

Цель этой работы состоит в приобретении студентом начальных навыков программирования микроконтроллера и в наблюдении за его работой в процессе взаимодействия с внешними по отношению к нему источниками сигналов (датчиками) и устройствами вывода (исполнительными механизмами).

Работа выполняется на макетной плате, в верхней части которой установлены микроконтроллер (рис. 1) и светодиодная матрица (рис. 2). Слева от микропроцессора находится плата сопряжения, подключаемая к параллельному порту компьютера посредством 14-проводного шлейфа. В средней части макетной платы размещены отдельные компоненты (кнопка, переменный резистор и др.), используемые при выполнении упражнений, предусмотренных заданием.

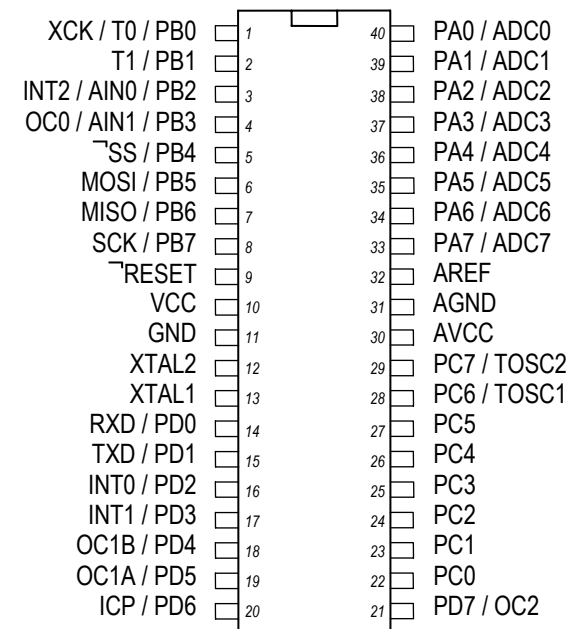


Рис. 1. Микроконтроллер ATmega8535

Используемый в данной работе микроконтроллер описан в литературе [1, 2]. Исчерпывающие сведения приведены в файле **Микроконтроллер ATmega8535 (англ.)** в директории **C:\Labworks\Lab_02**. Программирование микроконтроллера производится на ассемблере в программной среде **AVRStudio4**; подробные сведения об ассемблере можно найти в той же директории в файле **Система команд (англ.).pdf**. Загрузка программы в микроконтроллер осуществляется программой **PonyProg**.

- [1] Донов Г. И. Применение Микроконтроллеров. Учеб. пособие. – М.: МФТИ, 2007.
 [2] Евстифеев А. В. Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL. – М.: Издательский дом «Додэка-XXI», 2006.

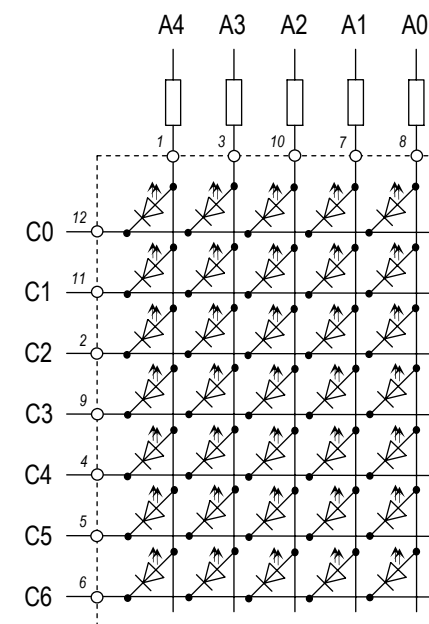


Рис. 2. Светодиодная матрица TA07-11SRWA.

Прежде чем приступить к выполнению упражнений, предусмотренных заданием, *создайте* в директории **c:\Program Files\Atmel\AVR Tools\AVRStudio4\Work** свою индивидуальную директорию, имя которой должно начинаться с номера учебной группы студента и содержать его фамилию латинскими буквами, и *скопируйте* все содержимое директории **c:\Program Files\Atmel\AVR Tools\AVRStudio4\References** в свою индивидуальную директорию.

Подключите 14-проводный шлейф к плате сопряжения на лабораторном макете. **Перед тем как подключить 14-проводный шлейф к плате сопряжения, убедитесь в том, что на макет не поступает никакое напряжение питания от находящегося на лабораторном столе источника питания.**

Порядок выполнения каждого упражнения заключается в следующем: (а) изучается готовая программа; (б) готовая программа загружается в микроконтроллер; (в) наблюдается работа микроконтроллера; (г) видоизменяется готовая программа; (д) производится – в случае необходимости – отладка видоизмененной программы; (е) видоизмененная программа загружается в микроконтроллер; (ж) экспериментально проверяется, что микроконтроллер действует в соответствии с видоизмененной программой.

Упражнение 1: Датчики и исполнительный механизм.

Роль датчиков в этом упражнении играют кнопка, подключенная к входу PD7, и производимое вручную замыкание входа PD6 на землю. Исполнительным механизмом служит светодиодная матрица, столбцы которой нумеруются справа налево, а строки – сверху вниз: светодиод, включенный между i -м столбцом ($i = 0, \dots, 4$) и j -й строкой ($j = 0, \dots, 6$), загорается, когда одновременно подано положительное напряжение (логическая 1) на i -й столбец и равно нулю напряжение (логический 0) на j -й строке.

Запустите программу AVRStudio4 и, поступая в соответствии с инструкцией по работе с AVRStudio4, вызовите из своей индивидуальной директории файл **task_1.asm**. Ознакомьтесь с программой и осуществите преобразование файла **task_1.asm** в файл **task_1.hex**, который может быть загружен в память программ микроконтроллера.

Убедившись, что 14-проводный шлейф подключен к плате сопряжения, включите напряжение питания +5 В. Обратитесь к программе PonyProg: поступая в соответствии с инструкцией по работе с PonyProg, вызовите файл **task_1.hex** и произведите программирование микроконтроллера.

Правильная работа программы **task_1** заключается в следующем.

При замыкании на землю входа PD6 на светодиодную матрицу путем сканирования по строкам выводится изображение, записанное в регистры r16, ..., r22 в виде комбинаций логических 0 и 1. Если вход PD6 не замкнут на землю, то при нажатии на кнопку в результате кратковременной подачи логического 0 на вход PD7 выводится очередная строка того же самого изображения.

Произведя необходимые изменения в программе, осуществите вывод на светодиодную матрицу заданного преподавателем символа (изображения).

task_1.asm

1	.EQU PORTA = \$1b	; здесь вводятся имена
2	.EQU DDRA = \$1a	; для используемых регистров
3	.EQU PORTC = \$15	; ввода-вывода
4	.EQU DDRC = \$14	;
5	.EQU PORTD = \$12	;
6	EQU DDRD = \$11	;
7	.EQU PIND = \$10	;
8		;
9	ldi r16, \$10	; комбинация '0' и '1' в 0-й строке
10	ldi r17, \$02	; комбинация '0' и '1' в 1-й строке
11	ldi r18, \$09	; комбинация '0' и '1' в 2-й строке
12	ldi r19, \$1d	; комбинация '0' и '1' в 3-й строке
13	ldi r20, \$09	; комбинация '0' и '1' в 4-й строке
14	ldi r21, \$02	; комбинация '0' и '1' в 5-й строке
15	ldi r22, \$10	; комбинация '0' и '1' в 6-й строке
16		;

task_1.asm

17		clr	r23	; в r23 хранятся числа; здесь [r23]=\$00
18		out	DDRD, r23	; все разряды порта D - входы
19		ser	r23	; [r23]=\$ff
20		out	DDRA, r23	; все разряды порта A - выходы
21		out	DDRC, r23	; все разряды порта C - выходы
22		out	PORTD,r23	; на всех входах порта D включены
23				; "подтягивающие" резисторы
24		clr	r24	; указатель строки с 0 в 7-м разряде
25		sec		; присвоение '1' флагу 'Carry'
26		clr	r27	; '0' в 8 старших разрядах 'X'
27				;
28	loop:	sbrc	r24, 7	; проверка перехода к 0-й строке
29		rjmp	label1	; переход к строке с номером, не равным 0
30		ldi	r24, \$fe	; установка номера 0 в указателе строки
31		ldi	r26, \$10	; адрес r16 в младшие разряды X
32				;
33	label1:	out	PORTC,r23	; все строки погашены
34		ld	r25, X+	; загрузка в r25 очередной строки
35		out	PORTA,r25	; выдача в PORTA очередной строки
36		out	PORTC,r24	; выдача в PORTC указателя строки
37		rol	r24	; следующее значение указателя строки
38				;
39	label2:	sbis	PIND, 6	; если на входе 6 порта D '0',
40		rjmp	loop	; то периодическое повторение цикла
41		sbic	PIND, 7	; если кнопка не нажата ('1' на входе 7 порта D),
42		rjmp	label2	; то ожидание '0' на входах 6 или 7 порта D
43				;
44	label3:	sbis	PIND, 7	; бесконечный цикл,
45		rjmp	label3	; пока нажата кнопка
46		rjmp	loop	; при отпускании кнопки переход к следующей
47				; строке

Упражнение 1а: «Крестики и нолики».

В результате выполнения этого упражнения учащийся приобретает начальный опыт создания чередующихся изображений на светодиодной матрице. Центральная часть программы **task_1a.asm** реализует точно тот же самый принцип сканирования изображения по строкам светодиодной матрицы, какой был применен в упражнении 1 [строки 28–37 в **task_1.asm** повторены в **task_1a.asm** (строки 36–45) с соответствующими предварительными установками].

Процедура поочередной смены изображений состоит из двух фаз. Длительность каждой фазы определяется временем прохождения трех вложенных циклов, начинающихся с метки **loop**;, то есть временем перебора всех возможных значений в счетчиках, реализованных на регистрах r13, r14 и r15 (строки 47–52). В частности, время, в течение которого одно и то же изображение удерживается на светодиодной матрице, зависит от значения константы, добавляемой к предыдущему содержимому регистра r15 в строке 51. Вход в систему вложенных циклов происходит либо после записи в регистры r16–r22 образов строк символа «крестик», начиная с метки **sign_x**;, либо после записи в эти регистры образов строк символа «нолик», начиная с метки **sign_o**;;

У студента не должно вызывать затруднений такое видоизменение программы **task_1a.asm** (по собственной инициативе или по указанию преподавателя), чтобы число чередующихся изображений, сами изображения и темп, в котором они сменяют друг друга, были бы другими, нежели при работе с исходным текстом этой программы.

task_1a.asm

1	.EQU PORTA = \$1b	; здесь вводятся имена
2	.EQU DDRA = \$1a	; для используемых регистров
3	.EQU PORTC = \$15	; ввода-вывода
4	.EQU DDRC = \$14	;
5	.EQU PORTD = \$12	;
6	.EQU DDRD = \$11	;
7	.EQU PIND = \$10	;
8		;
9	clr r23	; в r23 хранятся числа; здесь r23=\$00
10	out DDRD, r23	; все разряды порта D - входы
11	ser r23	; r23=\$ff
12	out DDRA, r23	; все разряды порта A - выходы
13	out DDRC, r23	; все разряды порта C - выходы
14	out PORTD, r23	; на всех входах порта D включены
15		; "подтягивающие" резисторы
16	clr r27	; '0' в 8 старших разрядах регистра X
17	sec	; присвоение '1' флагу 'Carry'
18	ldi r24, \$7f	; указатель строки с '0' в 7-м разряде
19		;
20	clr r15	; обнуление счетчиков
21	clr r14	;
22	clr r13	;

task_1a.asm

23	ldi	r31, 32	; в дальнейшем (в строке 51) константа 32
24			; добавляется в r15 и тем самым замедляет
25			; переход от одного изображения к другому
26			;
27	sign_x:	set	; T <- '1'
28	ldi	r16, \$00	; комбинация '0' и '1' в 0-й строке
29	ldi	r17, \$11	; комбинация '0' и '1' в 1-й строке
30	ldi	r18, \$0a	; комбинация '0' и '1' в 2-й строке
31	ldi	r19, \$04	; комбинация '0' и '1' в 3-й строке
32	ldi	r20, \$0a	; комбинация '0' и '1' в 4-й строке
33	ldi	r21, \$11	; комбинация '0' и '1' в 5-й строке
34	ldi	r22, \$00	; комбинация '0' и '1' в 6-й строке
35			;
36	loop:	sbrc	r24, 7 ; проверка перехода к 0-й строке
37		rjmp	label1 ; переход к строке с номером, не равным 0
38	ldi	r24, \$fe	; установка номера 0 в указателе строки
39	ldi	r26, \$10	; адрес r16 в младшие разряды X
40			;
41	label1:	out	PORTC, r23 ; все строки погашены
42	ld	r25, X+	; загрузка в r25 очередной строки
43	out	PORTA, r25	; выдача в PORTA очередной строки
44	out	PORTC, r24	; выдача в PORTC указателя строки
45	rol	r24	; следующее значение указателя строки
46			;
47	inc	r13	;
48	brne	loop	;
49	inc	r14	;
50	brne	loop	;
51	add	r15, r31	;
52	brne	loop	;
53			;
54	brtc	sign_x	;
55			;
56	sign_o:	clt	; T <- '0'
57	ldi	r16, \$00	; комбинация '0' и '1' в 0-й строке
58	ldi	r17, \$0e	; комбинация '0' и '1' в 1-й строке
59	ldi	r18, \$11	; комбинация '0' и '1' в 2-й строке
60	ldi	r19, \$11	; комбинация '0' и '1' в 3-й строке
61	ldi	r20, \$11	; комбинация '0' и '1' в 4-й строке
62	ldi	r21, \$0e	; комбинация '0' и '1' в 5-й строке
63	ldi	r22, \$00	; комбинация '0' и '1' в 6-й строке
64			;
65	rjmp	loop	;

Упражнение 2: Таймер и прерывания.

В этом упражнении с помощью 16-разрядного таймера 1 измеряется период, с которым следуют прямоугольные импульсы, подаваемые извне на вход INT1 (PD3). Переходы из 0 в 1 в этом сигнале вызывают прерывания, которые поочередно запускают счет в таймере и останавливают его. Результат счета выводится в шестнадцатеричном виде на 4 нижние строки светодиодной матрицы. При переполнении счетчика происходит внутреннее прерывание и загораются светодиоды в левом столбце.

Обратитесь с помощью AVRStudio4 к программе **task_2.asm** из своей индивидуальной директории, осуществите преобразование файла **task_2.asm** в файл **task_2.hex** и с помощью программы PonyProg загрузите его в микроконтроллер. Подайте на 17-й вывод микросхемы прямоугольные колебания, принимающие значения 0 В и +4.5 В; при подключении к макетной плате лабораторного генератора используйте имеющиеся на плате соединения; частоту колебаний предстоит изменять в пределах от долей герца до нескольких сотен килогерц.

Сравнивая период подаваемых извне колебаний с показаниями таймера, определите частоту тактового сигнала микроконтроллера. Найдите место в программе **task_2.asm**, где устанавливается коэффициент деления частоты тактового сигнала таймера 1, измените значение этого коэффициента, скомпилируйте заново файл с расширением **.hex** и загрузите его в микроконтроллер. Убедитесь экспериментально в правильности определения периода следования подаваемых извне импульсов в новом временном масштабе.

task_2.asm

1	.EQU SREG = \$3f	;
2	.EQU SPH = \$3e	;
3	.EQU SPL = \$3d	;
4	.EQU GICR = \$3b	; то же, что GIMSK
5	.EQU TIMSK = \$39	;
6	.EQU MCUCR = \$35	;
7	.EQU TCCR1B= \$2e	;
8	.EQU TCNT1H= \$2d	;
9	.EQU TCNT1L= \$2c	;
10	.EQU PORTA = \$1b	;
11	.EQU DDRA = \$1a	;
12	.EQU PORTC = \$15	;
13	.EQU DDRC = \$14	;
14	.EQU PORTD = \$12	;
15	.EQU DDRD = \$11	;
16		;
17	rjmp reset	;
18	reti	;
19	rjmp int1	;
20	reti	;

task_2.asm

21		reti	;
22		reti	;
23		reti	;
24		reti	;
25		rjmp	tov1 ;
26		reti	;
27		reti	;
28		reti	;
29		reti	;
30		reti	;
31		reti	;
32		reti	;
33		reti	;
34		reti	;
35		reti	;
36		reti	;
37		reti	;
38			;
39	reset:	clr	r0 ; в r0 '0' во всех разрядах
40		ldi	r23, \$ff ; здесь и всюду далее r23 используется для
41			; кратковременного хранения промежуточного
42			; результата
43		mov	r1, r23 ; в r1 '1' во всех разрядах
44			;
45		clr	r2 ; r2 - "переключатель" режимов работы таймера 1
46			; при [r2]=\$ff - счет, при [r2]=\$00 - останов счета
47			;
48		ldi	r23, \$0f ; в r7 '1' в 4-х младших разрядах
49		mov	r7, r23 ; – это маска младшего полубайта
50		ldi	r23, \$80 ;
51		mov	r8, r23 ; в r8 '1' в 7-м разряде
52			;
53		ldi	r23, \$f7 ; в r12, ..., r15 находятся указатели строк
54		mov	r12, r23 ; светодиодной матрицы с 3-й по 6-ю
55		ldi	r23, \$ef ;
56		mov	r13, r23 ;
57		ldi	r23, \$df ; в дальнейшем в регистрах r19, ..., r22
58		mov	r14, r23 ; помещаются комбинации '0' и '1', выводимые
59		ldi	r23, \$bf ; на 3-ю, ..., 6-ю строки светодиодной матрицы
60		mov	r15, r23 ; соответственно
61			;
62		ldi	r23, \$02 ; инициализация стека:
63		out	SPH, r23 ; дно стека помещается
64		ldi	r23, \$5f ; в самый конец
65		out	SPL, r23 ; памяти данных

task 2.asm

66				
67	out	DDRD, r0		; все разряды порта D - входы
68	out	DDRA, r1		; все разряды порта A - выходы
69	out	DDRC, r1		; все разряды порта C - выходы
70	out	PORTD, r1		; на всех входах порта D включены
71				; "подтягивающие" резисторы
72				;
73	ldi	r23, \$0c		; присвоение битам ISC11 и ISC10 (3-му и 2-му
74	out	MCUCR, r23		; разрядам регистра MCUCR) комбинации
75				; значений '11' означает, что прерывание INT1
76				; будет происходить по нарастающему фронту
77				; сигнала на входе INT1
78				;
79	out	GICR, r8		; запись '1' в 7-й разряд регистра GICR означает
80				; разрешение прерываний по входу INT1
81				;
82	ldi	r23, \$04		; запись '1' во 2-й разряд регистра TIMSK означает
83	out	TIMSK, r23		; разрешение прерываний по переполнению
84				; таймера 1
85				;
86	out	TCCR1B, r0		; останов счета в таймере 1
87	out	TCNT1H, r0		; обнуление старшего и младшего байтов
88	out	TCNT1L, r0		; счетчика в таймере 1
89				;
90	sei			; запись '1' в 7-й разряд регистра SREG означает
91				; глобальное разрешение прерываний
92				;
93	loop:	out	PORTC, r1	; цикл сканирования светодиодной матрицы
94		out	PORTA, r19	;
95		out	PORTC, r12	;
96		out	PORTC, r1	;
97		out	PORTA, r20	;
98		out	PORTC, r13	;
99		out	PORTC, r1	;
100		out	PORTA, r21	;
101		out	PORTC, r14	;
102		out	PORTC, r1	;
103		out	PORTA, r22	;
104		out	PORTC, r15	;
105		rjmp	loop	;
106				;
107	int1:	eor	r2, r1	; обработка прерывания по входу INT1 начинается
108		breq	stop	; с проверки значения «переключателя» r2: если
109				; до выполнения команды eor в r2 были все '1', то
110				; переход к метке stop, то есть останов счета

task_2.asm

111			
112	ldi	r23, \$01	; присвоение битам CS12, CS11 и CS10 (три млад-
113	out	TCCR1B, r23	; ших разряда регистра TCCR1B) какой-либо одной
114			; из комбинаций значений от '001' до '101' означает
115			; запуск счета в таймере 1 с одновременным
116			; указанием частоты сигнала на входе счетчика
117			; таймера 1 в долях тактовой частоты clk _{IO}
118			;
119	reti		; возврат из прерывания по INT1, начало счета
120			;
121	stop:	out TCCR1B, r0	; обнуление 3-х младших разрядов в регистре
122			; TCCR1B означает останов счета в таймере 1
123			;
124	in	r5, TCNT1L	; младший байт таймера 1 запоминается в r5
125	in	r6, TCNT1H	; старший байт таймера 1 запоминается в r6
126			
127	mov	r19, r6	; в этом блоке команд осуществляется перепись
128	swap	r19	; содержимого счетчика TCNT1H:TCNT1L
129	and	r19, r7	; в регистры r19, ..., r22, отведенные для хранения
130	mov	r20, r6	; "образов" 3-й, ..., 6-й строк светодиодной
131	and	r20, r7	; матрицы, с разбиением на полубайты:
132	mov	r21, r5	; результат измерения интервала времени
133	swap	r21	; между соседними прерываниями по входу INT1
134	and	r21, r7	; представлен в строках светодиодной матрицы
135	mov	r22, r5	; сверху вниз от старших разрядов к младшим
136	and	r22, r7	; в шестнадцатеричной записи
137			
138	out	TCNT1H, r0	; обнуление старшего и младшего байтов
139	out	TCNT1L, r0	; счетчика в таймере 1
140			;
141	reti		; возврат из процедуры обработки прерывания
142			; int1, счет остановлен
143			;
144	tov1:	ldi r19, \$10	; прерывание при переполнении таймера
145		ldi r20, \$10	;
146		ldi r21, \$10	;
147		ldi r22, \$10	;
148			;
149	out	TCCR1B, r0	; возврат из прерывания tov1 с остановом счета
150	out	TCNT1H, r0	;
151	out	TCNT1L, r0	;
152	clr	r2	;
153	reti		;

Упражнение 3: Аналого-цифровое преобразование.

К входу ADC7 (PA7) подведено постоянное напряжение с движка потенциометра, включенного в средней части макетной платы между землей и напряжением питания +5 В. На правый столбец светодиодной матрицы выводятся в двоичном виде 7 старших разрядов результата аналого-цифрового преобразования; младший из выводимых разрядов – вверху, старший – внизу.

Обратитесь с помощью AVRStudio4 к программе **task_3.asm** из своей индивидуальной директории, осуществите преобразование файла **task_3.asm** в файл **task_3.hex** и с помощью программы PonyProg загрузите его в микроконтроллер. Убедитесь в том, что при повороте ручки потенциометра происходит соответствующее изменение двоичного числа, выводимого на светодиодный индикатор.

Для нескольких положений движка потенциометра измерьте вольтметром или с помощью цифрового осциллографа значения постоянного напряжения, подаваемого на вход аналого-цифрового преобразователя, и зафиксируйте соответствующие этим значениям показания, выводимые на светодиодную матрицу. По результатам наблюдений определите шаг квантования ΔU вдоль оси напряжений при реализуемом способе преобразования.

task_3.asm

1	.EQU SREG = \$3f ;
2	.EQU SPH = \$3e ;
3	.EQU SPL = \$3d ;
4	.EQU PORTA = \$1b ;
5	.EQU DDRA = \$1a ;
6	.EQU PORTC = \$15 ;
7	.EQU DDRC = \$14 ;
8	.EQU ADMUX = \$07 ;
9	.EQU ADCSRA = \$06 ;
10	.EQU ADCH = \$05 ;
11	
12	rjmp reset ;
13	reti ;
14	reti ;
15	reti ;
16	reti ;
17	reti ;
18	reti ;
19	reti ;
20	reti ;
21	reti ;
22	reti ;
23	reti ;
24	reti ;
25	reti ;

task_3.asm

26	rjmp	adc_end	;
27	reti		;
28	reti		;
29	reti		;
30	reti		;
31	reti		;
32	reti		;
33			;
34	reset:	ldi r23, \$02	; инициализация стека:
35		out SPH, r23	; дно стека помещается
36		ldi r23, \$5f	; в самый конец
37		out SPL, r23	; памяти данных
38			;
39		ldi r23, \$7f	; разряды порта A с 0-го по 6-й - выходы,
40		out DDRA, r23	; 7-й разряд порта A - вход;
41		out DDRC, r23	; разряды порта C с 0-го по 6-й - выходы
42			;
43		ldi r23, \$67	; значениями '01' в 7-м и 6-м разрядах
44		out ADMUX, r23	; регистра ADMUX в качестве источника
45			; опорного напряжения выбирается AVCC;
46			; '1' в 5-м разряде регистра ADMUX
47			; помещает 8 старших разрядов результата
48			; преобразования в регистр ADCH;
49			; значениями '00111' в разрядах регистра ADMUX
50			; с 4-го по 0-й входом преобразователя
51			; назначается 7-й разряд порта A
52			;
53		ldi r23, \$ef	; '1' в разрядах регистра ADCSRA с 7-го по 3-й:
54		out ADCSRA, r23	; 7-й разряд - разрешение преобразования;
55			; 6-й разряд - запуск преобразования;
56			; 5-й разряд - автоповтор преобразования;
57			; 3-й разряд - разрешение прерывания
58			; в момент окончания преобразования;
59			; значениями '111' в разрядах регистра ADCSRA
60			; со 2-го по 0-й устанавливается равный 128
61			; коэффициент деления частоты сигнала
62			; на тактовом входе преобразователя
63			;
64		ldi r23, \$ff	;
65		mov r1, r23	; '1' во всех разрядах регистра r1
66		ldi r23, \$01	;
67		out PORTA, r23	; выбор 0-го столбца светодиодной матрицы
68			;
69		sei	; разрешение прерываний
70			;

task 3.asm

71	loop:	rjmp	loop	; бесконечный цикл
72				;
73	adc_end:	in	r23, ADCH	; прерывание в момент окончания аналого-
74				; цифрового преобразования, чтение 8 старших
75				; разрядов результата преобразования
76				;
77		eor	r23, r1	; представление в обратном коде
78		lsr	r23	; сдвиг вправо на 1
79		out	PORTC, r23	; выдача результата в 0-й столбец
80		reti		; выход из прерывания

Упражнение 3а:

Преобразование напряжения в частоту прямоугольных колебаний.

В микроконтроллере нетрудно осуществить согласованную параллельную работу различных функциональных блоков. В данном упражнении преобразование постоянного напряжения в цифровой эквивалент, реализуемое так, как это делается в упражнении 3, служит источником управляющего воздействия на работу таймера 1 в режиме генерирования прямоугольных колебаний на специально предназначенном для этого выводе.

В программе **task_3a.asm** результат АЦП, находящийся в регистре ADCH, запоминается в качестве старшего байта OCR1AH порога А таймера 1 (младший байт этого порога OCR1AL ради простоты заполняется единицами). Таймер 1 устанавливается в режим непрерывного счета, однако в момент, когда текущее содержимое счетчика становится равным значению, хранящемуся в пороге А, счетчик сбрасывается до нуля, значение логического сигнала, вырабатываемого на выходе OC1A (вывод 19), меняется на противоположное и счет возобновляется.

Частота прямоугольного колебания, наблюдаемого с помощью осциллографа на выводе 19, оказывается напрямую зависящей от напряжения, поступающего с движка потенциометра на вход ADC7 (вывод 33) аналого-цифрового преобразователя. Период прямоугольного колебания на выходе OC1A (вывод 19) линейно зависит от величины преобразуемого постоянного напряжения, причем коэффициент пропорциональности можно менять, задавая различную частоту тактового сигнала на входе счетчика таймера 1.

task_3a.asm

1	.EQU SREG = \$3f	;
2	.EQU SPH = \$3e	;
3	.EQU SPL = \$3d	;
4	.EQU TCCR1A= \$2f	; управляющий регистр А таймера 1
5	.EQU TCCR1B= \$2e	; управляющий регистр В таймера 1
6	.EQU OCR1AH= \$2b	; старший байт порога А в таймере 1
7	.EQU OCR1AL= \$2a	; младший байт порога А в таймере 1
8	.EQU PORTA = \$1b	;
9	.EQU DDRA = \$1a	;
10	.EQU PORTC = \$15	;
11	.EQU DDRC = \$14	;
12	.EQU DDRD = \$11	;
13	.EQU ADMUX = \$07	;
14	.EQU ADCSRA = \$06	;
15	.EQU ADCH = \$05	;
16		;
17	rjmp reset	;
18	reti	;
19	reti	;
20	reti	;

task_3a.asm

21		reti		;
22		reti		;
23		reti		;
24		reti		;
25		reti		;
26		reti		;
27		reti		;
28		reti		;
29		reti		;
30		reti		;
31		rjmp	adc_end	;
32		reti		;
33		reti		;
34		reti		;
35		reti		;
36		reti		;
37		reti		;
38				;
39	reset:	ldi	r23, \$02	; инициализация стека:
40		out	SPH, r23	; дно стека помещается
41		ldi	r23, \$5f	; в самый конец
42		out	SPL, r23	; памяти данных
43				;
44		ldi	r23, \$7f	; разряды порта A с 0-го по 6-й - выходы,
45		out	DDRA, r23	; 7-й разряд порта A - вход;
46		out	DDRC, r23	; разряды порта C с 0-го по 6-й - выходы
47				;
48		ldi	r23, \$67	; значениями '01' в 7-м и 6-м разрядах
49		out	ADMUX, r23	; регистра ADMUX в качестве источника
50				; опорного напряжения выбирается AVCC;
51				; '1' в 5-м разряде регистра ADMUX
52				; помещает 8 старших разрядов результата
53				; преобразования в регистр ADCH;
54				; значениями '00111' в разрядах регистра ADMUX
55				; с 4-го по 0-й входом преобразователя
56				; назначается 7-й разряд порта A
57				;
58		ldi	r23, \$ef	; '1' в разрядах регистра ADCSRA с 7-го по 3-й:
59		out	ADCSRA, r23	; 7-й разряд - разрешение преобразования;
60				; 6-й разряд - запуск преобразования;
61				; 5-й разряд - автоповтор преобразования;
62				; 3-й разряд - разрешение прерывания
63				; в момент окончания преобразования;
64				; значениями '111' в разрядах регистра ADCSRA
65				; со 2-го по 0-й устанавливается равный 128

task_3a.asm

66				; коэффициент деления частоты сигнала
67				; на тактовом входе преобразователя
68				;
69	ldi	r23, \$ff		;
70	mov	r1, r23		; '1' во всех разрядах регистра r1
71	ldi	r23, \$01		;
72	out	PORTA, r23		; выбор 0-го столбца светодиодной матрицы
73				;
74	sei			; разрешение прерываний
75				;
76	ldi	r16, \$ff		;
77	out	DDRD, r16		; все выводы порта D - выходы
78				;
79	ldi	r16, \$40		; биты COM1A1 и COM1A0 (7-й и 6-й разряды в
80	out	TCCR1A, r16		; регистре TCCR1A) со значениями '01' означают
81				; переключение сигнала на выходе OC1A (PD5)
82				; в момент совпадения текущего значения
83				; в таймере 1 с порогом A; биты WGM11 и WGM10
84				; (1-й и 0-й разряды регистра TCCR1A) – младшие
85				; 2 бита в 4-битовом номере режима работы
86				; таймера 1 (здесь им присваиваются нулевые
87				; значения)
88				:
89	ldi	r16, \$09		; биты WGM13 и WGM12 (4-й и 3-й разряды
90	out	TCCR1B, r16		; регистра TCCR1B) – старшие 2 бита в 4-битовом
91				; номере режима работы таймера 1 (здесь им
92				; присваиваются значения '01'); с учетом
93				; сказанного выше битами WGM13, WGM12,
94				; WGM11 и WGM10 устанавливается 4-й режим
95				; работы таймера 1 (CTC); значениями битов
96				; CS12, CS11 и CS10 (2-й, 1-й и 0-й разряды
97				; регистра TCCR1B) выбирается частота тактового
98				; сигнала на счетном входе таймера 1 (в данном
99				; случае комбинацией значений '001' этих битов
100				; выбрана наибольшая частота clk _{IO})
101				;
102	loop:	rjmp	loop	; бесконечный цикл
103				;
104	adc_end:	in	r23, ADCH	; прерывание в момент окончания аналого-
105				; цифрового преобразования, чтение 8 старших
106				; разрядов результата преобразования
107				;
108	eor	r23, r1		; представление в обратном коде
109	lsl	r23		; сдвиг вправо на 1
110	out	PORTC, r23		; выдача результата в 0-й столбец

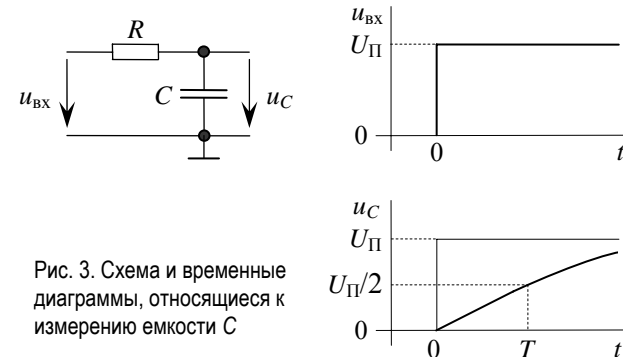
task_3a.asm

111	out	OCR1AH, r23	; результат АЦП вставляется в старший байт
112			; OCR1AH порога А таймера 1
113	ldi	r16, \$ff	;
114	out	OCR1AL, r16	; в младшем байте OCR1AL порога А таймера 1
115			; максимальное значение
116	reti		; выход из прерывания

Упражнение 4а:

Пример применения микроконтроллера для измерений.

На последовательно включенные R и C подается скачок напряжения (рис. 3) и по времени нарастания напряжения на конденсаторе от 0 до значения, равного половине напряжения скачка, определяется его емкость.



Скачок напряжения на входе RC -цепи, равный, грубо говоря, напряжению питания $U_{П}$, создается в результате вывода на выход (вывод 19) логической '1' после того, как в течение некоторого времени напряжение на этом выводе, а также на самом конденсаторе C , удерживается равным '0'. Определение момента достижения напряжением u_C значения, равного половине напряжения питания $U_{П}$, осуществляется с помощью встроенного в микроконтроллер аналогового компаратора. Напряжение u_C подводится к неинвертирующему входу компаратора AIN0 (PB2, вывод 3); напряжение $U_{П}/2$ образовано делителем из двух резисторов с одинаковыми сопротивлениями и подано на другой из входов компаратора AIN1 (PB3, вывод 4).

В средней части макета (слева иверху по отношению к потенциометру, снизу по отношению к плате сопряжения с компьютером) помещен предусмотренный схемой (рис. 3) горизонтально расположенный резистор R с сопротивлением, равным 51 кОм. Правый вывод этого резистора соединен с 19-м выводом микроконтроллера (выход PD5), а левый вывод резистора – с 3-м выводом микроконтроллера (входом/выходом PB2). Конденсатор, емкость которого измеряется в данном упражнении, следует включать между левым выводом резистора R и ближайшей землей.

Программой **task_4a**, реализующей измерение емкости конденсатора, предусмотрено переключение функции, выполняемой выводом 3 микроконтроллера. В пределах отрезка времени от 0 до T , отмеченного на нижней временной диаграмме (рис. 3), этот вывод назначен входом и играет роль одного из входов аналогового компаратора (AIN0), к которому подведено напряжение на верхней (на рисунке) обкладке конденсатора C . В другое время этот вывод становится выходом PB2, на котором устанавливается нулевое напряжение (точнее, выводится логический '0'), чтобы обеспечить разрядку конденсатора перед очередным измерением. Между верхней обкладкой конденсатора и 3-м выводом микрокон-

троллера включен резистор с сопротивлением порядка 300 Ом для ограничения тока, который в процессе разрядки конденсатора будет течь по этому выводу микроконтроллера.

Первое измерение осуществляется программой **task_4a**, сразу после включения питания. В дальнейшем, для того чтобы произвести очередное измерение, необходимо нажать на кнопку, то есть с помощью кнопки подать '0' на вход PD7, как в упражнении 1. До нажатия кнопки в следующий раз на светодиодной матрице удерживается результат предыдущего измерения.

Значение времени T определяется результатом счета тактовых импульсов таймером 1. Счетчик таймера 1 начинает свою работу в момент выдачи скачка напряжения на вывод 19 (выход PD5 порта D). При этом частота тактового сигнала clk с самого начала выбирается максимальной, то есть равной $clk_{I/O}$. Если до момента срабатывания аналогового компаратора, когда возникает прерывание a_comp , счетчик таймера 1 не оказывается переполненным, то на этом процедура определения значения T заканчивается и результат счета выводится в шестнадцатеричном виде на нижние 4 строки светодиодной матрицы точно так же, как это делается в упражнении 2.

Если до срабатывания компаратора происходит переполнение счетчика, то производится переключение делителя частоты тактового сигнала на счетном входе таймера 1 на меньшее значение. Переход ко все меньшим значениям частоты тактового сигнала на входе таймера 1 происходит до тех пор, пока счет не заканчивается в момент срабатывания аналогового компаратора в отсутствие переполнения.

Когда измерение заканчивается, то помимо результата счета, выводимого в шестнадцатеричном виде на 4 нижние строки светодиодной матрицы, в 3 младших разряда верхней строки матрицы помещаются значения битов CS12, CS11 и CS10 регистра TCCR1B, значениями которых определяется частота clk тактового сигнала на входе таймера 1 к моменту окончания измерения. В микроконтроллере аппаратно установлена следующая связь между значениями битов CS12, CS11 и CS10 и частотой тактового сигнала clk (предполагается, что значение $clk_{I/O}$ (I/O Clock) известно из упражнения 2):

CS12	CS11	CS10	Выбор тактового сигнала
0	0	0	Нет счета (останов)
0	0	1	$clk = clk_{I/O}$
0	1	0	$clk = clk_{I/O}/8$
0	1	1	$clk = clk_{I/O}/64$
1	0	0	$clk = clk_{I/O}/256$
1	0	1	$clk = clk_{I/O}/1024$
1	1	0	От T1 / PB1 / [2] Negative Edge
1	1	1	От T1 / PB1 / [2] Positive Edge

Пусть N – показания 16-разрядного счетчика таймера 1, а $t_0 = 1 / clk$ – период тактового сигнала на счетном входе таймера 1 на момент срабатывания аналогового компаратора. Тогда $T = N \cdot t_0$ и из равенства

$$U_{\Pi}/2 = U_{\Pi} \left(1 - e^{-T/(RC)} \right)$$

следует, что $C \approx T / (0.693 \cdot R)$, где $0.693 \approx -\ln 0.5$.

task_4a.asm

1	.EQU SPH = \$3e		; старший байт указателя стека
2	.EQU SPL = \$3d		; младший байт указателя стека
3	.EQU TIMSK = \$39		; маски прерываний таймеров
4	.EQU TIFR = \$38		; флаги прерываний таймеров
5	.EQU TCCR1B= \$2e		; второй из регистров управления таймером 1
6	.EQU TCNT1H= \$2d		; старший байт таймера 1
7	.EQU TCNT1L= \$2c		; младший байт таймера 1
8	.EQU PORTA = \$1b		;
9	.EQU DDRA = \$1a		;
10	.EQU PORTB = \$18		;
11	.EQU DDRB = \$17		;
12	.EQU PORTC = \$15		;
13	.EQU DDRC = \$14		;
14	.EQU PORTD = \$12		;
15	.EQU DDRD = \$11		;
16	.EQU PIND = \$10		;
17	.EQU ACSR = \$08		; регистр управления аналоговым компаратором
18			;
19	rjmp	reset	;
20	reti		;
21	reti		;
22	reti		;
23	reti		;
24	reti		;
25	reti		;
26	reti		;
27	reti		;
28	reti		;
29	reti		;
30	reti		;
31	reti		;
32	reti		;
33	reti		;
34	reti		;
35	rjmp	a_comp	; прерывание при срабатывании аналогового
36			; компаратора
37	reti		;
38	reti		;
39	reti		;
40	reti		;
41			;
42	reset:	clr	r0 ; в r0 нули во всех разрядах
43		ser	r31 ; в r31 единицы во всех разрядах
44		ldi	r16, \$0f ; в r7 маска младшего полубайта
45		mov	r7, r16 ; ('1' в разрядах с 3-го по 0-й)

task 4a.asm

46	ldi	r16, \$8b	; в r8 управление аналоговым компаратором
47	mov	r8, r16	; с '1' в 7-м разряде (аналоговый компаратор
48			; отключен)
49			;
50	ldi	r16, \$fe	; в регистрах r9, ..., r15 помещаются
51	mov	r9, r16	; указатели строк светодиодной матрицы
52	ldi	r16, \$fd	; (с 0-й по 7-ю), поочередно выводимые
53	mov	r10, r16	; на PORTC
54	ldi	r16, \$fb	;
55	mov	r11, r16	; вывод на PORTC содержимого одного из
56	ldi	r16, \$f7	; регистров r9, ..., r15 производится
57	mov	r12, r16	; после того, как на PORTA выведена
58	ldi	r16, \$ef	; комбинация '0' и '1',
59	mov	r13, r16	; хранящаяся к этому моменту времени в
60	ldi	r16, \$df	; в соответствующем из регистров r16, ..., r22,
61	mov	r14, r16	; а именно: r16 в PORTA, затем r9 в PORTC,
62	ldi	r16, \$bf	; сначала r17 в PORTA, затем r10 в PORTC, и т.д.,
63	mov	r15, r16	; сначала r22 в PORTA, затем r15 в PORTC
64			;
65	ldi	r16, \$02	; инициализация стека:
66	out	SPH, r16	; дно стека помещается
67	ldi	r16, \$5f	; в самый конец
68	out	SPL, r16	; памяти данных
69			;
70	out	DDRA, r31	; все выводы порта A выходы
71	out	DDRB, r0	; на время разряда конденсатора вывод PB2
72			; выход, остальные выводы порта B всегда входы
73	out	PORTB, r31	; на всех входах порта B включены
74			; «подтягивающие» резисторы, кроме вывода PB2,
75	cbi	PORTB, 2	; даже в том случае, когда он служит инвер-
76			; тирующим входом аналогового компаратора
77	out	DDRC, r31	; все выводы порта C выходы
78	out	DDRD, r0	; все выводы порта D входы, кроме вывода PD5
79	sbi	DDRD, 5	; вывод PD5 выход
80	out	PORTD, r31	; на всех входах порта D включены
81			; «подтягивающие» резисторы
82			;
83	clr	r10	; регистры r10, r11 и r17 используются для задания
84	clr	r11	; отрезка времени, в течение которого обеспечи-
85	mov	r17, r0	; вается разряд конденсатора, то есть удержание
86	ldi	r18, 16	; конденсатора с нулевым напряжением на нем;
87			; десятичное число 16 является константой, задаю-
88			; щей длительность указанного отрезка времени
89			;
90	sei		; глобальное разрешение прерываний

task_4a.asm

91				;
92	out	TIMSK, r0		; все прерывания таймеров запрещены
93				;
94	home:	clr	r16	; в 3-х младших разрядах r16 коэффициент
95				; деления частоты тактового сигнала таймера1;
96				; при нуле в r16 счет в таймере 1 остановлен
97				;
98	next_step:	inc	r16	; переход к следующему значению тактовой
99				; частоты на входе таймера 1
100	out	ACSR, r8		; аналоговый компаратор отключен
101	sbi	DDRB, 2		; вывод PB2 назначен выходом, на который из
102				; 2-го разряда регистра PORTB подан '0'
103	cbi	PORTD, 5		; на выход PD5 подан '0'
104	out	TCCR1B, r0		; счет в таймере 1 остановлен
105	out	TCNT1H, r0		; обнулен старший байт счетчика в таймере 1
106	out	TCNT1L, r0		; обнулен младший байт счетчика в таймере 1
107				;
108	delay:	inc	r10	; в течение отрезка времени, задаваемого этими
109		brne	delay	; 6-ю операторами, на конденсаторе удерживается
110		inc	r11	; нулевое напряжение
111		brne	delay	;
112		add	r17, r18	;
113		brne	delay	;
114				
115	cbi	ddrb, 2		; вывод PB2 назначен неинвертирующим входом
116				; аналогового компаратора
117	cbi	ACSR, 7		; аналоговый компаратор включен, разрешены
118				; прерывания по нарастающему фронту сигнала
119				; на его выходе
120	sbi	PORTD, 5		; на выход PD5 подана '1' (подан скачок
121				; напряжения на вход интегрирующей цепи)
122	out	TCCR1B, r16		; начат счет в таймере 1 с частотой тактового
123				; сигнала, заданной содержимым r16
124				
125	loop:			: вход в цикл сканирования светодиодной матрицы
126	out	PORTC, r31		; все строки погашены
127	out	PORTA, r16		; в 3-х младших разрядах 0-й строки матрицы
128	out	PORTC, r9		; содержится информация о частоте тактового
129				; сигнала на входе счетчика таймера 1
130	out	PORTC, r31		; все строки погашены
131	out	PORTA, r19		; в 4-ю строку матрицы выводится 1-й (старший)
132	out	PORTC, r12		; полубайт результата счета
133	out	PORTC, r31		; все строки погашены
134	out	PORTA, r20		; в 5-ю строку матрицы выводится 2-й полубайт
135	out	PORTC, r13		; результата счета

task 4a.asm

136		out	PORTC, r31	; все строки погашены
137		out	PORTA, r21	; в 6-ю строку матрицы выводится 3-й полубайт
138		out	PORTC, r14	; результата счета
139		out	PORTC, r31	; все строки погашены
140		out	PORTA, r22	; в 7-ю строку матрицы выводится 4-й (младший)
141		out	PORTC, r15	; полубайт результата счета
142		out	PORTC, r31	; все строки погашены
143				
144		in	r1, TIFR	; проверяется наличие переполнения в таймере 1
145		sbrs	r1, 2	; если флаг переполнения равен '1', то следующая
146				; команда rjmp пропускается
147		rjmp	button	; данное измерение закончено; переход
148				; к следующему измерению по нажатию кнопки
149		set		; вспомогательному разряду Т регистра состояния
150				; SREG задано единичное значение
151		bld	r1, 2	; значение разряда Т регистра SREG помещается
152				; во 2-й разряд регистра r1
153		out	TIFR, r1	; флаг переполнения счетчика в таймере 1 снят
154		rjmp	next_step	; переход к меньшему значению частоты clk
155				; тактового сигнала на входе счетчика таймера 1
156				;
157	button:	sbic	PIND, 7	; проверка, не нажата ли кнопка
158		rjmp	loop	; если кнопка не нажата, то возвращение к началу
159				; цикла сканирования светодиодной матрицы
160		rjmp	home	; если кнопка нажата, то
161				; переход к новому измерению
162				;
163	a_comp:			; обработка прерывания по срабатыванию
164				; аналогового компаратора
165		out	TCCR1B, r0	; останов счета в таймере 1
166		out	ACSR, r8	; отключение аналогового компаратора
167		cbi	PORTD, 5	; снятие единичного сигнала со входа
168				; интегрирующей цепи
169		in	r5, TCNT1L	; младший байт результата счета в таймере 1
170				; записывается в r5
171		in	r6, TCNT1H	; старший байт результата счета в таймере 1
172				; записывается в r6
173		mov	r19, r6	; старший байт результата счета в таймере 1
174				; помещен в r19
175		swap	r19	; в r19 полубайты поменяны местами
176		and	r19, r7	; в r19 оставлены только 4 младших разряда
177		mov	r20, r6	; старший байт результата счета в таймере 1
178				; помещен в r20
179		and	r20, r7	; в r20 оставлены только 4 младших разряда
180		mov	r21, r5	; младший байт результата счета в таймере 1

task_4a.asm

181			; помещен в r21
182	swap	r21	; в r21 полубайты поменяны местами
183	and	r21, r7	; в r21 оставлены только 4 младших разряда
184	mov	r22, r5	; младший байт результата счета в таймере 1
185			; помещен в r22
186	and	r22, r7	; в r22 оставлены только 4 младших разряда
187	reti		; возвращение из прерывания по срабатыванию
188			; аналогового компаратора