***МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ***

***Государственное образовательное учреждение высшего профессионального образования***

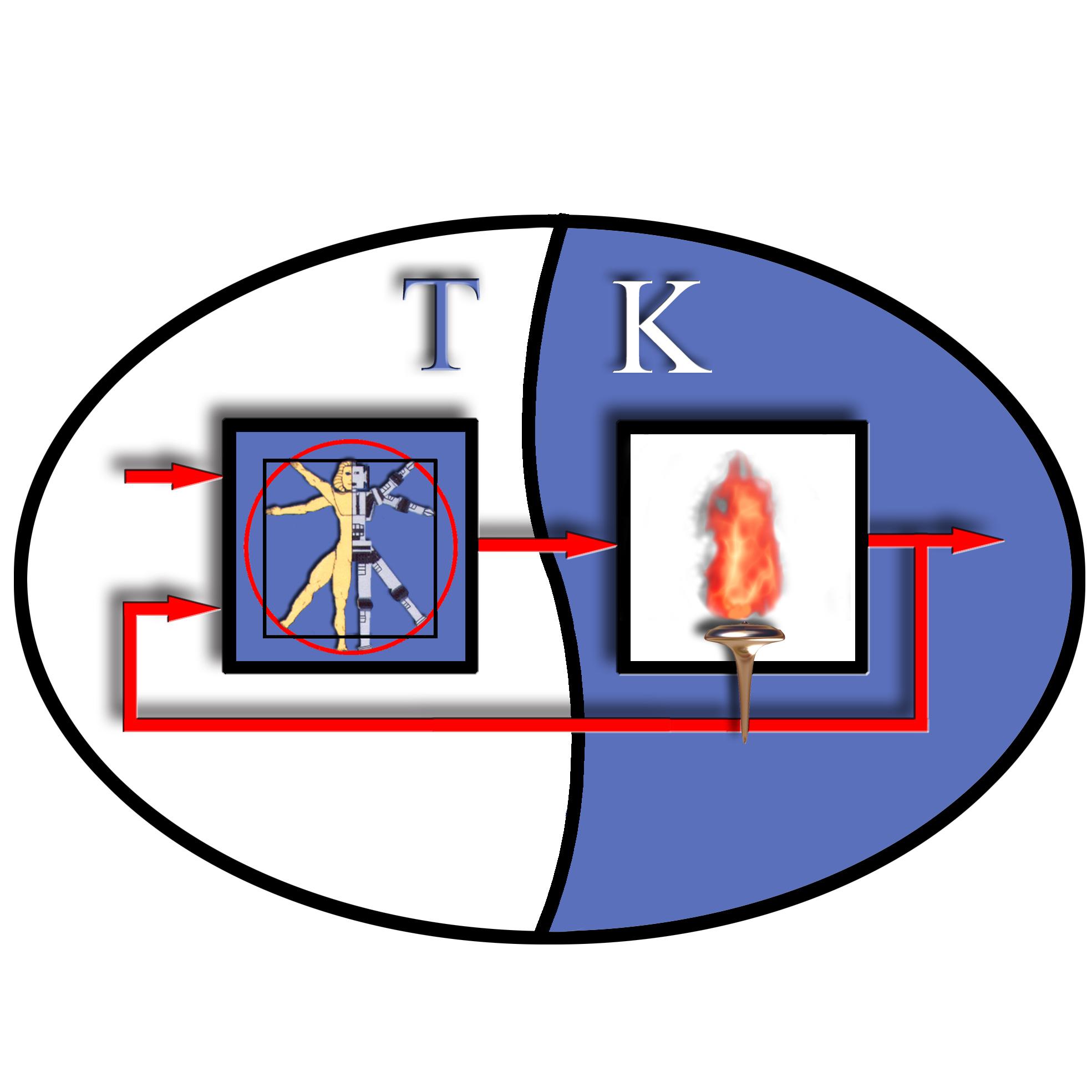
***«Белгородский государственный технологический университет***

***им. В.Г. Шухова»***

***(БГТУ им. В.Г. Шухова)***

***ИЭИТУС***

# *Кафедра «Техническая кибернетика»*



***Лабораторная работа №1***

*дисциплина:* ***«Микроконтроллеры в системах управления»***

***Выполнил:***

*студент группы МР-41*

*Славиогло Д.Ю.*

***Принял:***

*доц. Кижук А.С.*

*Белгород 2019*

Белгород 2011

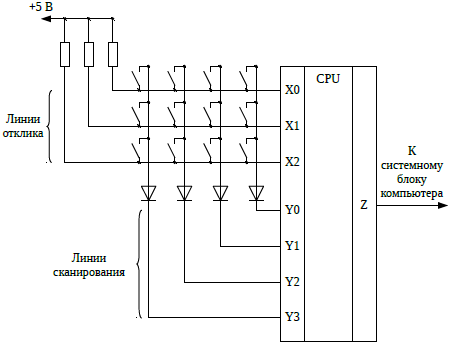
**Цель работы**

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

**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.

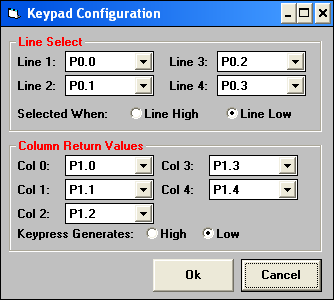
Горизонтальные линии матрицы соединены с линиями X0...X2 (линии отклика) входного порта X, значения каждой линии которого контроллер может считывать, определяя какое установлено на линии напряжение: соответствующее логическому «0» или «1». Кроме этого горизонтальные линии через подтягивающие резисторы 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.



**Рис. 2.** Настройки портов

**Код программы**

ORG 0h

MOV TMOD,#00100001b

;передача с частотой 9600гц

MOV TH1,#0FDh

;1-ый режим передачи

SETB SM1

CLR SM0

;опрос линии 0

L0:

;опрашиваем линию Р0.0

CLR P0.0

;изменение на линии P1 соответствует какому-то символу

CALL CH1

CALL CH2

CALL CH3

CALL CHC

CALL CHD

;заканчиваем опрашивать линию Р0.0

SETB P0.0

;если нажатий не было, то переход на следующую линию

SJMP L1

CH1:

JB P1.0, CH11

MOV A, 10h

CJNE A, #0h, ST1

;передача символа 1 на вывод

MOV A,#31h

CALL PR

MOV A, #01h

MOV 10h, A

SJMP ST1

CH11:

MOV 10h, #0h

ST1:

RET

CH2:

JB P1.1, CH21

MOV A, 11h

CJNE A, #0h, ST2

;передача символа 2 на вывод

MOV A,#32h

CALL PR

MOV A, #01h

MOV 11h, A

SJMP ST2

CH21:

MOV 11h, #0h

ST2:

RET

CH3:

JB P1.2, CH31

MOV A, 12h

CJNE A, #0h, ST3

;передача символа 3 на вывод

MOV A,#33h

CALL PR

MOV A, #01h

MOV 12h, A

SJMP ST3

CH31:

MOV 12h, #0h

ST3:

RET

CHC:

JB P1.3, CHC1

MOV A, 13h

CJNE A, #0h, STC

;передача символа C на вывод

MOV A,#43h

CALL PR

MOV A, #01h

MOV 13h, A

SJMP STC

CHC1:

MOV 13h, #0h

STC:

RET

CHD:

JB P1.4, CHD1

MOV A, 14h

CJNE A, #0h, STD

;передача символа D на вывод

MOV A,#44h

CALL PR

MOV A, #01h

MOV 14h, A

SJMP STD

CHD1:

MOV 14h, #0h

STD:

RET

;опрос линии 1

L1:

;опрашиваем линию Р0.1

CLR P0.1

;изменение на линии P1 соответсвует какому-то символу

CALL CH4

CALL CH5

CALL CH6

CALL CHE

CALL CHF

;заканчиваем опрашивать линию Р0.1

SETB P0.1

;если нажатий не было, то переход на следующую линию

SJMP L2

CH4:

JB P1.0, CH41

MOV A, 15h

CJNE A, #0h, ST4

;передача символа 4 на вывод

MOV A,#34h

CALL PR

MOV A, #01h

MOV 15h, A

SJMP STD

CH41:

MOV 15h, #0h

ST4:

RET

CH5:

JB P1.1, CH51

MOV A, 16h

CJNE A, #0h, ST5

;передача символа 4 на вывод

MOV A,#35h

CALL PR

MOV A, #01h

MOV 16h, A

SJMP ST5

CH51:

MOV 16h, #0h

ST5:

RET

CH6:

JB P1.2, CH61

MOV A, 17h

CJNE A, #0h, ST6

;передача символа 4 на вывод

MOV A,#36h

CALL PR

MOV A, #01h

MOV 17h, A

SJMP ST6

CH61:

MOV 17h, #0h

ST6:

RET

CHE:

JB P1.3, CHE1

MOV A, 18h

CJNE A, #0h, STE

;передача символа E на вывод

MOV A,#45h

CALL PR

MOV A, #01h

MOV 18h, A

SJMP STE

CHE1:

MOV 18h, #0h

STE:

RET

CHF:

JB P1.4, CHF1

MOV A, 19h

CJNE A, #0h, STD

;передача символа F на вывод

MOV A,#46h

CALL PR

MOV A, #01h

MOV 19h, A

SJMP STD

CHF1:

MOV 19h, #0h

STF:

RET

;опрос линии 2

L2:

;опрашиваем линию Р0.2

CLR P0.2

;изменение на линии P1 соответсвует какому-то символу

CALL CH7

CALL CH8

CALL CH9

CALL CHG

CALL CHH

;заканчиваем опрашивать линию Р0.2

SETB P0.2

;если нажатий не было, то переход на следующую линию

SJMP L3

CH7:

JB P1.0, CH71

MOV A, 1Ah

CJNE A, #0h, ST7

;передача символа 7 на вывод

MOV A,#37h

CALL PR

MOV A, #01h

MOV 1Ah, A

SJMP ST7

CH71:

MOV 1Ah, #0h

ST7:

RET

CH8:

JB P1.1, CH81

MOV A, 1Bh

CJNE A, #0h, ST8

;передача символа 8 на вывод

MOV A,#38h

CALL PR

MOV A, #01h

MOV 1Bh, A

SJMP ST8

CH81:

MOV 1Bh, #0h

ST8:

RET

CH9:

JB P1.2, CH91

MOV A, 1Ch

CJNE A, #0h, ST9

;передача символа 9 на вывод

MOV A,#39h

CALL PR

MOV A, #01h

MOV 1Ch, A

SJMP ST8

CH91:

MOV 1Ch, #0h

ST9:

RET

CHG:

JB P1.3, CHG1

MOV A, 1Dh

CJNE A, #0h, STG

;передача символа G на вывод

MOV A,#47h

CALL PR

MOV A, #01h

MOV 1Dh, A

SJMP STG

CHG1:

MOV 1Dh, #0h

STG:

RET

CHH:

JB P1.4, CHH1

MOV A, 1Eh

CJNE A, #0h, STH

;передача символа H на вывод

MOV A,#48h

CALL PR

MOV A, #01h

MOV 1Eh, A

SJMP STH

CHH1:

MOV 1Eh, #0h

STH:

RET

;опрос линии 3

L3:

;опрашиваем линию Р0.3

CLR P0.3

;изменение на линии P1 соответсвует какому-то символу

CALL CHA

CALL CH0

CALL CHB

CALL CHI

CALL CHJ

;заканчиваем опрашивать линию Р0.3

SETB P0.3

;если нажатий не было, то переход на следующую линию

LJMP L0

CHA:

JB P1.0, CHA1

MOV A, 1Fh

CJNE A, #0h, STA

;передача символа A на вывод

MOV A,#41h

CALL PR

MOV A, #01h

MOV 1Fh, A

SJMP STA

CHA1:

MOV 1Fh, #0h

STA:

RET

CH0:

JB P1.1, CH01

MOV A, 20h

CJNE A, #0h, ST0

;передача символа 0 на вывод

MOV A,#30h

CALL PR

MOV A, #01h

MOV 20h, A

SJMP ST0

CH01:

MOV 20h, #0h

ST0:

RET

CHB:

JB P1.2, CHB1

MOV A, 21h

CJNE A, #0h, STB

;передача символа B на вывод

MOV A,#42h

CALL PR

MOV A, #01h

MOV 20h, A

SJMP STB

CHB1:

MOV 20h, #0h

STB:

RET

CHI:

JB P1.3, CHI1

MOV A, 21h

CJNE A, #0h, STI

;передача символа I на вывод

MOV A,#49h

CALL PR

MOV A, #01h

MOV 21h, A

SJMP STI

CHI1:

MOV 21h, #0h

STI:

RET

CHJ:

JB P1.4, CHJ1

MOV A, 22h

CJNE A, #0h, STJ

;передача символа B на вывод

MOV A,#4Ah

CALL PR

MOV A, #01h

MOV 22h, A

SJMP STJ

CHJ1:

MOV 22h, #0h

STJ:

RET

;вывод

PR:

MOV SBUF,A

;переход на новую строку

MOV A,#13

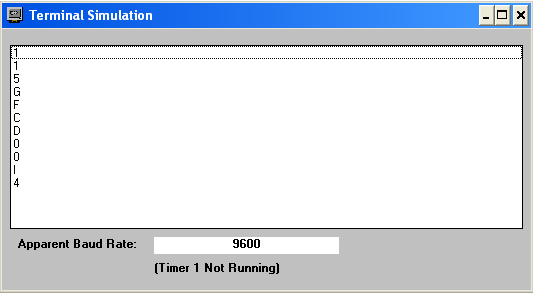
MOV SBUF,A

MOV A,#10

MOV SBUF,A

RET

END



**Рис. 3.** Пример работы

**Вывод**

В ходе выполнения данной лабораторной работы были написана программа для микроконтроллера i8051, которая позволяет считывать ASCI-код нажатой клавиши и отправлять его на другой контроллер с использованием УАПП.