Московский Физико-Технический Институт

Кафедра радиотехники

100 вопросов по микроконтроллерам

Выполнили
Ринат Валиев, 711 гр.
Кирилл Просвирин, 712 гр.
Никита Перепелицын, 713 гр.

Принял
Г.И. Донов – канд. техн. наук

1. Чем отличается микроконтроллер от микропроцессора?

   Микроконтроллер - интегральная схема, принимающая сигналы от различных датчиков, обрабатывающая их и выдающая управляющие сигналы на исполнительные механизмы.

   Микропроцессор - блок, находящийся внутри МК, регулирующий и обеспечивающий работу с информацией согласно системе команд (документации) МК.

2. Какие тактовые частоты могут быть у ATmega8535?

   Рабочий диапазон данного МК: 1 MHz, 2 MHz, 4 MHz, 8 MHz.

3. Какие таймеры есть у ATmega8535?

   - Таймер 0 (8 разрядов)
   - Таймер 1 (16 разрядов)
   - Таймер 2 (8 разрядов)
   - Программируемый сторожей таймер (WatchDog Timer) с генератором (также иногда называется счётчиком реального времени).

4. Внутренная структура МК?

   - Блок управления питанием (обеспечивает запуск МК).
   - Процессор (Обеспечивает обработку информации, выполняя команды в соответствии с системой команд МК, будь то CISK или RISK (Complex or Reduced Instruction Set Computing). Как правило, это RISK).
   - Програмная память (Хранит исполняемый код программы).
   - RAM (Хранит промежуточную информацию).
   - Reset блок (Устанавливает исходное состояние МК).
   - Блок синхронизации (Организует взаимодействие внутренних блоков МК посредством выработки тактовых сигналов).

5. Сколько прерываний есть у ATmega8535?

   21 прерывание: 4 внешних — INT0, INT 1, INT2, RESET и 17 внутренних.

6. Как организована система приоритетов при обработке прерываний?

   Все запросы поступают на блок обработки прерываний (Interrupt Unit). Обнаружив запрос, блок определяет его номер, затем проверяет соответствующий регистр и понимает, разрешено данное прерывание или нет. Если разрешено, то устанавливается блокировка на все остальные прерывания (обнуляется 7-ой разряд SREG (регистра состояний)). Текущее содержимое программного счетчика заносится в стек, а на его место в программный счётчик заносится адрес, приписанный данному прерыванию. Номера прерываний приведены в таблице векторов прерываний. Если одновременно возникло несколько прерываний, то первым обрабатывается имеющее наименьший номер, если другого не прудсмотрено пользователем или документацией.

7. Как можно запретить (разрешить) сразу все прерывания?

   Запись единицы в 7-ой разряд SREG разрешает все прерывания, 0, в свою очередь, запрещает. Для управления этим разрядом существует две команды:
   cli – общее запрещение прерываний (Clear Interrupt)
   sei – общее разрешение прерываний (Set Interrupt)

8. Таймер 0. Режимы работы, количество прерываний, регистры ввода-вывода, принадлежащие таймеру 0?

   Таймер 0 может работать в 4 режимах: 0, 1, 2, 3. Таймер 0 имеет 3 собственных регистра ввода-вывода и 2 регистра, доступ к которым есть у каждого из таймеров. Прерывания TIMER0_OVF и TIMER0_COMP.

9. Как можно разрешить (запретить) прерывания по переполнению таймера 0?

   Запись 0 или 1 в первый разряд OCIEO (Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable) регистра TIMSK (Timer/Counter Interrupt Mask Register) соответственно запрещают и разрешают прерывания.

10. Написать программу с использованием таймера 0, вырабатывающую на PORTA0 симметричное прямоугольное колебание с периодом 2ms.

```
.cseg
.org 0x0
rjmp reset
.org 0x13
rjmp T0_COMP

.org INT_VECTORS_SIZE
reset:
ldi r16, low(ramend)
out spl, r16

ldi r16, high(ramend) ;
out sph, r16 ; Setting the stack.

ser r16 ; r16 = 0b1...1
out DDRA, r16 ; PORTA is output

sei ; enabling all interrupts
ldi r16, 0x2
out TCCR0, r16 ; setting the freq of T0 equals to 1MHz/8
ldi r16, 0x2
out TIMSK, r16 ; enabling interrupts on compare (when TCNT0 == 250),
               ; because 250 * 8 = 2000us = 2ms
ldi r16, 0xFA ; 250
out OCR0, r16
cycle:
out PORTA, r17 ; sending 0 or 1 to PORTA0
rjmp cycle

T0_COMP: cpi r17, 0x0
breq T0_COMP_SET1
ldi r17, 0x0

reti
T0_COMP_SET1:
ldi r17, 0x1
reti
```

11. Какие коэффициенты деления частоты позволяет получать предварительный делитель таймера 0?

    Позволяет получать коэффициенты деления 8, 64, 256 и 1024.

12. Какой режим таймера 0 позволяет вырабатывать треугольные колебания, используя дополнительную интегрирующую цепочку?

    Режим 1 - ШИМ (Широтно-импульсная модуляция) - до достижения максимального значения ($FF) вверх (суммирование), потом вниз (вычитание).

13. Какие значения записаны в TCCR после сигнала RESET?

    Ноль по умолчанию.

14. Сколько прерываний и регистров ввода/вывода принадлежит порту A? Назначение этих регистров ввода/вывода.

    Три регистра:
    -DDRA – регистр выбора направлений выводов порта A. Если соответствующий бит в нём равен 1, то соответствующий вывод в нём будет настроен на выход, иначе – на вход.
    -PORTA – регистр данных порта A, доопределяет состояние выходов.
    -PINA – только для чтения, из которого можно получить то, что на данный момент подано на выводы порта A.

    Прерывания отсутствуют.

15. Почему после сигнала RESET все остальные прерывания запрещены?

    После RESET в 7-ой разряд SREG принудительно устанавливается 0, что и означает, что все остальные прерывания окажутся запрещены.

16. Регистр SREG. Назначение его разрядов.

    SREG (Status Register) - регистр флагов, регистр состояния ядра.

    - Bit 0 - C: Carry Flag - флаг переноса.
    - Bit 1 - Z: Zero Flag - флаг нулевого значения.
    - Bit 2 - N: Negative Flag - флаг отрицательного значения.
    - Bit 3 - V: Two's Complement Overflow Flag - увеличение размера флага переполнения до двух.
    - Bit 4 - S: Sign Flag - флаг знака.
    - Bit 5 - H: Half Carry Flag - Флаг полупереноса.
    - Bit 6 - T: Bit Copy Storage - бит временного хранения.
    - Bit 7 - I: Global Interrupt Enable - глобальное разрешение прерывания.

17. Приведите пример использования разряда T в регистре SREG?

    Пример: перемещение первого бита регистра r1 в третий бит регистра r0:

```
bst r1,1 ; write
bld r0,3 ; read
```

18. Как запрограммировать предварительный делитель таймера 0?

CSO2	CSO1	CSO0	Режим работы
0	0	0	Таймер остановлен, нет счёта
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	Сигнал с входа T0, счёт при переходе от 1 к 0
1	1	1	Сигнал с входа T0, счёт при переходе от 0 к 1

19. Режим 0 таймера 0.

    Режим 0 — Normal. Счётчик TCNT0 — простой суммирующий счётчик. По каждому импульсу тактового сигнала, поступающего с выхода предварительного делителя, его содержимое увеличивается на 1. При переходе TCNT0 через максимум 0xFF возникает переполнение, счёт продолжается с 0x0, флаг переполнения TOV0 устанавливается в единицу. Как только TCNT0 = OCR0, то флаг прерывания OCF0 в регистре TIFR выставляется в 1 и (если разрешено) генерируется соответствующее прерывание.

20. Режим 1 таймера 0.

    Режим 1 — Phase Correct PWN — режим с точной фазой. В нём TCNT0 функционирует как реверсивный счётчик. На каждом сигнале с предварительного делителя значение TCNT0 изменяется на 0x1 в пределах от 0x0 до 0xFF, потом происходит смена направления, а потому на каждом сигнале происходит вычитание, пока значение не достигнет 0x0. Тогда происходит смена направления счёта и устанавливается флаг прерывания TOV0 регистра TIFR. Каждый раз при достижении значения OCR0 устанавливается флаг OCF0, генерируется соответствующее прерывание (если оно разрешено) и изменяется состояние выхода OC0.

21. Режим 2 таймера 0.

    Режим 2 — CTC — режим счёта по модулю OCR0. TCNT0 движется в пределах от 0x0 до OCR0. При достижении OCR0 TCNT0 сбрасывается в ноль, при этом выставляется флаг OCF0, генерируется соответствующее прерывание (если оно разрешено) и может измениться состояние вывода OC0.

22. Режим 3 таймера 0.

    Режим 3 — Fast PWM — быстродействующий ШИМ. Похож на режим 0, но выход OC0 (если он изменяется) изменяется и при достижении порога OCR0, и при достижении 0xFF. Это нужно для генерации ШИМ сигнала на выходе OC0.

23. Можно ли писать в TCNT0 без остановки счёта?

    Такая возможность присутствует, однако это крайне редко используется на практике.

24. Как можно остановить счёт в таймере 0?

    Обнулить 3 последние значения в TCCR0.

25. Система прерываний микроконтроллера ATmega8535.

    21 прерывание. 4 внешних: INT0, INT1, INT2 и RESET. 17 внутренних, обслуживающих дополнительные блоки. Прерывания — некие внешние события, возникающие во время исполнения программы (нажатие кнопки или переполнение таймера). При обнаружении их микроконтроллер вызывает обработчик прерывания соответствующего прерывания. Этот обработчик выполняет действие и возвращается в основную программу командой reti (Return from Interrupt). За обработку прерываний отвечает Interrupt Unit.

26. В каких режимах таймера 0 порог изменяется не сразу (двойная буферизация записи) при записи нового значения в регистр порога с помощью команды OUT?

    В режимах ШИМ-PWR (Широтно-импульсной модуляции), т.е. 1 и 3 режимы таймера 0.

27. Откуда приходит сигнал на вход TCNT0?

    В качестве тактового сигнала счётчика можно использовать внешний сигнал микроконтроллера или внутренний генератор тактовых сигналов.

28. Какое минимальное время требуется для преобразования в АЦП?

    Минимальное время также часто называют периодом дискретизации АЦП, который связан с предельной частотой дискретизации.

    Предельная частота дискретизации определяет быстродействие АЦП и измеряется в герцах или количестве выборок в секунду (SPS – samples per second). Для микроконтроллеров AVR эта величина равна 15 kSPS (килло семплов в секунду). Практически АЦП AVRa работает и быстрее, но при этом его точность ухудшается.

    Теорема Котельникова (теорема Найквиста-Шеннона, теорема о выборке) гласит, что аналоговый сигнал имеющий ограниченный спектр, может быть восстановлен однозначно и без потерь по своим дискретным отсчётам, если частота выборки (дискретизации) превышает максимальную частоту спектра сигнала более чем в 2 раза. Выражаясь по-простому - если нужно оцифровать аналоговый сигнал с полосой спектра 0 - 7 КГц, то в идеальном случае частота дискретизации должна быть $>$ удвоенной максимальной частоты спектра этого сигнала, то есть $>$ 14 КГц. На практике все намного сложнее. Перед входом АЦП всегда ставят НЧ фильтр, чтобы ограничить спектр сигнала, а частота дискретизации выбирается еще более высокой.

29. Когда меняется порог в режиме 3 таймера 0?

    Особенность режима — двойная буферизация в регистр OCR0: записываемое число сохраняется в специальном буферном регистре. Изменение содержимого регистра порога меняется только при достижении счётчиком TNCT0 максимума в $FF. Благодаря этому исключается появление на выходе OC0 неправильных импульсов, которые были бы неизбежны при непосредственной записи в регистр порога без задержки. На практике всё может работать по другому, т.е. порог будет меняться не при достижении максимума счётчиком.

30. Чем сигнальный процессор отличается от МК?

    Сигнальный процессор, в отличие от микроконтроллера, оптимизирован специально под обработку сигналов. Для максимального ускорения подобных задач в сигнальных процессорах аппаратно реализуют циклы для многократного повторения заданного набора команд, в частности перемножение массивов и векторно-конвейерную обработку. Микроконтроллер также обладает системой сигналов (прерываний), однако у МК гораздо большие возможности благодаря наличию портов ввода-вывода, таймеров, а также возможности подключения внешних устройств.

31. Как организовать вложенные прерывания?

    По умолчанию при вызове обработчика прерываний все прерывания запрещаются, что делает невозможным вложенные прерывания. Но при вызове команды SEI в данном обработчике будет возможен вызов ещё одного прерывания, если таковое возникнет.

32. Зачем в программе надо устанавливать начальное значение Stack Pointer и чему оно должно быть равно?

    SP определяет положение вершины стека, а значит и всю последующую работу МК. Адрес вершины стека должен быть больше или равен \$60.

33. Сторожевой таймер и особенности его работы.

    Сторожевой таймер (WatchDog Timer) предназначен для ликвидации сбоев в работе МК, возникающих из-за различного рода помех. Включенный сторожевой таймер через некоторый промежуток времени вырабатывает RESET. Для обнаружения сбоев и предотвращения перезапуска при правильной работе в программу включают команду WDR (WD Reset), которая сбрасывает сторожевой таймер, то есть отсчёт времени до срабатывания RESET начинается заново. При сбое начнёт выполняться произвольный набор команд, и если WDR не встретится в течение периода CT, выполнится RESET.

34. Что такое SPI и зачем он нужен?

    SPI — последовательный синхронный интерфейс, он позволяет передавать данные с высокой скоростью (менее четверти частоты микроконтроллера) между микроконтроллероми различными внешними устройствами.

35. Как инициировать передачу байта в SPI?

    Байт записывается в SPDR ведущего МК: поступает в сдвиговый регистр, MASTER вырабатывает такты (8 тактов), байт от ведущего МК передается в сдвиговый регистр ведомого (одновременно из регистра сдвига ведомого байт из SPDR переходит в регистр ведущего). После передачи такты перестают вырабатываться, содержимое сдвигового регистра переписывается в RDB (Read Data Buffer — регистр промежуточного хранения). Чтение следует произвести до конца очередного цикла передачи.

36. Сколько прерываний и сколько регистров ввода/вывода принадлежит SPI?

    Всего одно прерывание, которое вырабатывается по окончании передачи по адресу $008.
    Три регистра:
    - SPDR ($0F) — Data Register , Read-Write.
    - SPCR ($0D) — Control Register, Read-Write.
    - SPSR ($0E) — Status Register, Read only.

37. Однопроводный интерфейс (сеть MicroLAN).

    1-Wire — двунаправленная шина связи для устройств с низкоскоростной передачей данных, в которой данные передаются по цепи питания (то есть всего используются два провода — один общий (GND), а второй для питания и данных). Сеть устройств 1-Wire со связанным основным устройством названа MicroLan.

38. Как выглядит физический ноль и физическая единица?

    Напряжение, не превышающее 0,7-0,8 В, соответствует логическому нулю, а напряжение больше 2,2-2,3 В — логической единице.

39. Как в однопроводном интерфейсе передаётся информационный ноль и информационная единица? Какова максимальная скорость передачи?

    В 1-Wire системе значения логического 0 и логической 1 представлены импульсами различной длительности. Продолжительность низкого уровня импульса записи 1 должна быть короче 15 мкс, для записи 0 продолжительность низкого уровня импульса должна быть по крайней мере 60 мкс.

    Обычный режим до 15,4 кбит/с и режим overdrive до 125 кбит/с (если стандарт не обновился за последние несколько лет).

40. Что такое серийный номер в однопроводном интерфейсе и какова его структура?

    Серийный номер - уникальный код устройства, с которым работают специальные ROM-команды. Каждое "одноварное"устройство имеет свой 64-битный код, состоящий из трёх частей: Младший байт – это код семейства, к которому относится устройство, 6 следующих байт – уникальный в семействе серийный номер, ну и наконец, старший байт – это CRC, который служит для проверки правильности приёма всего кода.

41. Какая команда позволяет Master определить номера всех Slave в сети MicroLAN?

    Команда SEARCH ROM 0xF0 обладает таким функционалом.

42. Как выглядит сигнал сброса в сети MicroLAN?

    Импульс сброса — низкий уровень на протяжении не менее 8 таймслотов, каждый из которых по 60 мкс, т.е. 480мкс. Далее MASTER устанавливает высокий уровень на время не менее 480 мкс, в течении которого SLAVE должен установить низкий уровень на 60 мкс — этим он показывает присутствие на линии.

Программа: кнопочные ковбои

Из названия программы сразу же понятен ее принцип работы.

После запуска программы через произвольное (2 – 7 секунд) время из пьезопищалки издается короткий звук. Затем игроки стараются как можно быстрее отреагировать на сигнал старта дуэли и нажать на свою кнопку. Кто первый нажимает кнопку, у того загорается светодиод на 1 секунду (сигнал победы). И так по кругу...

```
#define BUZZER_PIN   12  // пин с пищалкой
#define PLAYER_COUNT 2   // количество игроков-ковбоев

// объявляем пару списков: один с номерами пинов с кнопками, другой - со светодиодами
int buttonPins[PLAYER_COUNT] = {3, 13};
int ledPins[PLAYER_COUNT] = {9, 11};

void setup()
{
  pinMode(BUZZER_PIN, OUTPUT);
  for (int player = 0; player < PLAYER_COUNT; ++player)
  {
    pinMode(ledPins[player], OUTPUT);
    pinMode(buttonPins[player], INPUT_PULLUP);
  }
}

void loop()
{
  // даём сигнал «пли!», выждав случайное время от 2 до 7 сек
  delay(random(2000, 7000));
  tone(BUZZER_PIN, 3000, 250); // 3 килогерца, 250 миллисекунд

  for (int player = 0; ; player = (player + 1) % PLAYER_COUNT)
  {
    // если игрок номер «player» нажал кнопку...
    if (!digitalRead(buttonPins[player]))
    {
      // ...включаем его светодиод и сигнал победы на 1 сек
      digitalWrite(ledPins[player], HIGH);
      tone(BUZZER_PIN, 4000, 1000);
      delay(1000);
      digitalWrite(ledPins[player], LOW);
      break; // Есть победитель! Выходим
    }
  }
  delay(1000);
}
```