ГУАП

КАФЕДРА 44

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц, канд. тех. наук |  |  |  | Т.Н. Соловьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАЗРАБОТКА МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ СУСТРОЙСТВАМИ ВВОДА И ВЫВОДА |
| по курсу: МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4941 |  |  |  | Н.С. Горбунов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы:** приобретение навыков разработки микропроцессорной системы, оснащенной устройствами ввода и вывода информации.**Задание на лабораторную работу**

Требуется, используя результаты выполнения двух предыдущих лабораторных работ, разработать микропроцессорную систему, выводящую на экране ЖКИ две строки.

|  |  |  |
| --- | --- | --- |
| Номер варианта | Строка 1 | Строка 2 |
| 9 | N. S. Gorbunov | Row = X Col = 1 |

Листинг 1.

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \*

; Filename: LW5.asm

; Date: 2022/03/08

; File Version: 1

; Author: Gorbunov N. S.

; Company: SUAI

; Description: LW5

; \*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Variables

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

row\_n equ 41h

col\_n equ 42h

N equ 40h ;номер нажатой клавиши

map\_start equ 28h ;начало области хранения КС клавиатуры

switch equ 43h ;переключатель «команда-данные» (RS)

bte equ 44h ;выдаваемый на ЖКИ байт

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Reset Vector

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

org 0h ; processor reset vector

ajmp start ; go to beginning of program

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; MAIN PROGRAM

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

org 100h

start:

lcall indic\_init

loop: lcall klav ;получаем номер нажатой клавиши

mov switch,#0 ;переключатель – команда

mov bte,#0c9h ;позиция вывода первого символа

lcall indic\_wr

mov a,row\_n ;строка

mov b,#10

div ab ;получаем единицы

mov a, b

add a,#30h ;т.к. код цифры = цифра+30h по таблице

;кодов символов ЖКИ

mov switch,#1 ;переключатель – данные

mov bte,a

lcall indic\_wr

mov switch,#0 ;переключатель – команда

mov bte,#0d1h ;позиция вывода первого символа

lcall indic\_wr

mov a,col\_n ;строка

mov b,#10

div ab ;получаем единицы

mov a, b

;символ

add a,#30h

mov switch,#1 ;переключатель – данные

mov bte,a

lcall indic\_wr

jmp loop ;зацикливание опроса

;формирование КС

;установка "0" в начальные позиции

klav: mov a, #07Fh ;подготовка "бегущего нуля " (01111111)

mov r0, #map\_start ;адрес начала карты состояние

opros: mov p0, a ;"бегущего нуля" в порт 0

setb p0.1 ;настройка разрядов порта P1 на чтение 1

setb p0.0; 1

mov b, p0 ;чтение;1

anl b, #07h ;выделение значащих разрядов (у нас 3

;младших разряда, поэтому умножаем на

;00000111)

mov @r0, b ;записываем стоку карты

setb c ;подготовка нового опроса сдвиг "0"в

rrc a ;следующую позицию

inc r0 ;переходим к следующей ячейке КС

cjne a, #11111011b, opros

;b указывает на двоичный код

;пока ноль не сдвинется в перенос

;дешифрация карты

mov r0, #map\_start

dc: mov a, @r0 ;читаем очередную стоку карты

cjne a, #07h, dck ;если в значащих разрядах есть ноль

;(нажата клавиша), переходим к dck

inc r0 ;если не нажата - просмотр карты далее

cjne r0, #(map\_start+8), dc

;пока не закончились строки

mov row\_n, #4 ;если клавиша не нажата, устанавливаем

mov col\_n, #6 ;несуществующие значения

sjmp end1 ;и переходим в конец

dck: mov a, r0 ;клавиша нажата: в R0 – адрес ячейки

clr c

subb a, #map\_start

;вычитаем нач. адр. КС, чтобы узнать

mov col\_n, a ;номер строки

mov a, @r0 ;берем содержимое ячейки КС для

mov row\_n, #0 ;определения № столбца (сначала № = 0)

dloop1: rrc a ;последовательно сдвигаем вправо, т.к.

;значащие разряды - младшие

jnc end1 ;пока не ноль вытиснится в перенос

inc row\_n

mov r1, row\_n

cjne r1, #5, dloop1

;пока не сдвинем 5 раз

end1: lcall get\_num ;вызов подпрограммы опред. номера

ret;переход на конец программы

get\_num: push a ;спасаем аккумулятор

mov a, col\_n

cjne a, #8, gn\_end

mov N, #0 ;если col\_n = 8, то ничего не нажато

pop a

ret

gn\_end: clr c

mov b, #3d

MUL AB

;умножаем col\_n на 3 (т.к. 3 столбца)

add a, row\_n

inc a

mov N, a

pop a

ret

indic\_init: mov switch, #0;переключатель уст-ть на команду (RS=0)

mov bte, #38h ;байт – команда

lcall indic\_wr ;вызов подпрограммы передачи в ЖКИ

mov bte, #0ch ;активация всех знакомест

lcall indic\_wr

mov bte, #06h ;режим автом. перемещения курсора

lcall indic\_wr

mov bte, #83h ;установка адреса первого символа

lcall indic\_wr

;вывод строк

mov switch, #1 ;переключатель – данные (RS=1)

mov dptr, #0fd0h ;адрес, по которому расположены данные

;(см. конец программы)

indic\_data\_wr1: ;вывод символов первой строки

clr a

movc a, @a+dptr

ind\_row1: mov bte, a ;передаваемый байт – код символа

lcall indic\_wr

inc dptr

mov a, dpl ;младший байт указателя данных

cjne a, #0ddh, indic\_data\_wr1

;пока не введены 13 символов 1ой строки

mov switch, #0 ;RS=0 – команда

mov bte, #0C3h ;установка адреса первого символа

lcall indic\_wr ;второй строки

mov switch, #1 ;RS=1 - данные

indic\_data\_wr2: ;вывод символов второй строки

clr a

movc a, @a+dptr

ind\_row2: mov bte, a

lcall indic\_wr

inc dptr

mov a, dpl

cjne a, #0ECh, indic\_data\_wr2

;0DDh+0Fh=ECh – адр. конца второй стр.

ret ;переход на конец программы

;подпрограмма передачи в ЖКИ

indic\_wr: mov p2, bte ;передаваемый байт – в Р2

setb p1.6 ;E:=1

clr p1.4 ;RW:=0 (запись)

mov a, switch

mov c, acc.0 ;нам нужен 0-ой бит аккумулятора

mov p1.5, c ;RS:=switch (команда/данные)

lcall indic\_delay ;вызов подпрограммы задержки

clr p1.6 ;E:=0

lcall indic\_delay

setb p1.6 ;E:=1

ret

indic\_delay: ;подпрограмма задержки на 40мкс

push A ;сохраняем аккумулятор в стеке

mov A, #0Ah ; 40 = 2+2+1+А(1+2)+1+2+2

m: dec A

jnz m

nop

pop A ;восстанавливаем значение аккумулятора

ret

;данные располагаем в памяти программ

org 0FD0h

data: db 'N.S. Gorbunov'

db 'ROW = X COL = X' ;директива db помещает коды

;символов в последовательные ячейки памяти программ

finish: sjmp $ ;конец программы

end

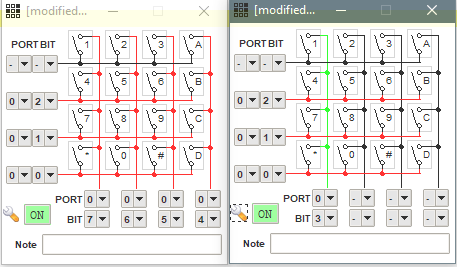


Рисунок 1. Настройки массива кнопок

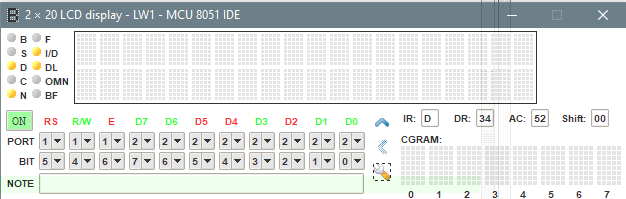


Рисунок 2. Настройки ЖК дисплея

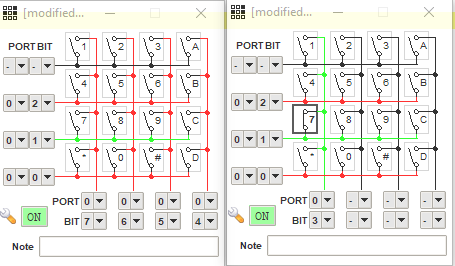


Рисунок 3. Входные данные

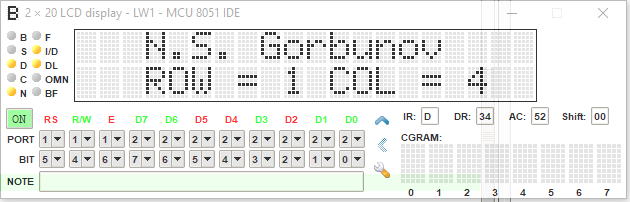


Рисунок 4. Результат работы программы

**Вывод:** в результате выполнения работы разработана программа на языке ассемблера SAB 80С515 для вывода на экране ЖКИ двух заданных строк. Программа выполняет опрос кнопок для отображения номеров строки и столбца нажатой кнопки на ЖКИ. Проверка работоспособности программы произведена в среде MCU 8051 IDE. Приобретены навыки организации взаимодействия микроконтроллера с устройствами вывода и вывода на примере жидкокристаллического знакосинтезирующего дисплея и массива кнопок 3х5.