углах поворота маховика. Обращаю внимание на значении 2,4 градуса. Это именно те потерянные ~3 градуса, про которые сказано в Главной разметочной таблице. предыдущей версии прошивки. **Теперь это менее 1 градуса без потери точности вычисления!** Безусловно, объем используемых ресурсов ОЗУ возрастает. Так при максимальных параметрах системы ОЗУ используется на 97%.

13. Все параметры Ядра теперь имеют значения по «умолчанию». Если пользователь в моторной части «забыл» задать параметр с помощью директивы **#define,** то будут действовать следующие значения по умолчанию:

Параметр	Назначение	Значение по умолчанию
_EngineInfo	"Нет информации"	-
_Base	Угол-база в градусах	360
_Delta	Угловой размер метки Delta	30
_Betta	Угловое расстояние от ВМТ до метки Delta.	0
_UOZMax	Максимальный угол опережения зажигания	_Betta + _Delta-1
_UOZMin	Минимальный угол опережения зажигания	_Betta
_Alfa	Угловое расстояние отключения сигнала зажигания после метки Delta	0.5
_3734Enable	Режим работы с коммутатором ХХ.37.34, при использовании которого скважность сигнала управления равна 3	отключен
_Fn	Частота тактовых импульсов, поступающих на таймер Т1	8
_Load	Угловая апертура корректора нагрузки в "-", град.	0
_LoadStartSelectr	Номер начального диапазона работы корректора нагрузки	31
_LoadBufferSize	Длина кольцевого буфера корректора нагрузки	64
_ProtectClk	Интервал дискретизации протокола защиты от пульсаций в мкс	6

Параметр	Назначение	Значение по умолчанию
_ProtectCount	Количество повторных чтений протоколом защиты с интервалом ProtectClk после первого изменения линии датчика	2
_ADCBufferSize	Длина буферов АЦП	8
_StopTime	Временной интервал для расчета времени установления режима СТОП, сек.	1
_Manual	Константа для ручного корректора, град.	0
_ADCCanCorSelectr	Номер канала АЦП Ручного корректора УОЗ	0
_DoInvertADCCorSelectr	Инверсия кода АЦП канала _ADCCanCorSelectr код = (255-код)	отключена
_TblCount	Количество таблиц УОЗ	1
_ADCCanTblSelectr	Номер канала АЦП выбора активной таблицы УОЗ	0
_DefaultTbl	Номер активной таблицы в случае отключенного канала АЦП _ADCCanTblSelectr и количества таблиц более 1	_TblCount-1
_DoInvertADCTblSelectr	Инверсия кода АЦП канала _ADCCanTblSelectr код = (255-код)	отключена
_ADCCanA	Номер канала АЦП свободного канала	0
_ADCCanB	Номер канала АЦП свободного канала	0
_DoAverage	Включено усреднение по двум отсчетам оценки скорости вращения	отключено
_DH2Disable	Принудительное (без маски) отключение входа датчика DH2 - PD3(int1)	отключено
_DH1Disable	Принудительное (без маски) отключение входа датчика DH1 – PD2(int0)	отключено

Пример полного набора параметров с учетом значений по умолчанию:

```
#define _EngineInfo "1Ц.4Т Индуктивный датчик с удлинением метки….."
#define _Delta              25.0
#define _Betta              6.0
#define _UOZMax             30.0
#define _UOZMin              6
#define _Fn                  18
#define _Load               3.0
#define _LoadStartSelectr 28
#define _LoadBufferSize   128
#define _DoAverage
```

14. Расширена логика управления корректора нагрузки. Введен начальный номер диапазона оборотов (Selectr), выше которых корректор работает. Введены два режима его работы: только до верхней границы таблиц УОЗ, и для всех оборотов. Режим выбирается значением бита №3 порта управления CoreSetup (п.2.)

15. Вспомогательные функции для событий ServiceA, ServiceB, Loop переведены в режим подключаемых с помощью механизма п.7. Если в моторной части будет строка

```
#define _ServiceA  ProgName
```

и где-то в листинге будет описана подпрограмма

```
ProgName:
        do something
        do something
        ret
```

то эта подпрограмма будет вызываться в соответствующий момент времени функционирования Ядра. То же справедливо и для событий ServiceB и Loop. Напомню, что ни какие регистры (кроме temp,

ZL, ZH) изменять в подпрограмме нельзя без предварительного сохранения в стеке и последующего их восстановления в стеке. Подпрограмма RESET, по прежнему обязательная должна быть в моторной части.

16. Весь механизм сравнения регистра событий Events с заранее известными кодами перенесен из моторной части в Ядро. В исходном коде Ядра определены 4 **потенциальных вызова** подпрограмм EventX (X=0..3). Для активации нужного события следует указать код события. Если в тексте моторной части будет подпрограмма

```
Event0: do something
        ret
```

то для того чтобы Ядро могло вызвать эту подпрограмму следует указать

```
#define  _Event0  0b00001010
```

Тем самым укажем Ядру, что при появления события с кодом 0b00001010 следует вызвать подпрограмму. Аналогично и с Event1, Event2, Event3.
Следующим новшество – введение для каждого события **отдельной маски** _EventMaskX(X=0..3). Этот механизм позволяет расширить логику входных сигналов за счет сокращения контролируемых состояний во времени. В случае , когда _EventMaskX не определена, то сравнение происходит по четырем моментам времени. Установив маску можно отключить отдельные моменты времени или вообще весь вход контроллера:

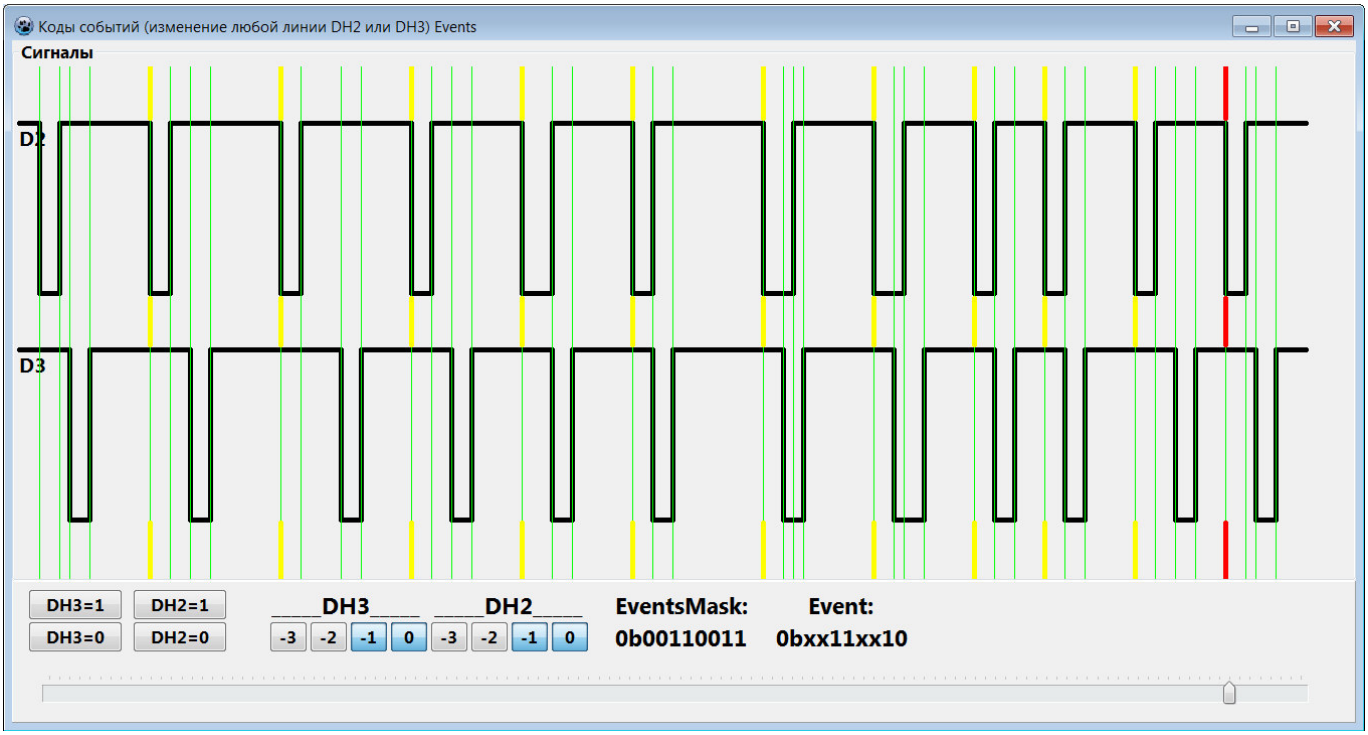
```
#define _EventMask0  0b00110011
#define _Event0      0b00110010
#define _EventMask1  0b00111000
#define _Event1      0b00101000
.
Event0:
  StartDelta
  ret

Event1:
  StopDelta
  ret
```

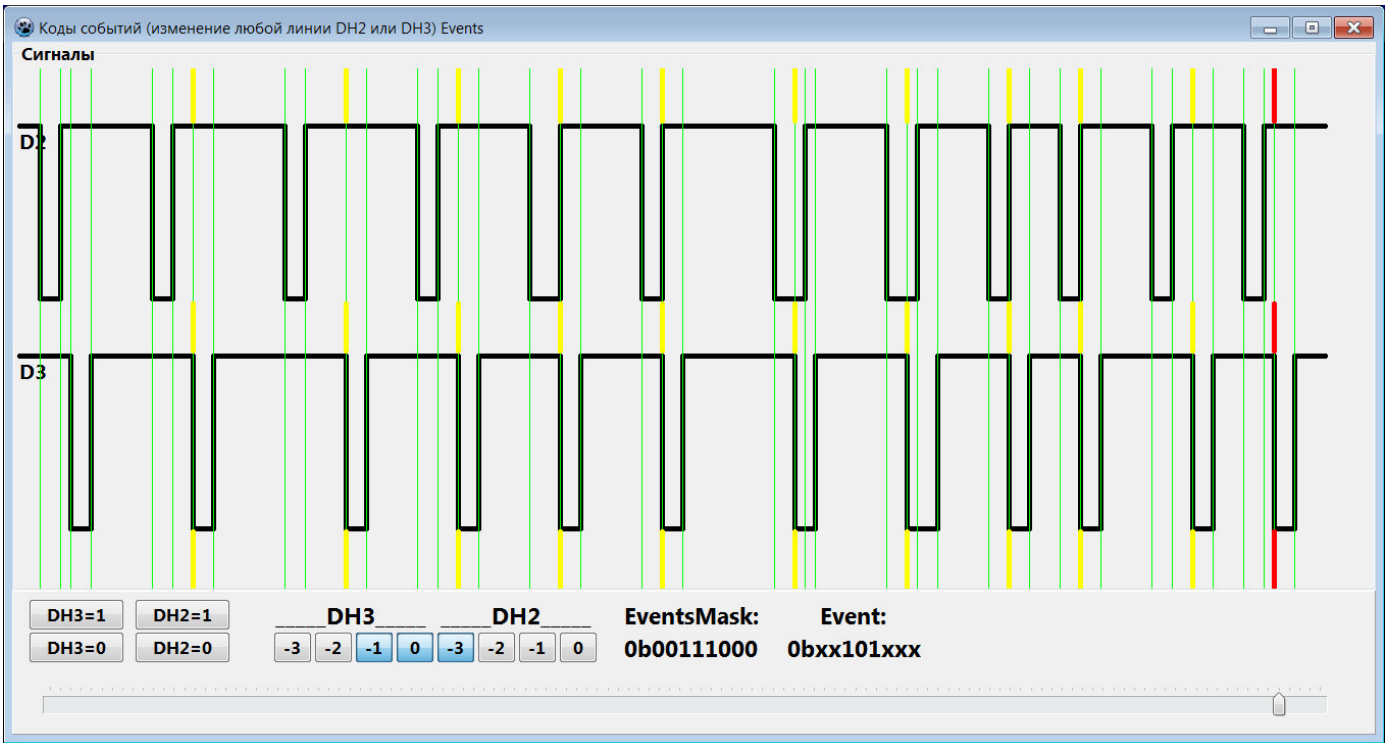
В приведенном примере событие **Event0** произойдет если при неизменном входном сигнале на входе Int1/PD3/3 и равным 1, на другом входе Int0/PD2/2 **в это время** произойдет переход из 1 в 0. В свою очередь, событие **Event1** произойдет если «три изменения состояний входных линий» назад на входе Int0/PD2/2 есть спад логического уровня на входе Int1/PD3/3. Понятно, что такое временное кодирование весьма трудноосуществимо. Для создания таких кодов была разработана программа EventMaker. В программе следует нарисовать диаграммы входных сигналов, «поиграться» с масками и кодами для обнаружения уникальных комбинаций.

Например, рассмотрим применения штатного индуктивного датчика, положительные и отрицательные импульсы которого включают диодную часть двух оптопар с триггером Шмидта на выходе. Справа представлена «собранная» из двух выборок осциллограмма этого случая. Данные любезно предоставил коллега fcftdbx. Из-за короткого расстояния между этими импульсами микроконтроллер может не успеть обработать входные линии с помощью протокола защиты от импульсной помехи на высоких оборотах без маскирования. На оборотах 10000 1/мин расстояние между фронтом первого импульса и спадом второго всего ~40 мкс.

Ниже приведены скриншоты программы EventMaker для вышеприведенного примера.



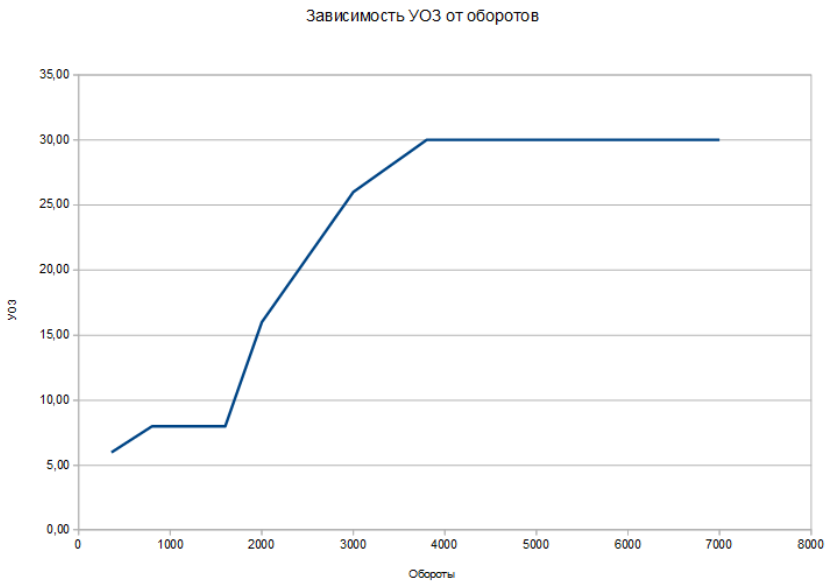
На диаграммах специально создается ситуация наложения импульсов. Красный маркер указывает на рассматриваемое событие. Желтые маркеры, указывают на моменты времени, где есть такие же события. Первые две пары изменений слева не учитываются. Зелеными вертикальными линиями обозначены моменты времени, когда происходит любое изменение на любом из входов.

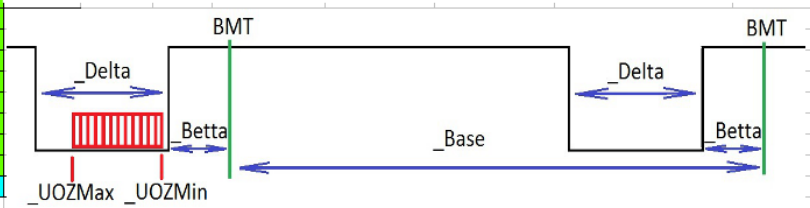


Вместе с программой выложен и файл «event128.dat», в котором программа EventMaker хранит результаты своей работы. Можно повторить мои изыскания и возможно найти еще комбинации кодов и масок. Отмечу, что рассматриваемый случай - чуть ли на самый сложный. В простейшем случае одного датчика Холла все будет гораздо проще.

17. Файл «Главная разметочная таблица» тоже не мог не поменяться. Сократил многие поля, добавил информационные.

Обороты, 1/мин	Период, сек	Интервал Base-Delta, сек	Selectr	Код таймера Index DEC	Код таймера Index HEX	В текст программы FillIndex	Базовая таблица относительно ВМТ, град	В текст программы FillTable
Менее 358	Зажигание по завершению метки Delta = 6 град.		1F					
358	0,1676	0,1560		65535	FFFF	FTW0xFFFF, 0x000000;358 1/мин	6,00	FTB0xF8;6 град.
800	0,0750	0,0698	1E	29386	72CA	FTW0x72CA, 0x0001D0;800 1/мин	8,00	FTB0xE4;8 град.
1600	0,0375	0,0349	1D	14693	3965	FTW0x3965, 0x000476;1600 1/мин	8,00	FTB0xE4;8 град.
1700	0,0353	0,0328	1C	13829	3605	FTW0x3605, 0x004BDA;1700 1/мин	10,00	FTB0xD1;10 град.
1800	0,0333	0,0310	1B	13060	3304	FTW0x3304, 0x005539;1800 1/мин	12,00	FTB0xBD;12 град.
1900	0,0316	0,0294	1A	12373	3055	FTW0x3055, 0x005F65;1900 1/мин	14,00	FTB0xA9;14 град.
2000	0,0300	0,0279	19	11754	2DEA	FTW0x2DEA, 0x0069E0;2000 1/мин	16,00	FTB0x95;16 град.
2100	0,0286	0,0266	18	11195	2BBB	FTW0x2BBB, 0x00753D;2100 1/мин	17,00	FTB0x8B;17 град.
2200	0,0273	0,0254	17	10686	29BE	FTW0x29BE, 0x0080C1;2200 1/мин	18,00	FTB0x81;18 град.
2300	0,0261	0,0243	16	10221	27ED	FTW0x27ED, 0x008CF0;2300 1/мин	19,00	FTB0x77;19 град.
2400	0,0250	0,0233	15	9795	2643	FTW0x2643, 0x0099D7;2400 1/мин	20,00	FTB0x6D;20 град.
2500	0,0240	0,0223	14	9404	24BC	FTW0x24BC, 0x00A79C;2500 1/мин	21,00	FTB0x63;21 град.
2600	0,0231	0,0215	13	9042	2352	FTW0x2352, 0x00B50A;2600 1/мин	22,00	FTB0x59;22 град.
2700	0,0222	0,0207	12	8707	2203	FTW0x2203, 0x00C3A1;2700 1/мин	23,00	FTB0x4F;23 град.
2800	0,0214	0,0199	11	8396	20CC	FTW0x20CC, 0x00D2BA;2800 1/мин	24,00	FTB0x46;24 град.
2900	0,0207	0,0193	10	8106	1FAA	FTW0x1FAA, 0x00E1FC;2900 1/мин	25,00	FTB0x3C;25 град.
3000	0,0200	0,0186	0F	7836	1E9C	FTW0x1E9C, 0x00F2BA;3000 1/мин	26,00	FTB0x32;26 град.
3200	0,0188	0,0174	0E	7346	1CB2	FTW0x1CB2, 0x0085BF;3200 1/мин	27,00	FTB0x28;27 град.
3400	0,0176	0,0164	0D	6914	1B02	FTW0x1B02, 0x0097B4;3400 1/мин	28,00	FTB0x1E;28 град.
3600	0,0167	0,0155	0C	6530	1982	FTW0x1982, 0x00AAAB;3600 1/мин	29,00	FTB0x14;29 град.
3800	0,0158	0,0147	0B	6187	182B	FTW0x182B, 0x00BF11;3800 1/мин	30,00	FTB0x0A;30 град.
4000	0,0150	0,0140	0A	5877	16F5	FTW0x16F5, 0x00D368;4000 1/мин	30,00	FTB0x0A;30 град.
4200	0,0143	0,0133	09	5597	15DD	FTW0x15DD, 0x00EA0F;4200 1/мин	30,00	FTB0x0A;30 град.
4400	0,0136	0,0127	08	5343	14DF	FTW0x14DF, 0x010204;4400 1/мин	30,00	FTB0x0A;30 град.
4600	0,0130	0,0121	07	5111	13F7	FTW0x13F7, 0x011A7C;4600 1/мин	30,00	FTB0x0A;30 град.
4800	0,0125	0,0116	06	4898	1322	FTW0x1322, 0x0133AE;4800 1/мин	30,00	FTB0x0A;30 град.
5000	0,0120	0,0112	05	4702	125E	FTW0x125E, 0x014E5E;5000 1/мин	30,00	FTB0x0A;30 град.
5200	0,0115	0,0107	04	4521	11A9	FTW0x11A9, 0x016A14;5200 1/мин	30,00	FTB0x0A;30 град.
5700	0,0105	0,0098	03	4124	101C	FTW0x101C, 0x00A514;5700 1/мин	30,00	FTB0x0A;30 град.
6300	0,0095	0,0089	02	3732	0E94	FTW0x0E94, 0x00A72F;6300 1/мин	30,00	FTB0x0A;30 град.
6500	0,0092	0,0086	01	3617	0E21	FTW0x0E21, 0x0239E1;6500 1/мин	30,00	FTB0x0A;30 град.
7000	0,0086	0,0080	00	3358	0D1E	FTW0x0D1E, 0x00FD09;7000 1/мин	30,00	FTB0x0A;30 град.
Более 7000	УОЗ не изменяется = 30 град.		FF	-	-	-	-	-





Константа Ядра	Значение	Краткое описание	Примечание
_Betta	6	Угловое смещение метки Delta относительно ВМТ	Ok
_Delta	25	Угловой размер метки в градусах	Ok
_Base	360	Угол-база в градусах	
_Fn	18	Код частоты тактовых импульсов системы F=0,4211 Мгц	
_UOZMax	30	Ограничение на УОЗ 'сверху' в градусах относительно ВМТ. Не более 30 грд.!	Ok
_UOZMin	6	Ограничение на УОЗ 'снизу' в градусах относительно ВМТ. Не менее 6 грд.!	Ok