Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №5

по дисциплине «Системы реального времени»

Выполнил: ст. гр. 45

Посевин Р.Р.

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар,

2021

**Лабораторная работа №5**

**Тема**

Организация обмена между датчиками, УВМ и исполнительными устройствами.

**Цель**

Изучение принципов организации инфообмена между ядром СРВ и периферийными устройствами; приобретение практических навыков разработки функциональных схем, алгоритмов и управляющих программ для управления технологическими процессами с использованием измерительных преобразователей и исполнительных устройств.

**Задание**

Вариант 10.

Управление магистральным агрегатом нефтепровода.

Транспортировка нефти по трубе обеспечивается магистральным агрегатом (МА), включающим в себя 2 бустерных насоса (основной и дополнительный), исполнительным механизмом которых выступает электродвигатель, управляемый дискретно.

Давление нефти в трубе на участке МА контролируется четырьмя аналоговыми датчиками (по 2 датчика до и после МА) с диапазоном измерения 0…20 атм и диапазоном выходного сигнала 0…320 мВ. Датчики опрашиваются поочередно с интервалом в 20 мс: сначала датчики до МА, затем через 0,4 с –после МА.

Если среднее давление в трубе до МА ниже 10,8 атм, то работает основной насос. При этом если разность среднего давления в трубе до МА и среднего давления в трубе после МА более 0,8 атм, то работает резервный насос.

Разрядность АЦП – 11.

Входной сигнал – -0,5 … 0,5 В.

Порядок выполнения

1. Сформировать список устройств для устройства связи с объектом (УСО).

2. Построить градуировочную характеристику для каждого аналогового датчика (при наличии).

3. Рассчитать двоичные эквиваленты контрольных и управляющих непрерывных величин.

4. Определить количество, разрядность и назначение портов ввода-вывода.

5. Определить назначение отдельных разрядов портов.

6. Разработать структурную схему интерфейса связи.

7. Разработать алгоритм функционирования объекта.

8. Рассчитать параметры процедуры временной задержки (при необходимости).

9. Составить программу управления (драйвер) на языке ассемблера.

10. На языке высокого уровня разработать приложение, в графическом режиме имитирующее функционирование СРВ. Обеспечить индикацию контрольных и управляющих величин, а также управление изменением режима работы объекта в режиме реального времени.

**Ход работы**

Построим градуировочную характеристику для датчиков давления нефти в трубе (рисунок 1).



Рисунок 1 – Градуировочная характеристика для датчика давления нефти в трубе

Рассчитаем контрольный уровень напряжения на выходе датчика давления, соответствующего давлению 10,8 атм и 0,8 атм:

U = k\*(T–Тн) + Uн; k = tg

K = tg = (320 + 320) / (20–0) = 32

U10,8 = 32 \* (10,8–0) - 320 = 25,6 мВ = 0,0256 В

U0,8 = 32 \* (0,8–0) - 320 = -294,4 мВ = -0,2944 В

Посчитаем цифровой эквивалент, соответствующий значению 0,0256 В и -0,2944 В:

D16 = 434

Аналогично для -0,2944 мВ:

D16 = 1A5

Далее рассмотрим список устройств для устройства связи с объектом.

1) 4 датчика давления нефти в трубе

2) Электродвигатель для основного насоса

3) Электродвигатель для дополнительного насоса

4) Коммутатор

5) АЦП

Так как коммутатор связан с 4 аналоговыми датчиками давления, в выходном порту 301h для него необходимо выделать 2 бита.

Разработаем структурную схему устройства связи с объектом (рисунок 2).

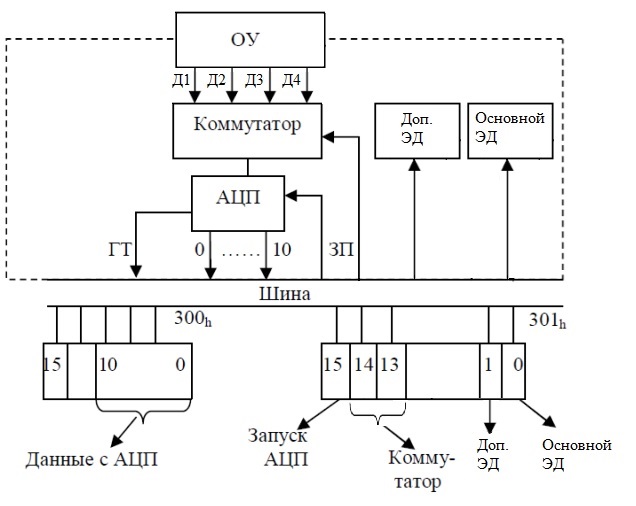


Рисунок 2 – Структурная схема интерфейса связи

Рассмотрим, какие порты вывода будут использоваться и для каких систем.

300h — Получение данных с АЦП

301h — Переключение коммутатора, запуск АЦП, включение/выключение основного и дополнительного электродвигателя.

Для выполнения необходимых временных задержек реализованы подпрограммы DELAY\_20\_MSEC и DELAY\_400\_MSEC для выполнения задержек в 20 и 400 мс соотвественно.

Программа, обеспечивающая выполнение поставленной задачи, приведена далее.

.686

include \masm32\include\io.asm

.data

d1 DW ?

d2 DW ?

d3 DW ?

d4 DW ?

avg\_before DW ?

avg\_after DW ?

.code

DELAY:

MOV BX, 4h

L1:

MOV CX, 0F2h

NOP

NOP

NOP

NOP

L2:

NOP

NOP

NOP

NOP

NOP

NOP

LOOP L2

DEC BX

JNZ L1

RET

ONE\_MSEC:

MOV DX, 100

L3:

CALL DELAY

DEC DX

JNZ L3

RET

DELAY\_20\_MSEC:

MOV DX, 20

L4:

PUSH DX

CALL ONE\_MSEC

POP DX

DEC DX

JNZ L4

RET

DELAY\_400\_MSEC:

MOV DX, 400

L5:

PUSH DX

CALL ONE\_MSEC

POP DX

DEC DX

JNZ L5

RET

LStart:

Met\_0:

mov ax, 0

mov dx, 301h ; Вывод 0 на коммутатор

out dx, ax

or ax, 8000h ; Запуск АЦП

out dx, ax

mov dx, 300h

D\_1:

in ax, dx ; Опрос порта с адресом 300h по флагу

test ax, 8000h

jz D\_1

and ax, 07FFh ; Выделение информационных разрядов

mov d1, ax

call DELAY\_20\_MSEC

; Вывод 1 на коммутатор

mov ax, 2000h

mov dx, 301h

out dx, ax

or ax, 8000h ; Запуск АЦП out dx, ax

mov dx, 300h

D\_2:

in ax, dx

test ax, 8000h

jz D\_2

and ax, 07FFh

mov d2, ax

call DELAY\_400\_MSEC

; Вывод 2 на коммутатор

mov ax, 4000h

mov dx, 301h

out dx, ax

or ax, 8000h

out dx, ax

mov dx, 300h

D\_3:

in ax, dx

test ax, 8000h

jz D\_3

and ax, 07FFh

mov d3, ax

call DELAY\_20\_MSEC

;Вывод 3 на коммутатор

mov ax, 6000h

mov dx, 301h

out dx, ax

or ax, 8000h ; Запуск АЦП

out dx, ax

mov dx, 300h

D\_4:

in ax, dx ; Опрос порта с адресом 300h по флагу

test ax, 8000h

jz D\_4

and ax, 07FFh ; Выделение информационных разрядов

mov d4, ax

mov cx, 0 ; Регистр для вкл двигателей

; Вычисление среднего давления до МФ

mov ax, d1

add ax, d2

shr ax, 1

mov avg\_before, ax

cmp ax, 434h

jge primary\_dont\_work

mov cx, 1

primary\_dont\_work:

; Вычисление среднего давления после МА

mov ax, d3

add ax, d4

shr ax, 1

mov avg\_after, ax

sub ax, avg\_before

cmp ax, 0

jge skip\_neg

neg ax

skip\_neg:

cmp ax, 1A5h

jle extra\_dont\_work

or cx, 2

extra\_dont\_work:

mov dx, 301h

mov ax, cx

out dx, ax ; вывод управляющих воздействий

call DELAY\_20\_MSEC

jmp Met\_0

end LStart

Была реализована программа, имитирующая действия СРВ с использованием языка Python (рисунок 3).

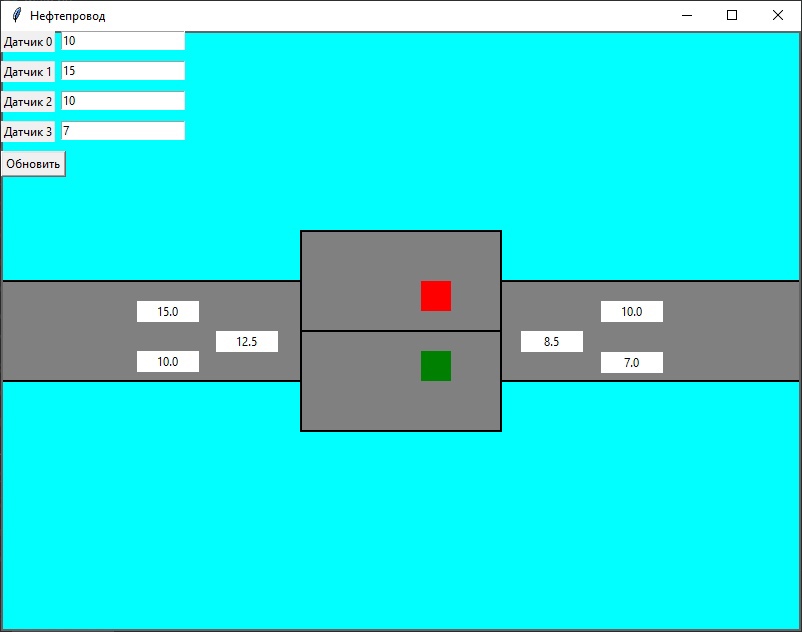


Рисунок 3 – Программа, имитирующая функционал СРВ

**Вывод**

В ходе лабораторной работы были получены навыки разработки программ управления (драйверов) на языке ассемблера, управляющих функционированием СРВ.