**ЭУМиРС (часть 2) модуль 4**

**Семинар 4**

**Изучаем микроконтроллер ATmega16**. Усложняем задание семинара3: При нажатии на кнопку "Случайное число" формируется случайное число, которое выводится насемисегментные индикаторы. Для этого кнопку "Случайное число" подключаем к выводу PD2/INT0, семиcегментные индикаторы к выводам PС0-PС7 (PC0-a,PC1-b,PC2-c,PC3-d,PC4-e,PC5-f,PC6-g,PC7-h),сигналы выбора индикатора подключаем к выводам PB2 - сотни,PB1-десятки,PB0 – единицы.

**Цель семинара**: 1. Изучение динамической индикации;

2. Изучение внешних прерываний

Принцип динамической индикации рассмотрен на лекциях.

Программа **Динамической индикации**.

Случайное число научились формировать в предыдущем семинаре. Случайное число – байт, значит максимальное значение 255. Три индикатора, на которые выводятся «сотни», «десятки», «единицы»

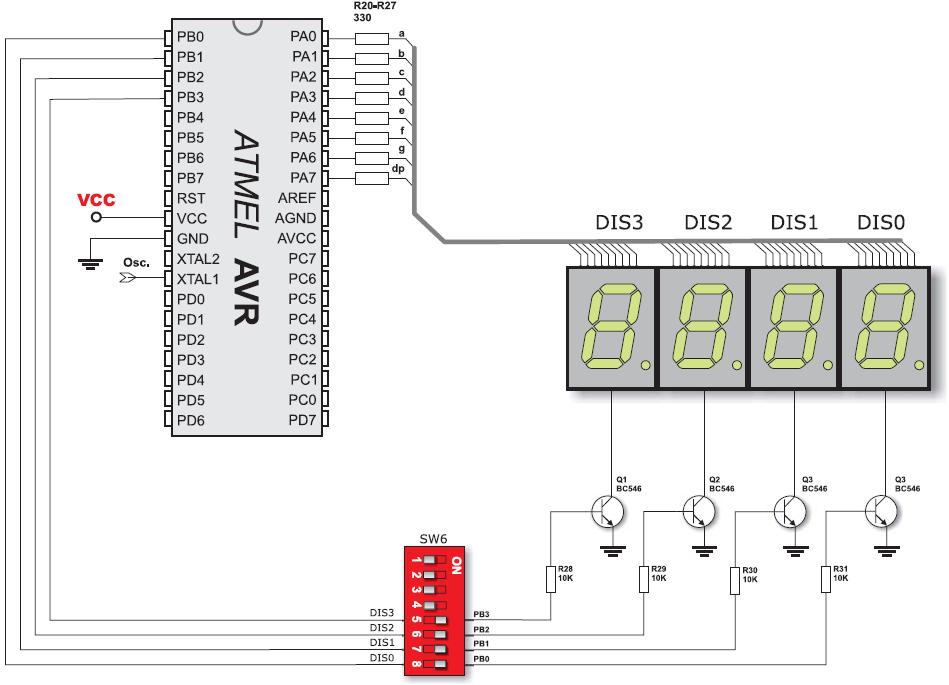


Схема подключения семисегментных индикаторов к портам МК

Поэтому выделены три РОНа

–**Hundreds, Tens, Ones** (R18, R19, R20), для хранения двоично-десятичного кода сформированного случайного числа.

Для создания указателя выбранного индикатора используем РОН

-**Disp\_Numb**.

Для этого используем косвенную адресацию к регистрамHundreds, Tens, Ones (R18, R19, R20), используя начальный адрес Hundreds 0х18. Указатель (адрес РОНа) формируем суммированием начального адреса со значением в Disp\_Numb. Disp\_Numb может принимать значения 0, 1, 2.

Для динамической индикациинеобходимо сформировать временной интервал для переключения индикаторов, для этого используем программный счетчик.

В РОН **Disp\_Count**(R23) загружаем константу Val\_dispCount, и организуем его работу как счетчика на вычитание. При достижении Disp\_Count нулевого значения, фиксируем, что временной интервал включенного индикатора истек, а значит необходимо перейти на индикацию следующего индикатора. Вот и весь незамысловатый алгоритм динамической индикации.

В основном цикле программы организуем динамическую индикацию, как подпрограмму.

Кнопку для формирования случайного числа подключили на вход внешнего прерывания INT0. Если кнопка не нажата, на этом входе «1» уровень. Поэтому, если при нажатии кнопки на этом входе изменяется сигнал (в нашем случае переход в «0» уровень), выставляется флаг прерывания INTF0. А поскольку при инициализации было разрешено прерывание (флаг INT0 в GICR и бит I в SREG), то перейдем на вектор **прерывания INT0addr**, где стоит команда относительного перехода к подпрограмме обработки внешнего прерывания (rjmp IN\_INT0).

В подпрограмме обработки внешнего прерывания INT0 формируется случайное число в двоичном коде. Для вывода на индикаторы двоичный код необходимо преобразовать в двоично-десятичный. Для этого значение случайного числа делится на 100 (определяется количество сотен), затем на 10 (определяется количество десятков), остающееся значение единицы.

Итак, программу необходимо отладить в симуляторе. Компилируем и запускаем программу на выполнение. Создаем окноWatch.

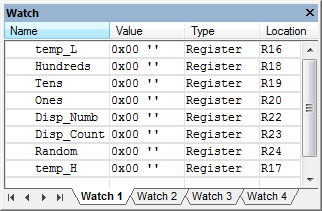


Рисунок 1 Окно Watch

По шагам начинаем отлаживать программу (F11),контролируя результаты отладки. Обращаем внимание,что указатель стека ATmega16 два байта (см.рисунок 2).

Инициализируем порты ввода/вывода. ВокнеI/OViewоткрываем окна портов и смотрим состояние трех регистров DDRn, PORТn, PINn.

Доходим до инициализации PORTD, тут наша кнопка PD2. Симулируем ее работу.

**Внимание**: Сами вручную нажимаем PIND2, черный квадратик – это «1» уровень. Это состояние не нажатой кнопки (см.рисунок 3).

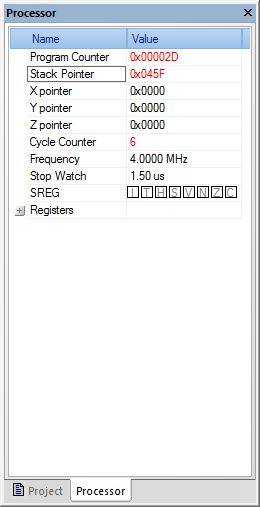


Рисунок 2 Окно Processor

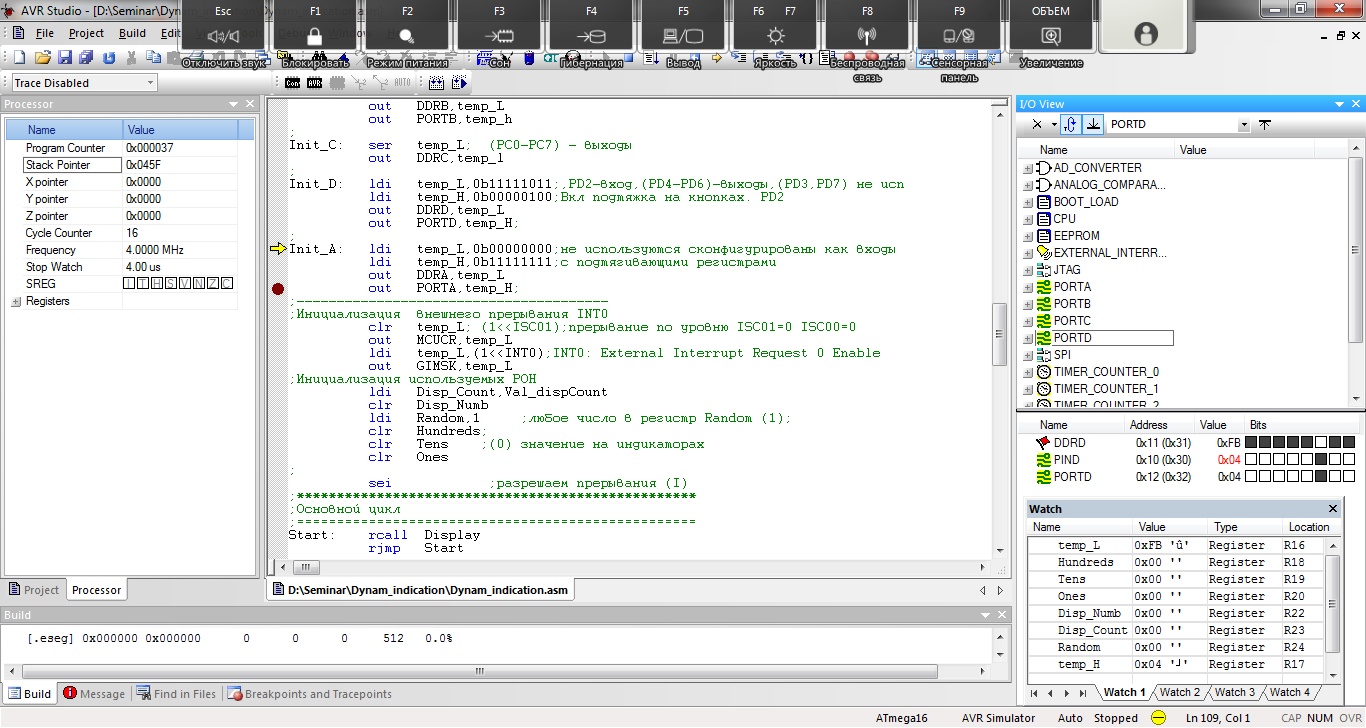


Рисунок 3 Окно инициализации порта D

Подпрограмма динамической индикации *Display*. Нажмите несколько раз кнопку F11, наблюдая за работой программы. Вы можете увидеть, как постепенно уменьшается значение регистра *Disp\_Count*, который отвечает за формирование времени активности семисегментного индикатора.

Поставить точку останова на метку Out\_disp. Нажмите несколько раз кнопку F5. Убедитесь, что программа производит изменение позиции выводимого символа на светодиодных индикаторах.

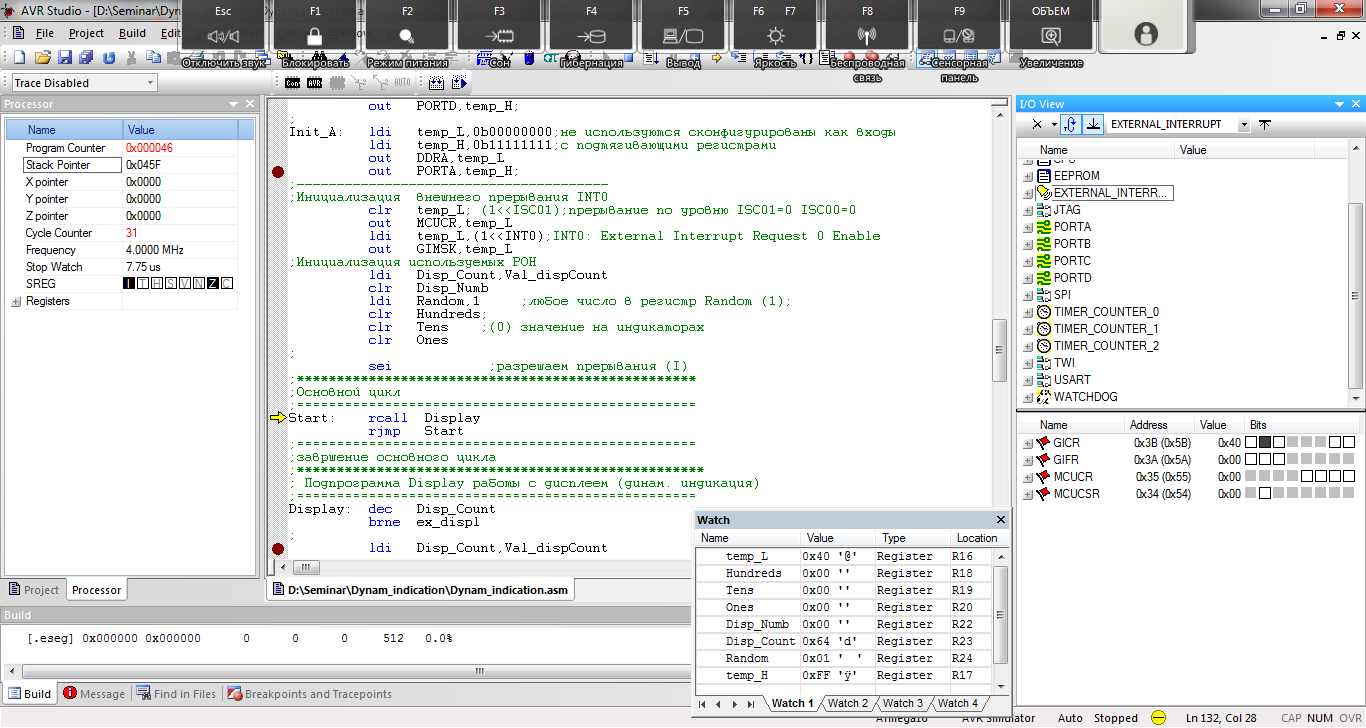


Рисунок 4Окно завершения инициализации внешнего прерывания INT0

Нажимая несколько раз кнопку F11, войдите в основной цикл программы. Симулируйте нажатие кнопки «"Случайное число"» переводом соответствующего бита регистра *PIND* (PD2) в ноль (светлый квадратик).

Нажмите кнопку F11. Перешли на вектор внешнего прерывания INT0 (см.рисунок 5).

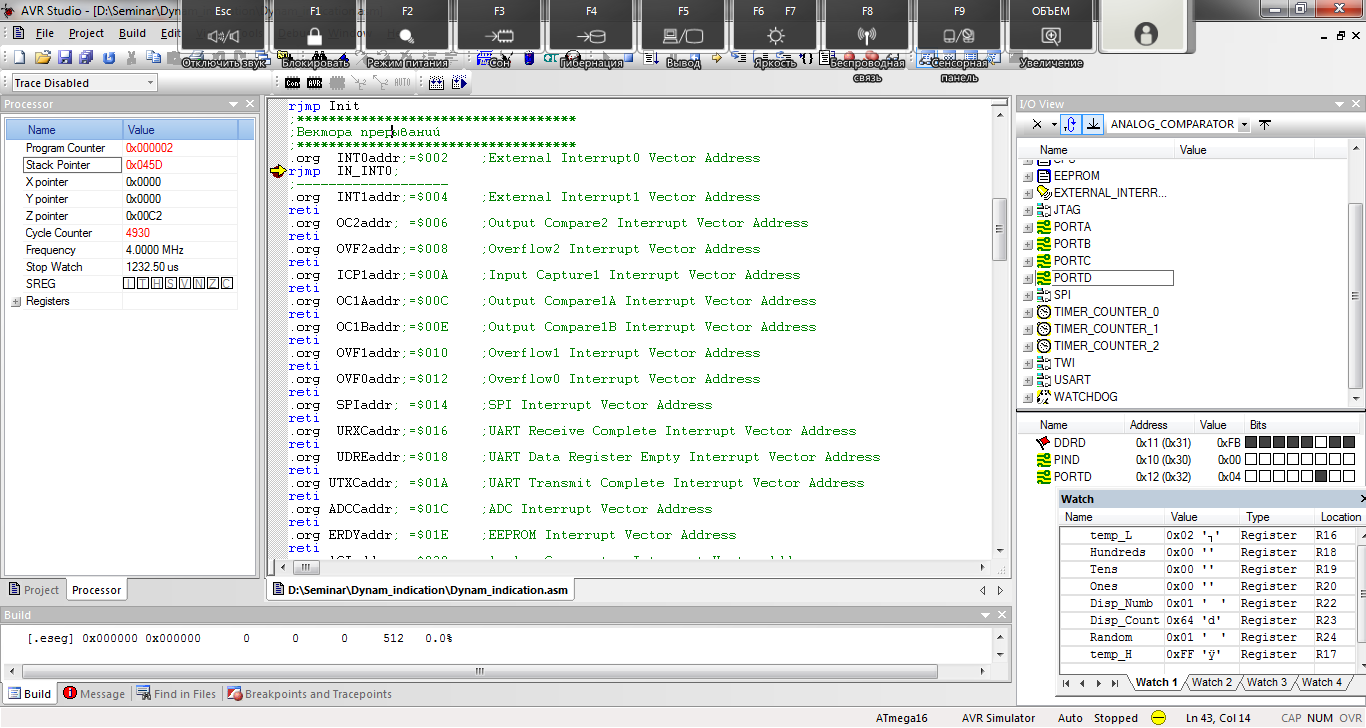


Рисунок 5Окноперехода на вектор внешнего прерывания INT0

Симулируйте отпускание кнопки «"Случайное число"» переводом соответствующего бита регистра *PIND* (PD2) в «1» уровень (черный квадратик).

Нажмите кнопку F11. Перешли в подпрограмму обработки внешнего прерывания INT0. Внимательно через окно Watchпроверьте формирование случайного числа, преобразование в двоично-десятичный код, изменение значений регистров –Hundreds, Tens, Ones.

Несколько раз симулируйте нажатие и отпускание кнопки,проверяя процесс формирования случайного числа и правильность вывода его на индикатор. Используйте точки останова в циклах.