

Разработка кодового замка на базе микроконтроллера AVR

Очередность этапов разработки

- 1. Постановка задачи
- 2. Разработка структурной схемы
- 3. Разработка программного обеспечения
- 4. Разработка электрической принципиальной схемы
- 5. Комплексная отладка

1. Постановка задачи

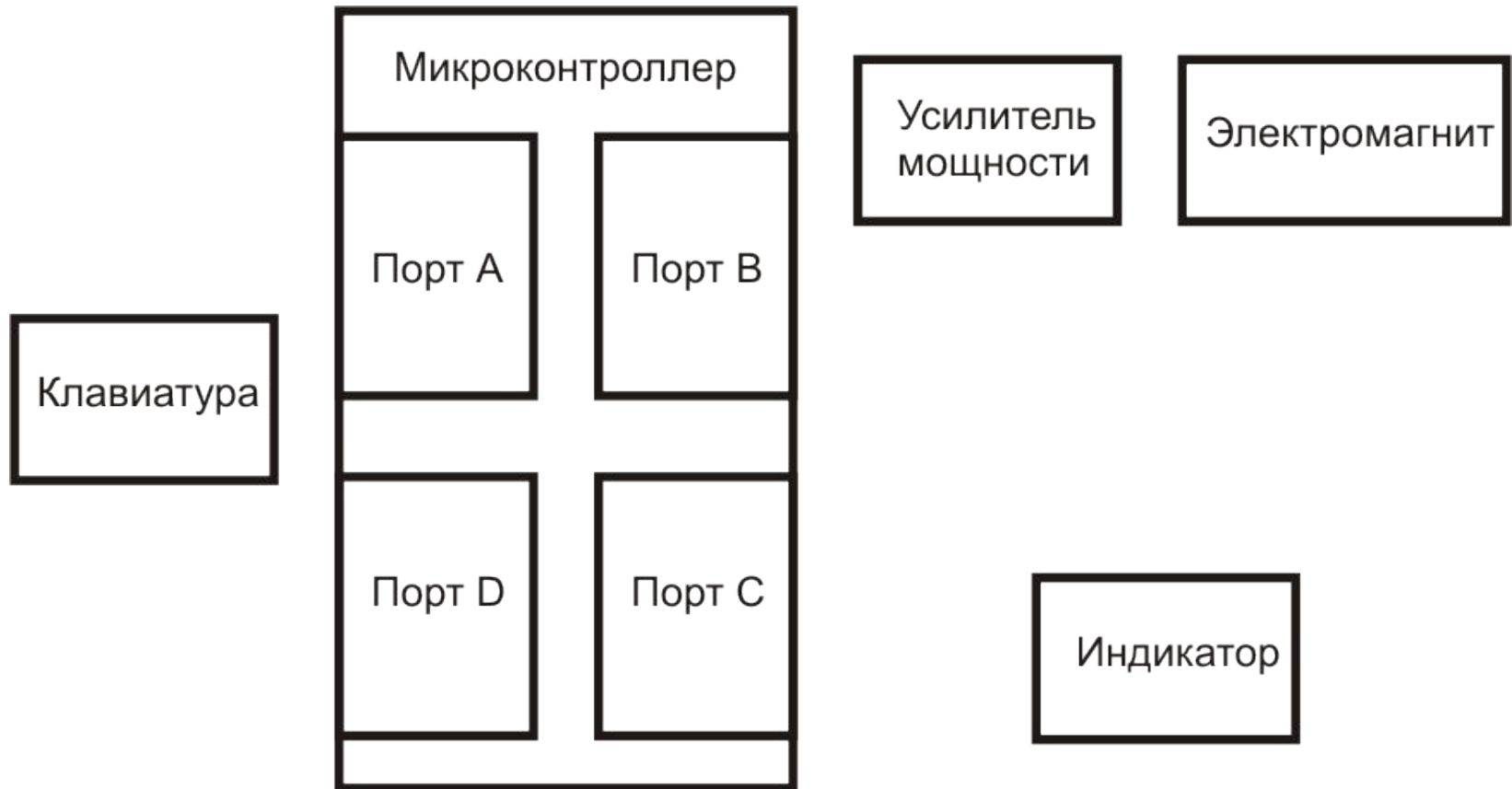
Описание устройства

- Кодовый замок должен иметь защиту от неправильно введенного кода.
- В случае если неправильный код водится три раза должна срабатывать сигнализация.
- Предусмотреть индикацию введенного кода и режимы работы кодового замка.

Предварительный выбор оборудования:

1. Выбираем микроконтроллер AVR имеющий четыре порта
2. Клавиатура
3. Усилитель мощности
4. Электромагнит замка
5. Индикатор

Состав оборудования кодового замка



Исходные данные

- Тип кода – двоичный;
- Количество комбинаций – 256;
- Количество попыток ввода кода – 3;
- Сигнализация неправильно набранного кода.

2. Разработка структурной СХЕМЫ

Подготовка к разработке структурной схемы

- Требуется выбрать конкретное оборудование (пока не выбирая микроконтроллер)
- Затем выбрать конкретный микроконтроллер
- Выполнить распределение ресурсов микроконтроллера

Распределение ресурсов микроконтроллера

- В данной разработке распределение ресурсов сводится к распределению портов ввода - вывода

Разделение системы на подсистемы

- Для удобства проектирования разделим все устройство на две системы:
- Система ввода
- Система вывода

Система ввода

Требования к системе ввода

- Требуется:
 - 1. Ввести код доступа
 - 2. Подтвердить, что код набран верно
 - 3. В случае неправильно набранного кода
- выполнить сброс кода

Выбор клавиатуры

- Выбираем двоичную клавиатуру
- В качестве кнопок ввода кода – 8 тумблеров с фиксацией
- Кнопки подтверждения и сброса без фиксации

КЛАВИАТУРА

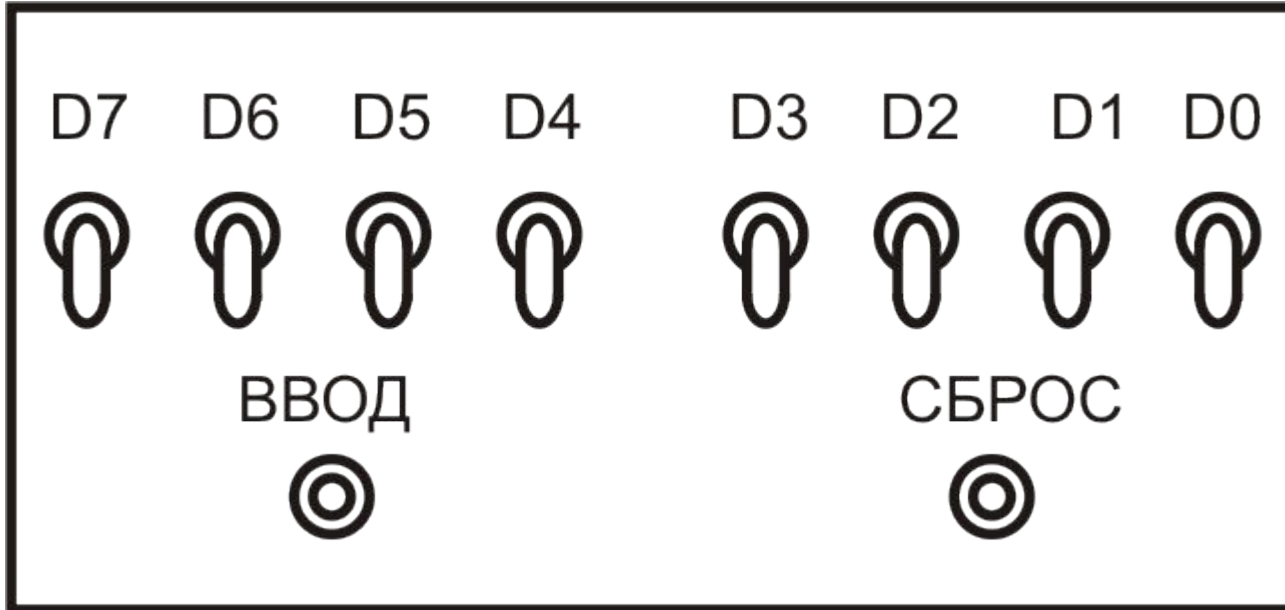
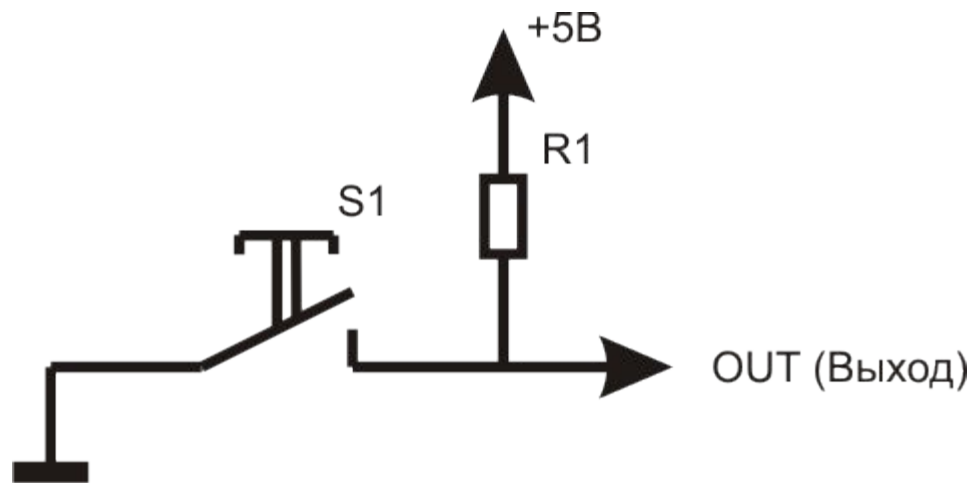


Схема включения кнопки



Эквивалентная схема линии порта

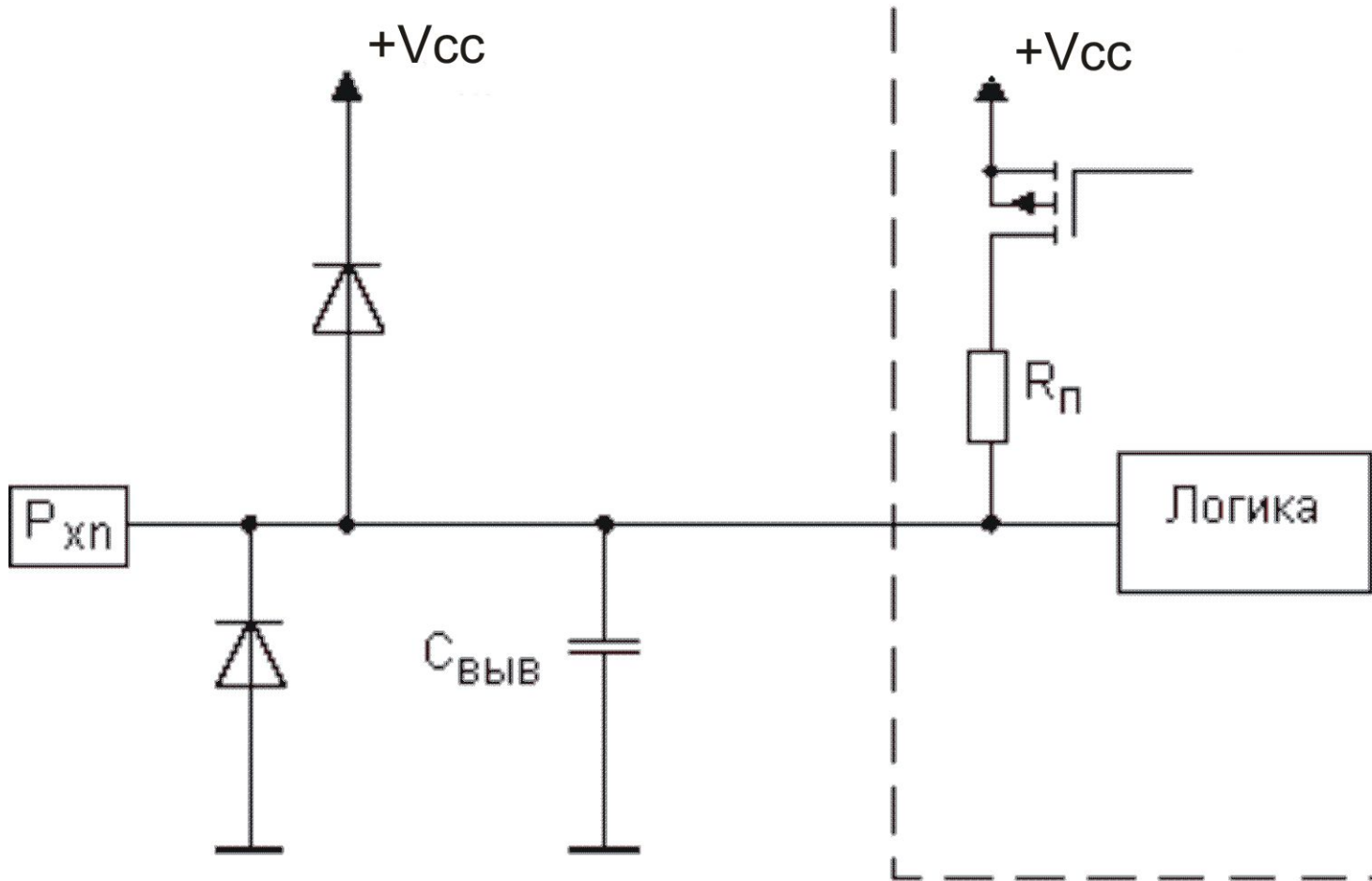
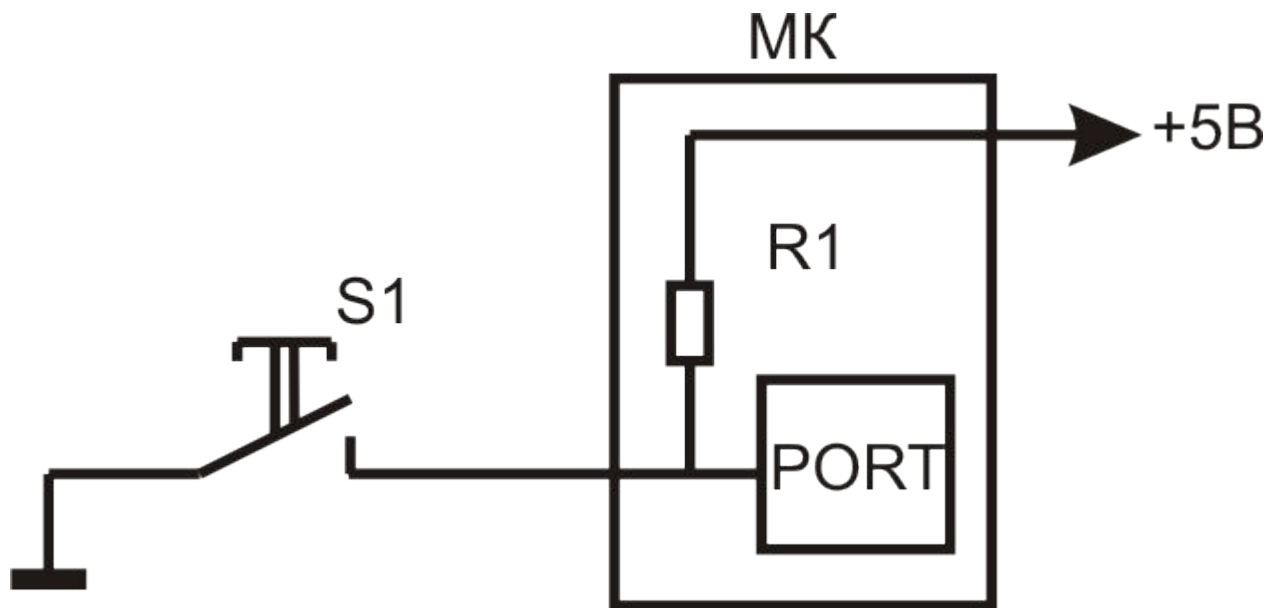


Схема включения кнопки к порту МК с Pull-up резистором



Система вывода

Требования к системе вывода

- Требуется:
- 1. Отображать на индикаторе набранный код доступа
- 2. Управлять электромагнитом замка
- 3. Возможно подключение звукового сигнала

Выбор устройства индикации

- Выберем двоичный индикатор отображения набранного кода на основе 8 светодиодов (LED)
- Добавим еще 2 светодиода:
индикатор правильно набранного кода и
индикатор неправильно набранного кода

Управление электромагнитом замка

- Требуется только открывать и закрывать замок
- Выберем схему с усилителем релейного типа
- Таким образом нам требуется одна линия вывода для управления электромагнитом замка

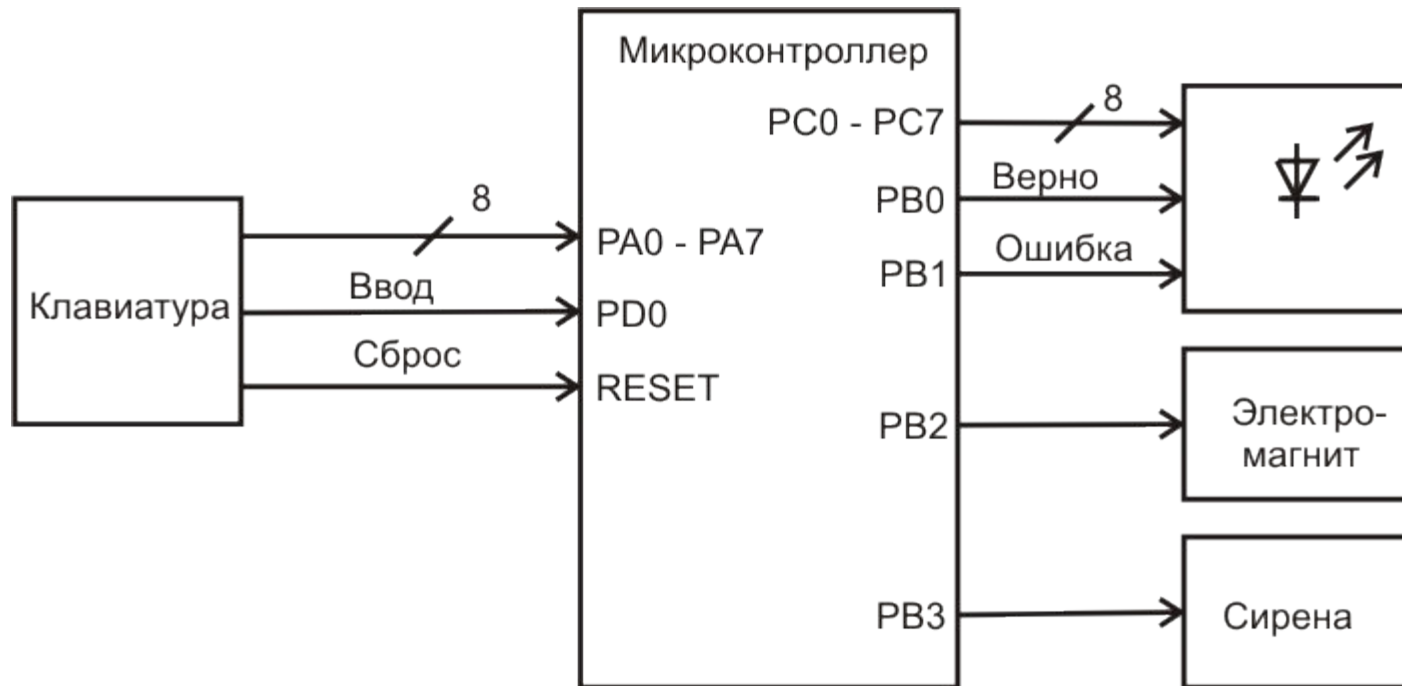
Настройка линий порта на ввод

- Порт A – ввод кода доступа
- Линия D0 – подтверждение правильно набранного кода

Настройка линий порта на вывод

- Порт С – отображение введенного кода
- Линия В0 – подтверждение правильно набранного кода
- Линия В1 – набран неправильный код
- Линия В2 – управление замком
- Линия В3 – управление звуковым сигналом

Структурная схема



Активный уровень сигнала

- В качестве активного уровня сигнала управления внешними устройствами выбираем ЛОГИЧЕСКИЙ НОЛЬ

3. Разработка программного обеспечения

Разработка алгоритма

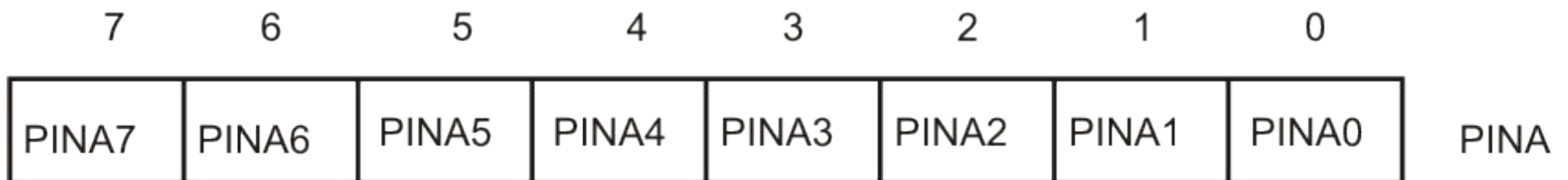
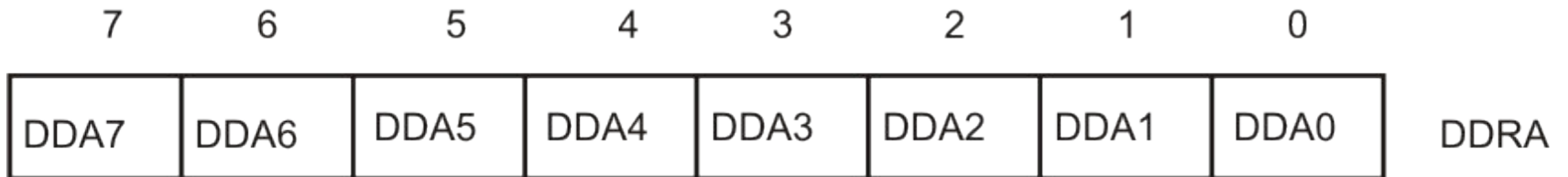
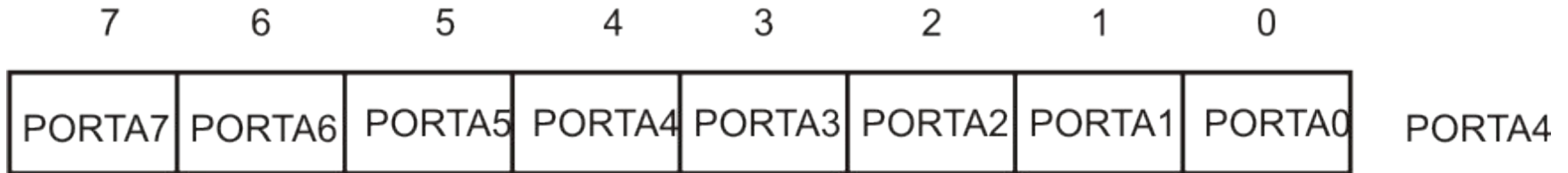
- Требуется разработать алгоритм работы кодового замка

Настройка портов ввода - вывода

Распределение линий портов

- Ввод:
- Порт А
- Порт D
- Вывод:
- Порт В
- Порт С

Регистры портов



Настройка порта

DDAn	PORTAn	I/O	Pull up	Комментарии
0	0	In	Нет	Z-состояние
0	1	in	Да	Rхп источник тока, если L
1	0	Out	Нет	Вывод лог. 0
1	1	Out	Нет	Вывод лог. 1

Выбор микроконтроллера

- На этом этапе выбираем микроконтроллер семейства AVR, имеющий четыре порта ввода-вывода.
- Предположим мы выбрали микроконтроллер ATmega32.

Написание текста программы на языке ассемблера:

- В соответствии с ранее разработанным алгоритмом пишется программа работы микроконтроллера
- Ниже будут приведены примеры типовых программных модулей

Типовые программные модули инициализации микроконтроллера:

1. Подключение стандартной библиотеки описания имен
2. Настройка портов ввода-вывода
3. Настройка указателя стека

Подключение библиотеки: «Описание имен для ATmega32»

```
.NOLIST
```

```
.include "m32def.inc"
```

```
.LIST
```

Настройка портов ввода-вывода

Настройка порта A на ввод

```
ldi r16, $FF      ; загрузка константы FFH в  
                  ; регистр R16  
out PORTA, r16; Загрузка содержимого R16 в  
              ; регистр данных порта A)
```

Настройка порта D на ввод

`ldi r16, $FF` ; загрузка константы FFH в
; регистр R16

`out PORTD, r16;` Загрузка содержимого R16 в
; регистр данных порта D)

Настройка порта В на вывод

```
ldi r16, $FF      ; загрузка константы FFH в  
    ; регистр R16  
out PORTB, r16    ; Загрузка содержимого R16 в  
    ; регистр данных порта В
```

```
ldi r16, $FF      ; загрузка константы FFH в  
    ; регистр R16  
out DDRB, r16     ; Загрузка содержимого R16 в  
    ; регистр направления порта В
```


Настройка порта C на вывод

```
ldi r16, $FF      ; загрузка константы FFH в  
    ; регистр R16  
out PORTC, r16    ; Загрузка содержимого R16 в  
    ; регистр данных порта C
```

```
ldi r16, $FF      ; загрузка константы FFH в  
    ; регистр R16  
out DDRC, r16     ; Загрузка содержимого R16 в  
    ; регистр направления порта C
```

Настройка стека

- Например, определяем адрес указателя стека SP на ячейку 085FH (последняя ячейка IRAM ATmega32)
- Содержимое регистра указателя стека определяется:

• Регистр	Н-адрес	Н-код
• SPH	3EH	08H
• SPL	3DH	5FH

Настройка указателя стека

```
ldi R16, low (RAMEND)
```

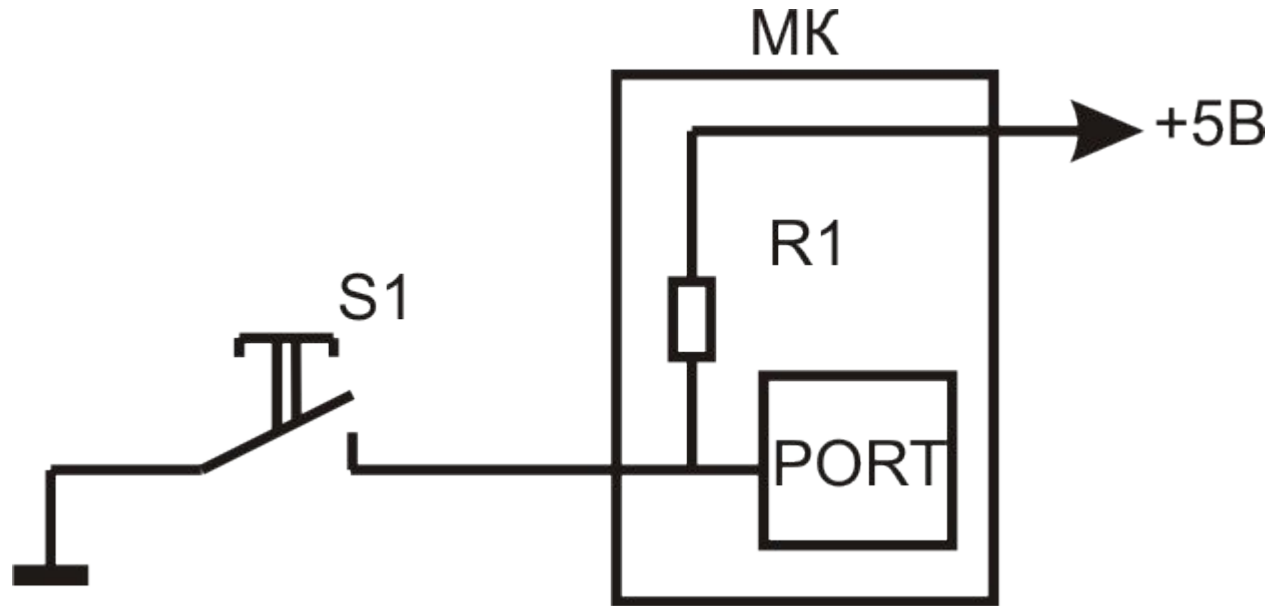
```
out SPL, R16
```

```
ldi R16, high (RAMEND)
```

```
out SPH, R16
```

Примеры рабочих программных модулей

Ожидание нажатия кнопки



Команды «СКИП»

`sbic PINx, n` – переход через следующую команду (скип) если бит порта ввода сброшен

`sbis PINx, n` – переход через следующую команду (скип) если бит порта ввода установлен

Ожидание нажатия кнопки (sbic)

- WAIT_KEY: wdr
sbic PIND, 0
rjmp WAIT_KEY
nop

Ожидание отпускания кнопки (sbis)

- WAIT_KEY: wdr
sbis PIND, 0
rjmp WAIT_KEY
nop

Использование команд ввода/вывода

- В командах ввода-вывода могут быть использованы любые регистры с именами от R0 до R31
- В примерах приведенных далее в качестве регистра будет использован только регистр R16

Чтение порта

in r16, PINA ; ввод информации из порта A
; в регистр r16

in r16, PIND ; ввод информации из порта D в
;регистр r16

Запись в порт

out PORTC, r16 ; вывод информации из
; регистра r16 в порт C

out PORTD, r16 ; вывод информации из
; регистра r16 в порт D

Включить или выключить устройство

- Для управления некоторым устройством требуется одна линия вывода
- Рекомендуется использовать команды операций с отдельными битами порта
- `cbi` – сбросить (в ноль) линию порта
- `sbi` – установить (е единицу) линию порта

Пример команды cbi

cbi PORTC, 2 ;Сбросить линию порта C
; с номером 2

Пример команды sbi

`sbi PORTC, 2` ;Установить линию порта C
;с номером 2

Пример сравнения содержимого регистров

Сравним содержимое двух регистров R18 и R20

- R18 – заданная величина
- R20 – неизвестная величина
- Требуется определить равны они или нет

Используем команду вычитания

`mov r19, r18 ; копирование заданного
; значения`

`sub r19, r20 ; вычитание неизвестной
; величины из копии заданного`

После операции вычитания в r19

помещается разность и устанавливаются
флаги (C, Z и т.д.)

Реализация ветвлений в программе

- В зависимости от того, что были равны или не равны значения в регистрах, требуется организовать ветвления в программе.
- Используем команды условных переходов

Условные переходы по флагу «Z»

`brne NE_RAVNO`; переход на метку

;«NE_RAVNO» если

; результат не равен нулю

; флаг «Z» сброшен

`breq RAVNO` ; переход на метку

;«RAVNO» если

; результат равен нулю

; флаг «Z» установлен