

Хартов В.Я.
Методические указания
Для проведения лабораторных работ по курсу
Микропроцессорные системы

Лабораторная работа 1. Программирование портов ввода-вывода

Цель работы: изучение системы команд микроконтроллеров AVR, приемов программирования на AVR Ассемблере, получение навыков отладки программ в среде отладки AVR Studio 4, работа со стартовым набором STK500.

Микроконтроллеры AVR фирмы Atmel обладают широкими возможностями по вводу и выводу данных. Микроконтроллеры моделей ATx8515 для этих целей имеют четыре параллельных 8-разрядных порта P_x (x=A, B, C, D) и один 3-разрядный порт PE (в модели ATmega8515). Все линии портов могут программироваться на ввод или вывод данных независимо друг от друга и имеют возможность подключения ко всем входам внутренних подтягивающих резисторов сопротивлением 35...120 кОм.

В состав каждого порта P_x входят три регистра с именами DDR_x, PORT_x и PIN_x. В микроконтроллере AT90S8515 регистр PIN_x не имеет аппаратной реализации. Это имя используется для чтения линий интерфейса. На рис. 1 приведена общая структурная схема 8-разрядных портов P_x и структурная схема одного разряда порта P_x.Y (Y=0, 1, ...7) микроконтроллера AT90S8515.

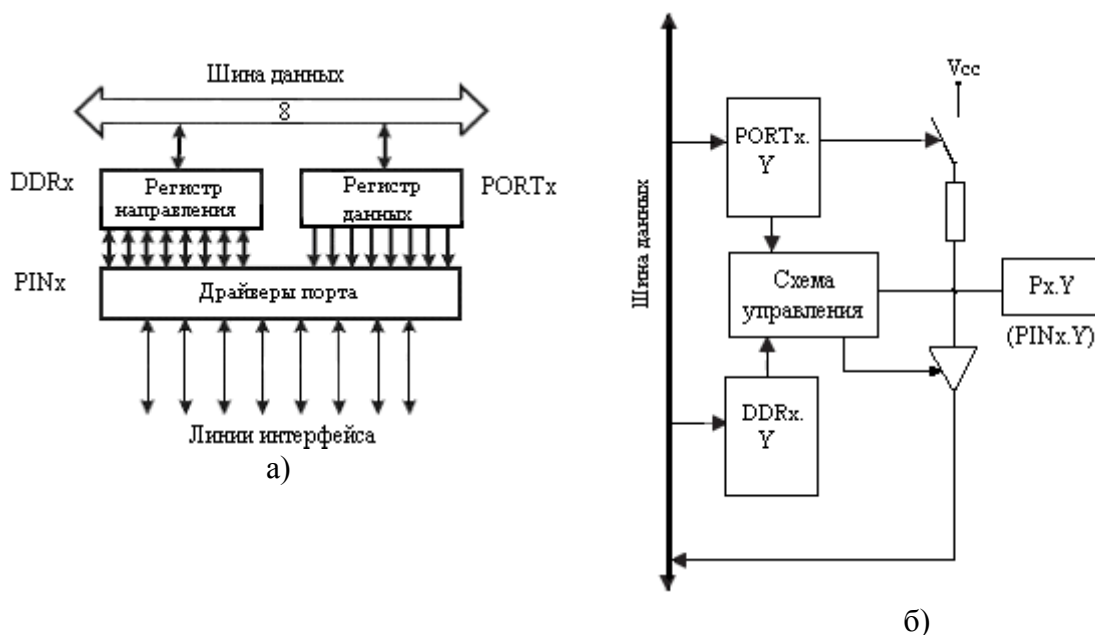


Рис.1. Структура порта P_x (а) и схема одного разряда порта (б)

Состояние разряда DDR_x.Y определяет направление передачи бита данных через вывод порта P_x.Y. При DDR_x.Y=0 вывод порта P_x.Y является входом, при DDR_x.Y=1 – выходом.

В режиме входа состояние разряда PORT_x.Y определяет состояние вывода P_x.Y. При PORT_x.Y=1 вывод порта через внутренний резистор подключается к шине питания Vcc. При PORT_x.Y=0 резистор отключен, вывод P_x.Y находится в высокоимпедансном состоянии (Z-состояние).

В режиме выхода состояние разряда PORTx.Y определяет значение сигнала на выводе Px.Y. При PORTx.Y=0 на выводе устанавливается напряжение низкого уровня, при PORTx.Y=1 - напряжение высокого уровня.

При пуске и перезапуске микроконтроллера все разряды регистров DDRx и PORTx сбрасываются в нулевое состояние, вследствие чего выходы портов работают в режиме входа и находятся в Z – состоянии.

При совместном использовании всех разрядов порта для ввода байта данных используются команды с мнемоникой IN Rd, PINx, для вывода - OUT PORTx, Rr (d, r = 0-31).

Значение выходного сигнала на отдельном выводе порта можно задать с помощью команд установки «0» (CBI PORTx.Y) и установки «1» (SBI PORTx.Y). Значение входного сигнала на отдельном выводе порта можно проверить, используя команды условного перехода SBIC PINx,Y или SBIS PINx,Y, которые предусматривают пропуск следующей команды по нулевому или по единичному значению PINx.Y.

Взаимодействие микроконтроллера с кнопками и светодиодами

Подготовить программу, которая при нажатии кнопки START выполняет поочередное переключение светодиодов, при нажатии кнопки STOP переключение останавливается и возобновляется при повторном нажатии кнопки START.

Пример программы приведен ниже. При программировании микроконтроллера AT90S8515 используется файл определений *8515def.inc*, ATmega8515 - *m8515def.inc*.

В программе линии порта PB использованы для индикации и, следовательно, проинициализированы на вывод, а линии 0 и 1 порта PD, соединяемые с кнопками, - на ввод. Затем ожидается нажатие кнопки START, после чего начинается последовательное переключение светодиодов с задержкой и проверка состояния кнопки STOP.

Программа 1

```

;*****
;Программа 1 для микроконтроллеров ATx8515:
;переключение светодиодов (СД) при нажатии на кнопку START (SW0),
;после нажатия кнопки STOP (SW1) переключение прекращается и
;возобновляется с места остановки при повторном нажатии на кнопку
;START
;*****
.include "8515def.inc"           ;файл определений для AT90S8515
.include "m8515def.inc"        ;файл определений для ATmega8515
.def temp = r16                ;временный регистр
.def reg_led = r20             ;состояние регистра светодиодов
.equ START = 0                 ;0-ой вывод порта PD
.equ STOP = 1                  ;1-ый вывод порта PD

.org $000
    rjmp init
;***Инициализация***
INIT:    ldi reg_led,0xFE        ;сброс reg_led.0 для включения LED0
        sec                    ;C=1
        set                    ;T=1 - флаг направления
        ser temp               ;инициализация выводов
        out DDRB,temp          ; порта PB на вывод
        out PORTB,temp         ;погасить СД
        clr temp               ;инициализация 0-ого и 1-ого выводов
        out DDRD,temp          ; порта PD на ввод
        ldi temp,0x03          ;включение 'подтягивающих'
        out PORTD,temp         ; резисторов порта PD
WAITSTART:

```

```

        sbic PIND,START      ; нажатия
        rjmp WAITSTART      ; кнопки START
LOOP:    out PORTB,reg_led   ; включение СД
;***Задержка (три вложенных цикла)***
        ldi r17,2
d1:      ldi r18,2
d2:      ldi r19,2
d3:      dec r19
        brne d3
        dec r18
        brne d2
        dec r17
        brne d1

        sbic PIND,STOP     ; если нажата кнопка STOP, то
        rjmp MM            ; переход
        rjmp WAITSTART     ; для ожидания нажатия START
MM:      ser temp          ; выключение
        out PORTB,temp     ; светодиодов
        brts LEFT         ; переход, если флаг T установлен
        sbrs reg_led,0    ; пропуск следующей команды,
                          ; если 0-й разряд reg_led установлен
        set               ; T=1 - переключение флага направления
        ror reg_led       ; сдвиг reg_led вправо на 1 разряд
        rjmp LOOP
LEFT:    sbrs reg_led,7    ; пропуск следующей команды,
                          ; если 7-й разряд reg_led установлен
        clt               ; T=0 - переключение флага направления
        rol reg_led       ; сдвиг reg_led влево на 1 разряд
        rjmp LOOP

```

Задание 1. Проверить работу программы в шаговом режиме работы с помощью симулятора AVR Studio 4. Симуляция замыкания и размыкания кнопок START и STOP осуществляется путем установки «0» (белым цветом) и «1» (черным цветом) в маленьких квадратиках порта линий интерфейса PIND. Перед началом прогона программы установите для обеих кнопок состояние логической «1» (кнопки отжаты).

Убедившись в правильной работе программы, введите новые коэффициенты цикла задержки в регистры r17, r18, r19, чтобы длительность цикла задержки составила 0,5с. Проверьте время задержки. Для этого установите контрольные точки (Debug/ Toggle Breakpoint) перед началом выполнения программного блока задержки и после выхода из него. Запустив программу в режиме прогона (Debug/Run) с остановом в контрольных точках, оцените время задержки, контролируя либо показания счетчика циклов Cycle Counter в окне Workspace AVR Studio 4 (вкладка I/O, секция Processor), либо показания Stop Watch.

Выполнив трансляцию программы, загрузите hex-файл в STK500. При программировании STK500 следите, чтобы тип целевого микроконтроллера, установленного на используемой плате, совпадал с типом микроконтроллера в поле Program. В процессе программирования в окне STK500 появляются сообщения о ходе загрузки программы.

Убедившись в правильности загрузки по выводимым сообщениям, проверьте работу программы на макете. Для этого, выключив питание STK500, с помощью 10-проводного шлейфа соединить выводы разъема порта PD с выводами разъема кнопок общего назначения. С помощью второго 10-проводного шлейфа соединить выводы разъема порта

РВ с выводами разъема светодиодов. Включить питание и проверить работу загруженной программы.

Обработка внешних прерываний

В качестве входов внешних прерываний используются входы портов с альтернативной функцией. Это входы PD2, PD3 для прерываний INT0, INT1 и вход PE0 для прерывания INT2 в микроконтроллере ATmega8515. Запросы внешних прерываний INT0, INT1 могут быть представлены низким уровнем сигнала прерывания (L), переходом от высокого уровня сигнала к низкому (HL - по отрицательному фронту), переходом от низкого уровня сигнала к высокому (LH - по положительному фронту), запрос INT2 только переходами (LH) и (HL). В зависимости от типа запроса в регистре управления микроконтроллера MCUCR необходимо установить биты ISCx0 и ISCx1 согласно табл.1 для каждого из прерываний INTx (x=0,1) и определить бит ICS2 в регистре EMCUCR для прерывания INT2. При ICS2=0 прерывание осуществляется по положительному фронту, при ICS2=1 – по отрицательному фронту.

Таблица 1. Таблица выбора типа запроса

IXCx1	ISCx0	Тип запроса
0	0	L
0	1	-
1	0	HL
1	1	LH

Подготовить программу переключения светодиодов с использованием внешнего прерывания от кнопки STOP.

Согласно поставленным требованиям в блок инициализации микроконтроллера внесем ряд изменений:

- добавляем вектор прерываний по адресу \$001;
- указатель стека устанавливаем на последнюю ячейку ОЗУ;
- разрешаем внешнее прерывание INT0 (по сигналу «0» на линии 2 порта PD) и прерывания вообще.

Так как внешнее прерывание INT0 представлено сигналом на входе порта PD2, в качестве кнопки STOP используем кнопку SW2 и программируем PD2 на ввод. Пример программы 2 приведен ниже. Задержка представлена подпрограммой DELAY. Программа работает аналогично программе 1, но нажатие кнопки STOP вызывает прерывание.

Программа 2

```

;*****
;Программа 2 для поочередного переключения светодиодов (СД)
;при нажатии на кнопку START.
;После нажатия на кнопку STOP переключение прекращается и
;возобновляется с места остановки при повторном нажатии
;на кнопку START
;*****
.include "8515def.inc"           ;файл определений для AT90S8515
.include "m8515def.inc"         ;файл определений для ATmega8515
.def temp = r16                 ;временный регистр
.def reg_led = r20              ;состояние регистра светодиодов
.equ START = 0                  ;0-ой вывод порта PD

.org $000
    ;***Векторы прерываний***

```

```

        rjmp INIT                ;обработка сброса
        rjmp STOP_PRESSED       ;обработка внешнего прерывания INT0 (STOP)
;***Инициализация МК***
INIT:    ldi reg_led,0xFE
        ldi temp,$5F           ;установка
        out SPL,temp           ; указателя стека
        ldi temp,$02           ; на последнюю
        out SPH,temp           ; ячейку ОЗУ
        sec                     ;C=1
        set                    ;T=1
        ser temp               ;инициализация выводов
        out DDRB,temp          ; порта PB на вывод
        out PORTB,temp         ;погасить СД
        clr temp               ;инициализация 0-ого и 2-ого выводов
        out DDRD,temp          ; порта PD на ввод
        ldi temp,0x05          ;включение 'подтягивающих'
        out PORTD,temp         ; резисторов порта PD
        ldi temp,0x40          ;разрешение
        out GIMSK,temp        ; прерывания INT0 (6 бит GIMSK или GICR)
        ldi temp,0x00          ;обработка прерывания INT0
        out MCUCR,temp         ; по низкому уровню
        sei                    ;глобальное разрешение прерываний
WAITSTART: sbic PIND,START     ; ожидание нажатия
        rjmp WAITSTART        ; кнопки START
LOOP:    out PORTB,reg_led     ;включение СД
        rcall DELAY           ;задержка
        ser temp              ;выключение
        out PORTB,temp        ; светодиодов
        brts LEFT             ;переход, если флаг T установлен
        sbrs reg_led,0        ;пропуск следующей команды,
        ; если 0-й разряд reg_led установлен
        set                   ;T=1
        ror reg_led           ;сдвиг reg_led вправо на 1 разряд
        rjmp LOOP
LEFT:    sbrs reg_led,7        ;пропуск следующей команды,
        ;если 7-й разряд reg_led установлен
        clt                   ;T=0
        rol reg_led           ;сдвиг reg_led влево на 1 разряд
        rjmp LOOP
;***Обработка прерывания от кнопки STOP***
STOP_PRESSED:
WAITSTART_2:                ;ожидание
        sbic PIND,START       ; нажатия
        rjmp WAITSTART_2     ; кнопки START
        reti
;*** Задержка ***
DELAY:   ldi r17,2
d1:      ldi r18,2
d2:      ldi r19,2
d3:      dec r19
        brne d3
        dec r18
        brne d2
        dec r17
        brne d1
        ret

```

Задание 2. Проверить работу программы с помощью отладчика AVR Studio 4. Установить в регистры r17, r18, r19 коэффициенты, обеспечивающие длительность задержки 0,5с. Запрограммировать микроконтроллер и проверить работу программы.

Задание 3. Изменить программу для включения (выключения) светодиодов в заданной последовательности:

а) 7-6-5-4-3-2-1-0-1-2-3-4-5-6-7-6-5....

б) последовательно по четным позициям, начиная с 6-й,

б) последовательно по нечетным позициям, начиная с 1-й,

в) последовательно увеличивая количество включенных (выключенных) светодиодов до восьми и затем уменьшая до нуля.

Отладив программу, загрузить ее в микроконтроллер и проверить ее работу

Задание 4. Изменить программу, добавив внешнее прерывание INT1 (сигнал на линии PD3, адрес прерывания \$002, бит 7 регистра маски GIMSK или GICR) от кнопки START. Открыв при отладке программы окно памяти данных, проверить работу стека, размещаемого в памяти с адреса \$025F. Отлаженную программу загрузить в микроконтроллер и проверить ее работу.

Контрольные вопросы

1. Какова роль ‘подтягивающих резисторов’? Каким образом можно подключить и отключить резисторы?

2. Какие линии портов микроконтроллера можно использовать для внешних прерываний? Каковы адреса векторов внешних прерываний? Как управлять внешними прерываниями? Какое из внешних прерываний имеет более высокий приоритет?

3. Каким сигналом (логического «0» или логической «1») можно включить светодиод STK500? Как должен выглядеть бит порта PINx.Y в AVR Studio 4 при замыкании (размыкании) кнопки?

4. Как повысить точность программной задержки?