

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

**Отчёт**  
**по лабораторной работе №5**  
**«Асинхронный обмен данными с ВУ»**

Вариант №806

Выполнил: Качанов Д.В., группа Р3106

Преподаватель: Афанасьев Д.Б.

Санкт-Петербург

~ 2023 ~

## Оглавление

Задание .....	3
Код составленной программы .....	3
Описание программы.....	4
Строка для трассировки.....	6
Трассировка .....	7
Вывод.....	8

## Задание

По выданному преподавателем варианту разработать программу асинхронного обмена данными с внешним устройством. При помощи программы осуществить ввод или вывод информации, используя в качестве подтверждения данных сигнал (кнопку) готовности ВУ.

- 1) Программа осуществляет асинхронный ввод данных с ВУ-3
- 2) Программа начинается с адреса  $57F_{16}$ . Размещаемая строка находится по адресу  $563_{16}$ .
- 3) Строка должна быть представлена в кодировке КОИ-8.
- 4) Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП\_СИМВ.
- 5) Ввод или вывод строки должен быть завершен по символу с кодом 0D (CR). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

## Код составленной программы

```
ORG 0x57F
START: LD RS      ;Чтение адреса первой ячейки результата
      ST R       ;Сохранение значения аккумулятора в ячейку указателя на результат
      CLA       ;Очистка аккумулятора
s1:    IN 0x7     ;Проверка готовности ВУ
      AND #0x40  ;Проверка наличия символа
      BEQ s1     ;Возврат к проверке готовности ВУ, если символа нет
      IN 0x6     ;Чтение символа в младший байт аккумулятора
      SWAB      ;Перемещение символа в старший байт аккумулятора
      ST (R)     ;Сохранение значения аккумулятора в ячейку результата
      ST T       ;Сохранение значения аккумулятора в ячейку временных данных
      SWAB      ;Перемещение символа в младший байт аккумулятора
      CMP S      ;Проверка, является ли введенный символ Стоп символом
      BEQ H      ;Переход на останов, если введенный символ это Стоп символ
      CLA       ;Очистка аккумулятора
s2:    IN 0x7     ;Проверка готовности ВУ
      AND #0x40  ;Проверка наличия символа
      BEQ s2     ;Возврат к проверке готовности ВУ, если символа нет
      LD (R)     ;Загрузка первого символа в аккумулятор (в старший байт)
      IN 0x6     ;Чтение символа в младший байт аккумулятора
      ST (R)+    ;Сохранение аккумулятора с двумя символами в ячейку результата
      SUB T      ;Удаление из аккумулятора первого символа
      CMP S      ;Проверка, является ли введенный символ Стоп символом
      BEQ H      ;Переход на останов, если введенный символ это Стоп символ
      JUMP s1    ;Переход на ожидание первого символа
H:     HLT       ;Останов
RS:    WORD 0x563 ;Указатель на адрес первой ячейки размещаемой строки
R:     WORD 0x0   ;Указатель на результат
S:     WORD 0x0D  ;Стоп символ
T:     WORD 0x00  ;Ячейка, используемая программой
```

## Описание программы

**Назначение программы** – реализация посимвольного асинхронного ввода символов из ВУ-3 и сохранение их в указанную область памяти. Прекращение ввода осуществляется после ввода Стоп символа.

### Описание и назначение исходных данных и результата:

Ячейки	Назначение	ОПИ	ОДЗ
	Строка, вводимая пользователем <i>Исходные данные</i>	Массив длины $L$ пар символов $K_i$ Каждый символ в 8-битной кодировке	Каждый символ $K_i$ : $0x0 \leq K_i \leq 0xFF$
<b>59A</b>  <i>RS</i>	Адрес начала хранимой строки  <i>Исходное данное</i>	11-битное беззнаковое число	$0x59E \leq RS \leq 0x7FF$  <i>или</i> $0x0 \leq RS \leq 0x57E$
	Длина строки	$2L$	Если $0x0 \leq RS \leq 0x57E$ , то $0 \leq L \leq 0x57E - RS$ Если $0x59E \leq RS \leq 0x7FF$ , то $0 \leq L \leq 0x59E + 0x7FF - RS$

<b><i>[RS; RS+L]</i></b>	<b><i>Результат</i></b>	Пары символов в формате:  АДР1: СИМВ1 СИМВ2  АДР2: СИМВ3 СИМВ4  ... СТОП_СИМВ	Каждый символ $K_i$ :  $0x0 \leq K_i \leq 0xFF$
--------------------------	-------------------------	---	---

***Расположение в памяти ЭВМ:***

- *Программы:* ячейки 57F– 599
- *Используемых программой ячеек памяти:* ячейки 59A-59D
- *Результата:* ячейки [RS; RS+L]

***Первая исполняема команда - 57F, последняя – 599***

# Строка для трассировки

GET OUT - известный...

КОИ-8

G	E	T		O	U	T		-		и	з	в	е	с	т	н	ы	й	.	.	.	
47	45	54	20	4f	55	54	20	2d	20	e8	e7	e2	e5	f1	f2	ed	fb	e9	2e	2e	2e	od

UTF-8

G	E	T		O	U	T		-		и	з	в	е	с	т	н
47	45	54	20	4f	55	54	20	2d	20	d0b8	d0b7	d0b2	d0b5	d181	d182	d0bd

ы	й	.	.	.	
d18b	d0b9	2e	2e	2e	0d

UTF-16

G	E	T		O	U	T		-		и	з	в	е
0047	0045	0054	0020	004f	0055	0054	0020	002d	0020	0438	0437	0432	0435

с	т	н	ы	й	.	.	.	
0441	0442	043d	044b	0439	002e	002e	002e	0d

## Трассировка

Выполняемая команда		Содержимое регистров процессора после выполнения команды								Ячейка, содержимое которой изменилось	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
57F	AE18	580	AE18	598	0563	000	0018	0563	0000		
580	EE18	581	EE18	599	0563	000	0018	0563	0000	599	0563
581	0200	582	0200	581	0200	000	0581	0000	0100		
582	1207	583	1207	582	1207	000	0582	0040	0100		
583	2F40	584	2F40	583	0040	000	0040	0040	0000		
584	F0FD	585	F0FD	584	F0FD	000	0584	0040	0000		
585	1206	586	1206	585	1206	000	0585	0047	0000		
586	0680	587	0680	586	0680	000	0586	4700	0000		
587	E811	588	E811	563	4700	000	0011	4700	0000	563	4700
588	EE12	589	EE12	59B	4700	000	0012	4700	0000	59B	4700
589	0680	58A	0680	589	0680	000	0589	0047	0000		
58A	7E0F	58B	7E0F	59A	000D	000	000F	0047	0001		
58B	F00B	58C	F00B	58B	F00B	000	058B	0047	0001		
58C	0200	58D	0200	58C	0200	000	058C	0000	0101		
58D	1207	58E	1207	58D	1207	000	058D	0040	0101		
58E	2F40	58F	2F40	58E	0040	000	0040	0040	0001		
58F	F0FD	590	F0FD	58F	F0FD	000	058F	0040	0001		
590	A808	591	A808	563	4700	000	0008	4700	0001		
591	1206	592	1206	591	1206	000	0591	4745	0001		
592	EA06	593	EA06	563	4745	000	0006	4745	0001	599 563	0564 4745

## **Вывод**

В ходе выполнения данной лабораторной работы была изучена система ввода-вывода БЭВМ.