



**Министерство науки и высшего образования Российской  
Федерации  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

**Отчёт по лабораторной работе № 2**

по дисциплине «Аппаратные средства вычислительной техники»

**Тема: «Автомобильная индикация»**

Вариант 3.

Преподаватель:  
Рафиков А.Г.

Студент: Веденеев А.А.

Группа: ИУ8-72

*Москва, 2023*

### **Цель работы:**

Изучение основ реализации таймера и освоение работы с прерываниями на микроконтроллере серии 8051 в рабочей среде Proteus.

### **Задание:**

Реализовать схему автомобильной световой индикации по следующим требованиям:

- Имеется 6 ламп (выходные сигналы) – левая и правая лампы, указатели на панели, и задние.
- Имеется 5 входных сигналов – педаль тормоза, аварийный сигнал, сигнал парковки, переключатель правого и левого поворота.
- Нажатие тормозной педали включает немигающий задний свет (BREAK -> L\_REAR and R\_REAR).
- Нажатие аварийного переключателя включает мигание всех шести ламп.
- Нажатие переключателя парковки включает приглушенные задние лампы.
- Замыкание контакта поворота включает мигание левых/правых ламп (L\_DASH/R\_DASH).

### Вариант 3:

- Отобразить на дисплее перегоревшие лампы.
- Скважность при парковке 30%.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

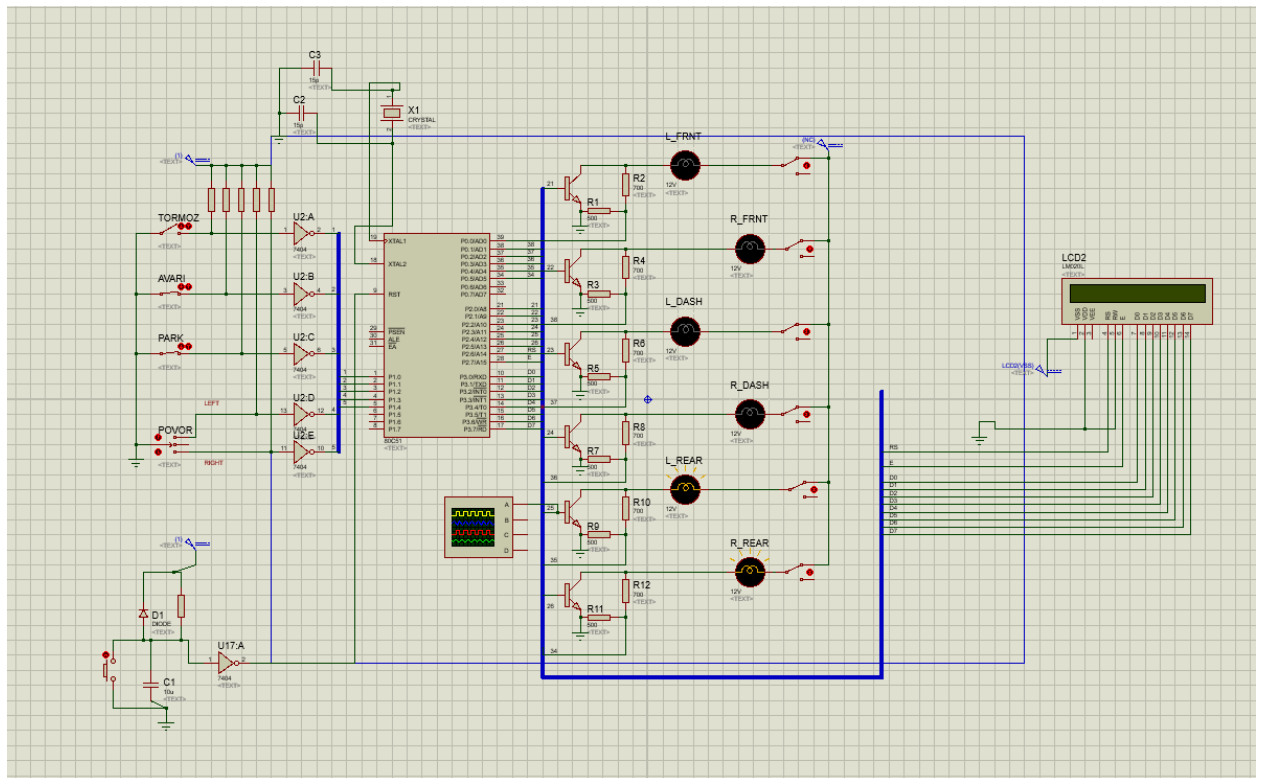


Рисунок 1 - Схема в Proteus

### Код микроконтроллера:

```
$NOMOD51  
$INCLUDE (8051.MCU)
```

```
; INPUTS  
BRAKE BIT P1.0  
EMERG BIT P1.1  
PARK BIT P1.2  
L_TURN BIT P1.3  
R_TURN BIT P1.4
```

```
; OUTPUTS  
L_FRNT BIT P2.0  
R_FRNT BIT P2.1  
L_DASH BIT P2.2
```

```

R_DASH BIT P2.3 ; ПРАВЫЙ ИНДИКАТОР НА ПАНЕЛИ
L_REAR BIT P2.4
R_REAR BIT P2.5
; LCD
RS BIT P2.6
E BIT P2.7
DD DATA P3
F DATA 21h
; ИСПОЛЬЗУЮТСЯ ДЛЯ ПОЛУЧЕНИЯ ЧАСТОТ
SUB_DIV DATA 20H ; ДЕЛИТЕЛЬ ЧАСТОТЫ ПЕРЕРЫВАНИЙ
HI_FREQ BIT SUB_DIV.0 ; БИТ ГЕНЕРАТОРА ВЫСОКОЙ ЧАСТОТЫ - ДЛЯ СТОЯНКИ
;ME_FREQ BIT SUB_DIV.5
LO_FREQ BIT SUB_DIV.7 ; БИТ ГЕНЕРАТОРА НИЗКОЙ ЧАСТОТЫ - ДЛЯ МИГАНИЯ
;=====
; RESET
;=====
ORG 0000H
ACALL INIT
JMP MAIN
;=====
; ОБРАБОТЧИК ПЕРЕРЫВАНИЯ ТАЙМЕРА_0
;=====
ORG 000BH ; ПО ЭТОМУ АДРЕСУ РАСПОЛАГАЕТСЯ ПОДПРОГРАММА ОБР-КИ ПРЕ-ИЯ
ACALL TMR0_PROCCESING
RETI
; ОСНОВНАЯ ПРОГРАММА
ORG 150H
MAIN:
    SJMP $ ;jump to currentline
; ПОДПРОГРАММЫ
ORG 200H
INIT:
; - регистр режимов - TMOD - от английского "Timer MODE";
; - регистр управления/статуса - TCON - от "Timer CONtrol"
MOV TMOD, #00000001B; ТАЙМЕР(С/Т0 = 0) В 1 РЕЖИМ М1.0=0 М0.0=1
MOV TL0, #0 ; ИНИЦИАЛИЗАЦИЯ РЕГИСТРОВ ТАЙМЕРА
MOV TH0, #-16 ; ИНИЦИАЛИЗАЦИЯ РЕГИСТРОВ ТАЙМЕРА
MOV SUB_DIV, #244 ; ДЕЛЕНИЕ ЧАСТОТЫ ПЕРЕРЫВАНИЯ НА 244
MOV R5, #61
SETB ET0 ; РАЗРЕШИТЬ ПЕРЕРЫВАНИЕ ОТ ТАЙМЕРА 0
SETB EA ; РАЗРЕШИТЬ ОБЩЕЕ ПЕРЕРЫВАНИЕ
SETB TR0 ; СТАРТ ТАЙМЕРА
CLR RS
CLR E
CLR F0
MOV R7, #14
MOV F, #255
MOV R6, #24
RET
TMR0_PROCCESING:
MOV TL0, #0
MOV TH0, #-16 ; ВНОВЬ ЗАПИСАТЬ ЧИСЛО, ЧТОБЫ ОН МОГ СНОВА СЧИТАТЬ
DJNZ SUB_DIV, T0_SERV ; уменьшение SUB_DIV на 1
MOV SUB_DIV, #244 ; если стал 0, то снова записать 244
; это как раз и нужно для генерации сигнала высокой частоты SUB_DIV.0
T0_SERV:
CLR E
MOV A, R7
JNZ INIT_LCD1
DJNZ R5, NEXT
MOV R5, #61
CHECK00:

```

```

CLR L_FRNT
CLR R_FRNT
CLR L_DASH
CLR R_DASH
CLR L_REAR
CLR R_REAR
MOV F, P0
NEXT:
    DJNZ R6, CHECK
    MOV R6, #24
CHECK:
    MOV A, R6
    ANL A, #00010000B
    JNZ CHECK_456
CHECK_123:
    MOV A, R6
    ANL A, #00001000B
    JNZ CHECK_23
CHECK_1:
    MOV A, R6
    ANL A, #00000010B
    JNZ SET_ADDR1
    JB F.0, PRINT_PLUS
    JMP PRINT_MINUS
SET_ADDR1:
    MOV A, #10000000B
    JMP SET_ADDR
CHECK_23:
    MOV A, R6
    ANL A, #00000100B
    JNZ CHECK_3
CHECK_2:
    MOV A, R6
    ANL A, #00000010B
    JNZ SET_ADDR2
    JB F.1, PRINT_PLUS
    JMP PRINT_MINUS
SET_ADDR2:
    MOV A, #10000010B
    JMP SET_ADDR
INIT_LCD1:
    LJMP INIT_LCD
CHECK_3:
    MOV A, R6
    ANL A, #00000010B
    JNZ SET_ADDR3
    JB F.2, PRINT_PLUS
    JMP PRINT_MINUS
SET_ADDR3:
    MOV A, #10000100B
    JMP SET_ADDR
CHECK_456:
    MOV A, R6
    ANL A, #00001000B
    JNZ CHECK_6
CHECK_45:
    MOV A, R6
    ANL A, #00000100B
    JNZ CHECK_5
CHECK_4:
    MOV A, R6
    ANL A, #00000010B

```

```

JNZ SET_ADDR4
JB F.3, PRINT_PLUS
JMP PRINT_MINUS
SET_ADDR4:
MOV A, #10000110B
JMP SET_ADDR
CHECK_5:
MOV A, R6
ANL A, #00000010B
JNZ SET_ADDR5
JB F.4, PRINT_PLUS
JMP PRINT_MINUS
SET_ADDR5:
MOV A, #10001000B
JMP SET_ADDR
CHECK_6:
MOV A, R6
ANL A, #00000010B
JNZ SET_ADDR6
JB F.5, PRINT_PLUS
JMP PRINT_MINUS
SET_ADDR6:
MOV A, #10001010B
JMP SET_ADDR
PRINT_PLUS:
SETB RS
MOV DD, #00101011B
MOV A, R6
ANL A, #00000001B
JNZ SAVE
JMP DEC1
PRINT_MINUS:
SETB RS
MOV DD, #00101101B
MOV A, R6
ANL A, #00000001B
JNZ SAVE
JMP DEC1
SET_ADDR:
CLR RS
MOV DD, A
MOV A, R6
ANL A, #00000001B
JNZ SAVE
JMP DEC1
SAVE:
SETB E
DEC1:
DEC R6
; включение парковых огней при парковке
PARK_SCRIPT:
JNB PARK, ORDINARY_MODE
CLR L_DASH
CLR R_DASH
CLR L_FRNT
CLR R_FRNT
; скважность 10
MOV C, SUB_DIV.0
ANL C, SUB_DIV.1
ANL C, SUB_DIV.2
ANL C, SUB_DIV.3
ORL C, SUB_DIV.4

```

```

ANL C, SUB_DIV.1
ANL C, SUB_DIV.2
ANL C, SUB_DIV.3

MOV L_REAR, C ; на заднюю лампочку
MOV R_REAR, C ; на заднюю лампочку
JMP STOP_SCRIPT
ORDINARY_MODE:
; выдать левый поворот, если он нажат
MOV C, L_TURN
ORL C, EMERG
ANL C, LO_FREQ
MOV L_DASH, C ; на панель
MOV L_FRNT, C ; на переднюю лампочку
MOV L_REAR, C ; на заднюю лампочку
; выдать правый поворот, если он нажат
MOV C, R_TURN
ORL C, EMERG
ANL C, LO_FREQ
MOV R_DASH, C ; на панель
MOV R_FRNT, C ; на переднюю лампочку
MOV R_REAR, C ; на заднюю лампочку
; выдать стоп, если он нажат
STOP_SCRIPT:
MOV C, BRAKE
JNC BRAKE_PRESSED
SETB L_REAR
SETB R_REAR
BRAKE_PRESSED:
RET
INIT_LCD: ;настройка экрана вывод плюсов
MOV A, R7
XRL A, #00001110B
JZ S0
MOV A, R7
XRL A, #00001101B
JZ S1
MOV A, R7
ANL A, #00000001B
JZ SPACE
MOV A, R7
XRL A, #00001011B
JZ S3
MOV A, R7
XRL A, #00001001B
JZ S5
MOV A, R7
XRL A, #00000111B
JZ S7
MOV A, R7
XRL A, #00000101B
JZ S9
MOV A, R7
XRL A, #00000011B
JZ S11
MOV A, R7
XRL A, #00000001B
JZ S13
BYE:
RET
SPACE:
SETB RS

```

```

MOV A, #00100000B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S0:
MOV A, #00111100B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S1:
MOV A, #00001100B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S3:
MOV A, #00110001B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S5:
MOV A, #00110010B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S7:
MOV A, #00110011B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S9:
MOV A, #00110100B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S11:
MOV A, #00110101B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
S13:
MOV A, #00110110B
MOV DD, A
JNB HI_FREQ, BYE
JMP LCD_CMD
LCD_CMD:
SETB E
DEC R7
RET
;=====
END

```



## **Вывод**

В ходе выполнения данной лабораторной работы продолжилось изучение микроконтроллера модели 8051, а также была разработана схема управления автомобильной индикации, а также контроля работы внешних и внутренних ламп автомобиля.

Во время работы были изучены способ использования таймера и прерываний для данного микроконтроллера.