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 СИСПОЛЬЗОВАНИЕМ ВНЕШНИХ ПРЕРЫВАНИЙ |
| по курсу: МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2022

**Цель работы:** изучение принципов работы системы прерываний микроконтроллера; приобретение навыков разработки микропроцессорных систем, использующих внешние прерывания.

**Задание на лабораторную работу**

|  |  |  |
| --- | --- | --- |
| Номер варианта | Кнопка | Событие |
| 9 | INT0 | На второй строке выводится число нажатий на кнопку |

Листинг 1.

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

; \*

; Filename: LW6.asm

; Date: 2022/03/30

; File Version: 1

; Author: Gorbunov N. S.

; Company: SUAI

; Description: LW6

; \*

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

; Variables

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

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

org 0003h ; processor interrupt vector

ajmp int\_0 ; go to int0 interrupt service routine

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

; MAIN PROGRAM

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

org 100h

start:

setb EX0 ;Разрешение прерывания от int0

setb EA

mov r0, #0h

jmp indic

loop:

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

mov bte, #1Ch ;байт – команда 18 влево 1C вправо

lcall indic\_wr ;вызов подпрограммы передачи в ЖКИ

setb P1.2 ;Ожидание выхода

jnb P1.2, finish

sjmp loop

finish:

sjmp $ ;конец программы

indic: 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, #80h ;установка адреса первого символа93

lcall indic\_wr

;вывод строк

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

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

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, #0Efh, indic\_data\_wr1

;пока не введены 26 символов строки

mov bte, #100b ;режим автом. перемещения курсора

lcall indic\_wr

jmp loop

indic\_row2:

push a

push p1

push p2

push bte

push switch

mov bte, #0c3h ;установка адреса первого символа

lcall indic\_wr;вывод строк

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

mov a, r0

add a, #30h

mov bte, a ;передаваемый байт – код символа

lcall indic\_wr

pop switch

pop bte

pop p2

pop p1

pop a

ret

;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

;Обработчик прерывания INT0

int\_0:

clr EX0 ;запрет INT0

chatter:

lcall indic\_delay

jnb p3.2, chatter

inc r0

lcall indic\_row2

setb EX0

reti

org 0FD0h

data: db 'Gorbunov Nikita Sergeevich 4941' ;директива db помещает коды 26 символов

end

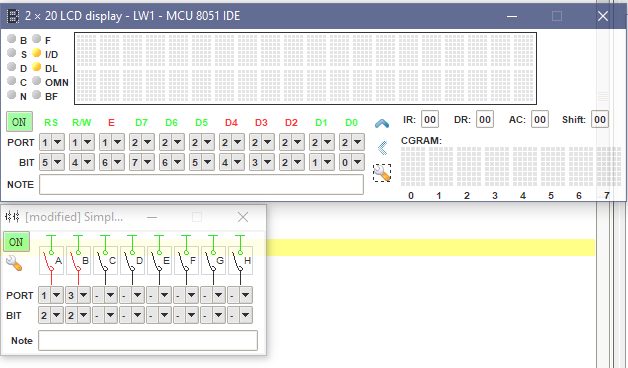


Рисунок 1. Настройки дисплея и кнопок

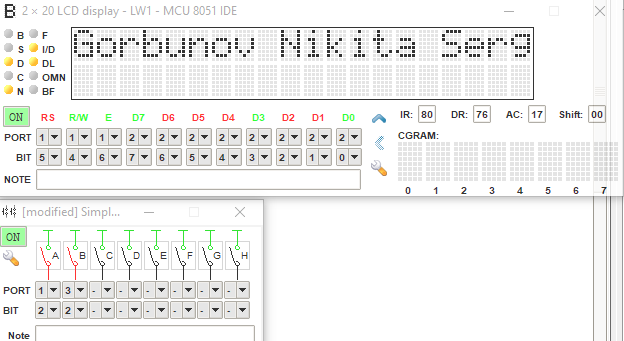


Рисунок 2. Заполнение строки

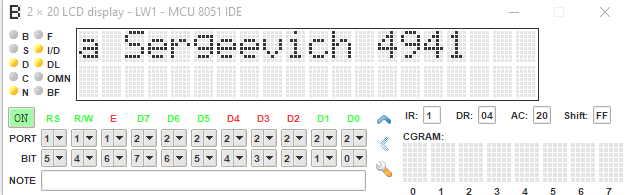


Рисунок 3. Бегущая строка

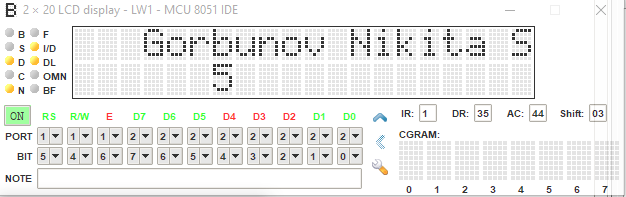


Рисунок 4. Вывод количества нажатий

**Вывод:** в результате выполнения работы создана программа на языке ассемблера SAB 80С515 для системных прерываний, при этом стирается отчество из движущейся строки ЖКИ. Проверка работоспособности программы произведена в среде MCU 8051 IDE. Изучены принципы работы системы прерываний микроконтроллера, приобретены навыки разработки микропроцессорных систем, использующих внешние прерывания.