

**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)**

Лабораторная работа №1
дисциплина «Программирование микроконтроллеров»
по теме «Написание программы обработки клавиатурной матрицы»

Выполнил: студент группы ВТ-31
Проверил:

Макаров Д.С.
Гольцов Ю. А.

Белгород 2020

Лабораторная работа №1

«Написание программы обработки клавиатурной матрицы»

Цель работы: Изучить устройство портов ввода вывода и универсального синхронно асинхронного приемопередатчика микроконтроллеров на архитектуре Intel 8051.

Вариант 6

Задание: Разработать программу обрабатывающую нажатия клавиш клавиатурной матрицы, преобразующей скан-коды в ASCII коды и выводящую через UART ASCII код нажатой клавиши.

Ход работы

Схема устройства

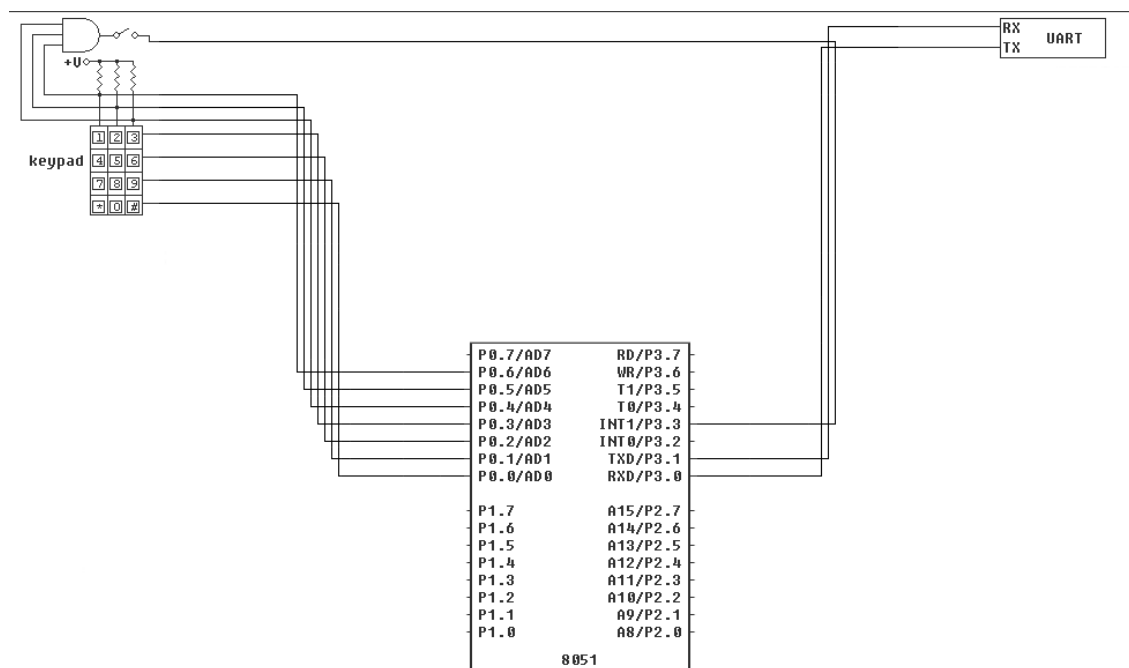


Рис. 1: Схема устройства

Блок схема алгоритма работы

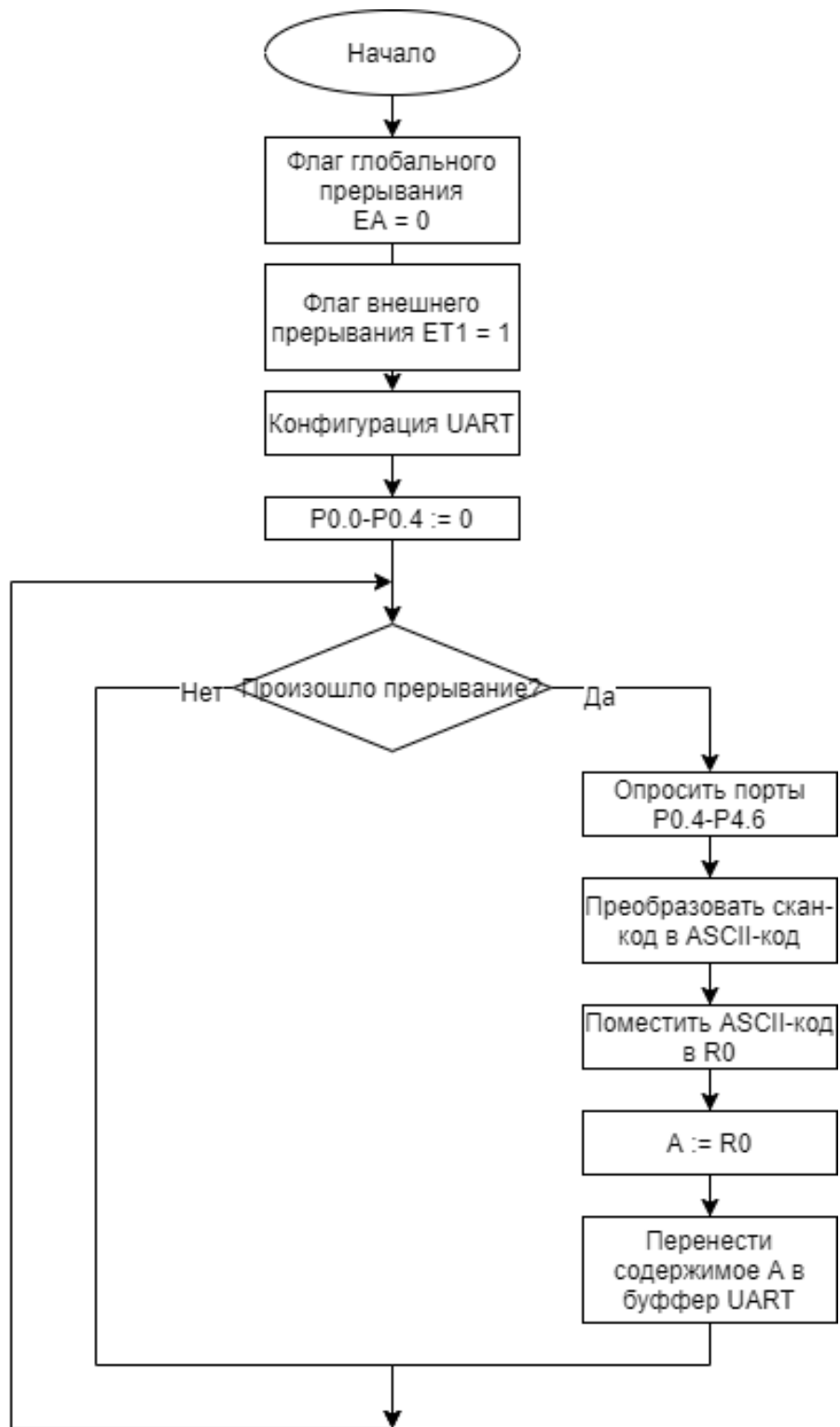


Рис. 2: Блок схема алгоритма работы

Приложение

Содержимое файла lab1irq.asm

```
ORG 0h
jmp start

ORG 13h
call setP0
call check
call resetP0
call printR0
CLR IE1
RETI

org 0030h
start:
    mov IE, #10001100b
    mov IP, #00000100b
    mov r5, #0h

    clr sm0
    setb sm1           ; формат кадра 8 бит

    clr a
    mov a, pcon
    setb acc.7
    mov pcon, a        ; установка скорости

    mov tmod, #20H      ; восьмибитный таймер 1 с авто сбросом
    mov th1, #243       ;
    mov tl1, #243       ; переполнение таймера каждые 13 микросекунд
    setb tr1
    call resetP0

wait:
    jmp wait

resetP0:
    CLR p0.0
    CLR p0.1
    CLR p0.2
    CLR p0.3
    ret

setP0:
    setb p0.0
    setb p0.1
    setb p0.2
    setb p0.3
    ret

check:
    MOV R0, #0
    SETB P0.3
    CLR P0.0
    CALL colScan4
    SETB P0.0
    CLR P0.1
    CALL colScan3
    SETB P0.1
```

```

    CLR P0.2
    CALL colScan2
    SETB P0.2
    CLR P0.3
    CALL colScan1
    RET

colScan1:
    JNB P0.4, gotKey3
    JNB P0.5, gotKey2
    JNB P0.6, gotKey1
    RET

colScan2:
    JNB P0.4, gotKey6
    JNB P0.5, gotKey5
    JNB P0.6, gotKey4
    RET

colScan3:
    JNB P0.4, gotKey9
    JNB P0.5, gotKey8
    JNB P0.6, gotKey7
    RET

colScan4:
    JNB P0.4, gotKeySharp
    JNB P0.5, gotKey0
    JNB P0.6, gotKeyAst
    RET

gotKey0:
    mov r0, #48
    RET

gotKey1:
    mov r0, #49
    RET

gotKey2:
    mov r0, #50
    RET

gotKey3:
    mov r0, #51
    RET

gotKey4:
    mov r0, #52
    RET

gotKey5:
    mov r0, #53
    RET

gotKey6:
    mov r0, #54
    RET

gotKey7:

```

```

    mov r0, #55
    RET

getKey8:
    mov r0, #56
    RET

getKey9:
    mov r0, #57
    RET

getKeyAst:
    mov r0, #42
    RET

getKeySharp:
    mov r0, #35
    RET

printR0:
    mov a, r0
        Jz print_exit
    mov sbuf, a
    jnb ti, $
    clr ti

print_exit:
    ret

```