МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

по дисциплине «Программирование микроконтроллеров»

тема: «Программирование часов на микроконтроллере MC8051»

Выполнил: ст. группы ВТ-32

Воскобойников И. С.

Проверил: Гольцов Ю. А.

Белгород 2021 г.

**Цель работы**: изучить принципы работы таймеров и организацию прерываний в микроконтроллерах серии MCS-51.

**Задание к работе**:

Создать программу для вывода часов на семисегментный индикатор секунды выводятся с помощью мигания точки.

Таблица соответствия P1 и ASCII

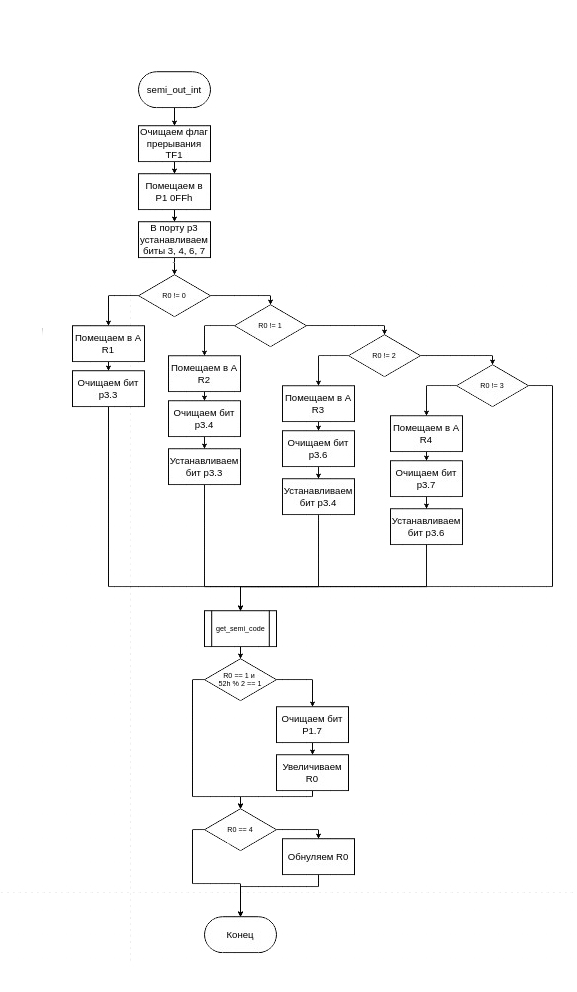
|  |  |
| --- | --- |
| P1 | ASCII |
| 1 1 0 0 0 0 0 0 | 0 |
| 1 0 0 1 0 0 0 0 | 9 |
| 1 0 0 0 0 0 0 0 | 8 |
| 1 1 1 1 1 0 0 0 | 7 |
| 1 0 0 0 0 0 1 0 | 6 |
| 1 0 0 1 0 0 1 0 | 5 |
| 1 0 0 1 1 0 0 1 | 4 |
| 1 0 1 1 0 0 0 0 | 3 |
| 1 0 1 0 0 1 0 0 | 2 |
| 1 1 1 1 1 0 0 1 | 1 |

Блок схемы разработанных подпрограмм

**step\_time\_int**

****

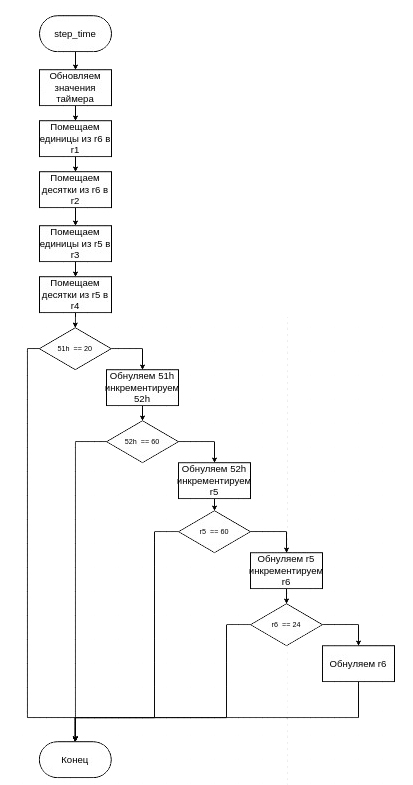
**semi\_out\_int**



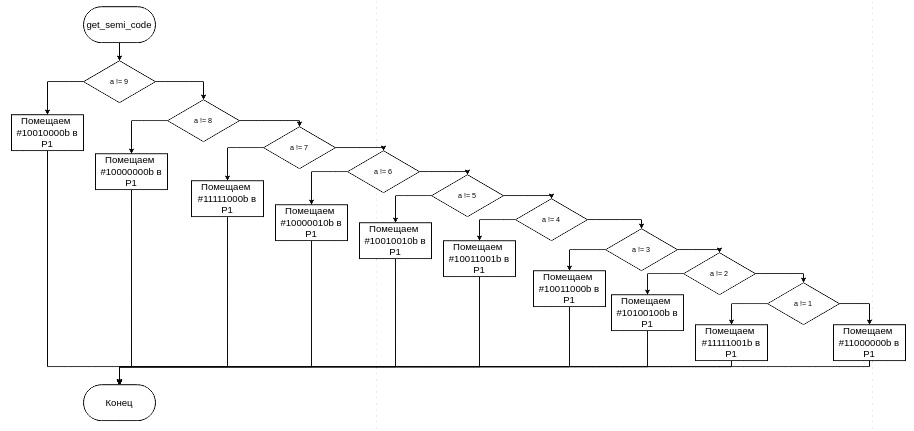
**main**



**step\_time**



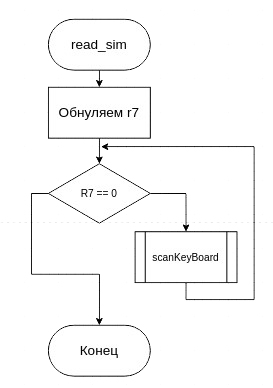
**get\_semi\_code**



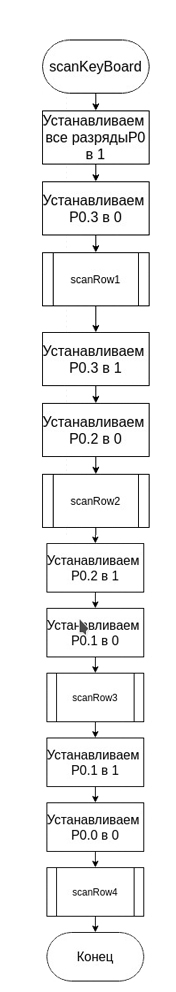
**init\_timer**



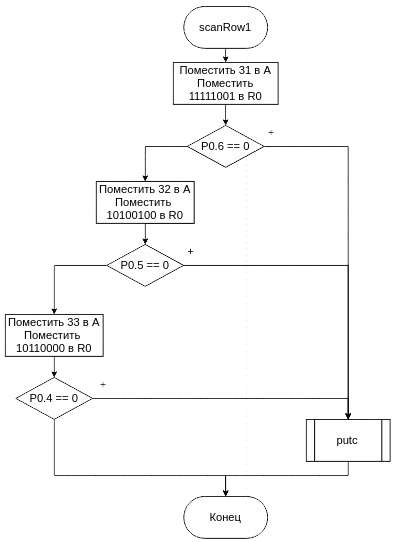
**read\_sim**



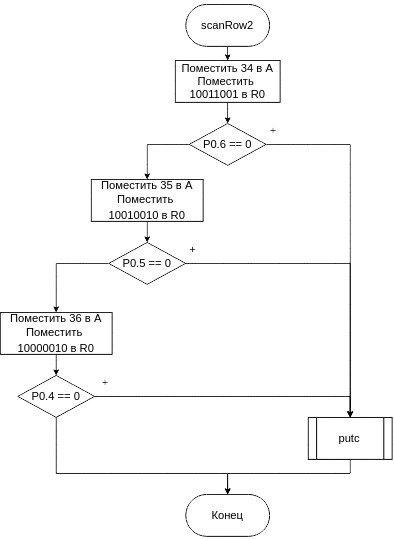
**scanKeyBoard**

****

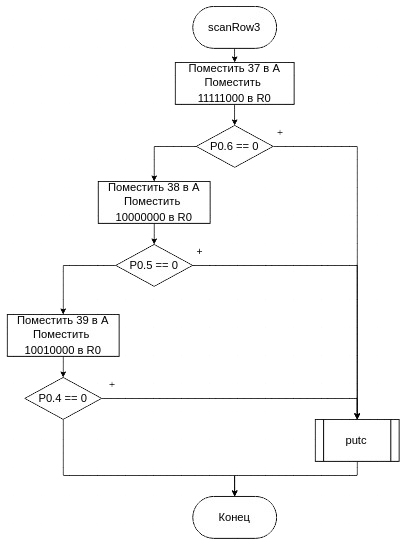
**scanRow1**



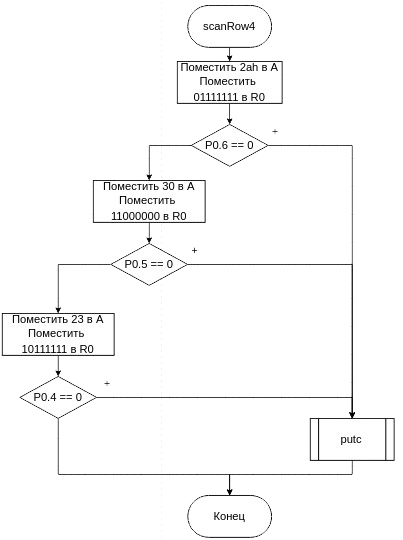
**scanRow2**



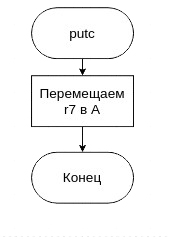
**scanRow3**



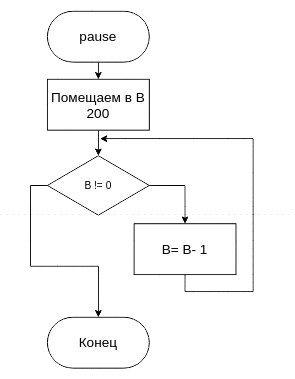
**scanRow4**

****

**putc**

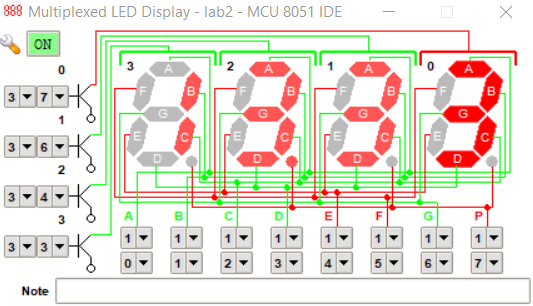
****

**pause**

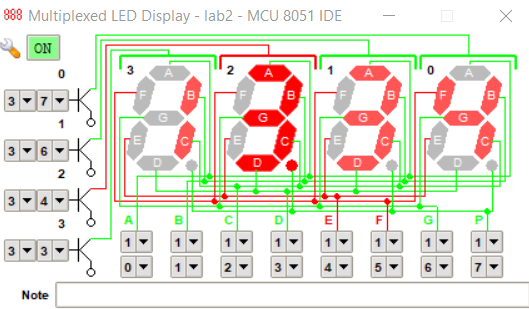


**Пример выполнения работы**

Задано начальное время 13:33



Результат через минуту



**lab2.asm**

ORG 0h

jmp main

step\_time\_int:

ORG 0Bh;Тут прерывание от таймера номер 0

CLR TF0;Очищаем флаг прерывания

call step\_time

RETI ;выход из перывания

semi\_out\_int:

ORG 1Bh;Тут прерывание от таймера номер 1

CLR TF1;Очистка флага прерывания

MOV P1, #0FFh

setb p3.3

setb p3.4

setb p3.6

setb p3.7

;первая лампочка

CJNE R0, #0h, lam\_skip\_1

mov A, r1;берем число соответствующее лампочке

clr p3.3;выбираем лампочку

lam\_skip\_1:

;вторая лампочка

CJNE R0, #1h, lam\_skip\_2

mov A, r2

setb p3.3

clr p3.4

lam\_skip\_2:

CJNE R0, #2h, lam\_skip\_3

mov A, r3

setb p3.4

clr p3.6

lam\_skip\_3:

CJNE R0, #3h, lam\_skip\_4

mov A, r4

setb p3.6

clr p3.7

lam\_skip\_4:

;Заносим цифру в A и вызываем функцию которая отображает ее в P1

call get\_semi\_code

CJNE R0, #1h, term;если вывод на вторую лампочку то там ставится точка, если секунд не четное кол-во

;она получится по центру

MOV A, 52h

MOV B, #2

DIV AB ;делим что бы получить остаток

MOV A, B

;делим на два и смотрим сколько получилось

CJNE A, #1, term ;если есть четное то прыгаем дальше

;иначе моргаем точкой

CLR P1.7

term:

inc R0 ;переходим к следующему сегменту

CJNE R0, #00000100b, end\_semo\_circle ;Если дошли до конца семи сегментного индикатора то обнуляем счетчик лампочек

MOV R0,#0b

end\_semo\_circle:

RETI;Выход из прерывания вывода

main:

;T1 M1=1 M0=0 по прерыванию первого таймера мы выводим в индикатор

;T0 M1=0 M0=1 по этому прерыванию увеличиваем время

MOV TMOD,#00100001b

MOV TH0, #0FFh

MOV TL0, #0E0h

;разрешаем прерывание от таймера 0

SETB ET0

;разрешаем прерывание от таймера 1

SETB ET1

;разрешаем выполнение прерываний

SETB EA

SETB TR1;запускаем таймер 1

;инициализация начальных значений

call init\_timer

SETB TR0;запускаем таймер 0

jmp $;Бесконечный цикл работы

;шаг таймера

step\_time:

MOV TH0, #0FFh;обновляем настройки таймера

MOV TL0, #0E0h

;получаем десятки часов в r1

MOV A, r6

MOV B, #0Ah

DIV AB

MOV r1,A

;получаем часов в r2

MOV r2,B

;получаем десятки минут в r3

MOV A, r5

MOV B, #0Ah

DIV AB

MOV r3,A

;получаем минуты в r4

MOV r4,B

MOV A, 51h;увеличиваем счетчик милисекунд

INC A

MOV 51h, A

CJNE A, #20, OUT ;если не накапала секудна прыгаем дальше

;иначе обнуляем счетчик милисекунд и увеличиваем секунду

MOV 51h, #0

;увеличиваем счетчик секунд

MOV A, 52h

INC A

MOV 52h, A

CJNE A, #60, OUT ;если не получили 60 секунд прыгаем дальше

;иначе обнуляем счетчик и увеличиваем минуты

MOV 52h, 0

INC r5

CJNE r5, #60, OUT;Если не получили 60 минут прыгаем дальше

;Если получили обновляем счетчик

MOV r5, #0

;увеличиваем часы

INC r6

CJNE r6, #24, OUT;Если не получили 24 часа прыгаем дальше

MOV r6, #0;Если получили обновляем счетчик

OUT:

RET

;отображение цифры из А в P1

get\_semi\_code:

cjne a,#9, num8

mov p1,#10010000b

ret

num8:

cjne a,#8, num7

mov p1,#10000000b

ret

num7:

cjne a,#7, num6

mov p1,#11111000b

ret

num6:

cjne a,#6, num5

mov p1,#10000010b

ret

num5:

cjne a,#5, num4

mov p1,#10010010b

ret

num4:

cjne a,#4, num3

mov p1,#10011001b

ret

num3:

cjne a,#3, num2

mov p1,#10110000b

ret

num2:

cjne a,#2, num1

mov p1,#10100100b

ret

num1:

cjne a,#1, num0

mov p1,#11111001b

ret

num0:

mov p1,#11000000b

ret

init\_timer:

;считываем кол-во десятков часов

call read\_sim

call pause

;закидываем считанное в соответствующие регистры

mov a, r7

mov r1, a

mov b, #0Ah

MUL ab

;переносим что бы оно не затерлось

mov 50h, a

;считываем кол-во часов

call read\_sim

call pause

;восстанавливаем значение a

mov a, 50h

;закидываем считанное в соответствующие регистры

mov b, r7

mov r2, b

add a, b

;приводим к максимальному кол-ву часов и сохраняем

MOV B, #24

DIV AB

mov r6, B

;считываем кол-во десятков минут

call read\_sim

call pause

;закидываем считанное в соответствующие регистры

mov a, r7

mov r3, a

mov b, #0Ah

MUL ab

;запоминаем a что б не потерлось

mov 50h, a

;считываем кол-во минут

call read\_sim

;восстанавливаем a

mov a, 50h

;закидываем считанное в соответствующие регистры

mov b, r7

mov r4, b

add a, b

;приводим к макс кол-ву минут и сохраняем

MOV B, #60

DIV AB

mov r5, B

ret

read\_sim:

mov r7, #0

read\_loop:

;пока не считываем число в не выходим из цикла

cjne r7, #0, read\_out

call scanKeyBoard

jmp read\_loop

read\_out:

ret

; Сканирование по рядам

scanKeyBoard:

mov p0, #11111111b

clr p0.3

CALL scanRow1

setb p0.3

clr p0.2

CALL scanRow2

setb p0.2

clr p0.1

CALL scanRow3

setb p0.1

clr p0.0

CALL scanRow4

ret

; Сканирование первого ряда

scanRow1:

MOV A, #1h

JNB P0.6, pressKeyR1

MOV A, #2h

JNB P0.5, pressKeyR1

MOV A, #3h

JNB P0.4, pressKeyR1

ret ; ключ не найден

pressKeyR1:

call putc

ret

; Сканирование второго ряда

scanRow2:

MOV A, #4h

JNB P0.6, pressKeyR2

MOV A, #5h

JNB P0.5, pressKeyR2

MOV A, #6h

JNB P0.4, pressKeyR2

ret ; ключ не найден

pressKeyR2:

call putc

ret

; Сканирование третьего ряда

scanRow3:

MOV A, #7h

JNB P0.6, pressKeyR3

MOV A, #8h

JNB P0.5, pressKeyR3

MOV A, #9h

JNB P0.4, pressKeyR3

ret ; ключ не найден

pressKeyR3:

call putc

ret

; Сканирование четвертого ряда

scanRow4:

MOV A, #0

JNB P0.5, pressKeyR4

ret ; ключ не найден

pressKeyR4:

call putc

ret

;перевеод числа из A в r7

putc:

mov r7, a

ret

;ожидание что бы цифра проявилась

pause:

mov b,#200

djnz b, $

ret

END

**Объектный код**

lab2 PAGE 1

1 ORG 0h

0000 020060 2 jmp main

3 step\_time\_int:

4 ORG 0Bh;Тут прерывание от таймера номер 0

000B C28D 5 CLR TF0;Очищаем флаг прерывания

6

000D 120078 7 call step\_time

8

0010 32 9 RETI ;выход из перывания

10

11 semi\_out\_int:

12 ORG 1Bh;Тут прерывание от таймера номер 1

001B C28F 13 CLR TF1;Очистка флага прерывания

001D 7590FF 14 MOV P1, #0FFh

0020 D2B3 15 setb p3.3

0022 D2B4 16 setb p3.4

0024 D2B6 17 setb p3.6

0026 D2B7 18 setb p3.7

19

20 ;первая лампочка

0028 B80003 21 CJNE R0, #0h, lam\_skip\_1

002B E9 22 mov A, r1;берем число соответствующее лампочке

002C C2B3 23 clr p3.3;выбираем лампочку

24

25 lam\_skip\_1:

26 ;вторая лампочка

002E B80105 27 CJNE R0, #1h, lam\_skip\_2

0031 EA 28 mov A, r2

0032 D2B3 29 setb p3.3

0034 C2B4 30 clr p3.4

31

32 lam\_skip\_2:

0036 B80205 33 CJNE R0, #2h, lam\_skip\_3

0039 EB 34 mov A, r3

003A D2B4 35 setb p3.4

003C C2B6 36 clr p3.6

37

38 lam\_skip\_3:

003E B80305 39 CJNE R0, #3h, lam\_skip\_4

0041 EC 40 mov A, r4

0042 D2B6 41 setb p3.6

0044 C2B7 42 clr p3.7

43

44 lam\_skip\_4:

45

46 ;Заносим цифру в A и вызываем функцию которая отображает ее в P1

0046 1200B1 47 call get\_semi\_code

48

0049 B8010D 49 CJNE R0, #1h, term;если вывод на вторую лампочку то там ставится точка, если секунд не четное

50 ;она получится по центру

004C E552 51 MOV A, 52h

004E 75F002 52 MOV B, #2

0051 84 53 DIV AB ;делим что бы получить остаток

0052 E5F0 54 MOV A, B

55 ;делим на два и смотрим сколько получилось

0054 B40102 56 CJNE A, #1, term ;если есть четное то прыгаем дальше

57 ;иначе моргаем точкой

0057 C297 58 CLR P1.7

59

60 term:

0059 08 61 inc R0 ;переходим к следующему сегменту

005A B80402 62 CJNE R0, #00000100b, end\_semo\_circle ;Если дошли до конца семи сегментного индикатора то обну

005D 7800 63 MOV R0,#0b

64

65 end\_semo\_circle:

005F 32 66 RETI;Выход из прерывания вывода

67

68 main:

69

70 ;T1 M1=1 M0=0 по прерыванию первого таймера мы выводим в индикатор

71 ;T0 M1=0 M0=1 по этому прерыванию увеличиваем время

0060 758921 72 MOV TMOD,#00100001b

73

0063 758CFF 74 MOV TH0, #0FFh

0066 758AE0 75 MOV TL0, #0E0h

76

77 ;разрешаем прерывание от таймера 0

0069 D2A9 78 SETB ET0

79 ;разрешаем прерывание от таймера 1

006B D2AB 80 SETB ET1

81 ;разрешаем выполнение прерываний

006D D2AF 82 SETB EA

83

84

006F D28E 85 SETB TR1;запускаем таймер 1

86

87 ;инициализация начальных значений

0071 1200F4 88 call init\_timer

89

0074 D28C 90 SETB TR0;запускаем таймер 0

91

0076 80FE 92 jmp $;Бесконечный цикл работы

93

94 ;шаг таймера

95 step\_time:

0078 758CFF 96 MOV TH0, #0FFh;обновляем настройки таймера

007B 758AE0 97 MOV TL0, #0E0h

98

99 ;получаем десятки часов в r1

007E EE 100 MOV A, r6

007F 75F00A 101 MOV B, #0Ah

0082 84 102 DIV AB

0083 F9 103 MOV r1,A

104

105 ;получаем часов в r2

0084 AAF0 106 MOV r2,B

107

108 ;получаем десятки минут в r3

0086 ED 109 MOV A, r5

0087 75F00A 110 MOV B, #0Ah

008A 84 111 DIV AB

008B FB 112 MOV r3,A

113

114 ;получаем минуты в r4

008C ACF0 115 MOV r4,B

116

008E E551 117 MOV A, 51h;увеличиваем счетчик милисекунд

0090 04 118 INC A

0091 F551 119 MOV 51h, A

120

0093 B4141A 121 CJNE A, #20, OUT ;если не накапала секудна прыгаем дальше

122 ;иначе обнуляем счетчик милисекунд и увеличиваем секунду

0096 755100 123 MOV 51h, #0

124 ;увеличиваем счетчик секунд

0099 E552 125 MOV A, 52h

009B 04 126 INC A

009C F552 127 MOV 52h, A

128

009E B43C0F 129 CJNE A, #60, OUT ;если не получили 60 секунд прыгаем дальше

130 ;иначе обнуляем счетчик и увеличиваем минуты

00A1 850052 131 MOV 52h, 0

00A4 0D 132 INC r5

133

00A5 BD3C08 134 CJNE r5, #60, OUT;Если не получили 60 минут прыгаем дальше

135 ;Если получили обновляем счетчик

00A8 7D00 136 MOV r5, #0

137 ;увеличиваем часы

00AA 0E 138 INC r6

139

00AB BE1802 140 CJNE r6, #24, OUT;Если не получили 24 часа прыгаем дальше

00AE 7E00 141 MOV r6, #0;Если получили обновляем счетчик

142

143 OUT:

00B0 22 144 RET

145

146 ;отображение цифры из А в P1

147 get\_semi\_code:

00B1 B40904 148 cjne a,#9, num8

00B4 759090 149 mov p1,#10010000b

00B7 22 150 ret

151 num8:

00B8 B40804 152 cjne a,#8, num7

00BB 759080 153 mov p1,#10000000b

00BE 22 154 ret

155 num7:

00BF B40704 156 cjne a,#7, num6

00C2 7590F8 157 mov p1,#11111000b

00C5 22 158 ret

159 num6:

00C6 B40604 160 cjne a,#6, num5

00C9 759082 161 mov p1,#10000010b

00CC 22 162 ret

163 num5:

00CD B40504 164 cjne a,#5, num4

00D0 759092 165 mov p1,#10010010b

00D3 22 166 ret

167 num4:

00D4 B40404 168 cjne a,#4, num3

00D7 759099 169 mov p1,#10011001b

00DA 22 170 ret

171 num3:

00DB B40304 172 cjne a,#3, num2

00DE 7590B0 173 mov p1,#10110000b

00E1 22 174 ret

175 num2:

00E2 B40204 176 cjne a,#2, num1

00E5 7590A4 177 mov p1,#10100100b

00E8 22 178 ret

179 num1:

00E9 B40104 180 cjne a,#1, num0

00EC 7590F9 181 mov p1,#11111001b

00EF 22 182 ret

183 num0:

00F0 7590C0 184 mov p1,#11000000b

00F3 22 185 ret

186

187 init\_timer:

188 ;считываем кол-во десятков часов

00F4 120136 189 call read\_sim

00F7 1201A7 190 call pause

191 ;закидываем считанное в соответствующие регистры

00FA EF 192 mov a, r7

00FB F9 193 mov r1, a

00FC 75F00A 194 mov b, #0Ah

00FF A4 195 MUL ab

196 ;переносим что бы оно не затерлось

0100 F550 197 mov 50h, a

198

199 ;считываем кол-во часов

0102 120136 200 call read\_sim

0105 1201A7 201 call pause

202 ;восстанавливаем значение a

0108 E550 203 mov a, 50h

204 ;закидываем считанное в соответствующие регистры

010A 8FF0 205 mov b, r7

010C AAF0 206 mov r2, b

010E 25F0 207 add a, b

208 ;приводим к максимальному кол-ву часов и сохраняем

0110 75F018 209 MOV B, #24

0113 84 210 DIV AB

0114 AEF0 211 mov r6, B

212

213 ;считываем кол-во десятков минут

0116 120136 214 call read\_sim

0119 1201A7 215 call pause

216 ;закидываем считанное в соответствующие регистры

011C EF 217 mov a, r7

011D FB 218 mov r3, a

011E 75F00A 219 mov b, #0Ah

0121 A4 220 MUL ab

221 ;запоминаем a что б не потерлось

0122 F550 222 mov 50h, a

223

224 ;считываем кол-во минут

0124 120136 225 call read\_sim

226 ;восстанавливаем a

0127 E550 227 mov a, 50h

228 ;закидываем считанное в соответствующие регистры

0129 8FF0 229 mov b, r7

012B ACF0 230 mov r4, b

012D 25F0 231 add a, b

232 ;приводим к макс кол-ву минут и сохраняем

012F 75F03C 233 MOV B, #60

0132 84 234 DIV AB

0133 ADF0 235 mov r5, B

236

0135 22 237 ret

238

239 read\_sim:

0136 7F00 240 mov r7, #0

241 read\_loop:

242 ;пока не считываем число в не выходим из цикла

0138 BF0005 243 cjne r7, #0, read\_out

013B 120141 244 call scanKeyBoard

013E 80F8 245 jmp read\_loop

246

247 read\_out:

0140 22 248 ret

249

250 ; Сканирование по рядам

251 scanKeyBoard:

0141 7580FF 252 mov p0, #11111111b

253

0144 C283 254 clr p0.3

0146 12015F 255 CALL scanRow1

256

0149 D283 257 setb p0.3

014B C282 258 clr p0.2

014D 120173 259 CALL scanRow2

260

0150 D282 261 setb p0.2

0152 C281 262 clr p0.1

0154 120187 263 CALL scanRow3

264

0157 D281 265 setb p0.1

0159 C280 266 clr p0.0

015B 12019B 267 CALL scanRow4

268

015E 22 269 ret

270

271 ; Сканирование первого ряда

272 scanRow1:

015F 7401 273 MOV A, #1h

0161 30860B 274 JNB P0.6, pressKeyR1

275

0164 7402 276 MOV A, #2h

0166 308506 277 JNB P0.5, pressKeyR1

278

0169 7403 279 MOV A, #3h

016B 308401 280 JNB P0.4, pressKeyR1

281

016E 22 282 ret ; ключ не найден

283

284 pressKeyR1:

016F 1201A5 285 call putc

0172 22 286 ret

287

288 ; Сканирование второго ряда

289 scanRow2:

0173 7404 290 MOV A, #4h

0175 30860B 291 JNB P0.6, pressKeyR2

292

0178 7405 293 MOV A, #5h

017A 308506 294 JNB P0.5, pressKeyR2

295

017D 7406 296 MOV A, #6h

017F 308401 297 JNB P0.4, pressKeyR2

298

0182 22 299 ret ; ключ не найден

300

301 pressKeyR2:

0183 1201A5 302 call putc

0186 22 303 ret

304

305 ; Сканирование третьего ряда

306 scanRow3:

0187 7407 307 MOV A, #7h

0189 30860B 308 JNB P0.6, pressKeyR3

309

018C 7408 310 MOV A, #8h

018E 308506 311 JNB P0.5, pressKeyR3

312

0191 7409 313 MOV A, #9h

0193 308401 314 JNB P0.4, pressKeyR3

315

0196 22 316 ret ; ключ не найден

317

318 pressKeyR3:

0197 1201A5 319 call putc

019A 22 320 ret

321

322 ; Сканирование четвертого ряда

323 scanRow4:

019B 7400 324 MOV A, #0

019D 308501 325 JNB P0.5, pressKeyR4

326

01A0 22 327 ret ; ключ не найден

328

329 pressKeyR4:

01A1 1201A5 330 call putc

01A4 22 331 ret

332

333 ;перевеод числа из A в r7

334 putc:

01A5 FF 335 mov r7, a

01A6 22 336 ret

337

338 ;ожидание что бы цифра проявилась

339 pause:

01A7 75F0C8 340 mov b,#200

01AA D5F0FD 341 djnz b, $

01AD 22 342 ret

343 END

ASSEMBLY COMPLETE, NO ERRORS FOUND, NO WARNINGS

SYMBOL TABLE:

??MCU\_8051\_IDE . . . . . . . . . . . N NUMB 8051H NOT USED

??VERSION. . . . . . . . . . . . . . N NUMB 0149H NOT USED

AC . . . . . . . . . . . . . . . . . B ADDR 00D6H NOT USED

ACC. . . . . . . . . . . . . . . . . D ADDR 00E0H NOT USED

ACSR . . . . . . . . . . . . . . . . D ADDR 0097H NOT USED

ADCF . . . . . . . . . . . . . . . . D ADDR 00F6H NOT USED

ADCLK. . . . . . . . . . . . . . . . D ADDR 00F2H NOT USED

ADCON. . . . . . . . . . . . . . . . D ADDR 00F3H NOT USED

ADDH . . . . . . . . . . . . . . . . D ADDR 00F5H NOT USED

ADDL . . . . . . . . . . . . . . . . D ADDR 00F4H NOT USED

AUXR . . . . . . . . . . . . . . . . D ADDR 008EH NOT USED

AUXR1. . . . . . . . . . . . . . . . D ADDR 00A2H NOT USED

B. . . . . . . . . . . . . . . . . . D ADDR 00F0H

BDRCON . . . . . . . . . . . . . . . D ADDR 009BH NOT USED

BDRCON\_1 . . . . . . . . . . . . . . D ADDR 009CH NOT USED

BRL. . . . . . . . . . . . . . . . . D ADDR 009AH NOT USED

CCAP0H . . . . . . . . . . . . . . . D ADDR 00FAH NOT USED

CCAP0L . . . . . . . . . . . . . . . D ADDR 00EAH NOT USED

CCAP1H . . . . . . . . . . . . . . . D ADDR 00FBH NOT USED

CCAP1L . . . . . . . . . . . . . . . D ADDR 00EBH NOT USED

CCAP2H . . . . . . . . . . . . . . . D ADDR 00FCH NOT USED

CCAP3H . . . . . . . . . . . . . . . D ADDR 00FDH NOT USED

CCAP4H . . . . . . . . . . . . . . . D ADDR 00FEH NOT USED

CCAPL2H. . . . . . . . . . . . . . . D ADDR 00FCH NOT USED

CCAPL2L. . . . . . . . . . . . . . . D ADDR 00ECH NOT USED

CCAPL3H. . . . . . . . . . . . . . . D ADDR 00FDH NOT USED

CCAPL3L. . . . . . . . . . . . . . . D ADDR 00EDH NOT USED

CCAPL4H. . . . . . . . . . . . . . . D ADDR 00FEH NOT USED

CCAPL4L. . . . . . . . . . . . . . . D ADDR 00EEH NOT USED

CCAPM0 . . . . . . . . . . . . . . . D ADDR 00DAH NOT USED

CCAPM1 . . . . . . . . . . . . . . . D ADDR 00DBH NOT USED

CCAPM2 . . . . . . . . . . . . . . . D ADDR 00DCH NOT USED

CCAPM3 . . . . . . . . . . . . . . . D ADDR 00DDH NOT USED

CCAPM4 . . . . . . . . . . . . . . . D ADDR 00DEH NOT USED

CCF0 . . . . . . . . . . . . . . . . B ADDR 00D8H NOT USED

CCF1 . . . . . . . . . . . . . . . . B ADDR 00D9H NOT USED

CCF2 . . . . . . . . . . . . . . . . B ADDR 00DAH NOT USED

CCF3 . . . . . . . . . . . . . . . . B ADDR 00DBH NOT USED

CCF4 . . . . . . . . . . . . . . . . B ADDR 00DCH NOT USED

CCON . . . . . . . . . . . . . . . . D ADDR 00D8H NOT USED

CFINT. . . . . . . . . . . . . . . . C ADDR 0033H NOT USED

CH . . . . . . . . . . . . . . . . . D ADDR 00F9H NOT USED

CKCON. . . . . . . . . . . . . . . . D ADDR 008FH NOT USED

CKCON0 . . . . . . . . . . . . . . . D ADDR 008FH NOT USED

CKRL . . . . . . . . . . . . . . . . D ADDR 0097H NOT USED

CKSEL. . . . . . . . . . . . . . . . D ADDR 0085H NOT USED

CL . . . . . . . . . . . . . . . . . D ADDR 00E9H NOT USED

CLKREG . . . . . . . . . . . . . . . D ADDR 008FH NOT USED

CMOD . . . . . . . . . . . . . . . . D ADDR 00D9H NOT USED

CPRL2. . . . . . . . . . . . . . . . B ADDR 00C8H NOT USED

CR . . . . . . . . . . . . . . . . . B ADDR 00DEH NOT USED

CT2. . . . . . . . . . . . . . . . . B ADDR 00C9H NOT USED

CY . . . . . . . . . . . . . . . . . B ADDR 00D7H NOT USED

DP0H . . . . . . . . . . . . . . . . D ADDR 0083H NOT USED

DP0L . . . . . . . . . . . . . . . . D ADDR 0082H NOT USED

DP1H . . . . . . . . . . . . . . . . D ADDR 0085H NOT USED

DP1L . . . . . . . . . . . . . . . . D ADDR 0084H NOT USED

DPH. . . . . . . . . . . . . . . . . D ADDR 0083H NOT USED

DPL. . . . . . . . . . . . . . . . . D ADDR 0082H NOT USED

EA . . . . . . . . . . . . . . . . . B ADDR 00AFH

EC . . . . . . . . . . . . . . . . . B ADDR 00AEH NOT USED

EECON. . . . . . . . . . . . . . . . D ADDR 0096H NOT USED

END\_SEMO\_CIRCLE. . . . . . . . . . . C ADDR 005FH

ES . . . . . . . . . . . . . . . . . B ADDR 00ACH NOT USED

ET0. . . . . . . . . . . . . . . . . B ADDR 00A9H

ET1. . . . . . . . . . . . . . . . . B ADDR 00ABH

ET2. . . . . . . . . . . . . . . . . B ADDR 00ADH NOT USED

EX0. . . . . . . . . . . . . . . . . B ADDR 00A8H NOT USED

EX1. . . . . . . . . . . . . . . . . B ADDR 00AAH NOT USED

EXEN2. . . . . . . . . . . . . . . . B ADDR 00CBH NOT USED

EXF2 . . . . . . . . . . . . . . . . B ADDR 00CEH NOT USED

EXTI0. . . . . . . . . . . . . . . . C ADDR 0003H NOT USED

EXTI1. . . . . . . . . . . . . . . . C ADDR 0013H NOT USED

F0 . . . . . . . . . . . . . . . . . B ADDR 00D5H NOT USED

FE . . . . . . . . . . . . . . . . . B ADDR 009FH NOT USED

GET\_SEMI\_CODE. . . . . . . . . . . . C ADDR 00B1H

IE . . . . . . . . . . . . . . . . . D ADDR 00A8H NOT USED

IE0. . . . . . . . . . . . . . . . . B ADDR 0089H NOT USED

IE1. . . . . . . . . . . . . . . . . B ADDR 008BH NOT USED

INIT\_TIMER . . . . . . . . . . . . . C ADDR 00F4H

INT0 . . . . . . . . . . . . . . . . B ADDR 00B2H NOT USED

INT1 . . . . . . . . . . . . . . . . B ADDR 00B3H NOT USED

IP . . . . . . . . . . . . . . . . . D ADDR 00B8H NOT USED

IPH. . . . . . . . . . . . . . . . . D ADDR 00B7H NOT USED

IPH0 . . . . . . . . . . . . . . . . D ADDR 00B7H NOT USED

IPH1 . . . . . . . . . . . . . . . . D ADDR 00B3H NOT USED

IPL0 . . . . . . . . . . . . . . . . D ADDR 00B8H NOT USED

IPL1 . . . . . . . . . . . . . . . . D ADDR 00B2H NOT USED

IT0. . . . . . . . . . . . . . . . . B ADDR 0088H NOT USED

IT1. . . . . . . . . . . . . . . . . B ADDR 008AH NOT USED

KBE. . . . . . . . . . . . . . . . . D ADDR 009DH NOT USED

KBF. . . . . . . . . . . . . . . . . D ADDR 009EH NOT USED

KBLS . . . . . . . . . . . . . . . . D ADDR 009CH NOT USED

LAM\_SKIP\_1 . . . . . . . . . . . . . C ADDR 002EH

LAM\_SKIP\_2 . . . . . . . . . . . . . C ADDR 0036H

LAM\_SKIP\_3 . . . . . . . . . . . . . C ADDR 003EH

LAM\_SKIP\_4 . . . . . . . . . . . . . C ADDR 0046H

MAIN . . . . . . . . . . . . . . . . C ADDR 0060H

NUM0 . . . . . . . . . . . . . . . . C ADDR 00F0H

NUM1 . . . . . . . . . . . . . . . . C ADDR 00E9H

NUM2 . . . . . . . . . . . . . . . . C ADDR 00E2H

NUM3 . . . . . . . . . . . . . . . . C ADDR 00DBH

NUM4 . . . . . . . . . . . . . . . . C ADDR 00D4H

NUM5 . . . . . . . . . . . . . . . . C ADDR 00CDH

NUM6 . . . . . . . . . . . . . . . . C ADDR 00C6H

NUM7 . . . . . . . . . . . . . . . . C ADDR 00BFH

NUM8 . . . . . . . . . . . . . . . . C ADDR 00B8H

OSCCON . . . . . . . . . . . . . . . D ADDR 0086H NOT USED

OUT. . . . . . . . . . . . . . . . . C ADDR 00B0H

OV . . . . . . . . . . . . . . . . . B ADDR 00D2H NOT USED

P. . . . . . . . . . . . . . . . . . B ADDR 00D0H NOT USED

P0 . . . . . . . . . . . . . . . . . D ADDR 0080H

P1 . . . . . . . . . . . . . . . . . D ADDR 0090H

P1M1 . . . . . . . . . . . . . . . . D ADDR 00D4H NOT USED

P1M2 . . . . . . . . . . . . . . . . D ADDR 00E2H NOT USED

P2 . . . . . . . . . . . . . . . . . D ADDR 00A0H NOT USED

P3 . . . . . . . . . . . . . . . . . D ADDR 00B0H

P3M1 . . . . . . . . . . . . . . . . D ADDR 00D5H NOT USED

P3M2 . . . . . . . . . . . . . . . . D ADDR 00E3H NOT USED

P4 . . . . . . . . . . . . . . . . . D ADDR 00C0H NOT USED

P4M1 . . . . . . . . . . . . . . . . D ADDR 00D6H NOT USED

P4M2 . . . . . . . . . . . . . . . . D ADDR 00E4H NOT USED

P5 . . . . . . . . . . . . . . . . . D ADDR 00E8H NOT USED

PAUSE. . . . . . . . . . . . . . . . C ADDR 01A7H

PC . . . . . . . . . . . . . . . . . B ADDR 00BEH NOT USED

PCON . . . . . . . . . . . . . . . . D ADDR 0087H NOT USED

PPCL . . . . . . . . . . . . . . . . B ADDR 00BEH NOT USED

PRESSKEYR1 . . . . . . . . . . . . . C ADDR 016FH

PRESSKEYR2 . . . . . . . . . . . . . C ADDR 0183H

PRESSKEYR3 . . . . . . . . . . . . . C ADDR 0197H

PRESSKEYR4 . . . . . . . . . . . . . C ADDR 01A1H

PS . . . . . . . . . . . . . . . . . B ADDR 00BCH NOT USED

PSL. . . . . . . . . . . . . . . . . B ADDR 00BCH NOT USED

PSW. . . . . . . . . . . . . . . . . D ADDR 00D0H NOT USED

PT0. . . . . . . . . . . . . . . . . B ADDR 00B9H NOT USED

PT0L . . . . . . . . . . . . . . . . B ADDR 00B9H NOT USED

PT1. . . . . . . . . . . . . . . . . B ADDR 00BBH NOT USED

PT1L . . . . . . . . . . . . . . . . B ADDR 00BBH NOT USED

PT2. . . . . . . . . . . . . . . . . B ADDR 00BDH NOT USED

PT2L . . . . . . . . . . . . . . . . B ADDR 00BDH NOT USED

PUTC . . . . . . . . . . . . . . . . C ADDR 01A5H

PX0. . . . . . . . . . . . . . . . . B ADDR 00B8H NOT USED

PX0L . . . . . . . . . . . . . . . . B ADDR 00B8H NOT USED

PX1. . . . . . . . . . . . . . . . . B ADDR 00BAH NOT USED

PX1L . . . . . . . . . . . . . . . . B ADDR 00BAH NOT USED

RB8. . . . . . . . . . . . . . . . . B ADDR 009AH NOT USED

RCAP2H . . . . . . . . . . . . . . . D ADDR 00CBH NOT USED

RCAP2L . . . . . . . . . . . . . . . D ADDR 00CAH NOT USED

RCLK . . . . . . . . . . . . . . . . B ADDR 00CDH NOT USED

RD . . . . . . . . . . . . . . . . . B ADDR 00B7H NOT USED

READ\_LOOP. . . . . . . . . . . . . . C ADDR 0138H

READ\_OUT . . . . . . . . . . . . . . C ADDR 0140H

READ\_SIM . . . . . . . . . . . . . . C ADDR 0136H

REN. . . . . . . . . . . . . . . . . B ADDR 009CH NOT USED

RESET. . . . . . . . . . . . . . . . C ADDR 0000H NOT USED

RI . . . . . . . . . . . . . . . . . B ADDR 0098H NOT USED

RS0. . . . . . . . . . . . . . . . . B ADDR 00D3H NOT USED

RS1. . . . . . . . . . . . . . . . . B ADDR 00D4H NOT USED

RXD. . . . . . . . . . . . . . . . . B ADDR 00B0H NOT USED

SADDR. . . . . . . . . . . . . . . . D ADDR 00A9H NOT USED

SADDR\_0. . . . . . . . . . . . . . . D ADDR 00A9H NOT USED

SADDR\_1. . . . . . . . . . . . . . . D ADDR 00AAH NOT USED

SADEN. . . . . . . . . . . . . . . . D ADDR 00B9H NOT USED

SADEN\_0. . . . . . . . . . . . . . . D ADDR 00B9H NOT USED

SADEN\_1. . . . . . . . . . . . . . . D ADDR 00BAH NOT USED

SBUF . . . . . . . . . . . . . . . . D ADDR 0099H NOT USED

SCANKEYBOARD . . . . . . . . . . . . C ADDR 0141H

SCANROW1 . . . . . . . . . . . . . . C ADDR 015FH

SCANROW2 . . . . . . . . . . . . . . C ADDR 0173H

SCANROW3 . . . . . . . . . . . . . . C ADDR 0187H

SCANROW4 . . . . . . . . . . . . . . C ADDR 019BH

SCON . . . . . . . . . . . . . . . . D ADDR 0098H NOT USED

SEMI\_OUT\_INT . . . . . . . . . . . . C ADDR 001BH NOT USED

SINT . . . . . . . . . . . . . . . . C ADDR 0023H NOT USED

SM0. . . . . . . . . . . . . . . . . B ADDR 009FH NOT USED

SM1. . . . . . . . . . . . . . . . . B ADDR 009EH NOT USED

SM2. . . . . . . . . . . . . . . . . B ADDR 009DH NOT USED

SP . . . . . . . . . . . . . . . . . D ADDR 0081H NOT USED

SPCON. . . . . . . . . . . . . . . . D ADDR 00C3H NOT USED

SPCR . . . . . . . . . . . . . . . . D ADDR 00D5H NOT USED

SPDAT. . . . . . . . . . . . . . . . D ADDR 00C5H NOT USED

SPDR . . . . . . . . . . . . . . . . D ADDR 0086H NOT USED

SPSR . . . . . . . . . . . . . . . . D ADDR 00AAH NOT USED

SPSTA. . . . . . . . . . . . . . . . D ADDR 00C4H NOT USED

STEP\_TIME. . . . . . . . . . . . . . C ADDR 0078H

STEP\_TIME\_INT. . . . . . . . . . . . C ADDR 000BH NOT USED

T0 . . . . . . . . . . . . . . . . . B ADDR 00B4H NOT USED

T1 . . . . . . . . . . . . . . . . . B ADDR 00B5H NOT USED

T2CON. . . . . . . . . . . . . . . . D ADDR 00C8H NOT USED

T2MOD. . . . . . . . . . . . . . . . D ADDR 00C9H NOT USED

TB8. . . . . . . . . . . . . . . . . B ADDR 009BH NOT USED

TCLK . . . . . . . . . . . . . . . . B ADDR 00CCH NOT USED

TCON . . . . . . . . . . . . . . . . D ADDR 0088H NOT USED

TERM . . . . . . . . . . . . . . . . C ADDR 0059H

TF0. . . . . . . . . . . . . . . . . B ADDR 008DH

TF1. . . . . . . . . . . . . . . . . B ADDR 008FH

TF2. . . . . . . . . . . . . . . . . B ADDR 00CFH NOT USED

TH0. . . . . . . . . . . . . . . . . D ADDR 008CH

TH1. . . . . . . . . . . . . . . . . D ADDR 008DH NOT USED

TH2. . . . . . . . . . . . . . . . . D ADDR 00CDH NOT USED

TI . . . . . . . . . . . . . . . . . B ADDR 0099H NOT USED

TIMER0 . . . . . . . . . . . . . . . C ADDR 000BH NOT USED

TIMER1 . . . . . . . . . . . . . . . C ADDR 001BH NOT USED

TIMER2 . . . . . . . . . . . . . . . C ADDR 002BH NOT USED

TL0. . . . . . . . . . . . . . . . . D ADDR 008AH

TL1. . . . . . . . . . . . . . . . . D ADDR 008BH NOT USED

TL2. . . . . . . . . . . . . . . . . D ADDR 00CCH NOT USED

TMOD . . . . . . . . . . . . . . . . D ADDR 0089H

TR0. . . . . . . . . . . . . . . . . B ADDR 008CH

TR1. . . . . . . . . . . . . . . . . B ADDR 008EH

TR2. . . . . . . . . . . . . . . . . B ADDR 00CAH NOT USED

TXD. . . . . . . . . . . . . . . . . B ADDR 00B1H NOT USED

WDTCON . . . . . . . . . . . . . . . D ADDR 00A7H NOT USED

WDTPRG . . . . . . . . . . . . . . . D ADDR 00A7H NOT USED

WDTRST . . . . . . . . . . . . . . . D ADDR 00A6H NOT USED

WR . . . . . . . . . . . . . . . . . B ADDR 00B6H NOT USED

**Вывод**

В ходе выполнения лабораторной работы был получен опыт работы с таймерами, прерываниями и семисегментным индикатором кристала.