КИПиА ЯМАЛ
6-й разряд · МДК.01.02

Полный конспект лекции на 2 часа

Тема 2.4. Системы управления на базе микроконтроллеров

ATmega8535, периферия, AVR и программирование на C

КУПД «Слесарь по КИПиА» от 21.04.2026 № УПД-15

Параметр	**Значение**
Тема	2.4
Объём	2 часа теории + 2 часа ЛПЗ
Учебная модель	ATmega8535
Основа	КУПД «Слесарь по КИПиА» от 21.04.2026 № УПД-15

СОДЕРЖАНИЕ

1. Однокристальные микроконтроллеры

2. Архитектура AVR и память

3. Порты ввода/вывода

4. Таймеры T0/T1/T2 и watchdog

5. Прерывания и энергосбережение

6. Аналоговый компаратор и АЦП

7. SPI и USART

8. Программирование и AVR-ассемблер

9. Язык C для AVR

10. Система управления

11. Диагностика

12. Итоги

1. Однокристальные микроконтроллеры

Микроконтроллер объединяет CPU, память и периферию на одном кристалле. ATmega8535 — 8-разрядный AVR с 8 КБ Flash, 512 Б SRAM, 512 Б EEPROM, 32 GPIO, тремя таймерами, SPI, USART, 10-битным АЦП и watchdog.

Микроконтроллер как однокристальная система

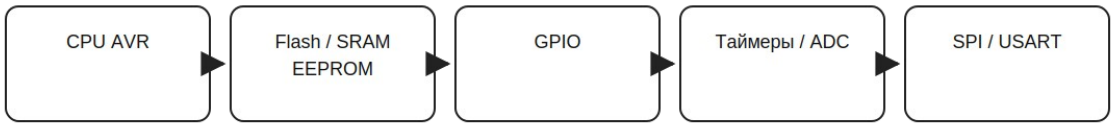


Рисунок 1. Однокристальная система.

2. Архитектура AVR и память

AVR содержит 32 рабочих регистра, АЛУ и SREG. Flash хранит программу, SRAM — переменные и стек, EEPROM — энергонезависимые параметры.

Упрощённая архитектура AVR

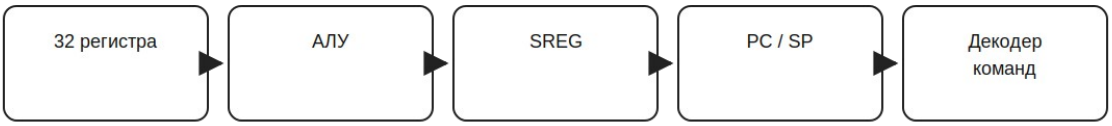


Рисунок 2. Ядро AVR.

Память ATmega8535

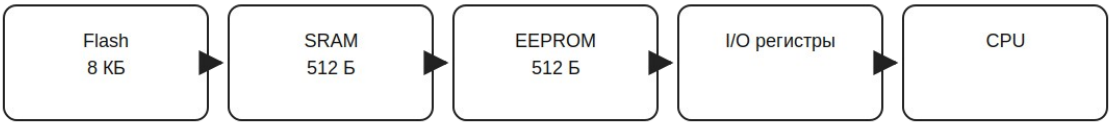


Рисунок 3. Память ATmega8535.

3. Порты ввода/вывода

DDRx задаёт направление, PORTx — выходной уровень или подтяжку, PINx — чтение входа.

Порт ввода/вывода AVR

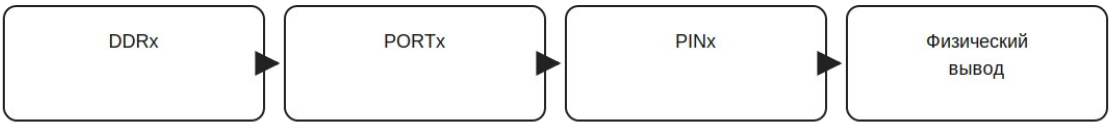


Рисунок 4. GPIO AVR.

```
DDRB |= (1 << PB0);
PORTB |= (1 << PB0);
PORTB &= ~(1 << PB0);
```

4. Таймеры T0/T1/T2 и watchdog

T0 и T2 — 8-битные, T1 — 16-битный. Предделитель уменьшает частоту счёта. Таймеры используются для периодических событий, измерения времени и PWM. Watchdog перезапускает систему, если программа перестала его обслуживать.

8-битный таймер T0

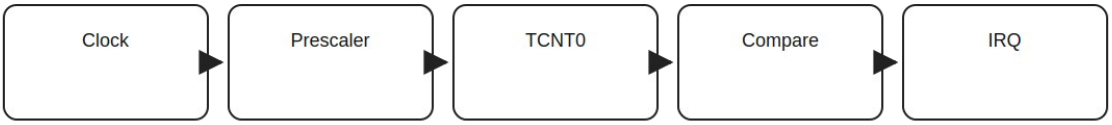


Рисунок 5. T0.

16-битный таймер T1

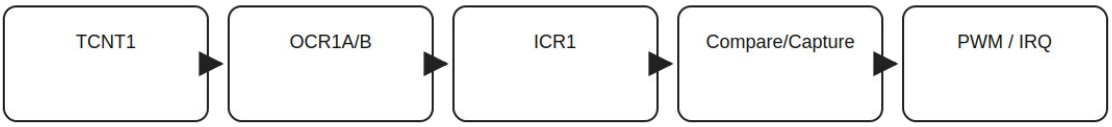


Рисунок 6. T1.

8-битный таймер T2

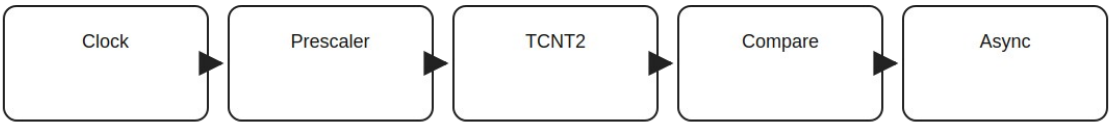


Рисунок 7. T2.

Сторожевой таймер

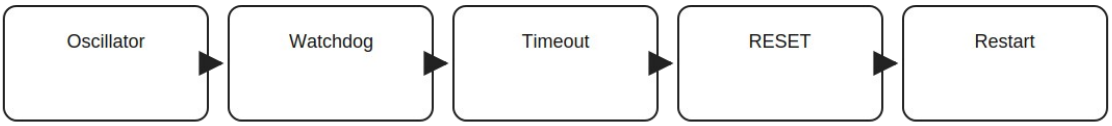


Рисунок 8. Watchdog.

5. Прерывания и энергосбережение

Событие устанавливает флаг и при разрешении прерываний вызывает ISR по вектору. Возврат выполняется RETI. Режимы сна позволяют отключать части ядра и периферии.

Прерывания AVR



Рисунок 9. Прерывание.

Режимы энергосбережения

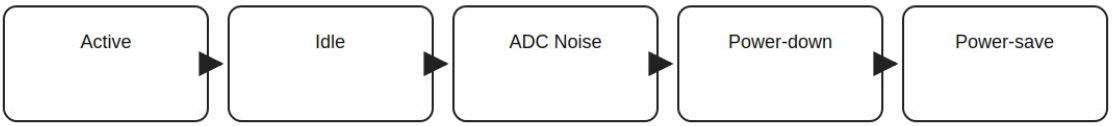


Рисунок 10. Режимы сна.

6. Аналоговый компаратор и АЦП

Компаратор выдаёт пороговый цифровой результат. 10-битный АЦП формирует код 0…1023. Для идеального случая D ≈ Uвх/Vref × 1023.

Uвх при Vref=5 В	Код
0	0
1,25 В	≈256
2,50 В	≈512
3,75 В	≈767
5,00 В	1023

Аналоговый компаратор

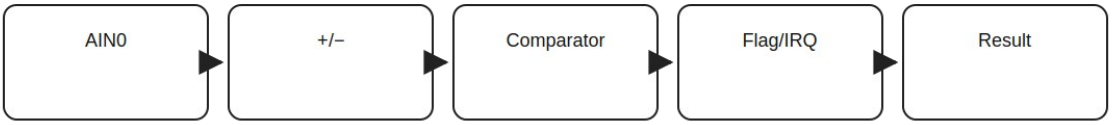


Рисунок 11. Компаратор.

10-битный АЦП

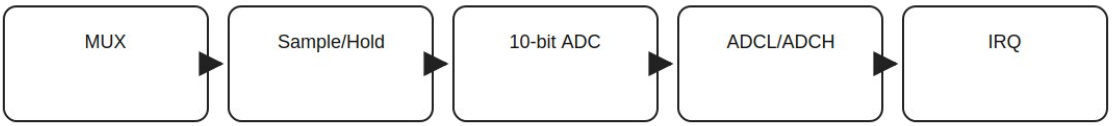


Рисунок 12. АЦП.

7. SPI и USART

SPI использует MOSI, MISO, SCK и SS. USART обеспечивает последовательный синхронный или асинхронный обмен и требует согласования скорости и формата кадра.

Интерфейс SPI

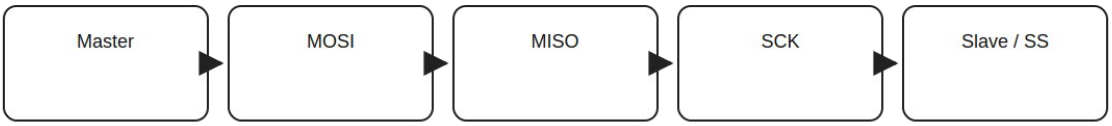


Рисунок 13. SPI.

Интерфейс USART

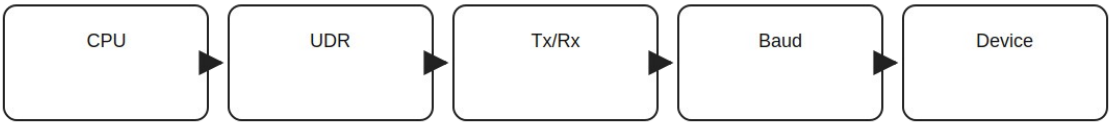


Рисунок 14. USART.

8. Программирование и AVR-ассемблер

Исходный код преобразуется в машинный образ и записывается во Flash. Отдельно программируются EEPROM и конфигурация. Неверные настройки источника тактирования могут привести к отсутствию запуска. В AVR-ассемблере рассматриваются SREG, адресация и группы команд: управление, логика, арифметика, биты и пересылка.

Программирование

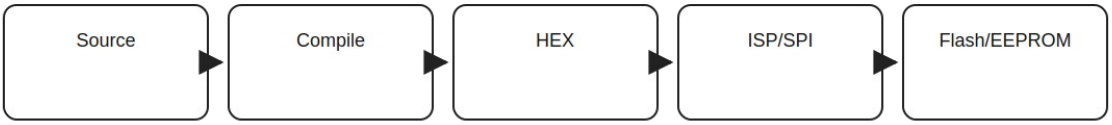


Рисунок 15. Программирование.

9. Язык C для AVR

Программа содержит заголовки, функции, переменные, операторы, циклы и массивы. Побитовые операции применяются для настройки периферии.

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

int main(void)
{
    DDRB |= (1 << PB0);
    while (1) {
        PORTB ^= (1 << PB0);
    }
}
```

Структура программы C

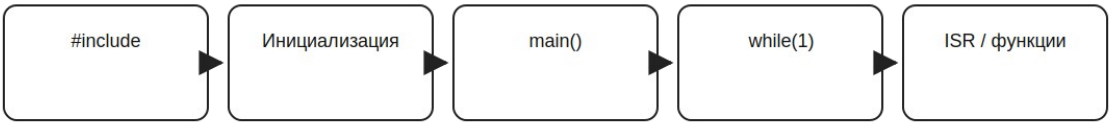


Рисунок 16. Структура C-программы.

10. Система управления

Типовая цепочка: датчик → АЦП → алгоритм → GPIO/PWM → исполнитель. Аппаратные узлы и программа работают как единая система.

Система управления



Рисунок 17. Система управления.

11. Диагностика

Проверка ведётся от питания и RESET к источнику тактирования и конфигурации, затем к GPIO, периферии и программе. Для АЦП проверяют AVCC/Vref и выбранный канал, для USART — clock, baud и формат кадра.

Признак	Что проверить
МК не стартует	питание, RESET, clock/fuse
GPIO не работает	DDRx/PORTx и нагрузку
Таймер неверен	clock/prescaler/mode
ADC неверен	Vref, канал, AVCC
USART ошибается	baud/frame/clock
Сбросы	watchdog, питание, помехи

Диагностика

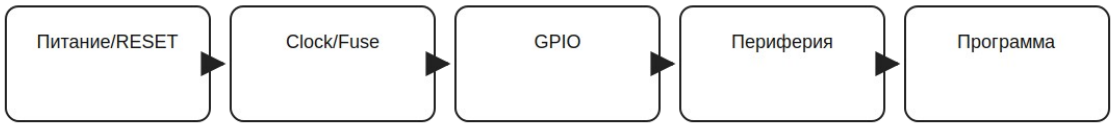


Рисунок 18. Диагностика.

12. Итоги

Микроконтроллерная система — совместная работа аппаратных ресурсов и программы. Для диагностики отказ локализуют между физическим сигналом, периферийным модулем, регистром и алгоритмом.

1. Чем МК отличается от МП?
2. Назначение Flash/SRAM/EEPROM?
3. Как работают DDRx/PORTx/PINx?
4. Чем T1 отличается от T0/T2?
5. Зачем watchdog?
6. Как работает прерывание?
7. Каков диапазон 10-битного ADC?
8. Линии SPI?
9. Назначение USART?
10. Что проверять при отсутствии запуска?

Источники: КУПД от 21.04.2026 № УПД-15; ATmega8535(L) Datasheet, document 2502.