Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

**по лабораторной работе №5**

**«Асинхронный обмен данными с ВУ»**

по дисциплине «Основы профессиональной деятельности»

вариант 760

Выполнил: Тахватулин М. В., группа Р3107

Преподаватель: Ткешелашвили Н.М.

г. Санкт-Петербург

~ 2023 ~

**Оглавление**

[**Задание 1** 2](#_Toc127327027)

[**Описание программы** 3](#_Toc127327028)

# **Задание 1**

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 4F916. Размещаемая строка находится по адресу 62F16.
3. Строка должна быть представлена в кодировке Windows-1251.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу c кодом 0A (NL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

# **Программа**

ORG 0x4F9 ; начало

RES: WORD 0x62F ; результат

FINISH: WORD 0x0A ; символ окончания

TEMP: WORD ? ; запись нечет

START: CLA ; очистка аккум

S1: IN 7 ; ввод первого символа из пары

AND #0x40 ; проверка на 1 в статусном регистре

BEQ S1 ; спин луп если 0 в ст. регистре

IN 6 ; байт в аккум

ST (RES) ; символ в результат

ST TEMP ; держим в памяти

CMP FINISH ; проверка на символ окончания

BEQ EXIT ; выход если окончание

CLA ; очистка аккум

S2: IN 7 ; ввод второго символа из пары

AND #0x40 ; проверка на 1 в статусном регистре

BEQ S2 ; спин луп если 0 в ст. регистре

IN 6 ; байт в аккум

SWAB ; скидываем второй символ в старший байт

OR TEMP ; соединяем с первым символом

ST (RES) ; сохраняем в результат

SUB TEMP ; убираем первый символ

SWAB ; возвращаем второй символ в младший байт

CMP FINISH ; проверка на символ окончания

BEQ EXIT ; выход если окончание

LD (RES)+ ; ссылка на результат+1

CLA ; очистка аккум

JUMP S1 ; прыгаем в начало

EXIT: LD (RES)+ ; ссылка на результат+1

HLT ; остановка

# **Описание программы**

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-3, посимвольно записывает их в память. Программа будет продолжать вводить символы, пока не будет введено 0A.

**Область допустимых значений:**

Максимально можно ввести 2\*(2047(7FF) – 1583(62F) + 1273(4F9) – 1(0A)) = 3472 символа.

**Слово**: СИРЕНЬ

**В кодировке Windows-1251**: D1 C8 D0 C5 CD DC

**В кодировке UTF-8**: D0A1 D098 D0A0 D095 D09D D0AC

**В кодировке UTF-16**: U+0421 U+0418 U+0420 U+0415 U+041D U+042C

Изображение выглядит как текст, число, прямоугольный, кроссворд

Автоматически созданное описание

# **Трассировка**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адр | Знчн |
| 4FC | 0200 | 4FC | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 004 | 0100 |  |  |
| 4FC | 0200 | 4FD | 0200 | 4FC | 0200 | 000 | 04FC | 0000 | 004 | 0100 |  |  |
| 4FD | 1207 | 4FE | 1207 | 4FD | 1207 | 000 | 04FD | 0000 | 004 | 0100 |  |  |
| 4FE | 2F40 | 4FF | 2F40 | 4FE | 0040 | 000 | 0040 | 0000 | 004 | 0100 |  |  |
| 4FF | F0FD | 4FD | F0FD | 4FF | F0FD | 000 | FFFD | 0000 | 004 | 0100 |  |  |
| 4FD | 1207 | 4FE | 1207 | 4FD | 1207 | 000 | 04FD | 0040 | 004 | 0100 |  |  |
| 4FE | 2F40 | 4FF | 2F40 | 4FE | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 4FF | F0FD | 500 | F0FD | 4FF | F0FD | 000 | 04FF | 0040 | 000 | 0000 |  |  |
| 500 | 1206 | 501 | 1206 | 500 | 1206 | 000 | 0500 | 00D1 | 000 | 0000 |  |  |
| 501 | E8F7 | 502 | E8F7 | 62F | 00D1 | 000 | FFF7 | 00D1 | 000 | 0000 | 62F | 00D1 |
| 502 | EEF8 | 503 | EEF8 | 4FB | 00D1 | 000 | FFF8 | 00D1 | 000 | 0000 | 4FB | 00D1 |
| 503 | 7EF6 | 504 | 7EF6 | 4FA | 000A | 000 | FFF6 | 00D1 | 000 | 0001 |  |  |
| 504 | F00F | 505 | F00F | 504 | F00F | 000 | 0504 | 00D1 | 000 | 0001 |  |  |
| 505 | 0200 | 506 | 0200 | 505 | 0200 | 000 | 0505 | 0000 | 000 | 0101 |  |  |
| 506 | 1207 | 507 | 1207 | 506 | 1207 | 000 | 0506 | 0040 | 000 | 0101 |  |  |
| 507 | 2F40 | 508 | 2F40 | 507 | 0040 | 000 | 0040 | 0040 | 000 | 0001 |  |  |
| 508 | F0FD | 509 | F0FD | 508 | F0FD | 000 | 0508 | 0040 | 000 | 0001 |  |  |
| 509 | 1206 | 50A | 1206 | 509 | 1206 | 000 | 0509 | 00C8 | 000 | 0001 |  |  |
| 50A | 0680 | 50B | 0680 | 50A | 0680 | 000 | 050A | C800 | 000 | 1001 |  |  |
| 50B | 3EEF | 50C | 3EEF | 4FB | 00D1 | 000 | 372E | C8D1 | 000 | 1001 |  |  |
| 50C | E8EC | 50D | E8EC | 62F | C8D1 | 000 | FFEC | C8D1 | 000 | 1001 | 62F | C8D1 |
| 50D | 6EED | 50E | 6EED | 4FB | 00D1 | 000 | FFED | C800 | 000 | 1001 |  |  |
| 50E | 0680 | 50F | 0680 | 50E | 0680 | 000 | 050E | 00C8 | 000 | 0001 |  |  |
| 50F | 7EEA | 510 | 7EEA | 4FA | 000A | 000 | FFEA | 00C8 | 000 | 0001 |  |  |
| 510 | F003 | 511 | F003 | 510 | F003 | 000 | 0510 | 00C8 | 000 | 0001 |  |  |