

Описание изменений

Поддержка номера COM порта, указанного как параметр при запуске. Ускоряет запуск при больших номерах COM портов. Поддержка – до COM15.

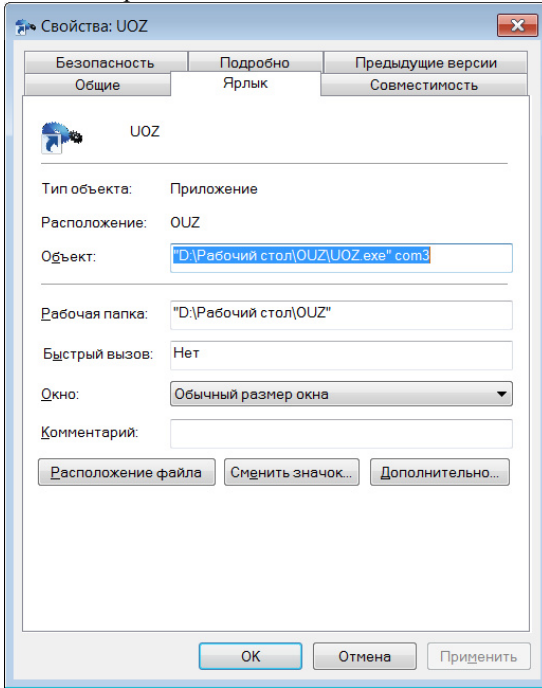


Рис.1

Индикатор соединения с блоком ФУОЗ.

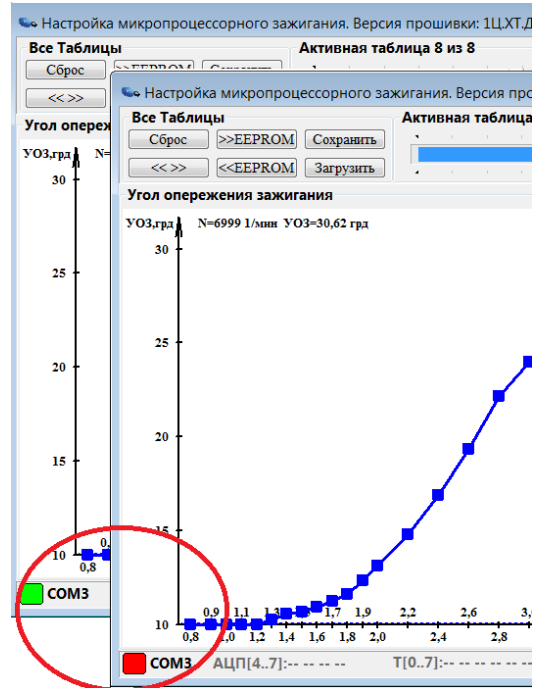


Рис.2

Размер Flash памяти (энергонезависимой памяти EEPROM) микроконтроллера 1024 байта. На каждую страницу отводится 32 байта. В случае, если используется менее максимального количества (32) таблиц, есть свободная половина EEPROM, а то и больше. Даже в случае использования максимального количества страниц можно найти точку в наборе таблиц, выбор которой Ядром будет маловероятным. Для достижения этой цели введены несколько новых макросов в Ядро, а так же создана система команд по управлению Flash памяти – система Flash команд по интерфейсу UART. В Ядро добавлен признак поддержки этих команд.

Система Flash команд приведена в файле «команды.ods» - выделено зеленым цветом. К сожалению, из-за необходимости сохранения обратной совместимости версий и нежелательности увеличения трафика по интерфейсу UART работы с этими командами носят весьма сложный характер. Внесены изменения и в программу OUZ. Она теперь различает Ядра с поддержкой Flash команд и при включении дополнительного функционала, описанного в предыдущем посте (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 вместо DIP переключателей на плате (EEPROM аналог DIP переключателей). Четыре правых бита предназначены для управления светодиода на плате и для удобства объединены в один орган контроля.

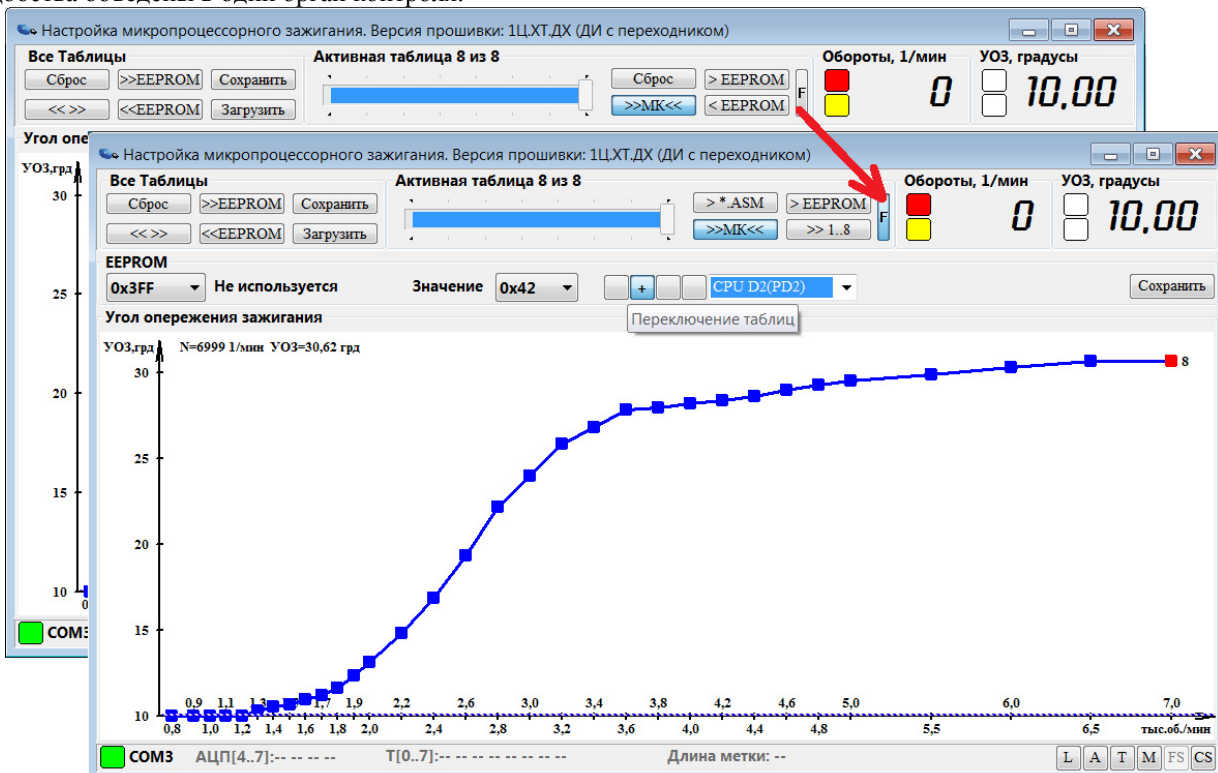


Рис.3

Дополнительно расширил действие программы при нажатии на кнопку “> *.ASM”. В конец листинга исходного файла теперь сохраняется и весь набор таблиц (с контрольной суммой) для восстановления без компьютера (механизм далее) при аварийной ситуации на энергонезависимой памяти микроконтроллера или просто как второй альтернативный набор для режима отработки наборов на устройстве.

FillTable:

```
FTB      0xF0      ;10,01 град.
FTB      0xF0      ;10,01 град.
.
.
.
FTB      0x15      ;30,62 град.
FTB      0x15      ;30,62 град.
ret

.db 0x15, 0x15, 0x19, 0x1D, 0x21, 0x24, 0x27, 0x2B, 0x2D, 0x2F, 0x32, 0x33, 0x3E, 0x48, 0x5C, 0x6F, 0x8D, 0xA7, 0xBD, 0xCF, 0xD7, 0xDF, 0xE3, 0xE6, 0xE9,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x15, 0x17, 0x22, 0x27, 0x28, 0x2D, 0x2E, 0x32, 0x35, 0x34, 0x39, 0x45, 0x4E, 0x65, 0x7A, 0x9C, 0xB1, 0xC1, 0xD8, 0xDD, 0xDF, 0xE4, 0xE5, 0xE8,
  0xEC, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x15, 0x17, 0x20, 0x2B, 0x2D, 0x31, 0x33, 0x39, 0x3B, 0x3F, 0x40, 0x4D, 0x5F, 0x71, 0x89, 0xA3, 0xB6, 0xC7, 0xDC, 0xDF, 0xE1, 0xE6, 0xE9, 0xEA,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x15, 0x1C, 0x24, 0x29, 0x2F, 0x32, 0x39, 0x3C, 0x3E, 0x41, 0x49, 0x54, 0x67, 0x79, 0x95, 0xA9, 0xBB, 0xCF, 0xD9, 0xDB, 0xDF, 0xE2, 0xE5, 0xE8,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x1A, 0x1E, 0x26, 0x2F, 0x32, 0x37, 0x3B, 0x3E, 0x40, 0x46, 0x4E, 0x5C, 0x6D, 0x87, 0x9D, 0xAC, 0xC4, 0xD1, 0xD9, 0xDB, 0xE0, 0xE2, 0xE5, 0xE8,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x1F, 0x22, 0x2B, 0x33, 0x35, 0x3A, 0x3D, 0x3E, 0x43, 0x4A, 0x53, 0x64, 0x74, 0x8F, 0x9F, 0xB5, 0xC8, 0xD5, 0xDC, 0xDE, 0xE0, 0xE2, 0xE5, 0xE8,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x18, 0x1D, 0x25, 0x2C, 0x35, 0x37, 0x3D, 0x3F, 0x43, 0x48, 0x4D, 0x57, 0x6C, 0x7F, 0x9A, 0xA6, 0xB9, 0xCC, 0xD9, 0xDE, 0xDF, 0xE4, 0xE6, 0xE8, 0xE9,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.db 0x15, 0x25, 0x2A, 0x32, 0x3B, 0x3B, 0x3F, 0x44, 0x46, 0x4E, 0x51, 0x5E, 0x77, 0x8A, 0x9F, 0xAC, 0xC0, 0xCD, 0xDC, 0xE2, 0xE2, 0xE6, 0xE8, 0xE8, 0xE8,
  0xEA, 0xED, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0
.dw 0x90FA
```

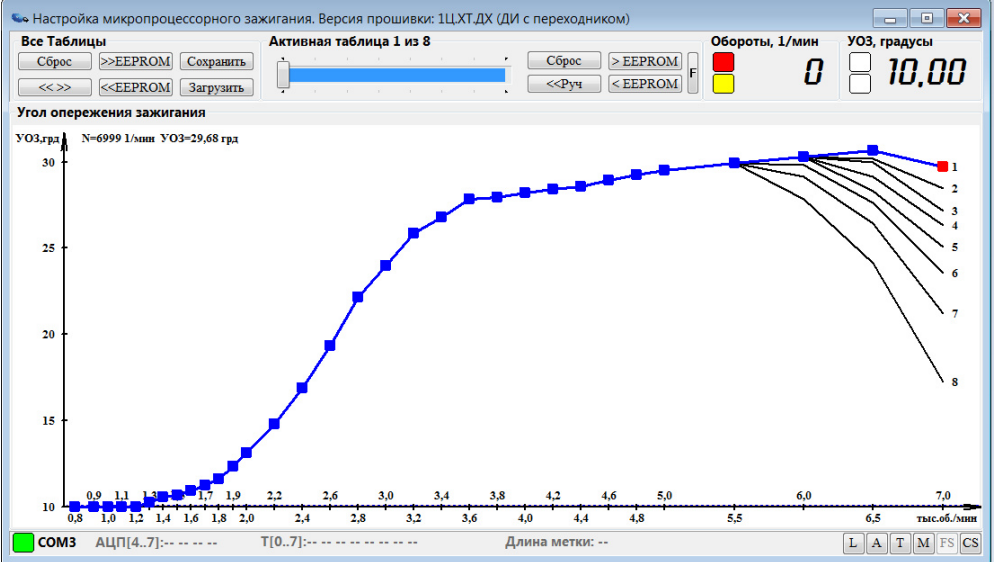


Рис. 4 Из EEPROM

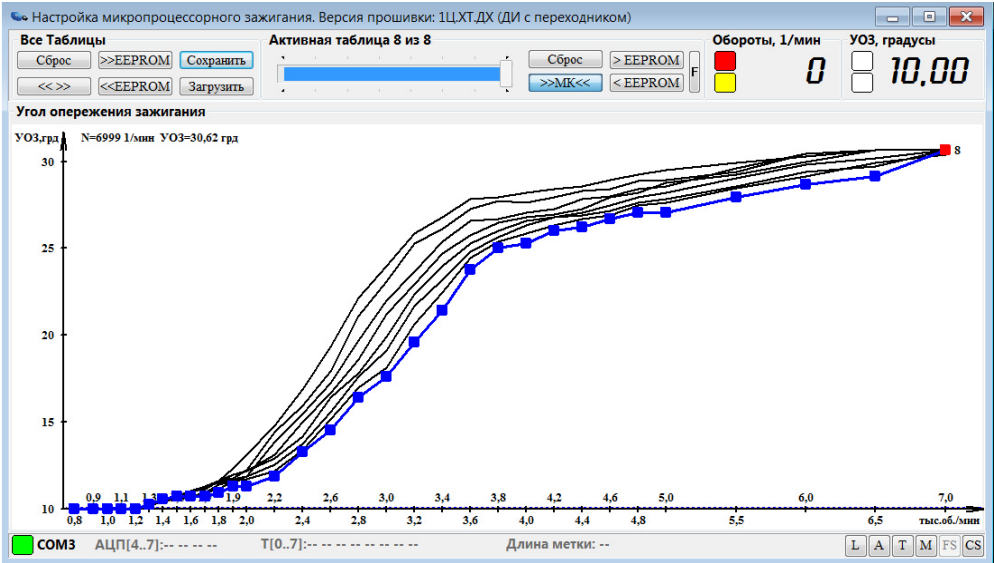


Рис. 5 Из программы

Управление переключателями

Анализ схем моих коллег дал возможность постулировать общий подход к построению вариантов управления. Так DIP переключатели при желании следуют подключить как на Рис.6-Рис.8:

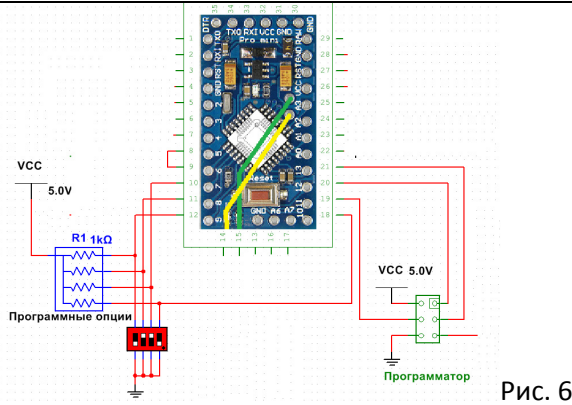


Рис. 6

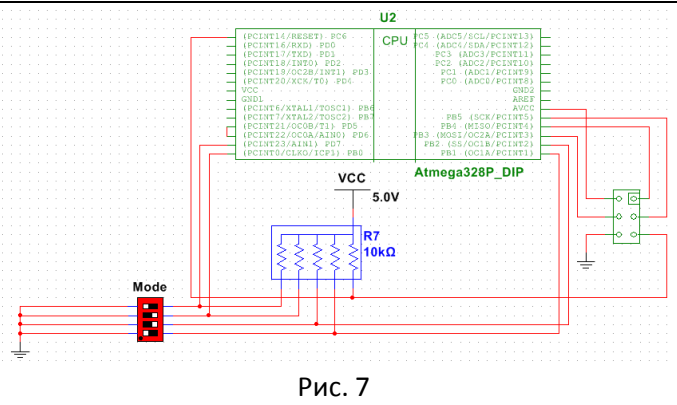


Рис. 7

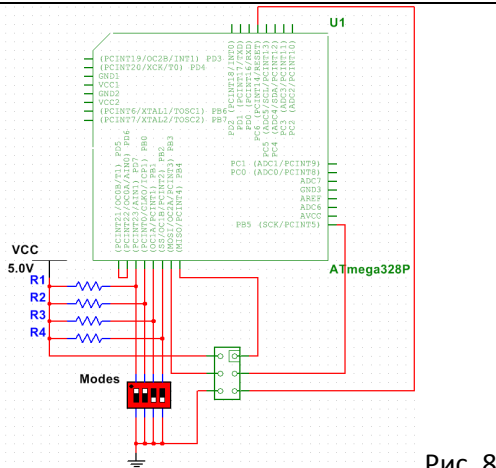


Рис. 8

Видно, что общей чертой схем является и разъем программирования. Этот разъем предлагается использовать еще и как выбор режима DIP переключателей и системы альтернативного набора таблиц. Операция осуществляется с помощью перемычек.

Первая – желтая перемычка (Рис.9-Рис.11). При наличии этой перемычки набор таблиц будет считан из области программы (Рис. 5). Это и есть аварийный режим работы без компьютера. При попытке чтения Ядро сверит контрольную сумму. В случае ошибки таблицы будут загружены как обычно из EEPROM (как без желтой перемычки). Для обеспечения возможности такого режима следует после отладки набора таблиц нажать на кнопку “> .ASM” и произвести повторное программирование МК измененной прошивкой.

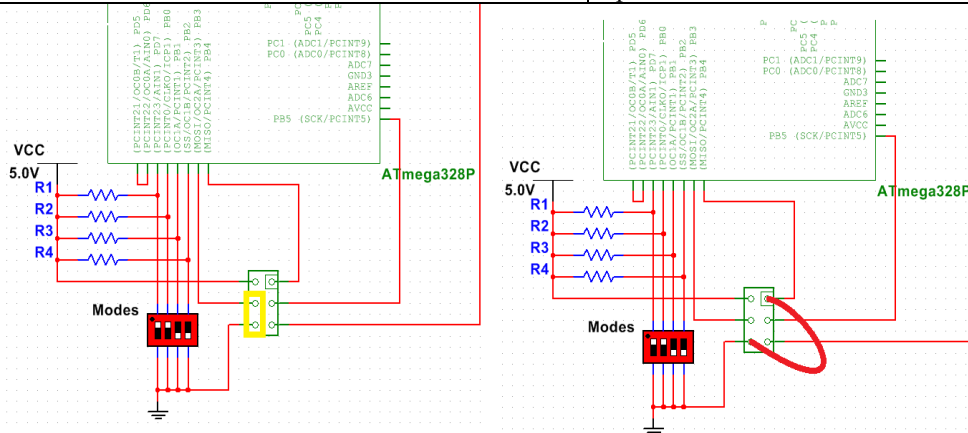


Рис.9

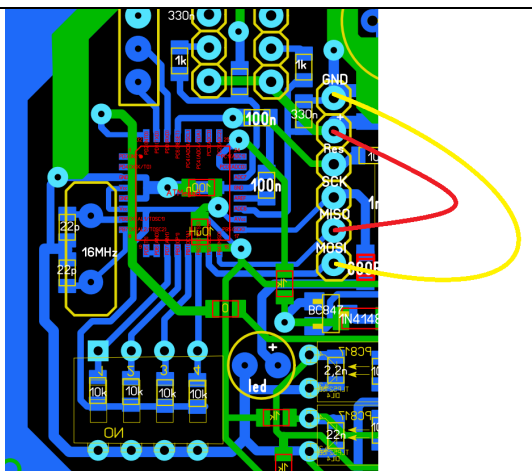


Рис.10

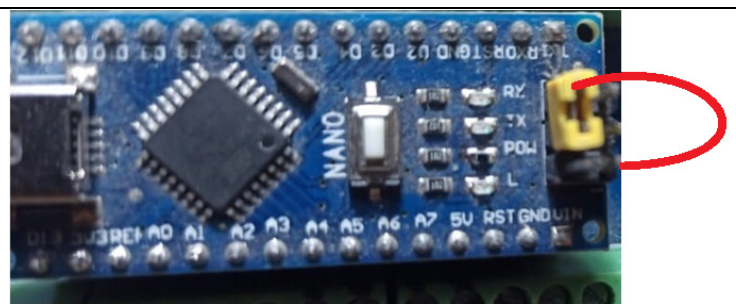


Рис.11

Вторая перемычка – красная. При её наличии программа МК будет использовать DIP переключатели на плате для коррекции режима работы Ядра. Однако следует принять, что при наличии этой перемычки режим работы светодиода, подключенного к линии PB.5, будет определяться только моторной частью программы. Для изменения режима его работы каждый раз придется «перешивать» прошивку.

reset:	sbi	DDRC,CmdLineBit	;Назначение линии PC0(adc0) контролера для работы на выход
	cbi	PORTC,CmdLineBit	;Деактивировать выходную линию
	sbi	DDRD,ResetLineBit	;Назначение линии PD4(t0) контролера для работы на выход
	cbi	PORTD,ResetLineBit	;Включить сигнал сброса входного RS триггера

load	A,CoreSetup	;Начальное значение 0b0000000X0 (Все включено), ;где X=ADCDisableBit устанавливает Ядром при анализе параметров ;ADCCanTb1Selectr, ADCCanCorSelectr, ADCCanA, ADCCanB
	76543210	
ori	A,0b10111000	
store	CoreSetup,A	

- 7 - Отключен датчик DH2
- 6 - Отключено усреднение по двум отсчетам T1
- 5 - Отключен вызов события ServiceA
- 4 - Отключен вызов события ServiceB
- 3 - Отключен корректор для коммутатора XX.37.34
- 2 - Отключен корректор нагрузки
- 1 - Отключен АЦП
- 0 - Отключен UART

Кодирование управления светодиодом на линии PB5 и копирование DIP в регистр A

0 - Выключен	
1 - Delta	
2 - CPU D2(PD2)	;Светодиод отражает состояние основного входного датчика;
3 - CPU D3(PD3)	
4 - CPU A3(PC3)	
5 - CPU A2(PC2)	
6 - CPU A1(PC1)	
7 - CPU A0(PC0)	
8 - Остановка	
9 - Низкие обороты	
10- УОЗ макс.	
11- УОЗ мин.	
12- Флаг Ign1	
13- Флаг Ign2	
14- UARTCtrl	
15- Включен	

В случае использования только DIP переключателей подсекция «Коррекция параметров Ядра» имеет вид:

Красную перемычку в этом случае использовать не надо.

[illegible]

Естественно, что секцию «Коррекция параметров Ядра» можно вообще упростить, оставив лишь функцию светодиода:

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

Назначение EEPROM аналогов DIP переключателей 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 – Резерв

Обращаю внимание, что если соответствующие каналы АЦП для обеспечения функций ручного корректора и переключателя таблиц не включены, то все манипуляции с DIP переключателем, в том числе с их EEPROM аналогами, не будут приводить ни к каким результатам. Тот же результат будет, если корректор нагрузки отключен в секции «Настройка параметров Ядра» при попытке его включить в секции «Коррекция параметров Ядра». **Выключатель может отключить, включенную ранее функцию!**