**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Программирование микроконтроллеров**

**Тема: Обработка кнопок клавиатуры**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: ст. пр. кафедры ТК

Гольцов Юрий Александрович

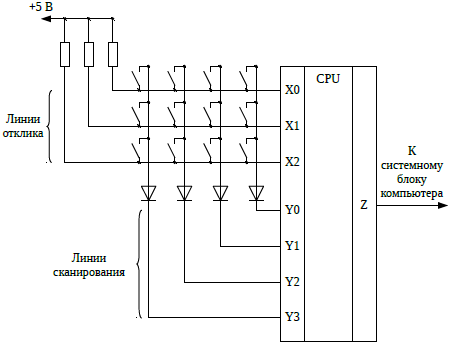
**Белгород 2020**

**Цель работы:** Необходимо написать программу для микроконтроллера i8051, которая позволяет считывать ASCI-код нажатой клавиши и отправлять его на другой контроллер с использованием УАПП.

**Краткие теоретические сведения**

8-битное АЛУ может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. В АЛУ имеются программно недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции и схема формирования признаков.

Если рассмотреть сильно упрощенную принципиальную схему клавиатуры (для простоты представлена клавиатура с небольшим количеством клавиш) можно заметить, что все клавиши находятся в узлах матрицы.



**Рис. 1.** Упрощённая схема клавиатуры

Вертикальные линии матрицы клавиатуры подключены через диоды к выходным линиям Y0...Y3 (линии сканирования) порта Y, который является выходным портом для контроллера в том плане, что контроллер может устанавливать на линиях, связанных с портом, сигналы низкого и высокого напряжения, т. е. логические ноль «0» и единицу «1». Диоды предназначены для предотвращения коротких замыканий при одновременном нажатии нескольких клавиш.

Горизонтальные линии матрицы соединены с линиями X0...X2 (линии отклика) входного порта X, значения каждой линии которого контроллер может считывать, определяя какое установлено на линии напряжение: соответствующее логическому «0» или «1». Кроме этого горизонтальные линии через подтягивающие резисторы соединены с напряжением питания +5 В, соответствующим логической «1», поэтому, когда ни одна клавиша не нажата и все контакты в узлах матрицы разомкнуты, на входах X0...X2 контроллера установлены «1».

В процессе функционирования контроллер устанавливает поочередно на каждой линии сканирования уровень напряжения логического «0», оставляя на остальных, кроме одной, уровень напряжения логической «1». Таким образом, если нажата клавиша, установка «0» на соответствующей вертикальной линии матрицы, приведет к появления сигнала «0» на соответствующей горизонтальной линии матрицы, что будет обнаружено контроллером, считывающим значения линий порта X. Зная на какой из сканирующих линий установлен в данный момент «0» и на какой линии отклика получен «0», контроллер клавиатуры определяет номер нажатой клавиши в матрицы. Также легко определяется, когда нажатая ранее клавиша отпускается.

Как только контроллер определил нажатие или отпускание клавиши, он посылает в центральный компьютер запрос на прерывание и номер клавиши в матрице, который однозначно зависит от схемы клавиатурной матрицы, но не от обозначений, нанесенных на поверхность клавиш. Этот номер называется скан-кодом (Scan Code). Слово scan «сканирование», подчеркивает тот факт, что клавиатурный компьютер сканирует клавиатуру для поиска нажатой клавиши.

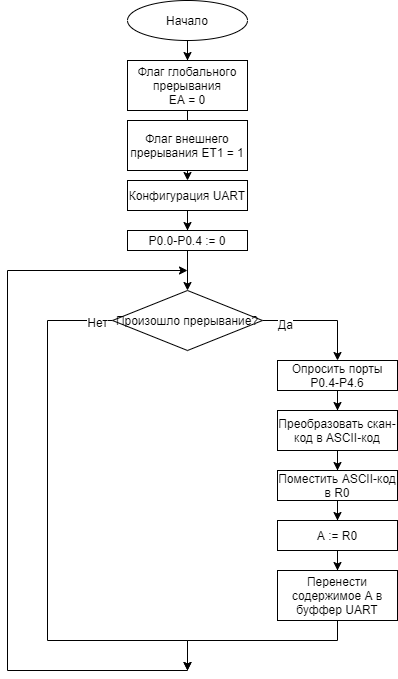
Обычно программе нужен не порядковый номер нажатой клавиши, а код, соответствующий обозначению на этой клавише, то есть код ASCII. Код ASCII не связан напрямую со скан-кодом, так как одной и той же клавише могут соответствовать несколько значений кода ASCII в зависимости от состояния других клавиш. Например, клавиша <1> используется для ввода символов '1' и '!', если она была нажата вместе с клавишей <Shift>. Все преобразования скан-кода в код ASCII выполняются программно.

Преобразование начинается после того, как клавиатурный процессор отошлет центральному процессору запрос на прерывание и скан-код клавиши. При этом центральный процессор прерывает исполняемую программу и переходит на подпрограмму обработки прерывания от клавиатуры. Эта подпрограмма формирует согласно значению скан-кода двухбайтовый код с последующей засылкой его в буфер ввода данных с клавиатуры, заполнение которого происходит по мере нажатия клавиш и никак не связано с выполняемой программой.

Задание к работе:

1. Создать программу для считывания скан-кода с клавиатуры, перевод его в ASCII код, и передачи полученного символа в терминал через UART путем постоянного опроса портов Р0, к которому подключена клавиатура.
2. Создать программу для считывания скан-кода с клавиатуры, перевод его в ASCII код, и передачи полученного символа в терминал через UART используя прерывания клавиатуры.

**Блок-схема разработанной программы**



**Код программы**

ORG 0h

jmp start

ORG 13h

call setP0

call check

call resetP0

call printRO

CLR IE1

RETI

org 0030h

start:

mov IE, #10001100b

mov IP, #00000100b

mov r5, #0h

clr sm0

setb sm1

clr a

mov a, pcon

setb acc.7

mov pcon, a

mov tmod, #20H

mov th1, #243

mov tl1, #243

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

gotKey8:

mov r0, #56

RET

gotKey9:

mov r0, #57

RET

gotKeyAst:

mov r0, #42

RET

gotKeySharp:

mov r0, #35

RET

printRO:

mov a, r0

Jz printExit

mov sbuf, a

jnb ti, $

clr ti

printExit:

ret

**Результат работы программы**

**Изображение выглядит как снимок экрана, компьютер, ноутбук

Автоматически созданное описание**