

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ФИЗТЕХ-ШКОЛА РАДИОТЕХНИКИ И КИБЕРНЕТИКИ

Ответы на 100 вопросов.

Работу выполнили:
Державин Андрей Андреевич, группа Б01-901
Хайдарй Фарид Гулович, группа Б01-901
Шурыгин Антон Алексеевич, группа Б01-909
Лиризмáн Карина Сергеевна, группа Б03-001

Долгопрудный, 2021

Содержание

1	Ответы:	4
1.1	Ваши фамилия, имя, отчество, номер группы.	4
1.2	Фамилия, имя, отчество лектора.	4
1.3	Чем отличается микроконтроллер от микропроцессора.	4
1.4	Какие тактовые частоты могут быть у ATmega8535.	4
1.5	Какие таймеры есть у ATmega8535.	4
1.6	Внутренняя структура МК.	4
1.7	Какие значения записаны в TCCR после сигнала RESET.	5
1.8	Порт А. Сколько прерываний и сколько регистров ввода/вывода принадлежит порту А. Назначение этих регистров ввода/вывода.	5
1.9	Регистр SREG. Назначение его разрядов.	5
1.10	Почему после сигнала RESET все прерывания запрещены.	6
1.11	Приведите пример использования разряда Т в регистре SREG.	6
1.12	Таймер 0. Режимы работы, количество прерываний, регистры ввода/вывода, принадлежащие таймеру 0.	6
1.13	В каких режимах таймера 0 порог изменяется не сразу (двойная буферизация записи) при записи нового значения в регистр порога с помощью команды OUT.	7
1.14	Откуда приходит сигнал на вход TCNT0.	7
1.15	Как можно разрешить (запретить) прерывания по переполнению таймера 0.	7
1.16	Написать программу с использованием таймера 0, вырабатывающую симметричное прямоугольное колебание на одном из выходов порта А.	7
1.17	Какие коэффициенты деления частоты позволяет получать предварительный делитель таймера 0.	8
1.18	Какой режим таймера 0 позволяет вырабатывать треугольные колебания, используя дополнительную интегрирующую цепочку.	8
1.19	Как запрограммировать предварительный делитель таймера 0.	10
1.20	Режим 0 таймера 0.	10
1.21	Режим 1 таймера 0.	10
1.22	Режим 2 таймера 0.	10
1.23	Режим 3 таймера 0.	10
1.24	Когда меняется порог в режиме 3 таймера 0.	10
1.25	Можно ли писать в TCNT0 без остановки счета.	10
1.26	Как можно остановить счет в таймере 0.	10
1.27	Система прерываний микроконтроллера ATmega8535.	10
1.28	Сколько всего прерываний у ATmega8535.	10
1.29	Как организовать вложенные прерывания.	10
1.30	Как можно разрешить (запретить) одновременно все прерывания.	10
1.31	Как организована система приоритетов при обработке прерываний.	10
1.32	Какое минимальное время требуется для преобразования в АЦП.	10
1.33	Чем сигнальный процессор отличается от МК.	10
1.34	Зачем в программе надо устанавливать начальное значение Stack Pointer и чему это значение должно быть равно.	10

1.35	Сторожевой таймер и особенности его работы.	10
1.36	Что такое SPI и зачем он нужен.	10
1.37	Как инициировать передачу байта в SPI.	10
1.38	Сколько прерываний и сколько регистров ввода/вывода принадлежит SPI.	10
1.39	Далее пойдут вопросы про однопроводный интерфейс (сеть MicroLAN).	10
1.40	Сколько проводов необходимо для реализации однопроводного интерфейса.	10
1.41	Как выглядит физический ноль и физическая единица.	10
1.42	Как в однопроводном интерфейсе передается информационный ноль и информационная единица? Какова максимальная скорость передачи?	10
1.43	Что такое серийный номер в однопроводном интерфейсе и какова его структура.	10
1.44	Какая команда позволяет Master определить номера всех Slave в сети MicroLAN.	10
1.45	Как выглядит сигнал сброса в сети MicroLAN.	10

1 Ответы:

1.1 Ваши фамилия, имя, отчество, номер группы.

- Державин Андрей Андреевич, группа В01-901
- Хайдарі Фарид Гулович, группа В01-901
- Шурыгин Антон Алексеевич, группа В01-909
- Лирисман Карина Сергеевна, группа В03-001

1.2 Фамилия, имя, отчество лектора.

Донов Геннадий Иннокентьевич.

1.3 Чем отличается микроконтроллер от микропроцессора.

1.4 Какие тактовые частоты могут быть у ATmega8535.

1, 2, 4 МГц от внутреннего генератора. 0.1 - 16 МГц от внешнего генератора.

1.5 Какие таймеры есть у ATmega8535.

У ATmega8535 есть следующие таймеры:

- два 8-разрядных таймера
- один 16-разрядный таймер

1.6 Внутренняя структура МК.

Многие современные МК имеют структуру, приведённую на рис. 1. Отмеченные на рисунке блоки, входящие в состав микроконтроллера, выполняют следующие функции:

- *Процессор*
Обеспечение обработки информации путём выполнения команды в соответствии с системой команд микроконтроллера.
- *Память программ*
Хранение программы, в соответствии с которой работает микроконтроллер.
- *ОЗУ*
Другое название — **RAM** (Random Access Memory). Хранение промежуточных результатов.
- *Порты ввода/вывода*
Осуществление обмена информацией с внешним миром.
- *Блок управления питанием*
Обеспечение правильности запуска микроконтроллера после включения питания.
- *Блок управления сбросом*
Установка вместе с входом RESET микроконтроллера в некоторое исходное состояние.
- *Блок синхронизации*
Выработка тактовых сигналов, необходимых для правильного взаимодействия всех внутренних блоков микроконтроллера.



Рис. 1 Внутренняя структура микроконтроллера.

1.7 Какие значения записаны в TCCR после сигнала RESET.

После сигнала RESET все разряды будут установлены в нулевое значение.

1.8 Порт А. Сколько прерываний и сколько регистров ввода/вывода принадлежит порту А. Назначение этих регистров ввода/вывода.

1.9 Регистр SREG. Назначение его разрядов.

Регистр SREG — 8-разрядный регистр признаков (регистр флагов). Назначение разрядов приведено на рис. 2 Рассмотрим подробнее назначение разрядов:

7	6	5	4	3	2	1	0	BIT
I	T	H	S	V	N	Z	C	Address \$3F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Read/Write
0	0	0	0	0	0	0	0	Initial value

Рис. 2 Назначение разрядов регистра SREG

- Бит 7 – I

Глобальное разрешение прерываний. Если в этом разряде нуль, то никакие прерывания не будут обрабатываться. Бит обнуляется при возникновении любого прерывания, и автоматически выставляется в единицу при выходе из прерывания.

- Бит 6 – T
Временное хранение бита. С помощью команд BST и BLD позволяет передавать бит из одного регистра общего назначения в другой. Например, следующий код:

 bst r31, 7 ; запись значения седьмого разряда регистра r31 в T
 bld r0, 3 ; запись из T в третий разряд регистра r0
- Бит 5 – H
Признак переноса между полубайтами.
- Бит 4 – S
Равен сумме по модулю 2 содержимого третьего и второго разряда регистра SREG.
- Бит 3 – V
Признак переполнения.
- Бит 2 – N
Признак отрицательного результата.
- Бит 1 – Z
Признак нуля.
- Бит 0 – C
Признак переноса.

1.10 Почему после сигнала RESET все прерывания запрещены.

Для обеспечения корректной инициализации работы микроконтроллера.

1.11 Приведите пример использования разряда T в регистре SREG.

bst r30, 5 ; запись значения пятого разряда регистра r30 в T

1.12 Таймер 0. Режимы работы, количество прерываний, регистры ввода/вывода, принадлежащие таймеру 0.

Режимы работы:

- Normal (0)
Счётчик TCNT0 функционирует как обычный суммирующий счётчик.
- PWM Phase Correct (1)
Режим ШИМ с точной фазой. Предназначен для генерации сигналов с широтно-импульсной модуляцией.
- CTC – Clear Timer on Compare match (2)
Режим счёта по модулю, который определяется содержимым регистра OCR0.
- Fast PWM (3)
Быстродействующий ШИМ. Позволяет генерировать высокочастотный сигнал с широтно-импульсной модуляцией.

Прерывания:

- TIMERO_OVF – переполнение таймера
- TIMERO_COMP – содержимое счётчика TCNT0 совпадает с содержимым регистра OCR0.

Имеет 3 регистра ввода-вывода. Ещё 2 регистра используются совместно с таймерами 1 и 2:

- TCCR0 – Регистр контроля
- SFIOR – Регистр обнуления
- TMSK – Регистр прерывания
- TIFR – Регистр флагов прерываний

Также есть возможность использования двух выводов микроконтроллера:

- вход T0 — Timer/Counter0 External Counter Input – вывод PB0
- выход OC0 — Timer/Counter0 Output Compare Match Output – вывод PB3

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

- PWM Phase Correct (1)
Режим ШИМ с точной фазой. Предназначен для генерации сигналов с широтно-импульсной модуляцией.
- Fast PWM (3)
Быстродействующий ШИМ. Позволяет генерировать высокочастотный сигнал с широтно-импульсной модуляцией.

1.14 Откуда приходит сигнал на вход TCNT0.

Сигналы на вход TCNT0 приходят с выхода управляемого предварительного делителя частоты.

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

Разрешить:

```
ldi r16, 1 << TOIE0
out TMSK, r16
```

Запретить:

```
ldi r16, 0 << TOIE0
out TMSK, r16
```

1.16 Написать программу с использованием таймера 0, вырабатывающую симметричное прямоугольное колебание на одном из выходов порта A.

```
#include <avr/io.h>
#include <avr/interrupt.h>

.global TIMERO_COMP_vect
TIMERO_COMP_vect:
    in r16, PORTA
    eor r16, r17
    out PORTA, r16
    reti
```

```
.global main
main:

    sei ; разрешить прерывания

    sbi DDRA, DDA0 ; PA0 - выход
    cbi PORTA, PORTA0 ; PA0 = 0

    ldi r17, 1 << PORTA0
    ldi r16, 1 << OCIE0 ; разрешить прерывание по совпадению
    out TIMSK, r16

    ldi r16, 0x7f ; threshold on half-way
    out OCR0, r16

    ldi r16, 1 << WGM00 | 1 << CS00 ; phase-correct PWM
    out TCCR0, r16

loop:

    nop
    nop
    rjmp loop
```

1.17 Какие коэффициенты деления частоты позволяет получать предварительный делитель таймера 0.

1, 8, 64, 256, 1024

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

В режимах Normal и CTC – нужно поставить OSC0 изменяется при совпадении с порогом.

В ШИМ режимах – выставить порог в половину максимального.

- 1.19 Как запрограммировать предварительный делитель таймера 0.
- 1.20 Режим 0 таймера 0.
- 1.21 Режим 1 таймера 0.
- 1.22 Режим 2 таймера 0.
- 1.23 Режим 3 таймера 0.
- 1.24 Когда меняется порог в режиме 3 таймера 0.
- 1.25 Можно ли писать в TCNT0 без остановки счета.
- 1.26 Как можно остановить счет в таймере 0.
- 1.27 Система прерываний микроконтроллера ATmega8535.
- 1.28 Сколько всего прерываний у ATmega8535.
- 1.29 Как организовать вложенные прерывания.
- 1.30 Как можно разрешить (запретить) одновременно все прерывания.
- 1.31 Как организована система приоритетов при обработке прерываний.
- 1.32 Какое минимальное время требуется для преобразования в АЦП.
- 1.33 Чем сигнальный процессор отличается от МК.
- 1.34 Зачем в программе надо устанавливать начальное значение Stack Pointer и чему это значение должно быть равно.
- 1.35 Сторожевой таймер и особенности его работы.
- 1.36 Что такое SPI и зачем он нужен.
- 1.37 Как инициировать передачу байта в SPI.
- 1.38 Сколько прерываний и сколько регистров ввода/вывода принадлежит SPI.
- 1.39 Далее пойдут вопросы про однопроводный интерфейс (сеть MicroLAN).
- 1.40 Сколько проводов необходимо для реализации однопроводного интерфейса.
- 1.41 Как выглядит физический ноль и физическая единица.
- 1.42 Как в однопроводном интерфейсе передается информационный ноль и информационная единица? Какова максимальная скорость передачи?
- 1.43 Что такое серийный номер в однопроводном интерфейсе и какова его структура.
- 1.44 Какая команда позволяет Master определить номера всех Slave в сети MicroLAN.
- 1.45 Как выглядит сигнал сброса в сети MicroLAN.