

DR. BOB DAVIDOV

Система термостатирования на базе промышленного модуля дискретного ввода-вывода ОВЕН МК110 (RS-485, ModBus)

Цель работы: Освоить канал связи среды разработки системы управления с внешней средой компьютера для построения быстродействующих систем реального времени.

Задача работы: Построить действующую систему термостатирования на базе модели промышленного модуля дискретного ввода-вывода ОВЕН МК110 (RS-485, ModBus) и средств управления: OPC сервер, m – программа MatLAB, Simulink - OPC сервер и MasterSCADA – OPC сервер.

Приборы и принадлежности: Модуль дискретного ввода-вывода ОВЕН МК110; Датчик температуры с частотным выходом МАХ6577; Преобразователя интерфейса UPort 1150 для связи устройств через USB и RS-232/422/485 порты; Нагревательный элемент от 60 до 800 Вт (например, лампа накаливания). Персональный компьютер.

ОБЩИЕ СВЕДЕНИЯ

- Система термостатирования включает следующие элементы.
 - Модуль дискретного ввода-вывода ОВЕН МК110;
 - Датчик температуры с частотным выходом МАХ6577;
 - Преобразователя интерфейса UPort 1150 для связи устройств через USB и RS-232/422/485 порты;
- Структурная схема системы термостатирования

МОДУЛЬ ДИСКРЕТНОГО ВВОДА-ВЫВОДА ОВЕН МК110;

Прибор предназначен для сбора данных со встроенных дискретных входов с последующей их передачей в сеть RS-485 и управления встроенными дискретными ВЭ, используемыми для подключения исполнительных механизмов с дискретным управлением, по сигналам из сети RS-485 или в зависимости от состояния дискретных входов.



Рис. 1. Общий вид модуля дискретного ввода-вывода ОВЕН МК110-224.8Д.4Р

Модули Мх110 могут применяться:

- Для увеличения числа входов-выходов ПЛК.
- Для удаленного ввода и вывода сигналов при подключении к SCADA-системам и другому ПО.
- Для приема и передачи данных через радиомодемы или сети GSM.
- Для передачи данных на панели оператора.
- Для работы с любым оборудованием, поддерживающим интерфейс RS-485 и протоколы обмена ModBus-RTU/ ASCII, DCON, ОВЕН. Тип протокола определяется прибором автоматически.

ОВЕН МК110 имеет встроенный источник питания для подключенных ко входам модуля датчиков.

Конфигурирование МК110 осуществляется на ПК через адаптер интерфейса RS-485/RS-232 или RS-485/USB (например ОВЕН АС3-М или АС4) с помощью программы «Конфигуратор М110», входящей в комплект поставки.

Табл. 1. Технические характеристики модуля дискретного ввода-вывода ОВЕН МК110-224.8Д.4Р

| Дискретные входы | |
|--|--|
| Количество дискретных входов | 8 (две группы по 4 входа, каждая группа гальванически изолирована) |
| Максимальная частота сигнала, подаваемого на дискретный вход | 1 кГц |
| Максимальный входной ток дискретного входа | не более 7 мА |
| Сопротивление контакта (ключа) и соединительных проводов, подключаемых к дискретному входу | не более 100 Ом |
| Дискретные выходы | |

| | |
|--|--|
| Количество дискретных выходных элементов | 4 2 выхода имеют нормально замкнутые и нормально разомкнутые клеммы; 2 выхода имеют только нормально разомкнутые клеммы |
| Параметры дискретных выходов (электромагнитных реле) | 4 А при напряжении не более 250 В 50 Гц и $\cos\varphi > 0,4$ или 4 А при постоянном напряжении не более 24 В |
| Питание | |
| Напряжение питания | 90...264 В переменного тока (номинальное напряжение 220В) частотой 47...63 Гц или 20..235 В постоянного тока (номинальное напряжение 24 В) |
| Потребляемая мощность | не более 6 ВА |
| Интерфейс | |
| Интерфейс связи с компьютером | RS-485 |
| Скорость обмена по интерфейсу RS-485 | от 2400 до 115200 бит/сек |
| Протокол связи, используемый для передачи информации | ОВЕН; ModBus-RTU; ModBus-ASCII; DCON |

Работа по протоколу ModBus

Работа по протоколу ModBus может идти в режимах ASCII или RTU. Частичный список регистров ModBus модуля приведен в следующей таблице

Табл. 2. Регистры протокола ModBus

| Параметр | Ед. измерен | Значение | Тип | Адрес регистра | |
|----------------------------|--------------|------------|--------|----------------|-------|
| | | | | (hex) | (dec) |
| Импульс ШИМ на выход №1 | 0.1 % | 0... 1000 | uint16 | 0000 | 0000 |
| Импульс ШИМ на выход №2 | 0.1 % | 0... 1000 | uint16 | 0001 | 0001 |
| Импульс ШИМ на выход №3 | 0.1 % | 0... 1000 | uint16 | 0002 | 0002 |
| Импульс ШИМ на выход №4 | 0.1 % | 0... 1000 | uint16 | 0003 | 0003 |
| Период ШИМ на выходе №1 | сек | 1... 900 | uint16 | 0020 | 0032 |
| Период ШИМ на выходе №2 | сек | 1... 900 | uint16 | 0021 | 0033 |
| Период ШИМ на выходе №3 | сек | 1... 900 | uint16 | 0022 | 0034 |
| Период ШИМ на выходе №4 | сек | 1... 900 | uint16 | 0023 | 0035 |
| Макс. сетевой тайм-аут | сек | 0... 600 | uint16 | 0030 | 0048 |
| Значение счетчика входа №1 | срабатывание | 0... 65535 | uint16 | 0040 | 0064 |

| | | | | | |
|----------------------------|--------------|------------|--------|------|------|
| Значение счетчика входа №2 | срабатывание | 0... 65535 | uint16 | 0041 | 0065 |
| ... | | | | | |
| Значение счетчика входа №8 | срабатывание | 0... 65535 | uint16 | 0043 | 0071 |

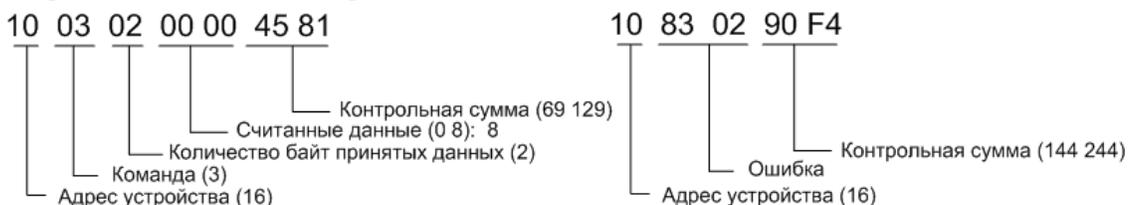
Примечание.

- 1) Запись в регистры осуществляется командой 16 (0x10), чтение – командами 03 или 04 (прибор поддерживает обе команды).
- 2) Обнуление счетчиков делается записью 0 в регистры хранения результатов счета.
- 3) Тип данных «uint16» в таблице является сокращением от «unsigned int16».
- 4) Каждый прибор в сети RS-485 должен иметь свой уникальный базовый адрес.

Формат modbus RTU: Передача данных



Формат modbus RTU: Запрошенные данные



Сброс или предустановка счетчика выполняется записью 0 или соответствующего значения в регистр счетчика: 064₁₀ .. 071₁₀

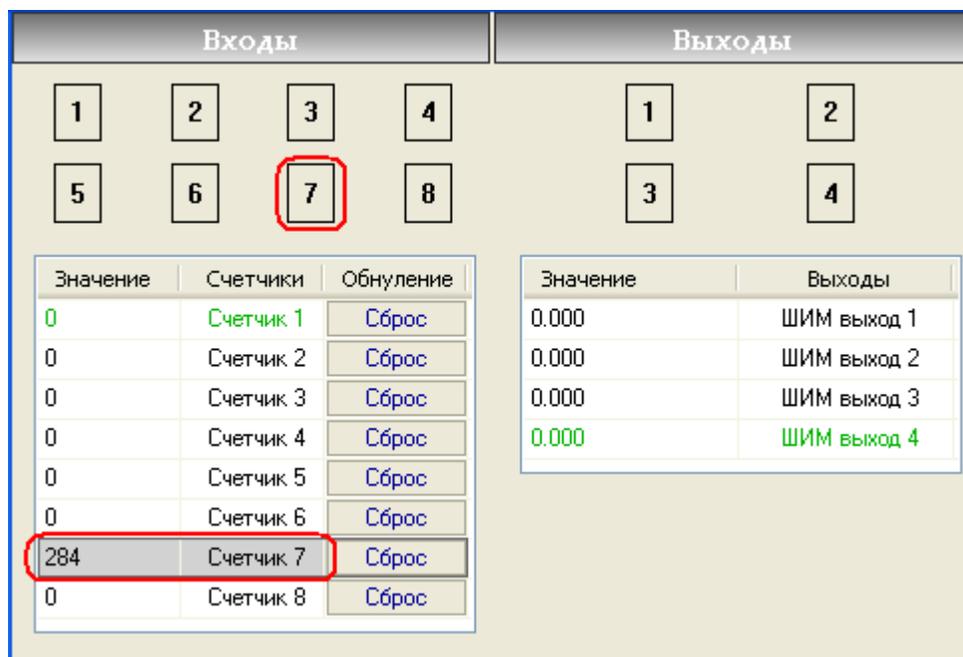


Рис. 2. Наблюдение за состоянием входов/выходов модуля ОВЕН МК110 при помощи программы «Конфигуратор М110».

Подключение модуля МК110 показано на структурной схеме системы термостатирования, Рис. 7.

ДАТЧИК ТЕМПЕРАТУРЫ С ЧАСТОТНЫМ ВЫХОДОМ МАХ6577;

Микросхема МАХ6577 это дешёвый, слаботочный температурный датчик с однопроводным выходом. Микросхема МАХ6577 преобразует окружающую температуру в меандр с частотой пропорциональной абсолютной температуре.

Основные характеристики температурного датчика с частотным выходом МАХ6577:

- Стандартный рабочий температурный диапазон от -40°C до $+125^{\circ}\text{C}$.
- Точность $\pm 0.8^{\circ}\text{C}$ при $+25^{\circ}\text{C}$ ($\pm 3^{\circ}\text{C}$ макс)
- Диапазон напряжения питания от $+2.7\text{В}$ до $+5.5\text{В}$
- Низкое типовое потребление тока 140мкА .
- Не требуют внешних компонентов.
- Маленький 6-контактный корпус SOT23.

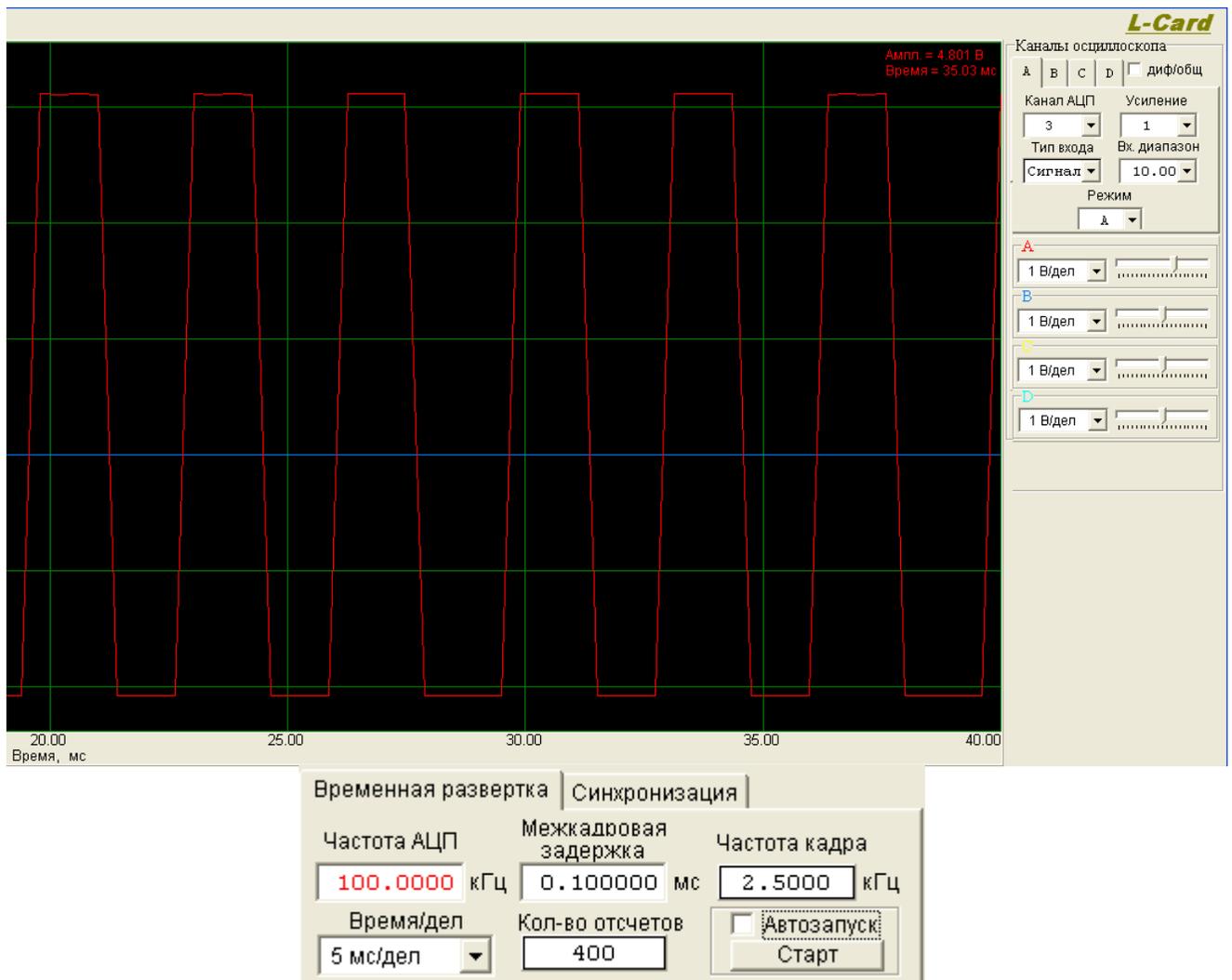


Рис. 3. Выходной сигнал датчика температуры - 4.8В при напряжении питания +5.2В.

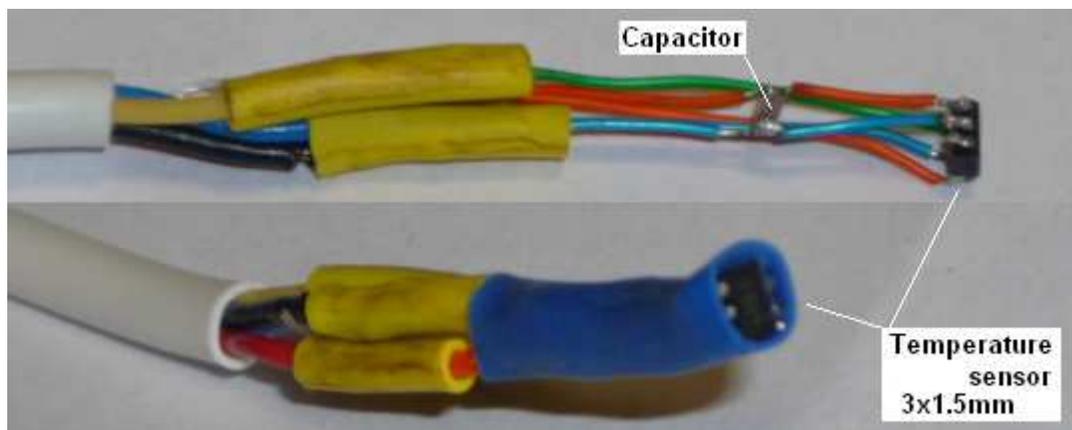


Рис. 4. Датчик температуры

Подключение датчика температуры показано на структурной схеме системы термостатирования, Рис. 7.

Подключение Modbus канала к COM порту ПРЕОБРАЗОВАТЕЛЯ ИНТЕРФЕЙСА UPORT 1150 ДЛЯ СВЯЗИ УСТРОЙСТВ ЧЕРЕЗ USB И RS-232/422/485 ПОРТЫ;

Простейшее подключение modbus канала к последовательному порту показано на Рис. 5. Типовая скорость передачи 9600 бод (бит в сек.). Максимальная скорость обмена по интерфейсу RS-485 составляет 115200бит/сек.

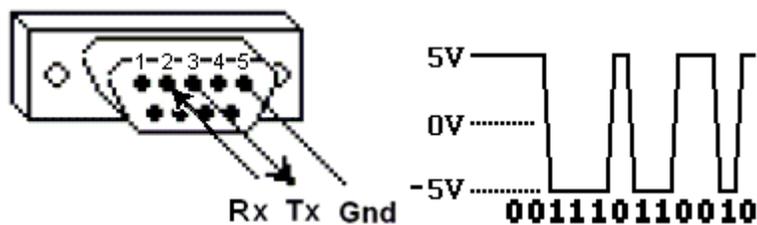


Рис. 5. Простейшее подключение modbus канала к последовательному порту

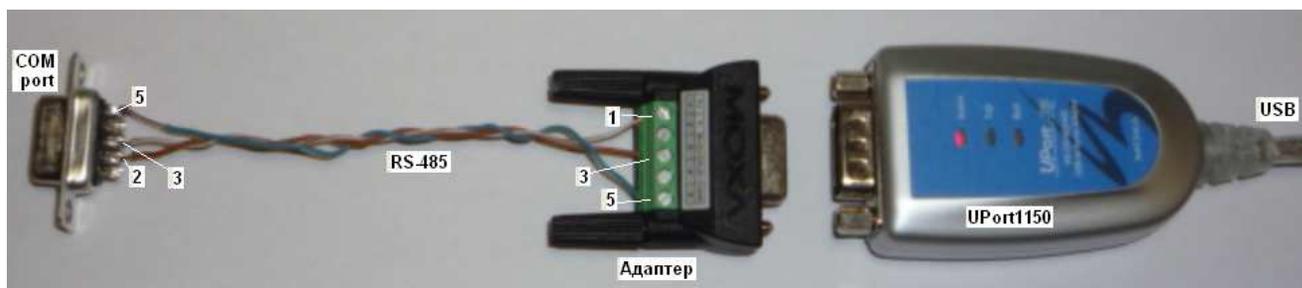


Рис. 6. Интерфейс RS-485 modbus канала преобразователя UPORT 1150

ПРЕОБРАЗОВАТЕЛЯ ИНТЕРФЕЙСА UPORT 1150 ДЛЯ СВЯЗИ УСТРОЙСТВ ЧЕРЕЗ USB И RS-232/422/485 ПОРТЫ

UPort 1150 имеет следующие характеристики:

- скорость USB операций до 12 Мбс
- Дополнительные прерывания не требуются
- скорость последовательной передачи данных до 921.6 Кбс
- FIFO 64-byte
- Защита от статического электричества 15 КВ
- Поддерживает RS-422, 2-проводной RS-485, и 4-проводной RS-485
- Имеет терминальный адаптер для подключения линий к порту
- Оптронная защитная развязка 2 KV

Табл. 3. Сравнение интерфейсов RS-232, RS-422 и RS-485

| Параметр | RS-232 | RS-422 | RS-485 |
|---|---------------------|------------------|------------------|
| Способ передачи сигнала | Однофазный | Дифференциальный | Дифференциальный |
| Максимальное количество приемников | 1 | 10 | 32 |
| Максимальная длина кабеля | 15 м | 1200 м | 1200 м |
| Максимальная скорость передачи | 460 кбит/с | 10 Мбит/с | 30 Мбит/с** |
| Синфазное напряжение на выходе | ± 25 В | -0,25...+6 В | -7...+12 В |
| Напряжение в линии под нагрузкой | $\pm 5... \pm 15$ В | ± 2 В | $\pm 1,5$ В |
| Импеданс нагрузки | 3...7 кОм | 100 Ом | 54 Ом |
| Ток утечки в "третьем" состоянии | - | - | ± 100 мкА |
| Допустимый диапазон сигналов на входе приемника | ± 15 В | ± 10 В | -7...+12 В |
| Чувствительность приемника | ± 3 В | ± 200 мВ | ± 200 мВ |
| Входное сопротивление приемника | 3...7 кОм | 4 кОм | ≥ 12 кОм |

Примечание. **Скорость передачи 30 Мбит/с обеспечивается элементной базой, но не является стандартной.

Для нормальной работы преобразователя UPort 1150 необходимо установить его драйвер и номер СОМ порта.

Подключение датчика температуры показано на структурной схеме системы термостатирования, Рис. 7.

СТРУКТУРНАЯ СХЕМА СИСТЕМЫ ТЕРМОСТАТИРОВАНИЯ

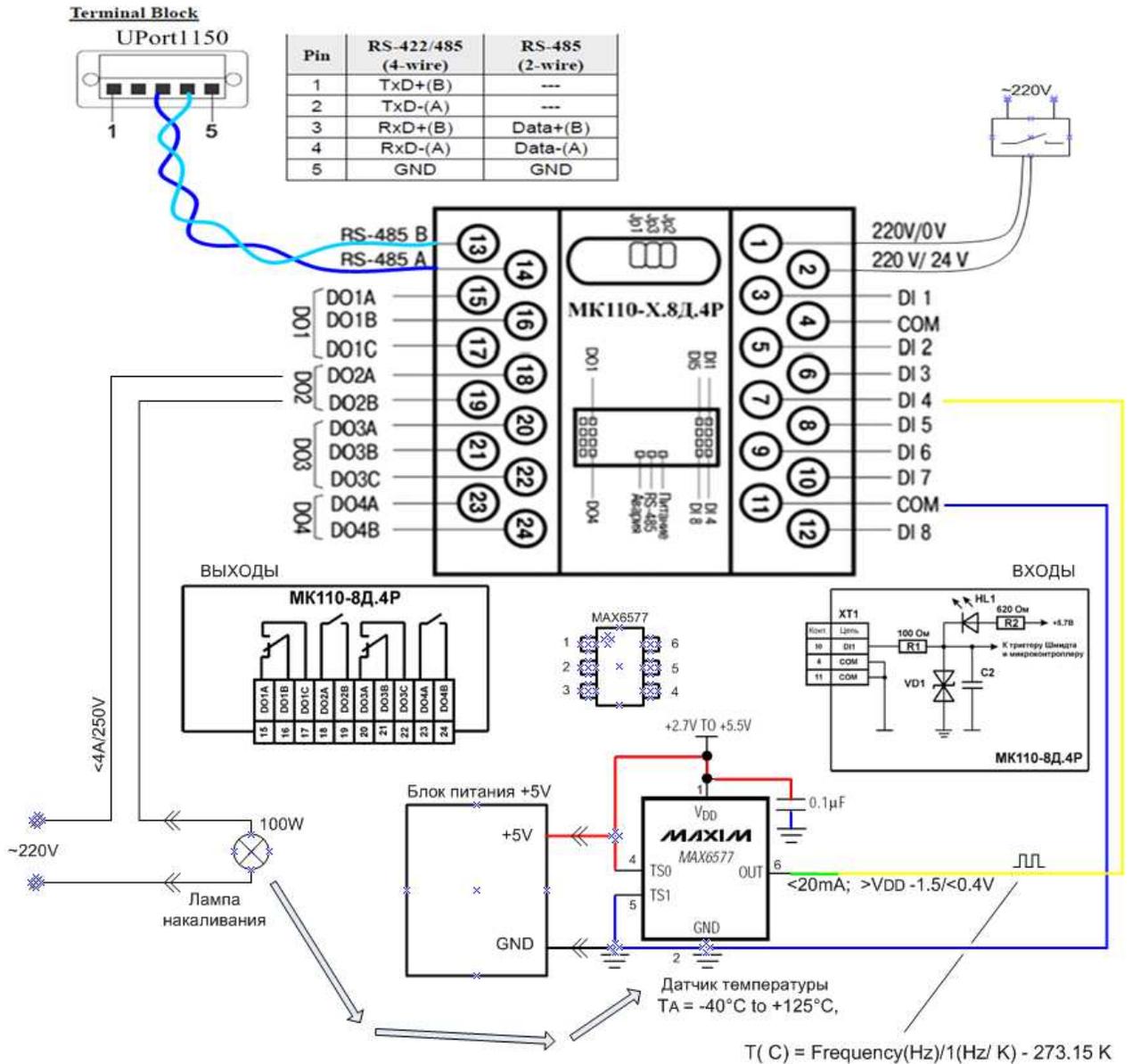


Рис. 7. Схема подключения внешних (физических) компонентов системы термостатирования.


```

% Matlab v7.0 (R14) SP 1
% Bob Davidov
% 25 February 2012
%
% CRC algorithm
% calculates check sum of Modbus RTU sequence
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function output_hex_string = crc_calculator (Input_hex);
%Input_hex = 'F70302640008'; % <= 2 * 16 Char
F = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
xor_constant = [1 0 1 0 0 0 0 0 0 0 0 0 0 0 1];
for i = 1 : length (Input_hex) / 2;
    A = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
    if ~(i > length (Input_hex)/2)
        A_hex = Input_hex ((i-1)*2+1:i*2); % Two HEX bytes
        A_bin = dec2bin (hex2dec (A_hex));
        length_A_bin = length (A_bin);
        for j = 0 : length_A_bin - 1
            A (16 - j) = str2num(A_bin (length_A_bin - j));
        end
    end
    F = xor (F,A);
    for ii = 1 : 8
        if F(16) ==1
            if xor_constant (1) == 0
                F_shift (1) = 0;
            else
                F_shift (1) = 1;
            end
            for j = 2 : 16;
                if xor_constant (j) == F (j-1);
                    F_shift (j) = 0;
                else
                    F_shift (j) = 1;
                end
            end
            else
                F_shift = circshift(F',1)';
            end
        F = F_shift;
    end
end
h = num2str(F);
h = h(1:3:length(h));
output_hex_string = num2str([dec2hex(bin2dec(h(9:12)))
dec2hex(bin2dec(h(13:16))) dec2hex(bin2dec(h(1:4)))
dec2hex(bin2dec(h(5:8)))]);
% End of crc_calculator.m

```

Программа связи с МК110 по Modbus RTU:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MatLAB_RTU_commander.m v1.0a
% Matlab v7.0 (R14) SP 1
% Bob Davidov
% 28 February 2012
%
```

```
clear all;
```

```
%Useful function:
```

```
%>>hex2dec('7531')
```

```
%>>char(36) => '%'
```

```
% Input data
```

```
Com_Port_Num = 'COM1';
```

```
Address = 16; % Station Address 8 bit: 1..255
```

```
Function = 16; % 3 or 4 is read; 16 is write;
```

```
PWM = 00; % 0 .. 1000
```

```
Data_First_Address = 3; % First address of Module Rg
```

```
Address_Range = 1; %
```

```
% End of Input data
```

```
dfa = dec2hex(Data_First_Address);
```

```
l = length(dfa);
```

```
str = '0000';
```

```
str(4-l+1:4) = dfa;
```

```
Data_First_Address_Bytes = [hex2dec(str(1:2)) hex2dec(str(3:4))];
```

```
dfa = dec2hex(Address_Range);
```

```
l = length(dfa);
```

```
str = '0000';
```

```
str(4-l+1:4) = dfa;
```

```
Address_Range_Bytes = [hex2dec(str(1:2)) hex2dec(str(3:4))];
```

```
if Function == 16
```

```
    dfa = dec2hex (PWM);
```

```
    l = length(dfa);
```

```
    str = '0000';
```

```
    str(4-l+1:4) = dfa;
```

```
    PWM_Bytes = [2 hex2dec(str(1:2)) hex2dec(str(3:4))];
```

```
else
```

```
    PWM_Bytes = [];
```

```
end
```

```
% Master's Tx data without Check sum
```

```
Code = [Address Function Data_First_Address_Bytes Address_Range_Bytes PWM_Bytes];
```

```
Code_Char = dec2hex(Code);
```

```
Code_Char_line = [];
```

```
for l = 1:length(Code)
```

```
    Code_Char_line = [Code_Char_line Code_Char(l,1:2)];
```

```
end
```

```
% Check sum calculation
```

```
Check_Sum = crc_calculator(Code_Char_line)
```

```
% Master's Tx data with Check sum
```

```
RTU_Code = [Code hex2dec(Check_Sum(1:2)) hex2dec(Check_Sum(3:4))]
```

```
COM1 = serial (Com_Port_Num);
```

```
fopen (COM1);
```

```
fwrite(COM1, RTU_Code);
```

```
pause(0.002);
```

```
BytesAvailable = get(COM1, 'BytesAvailable');
```

```
if BytesAvailable > 0
```

```
    Rx = fread(COM1, BytesAvailable)'
```

```
end
```

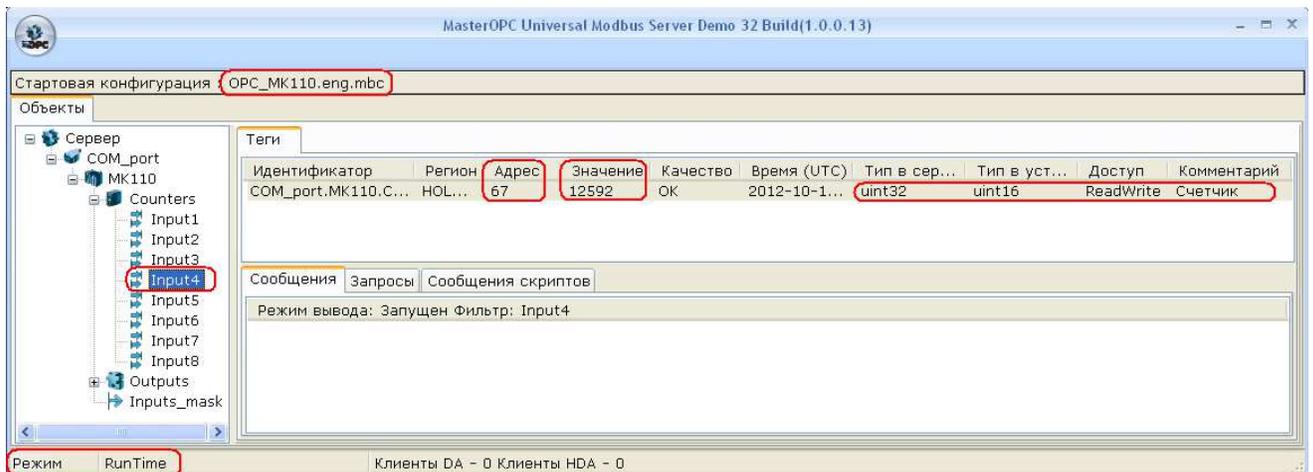
```
fclose (COM1);
```

```
delete (COM1);
```

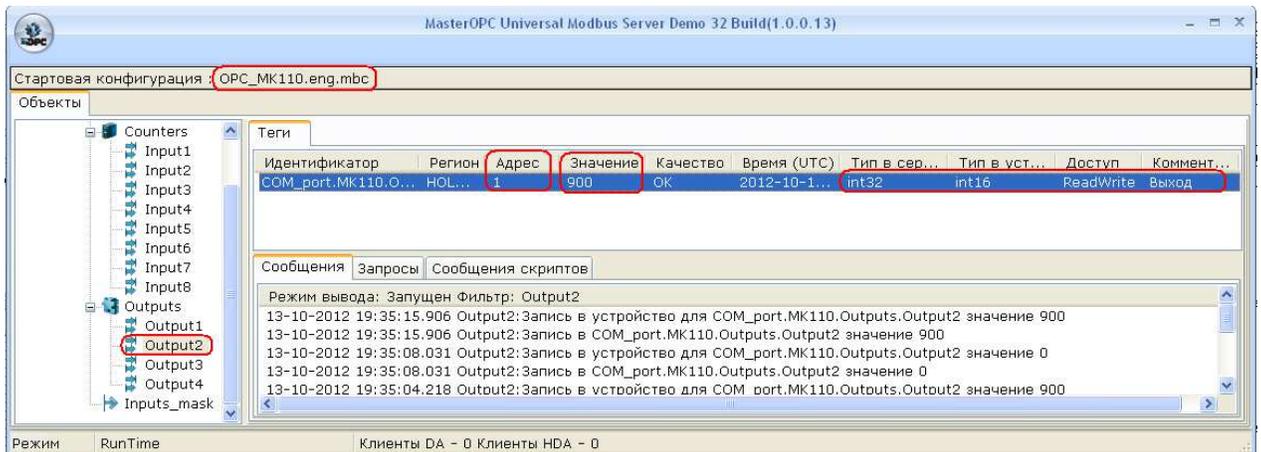
```
% End of MatLAB_RTU_commander.m.m
```

4. Проверьте возможность связи OPC сервера с модулем МК110

5. Считайте показания счетчика N4.



6. Включите и выключите нагреватель посылая на Выход N2 код 900 и 0.



Задание 2. Построение системы реального времени поддержания заданной температуры в среде MatLAB (m File)

Команд, полезные для выполнения задания:

Команды приема/передачи данных модуля МК110

| RTU_Code | Отклик Rx | Описание. |
|------------------------------|-----------------------|-----------------------------------|
| 16 16 0 1 0 1 2 3 132 103 66 | 16 16 0 1 0 1 83 72 | Замыкание контактов, выход №2 |
| 16 16 0 1 0 1 2 0 0 103 209 | 16 16 0 1 0 1 83 72 | Размыкание контактов, выход №2 |
| 16 16 0 67 0 1 2 0 0 104 243 | 16 16 0 67 0 1 243 92 | Сброс (предустановка) счетчика №4 |
| 16 3 0 67 0 1 118 159 | 16 3 2 0 0 | Чтение счетчика №4 |

Команды MatLAB

| | |
|-----|---|
| tic | Начало измеряемого временного интервала |
| toc | Время от начала временного интервала |

1. Разработайте программу термостатирования на языке MatLAB, на основе m-кода, приведенного ниже.

Программа термостатирования в релейном режиме на базе промышленного модуля ввода-вывода МК100 и канала передачи modbus RTU:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RT_Thermostatic_System.m  v1.0a
% Matlab v7.0 (R14) SP 1
% Bob Davidov
% 13 October 2012
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;

%Useful function:
%>hex2dec('7531')
%>char(36) => '%'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t_target = 35; % target temperature
t_task = 60; % Time of the thermostatic in sec

Com_Port_Num = 'COM2';
Address = 16; % Station Address 8 bit: 1..255
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Input data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Lamp_ON = [16 16 0 1 0 1 2 3 132 103 66]; % Lamp ON (OUTPUT N2); Write PWM 900;
Lamp_OFF = [16 16 0 1 0 1 2 0 0 103 209]; % Lamp OFF (OUTPUT N2); Write PWM 0;
Counter_Reset = [16 16 0 67 0 1 2 0 0 104 243]; % RESET Counter N4
%Counter_Read = [16 3 0 67 0 1 118 159]; % READ Counter N4
Counter_Read = [16 4 0 67 0 1 195 95];

% reserve memory for arrays
Temperature_C (1:t_task) = 0;

```

```

Time_period (1:t_task) = 0;
Pulse (1:t_task) = 0;

for i = 1:t_task
    % Open COM port
    COM1 = serial (Com_Port_Num);
    fopen (COM1);

    fwrite(COM1, Counter_Reset);
    tic;          % start of period
    pause(1);
    fwrite(COM1, Counter_Read);
    period = toc; % end of period

    pause(0.1);
    BytesAvailable = get(COM1, 'BytesAvailable');
    if BytesAvailable > 6
        Rx = fread(COM1, BytesAvailable);
        Temperature_C (i) = (Rx(BytesAvailable-3)*256 + Rx(BytesAvailable-2))/period - 273.15;
        Time_period(i) = period;
        i
    end

    if Temperature_C(i) > t_target
        fwrite(COM1, Lamp_OFF);
        Pulse(i) = 0;
    else
        fwrite(COM1, Lamp_ON);
        Pulse(i) = 1;
    end
    pause(0.1);

    % Close COM Port
    fclose (COM1);
    delete (COM1);
end

figure
time = cumsum(Time_period);
plot(time, Temperature_C, 'c', 'LineWidth', 2);
hold on
plot([0 max(time)], [t_target t_target], 'y', 'LineWidth', 2);
hold on
plot(time, Pulse * 10, 'r', 'LineWidth', 1);
hold on
plot(time, Pulse * 10, 'xr', 'LineWidth', 2);
grid
legend('Current temp., C', 'Target temp., C', 'Pulse, bitx10', 0);
xlabel (Time, sec);
ylabel ('Temperature [C]', 'Pulse');
title ('Temperature against Time')

% End of RT_Thermostatic_System.m

```

2. Соберите систему термостатирования по схеме Рис. 7.

3. Настройте программную часть для работы системы не менее одной минуты.
4. Снимите характеристики системы. Сравните с характеристиками, представленными на Рис. 9.

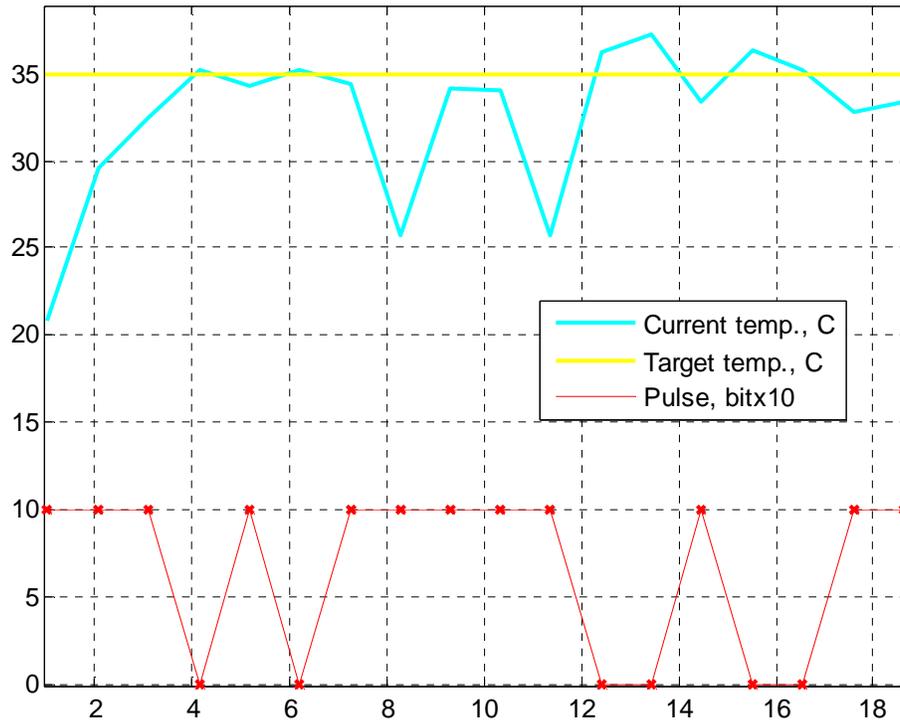


Рис. 9. Пример термостатирования на базе m-кода МатЛАБ.

Задание 3. Построение системы реального времени в Simulink на базе OPC сервера Master OPC Universal Modbus Server Demo 32 Build (1.0.0.13)

1. Загрузите конфигурацию OPC_MK110.eng.mbc в OPC сервер.
2. Запустите сервер
3. Загрузите МатЛАБ, например R2007.
4. Используя OPC Tool МатЛАБ (MatLAB > Start > Toolboxes > OPC > OPC Tool) подсоедините МатЛАБ к OPC серверу.

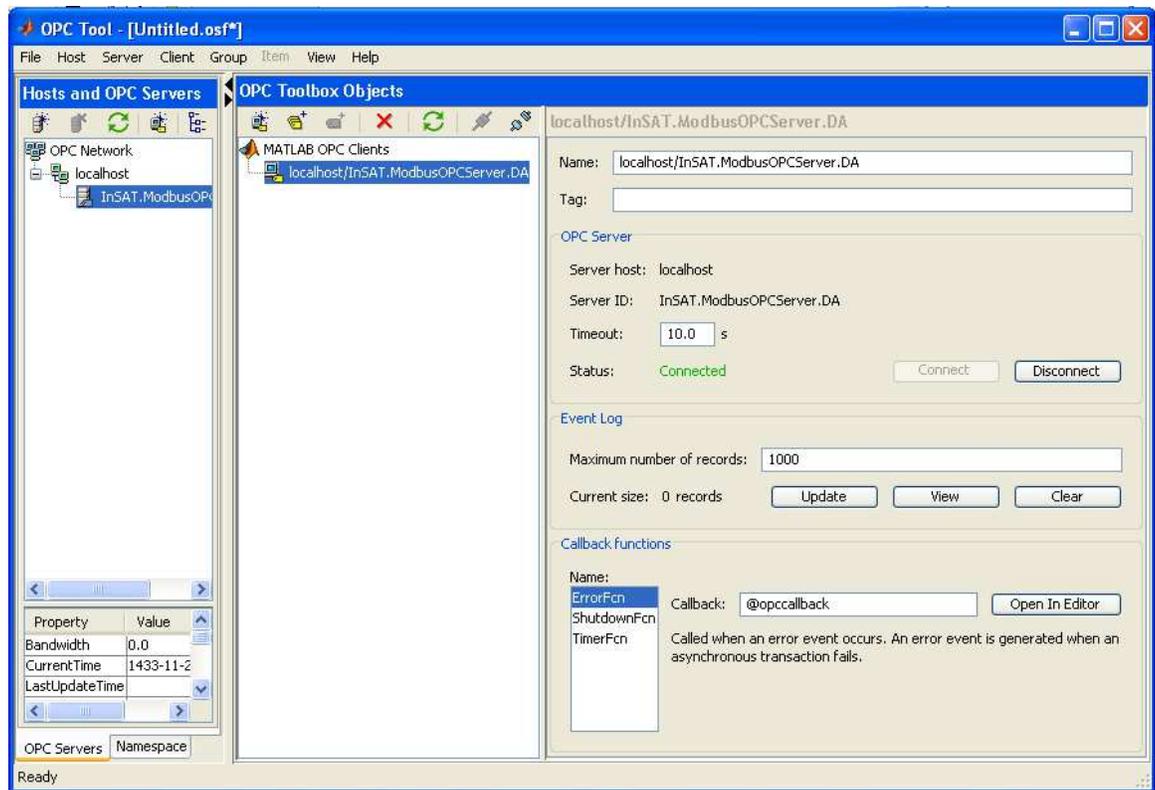


Рис. 10. Подключение МатЛАБ к OPC серверу.

5. Создайте или загрузите модель МатЛАБ (см. Рис. 11).

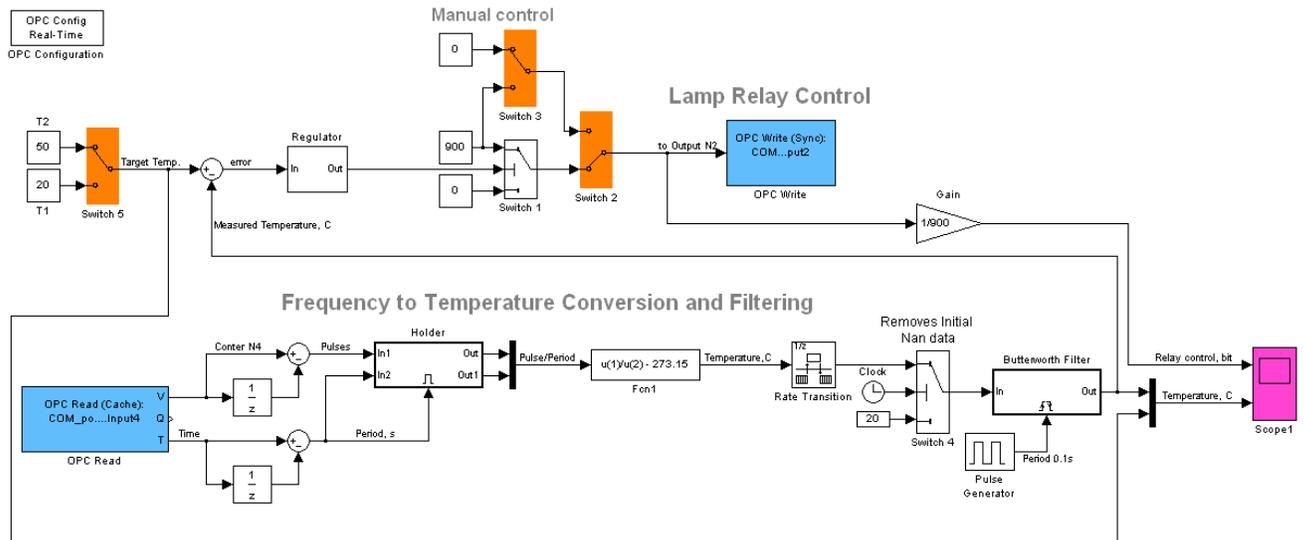


Рис. 11. RT simulink модель виртуальной части системы термостатирования на базе OPC сервера и промышленного модуля ввода/вывода ОВЕН МК110.

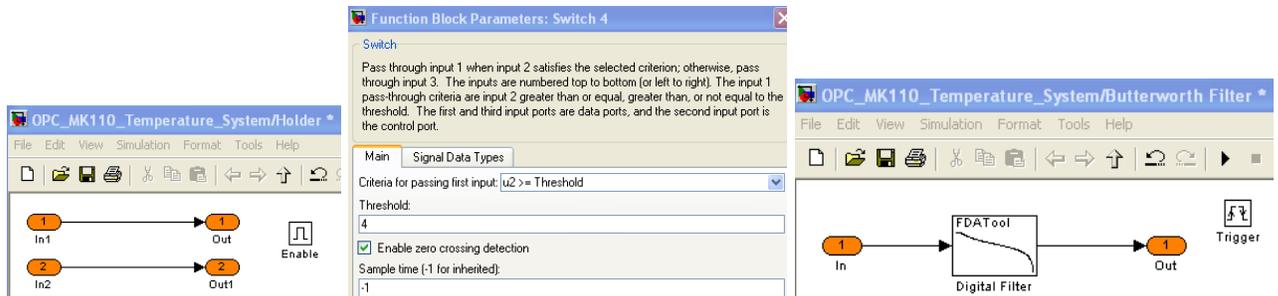
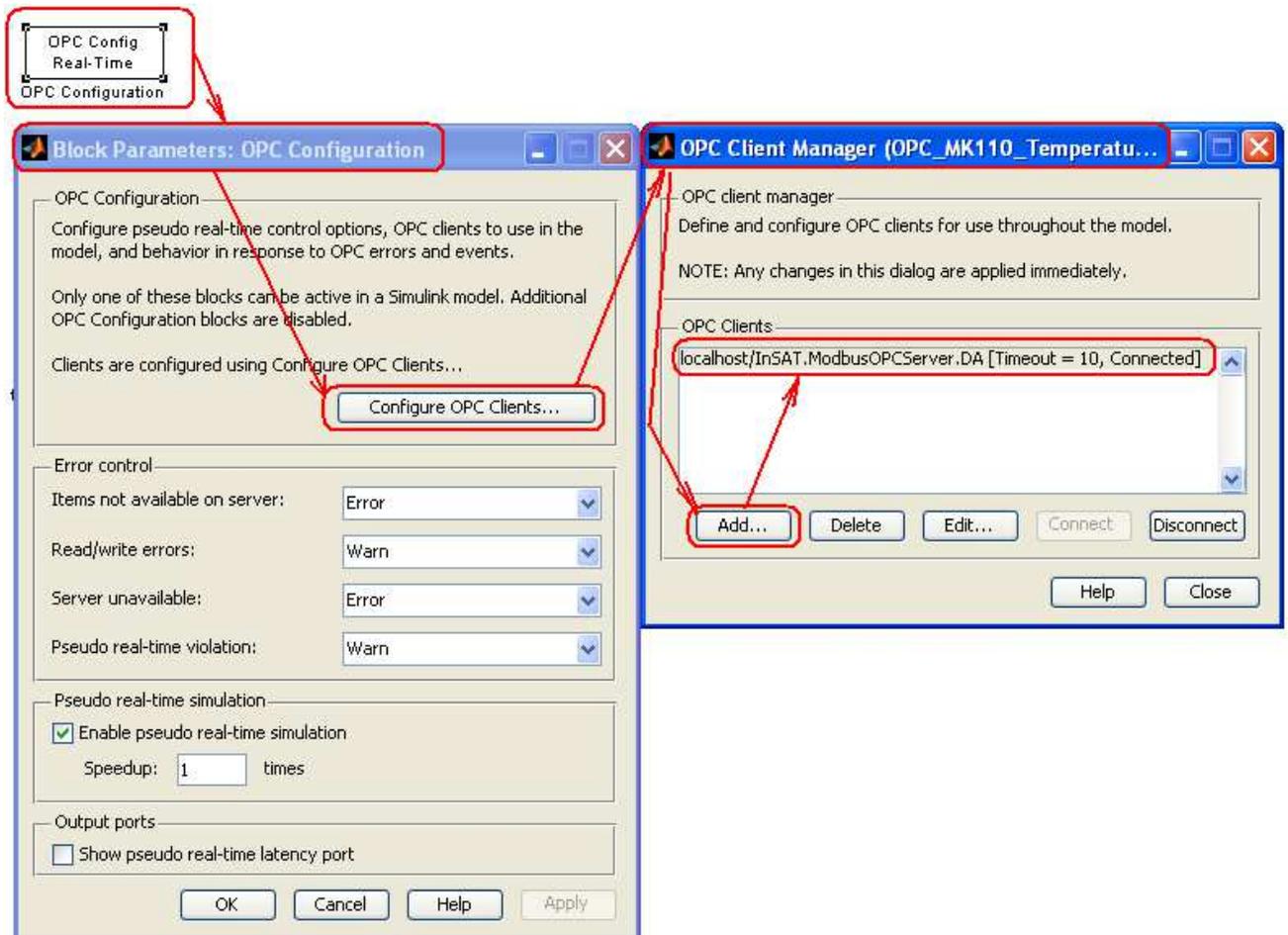


Рис. 12. Субблоки и параметры модели.



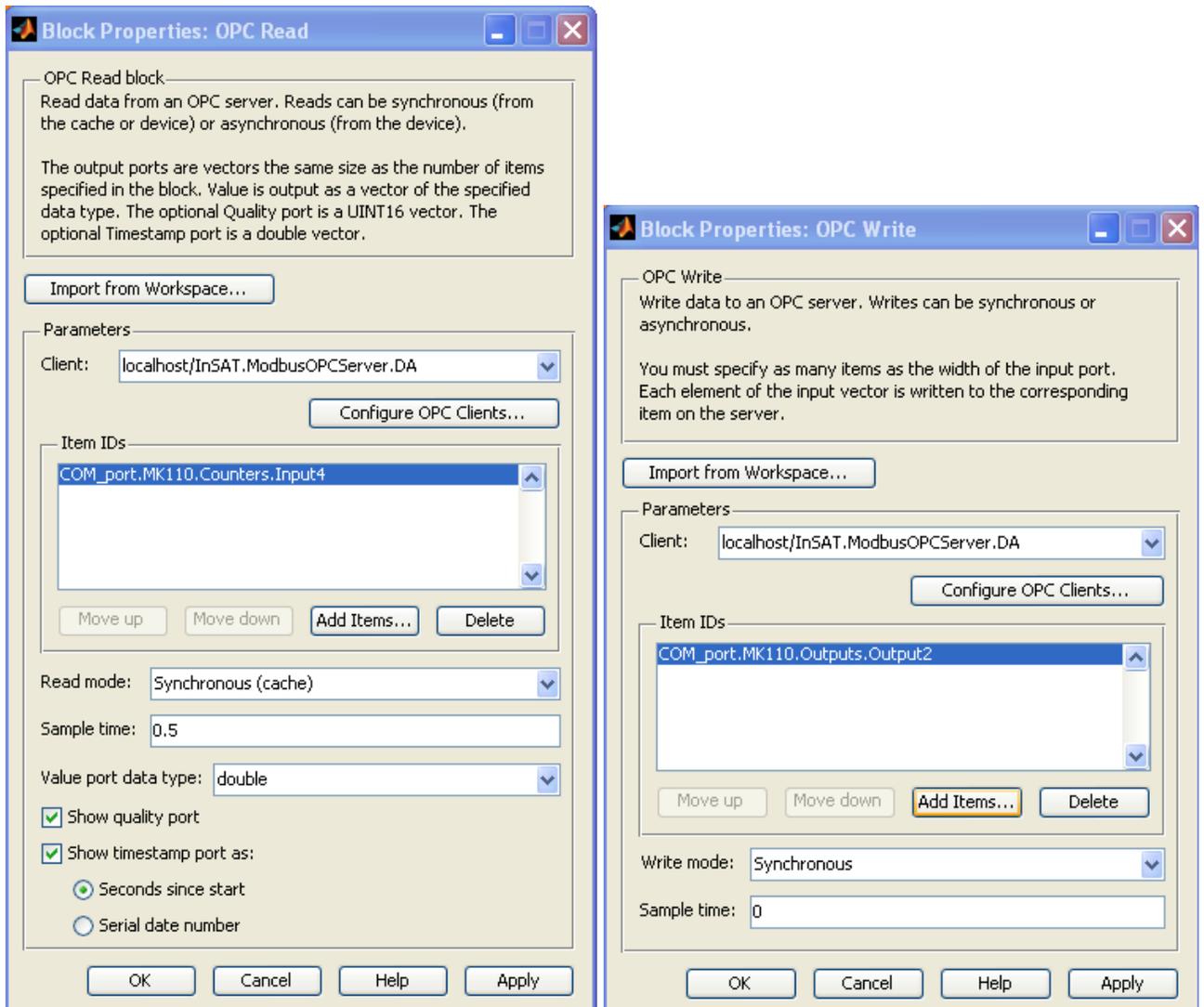


Рис. 13. Параметры блоков simulink модели.

6. Убедитесь, что система термостатирования на базе OPC сервера и Simulink модели работает.
7. Экспериментально определите средний период получения значений счетчика импульсов температурного датчика в simulink модели. Вычислите период опроса и сравните его с экспериментальным значением.
8. Найдите пути повышения частоты отклика датчика и увеличения точности датчика (например, уменьшением пакета опроса модуля OPC сервером и/или увеличением частоты канала RS-485)
9. Остановите OPC сервер.

10. Сохраните переходный процесс системы в mat файле. Сравните полученный процесс с примером, показанным на Рис. 14.

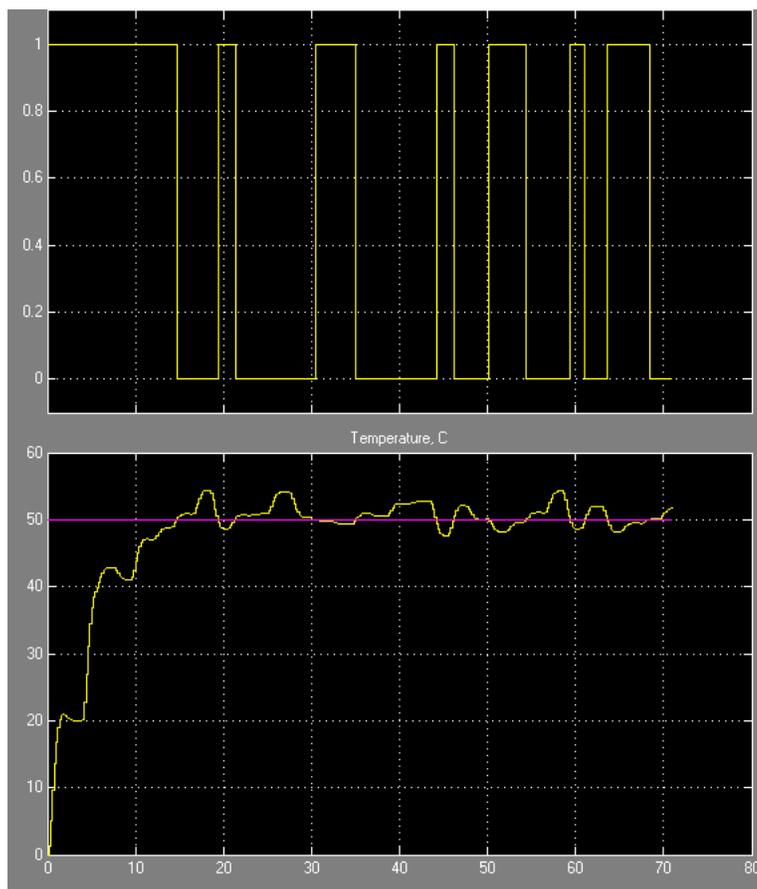


Рис. 14. Пример переходного температурного процесса. Заданная температура: 50 град. С.

Задание 4. Построение системы термостатирования на базе OPC сервера и MasterSCADA

ВНИМАНИЕ. Для выполнения задания потребуется демонстрационная версия MasterSCADA. Ее можно установить в последовательности, приведенной в работе 14.1. “Освоение MasterSCADA”.

1. Загрузите OPC сервер компании ИнСАТ с конфигурацией OPC_MK110_brief.eng.mbc.
2. Откройте конфигурацию MK110_ENG.mbc устройства дискретного ввода-вывода MK110.
3. Определите назначение скрипта переменной “Сервер>COM_port>MK110>Counters>Input4_my”, который запускается после считывания OPC сервером 4-го счетчика модуля MK110.

```
local cnt_cntr4, old_cntr4, inc_cntr4, cnt_time, old_time, dt, freq, old_freq, Error;  
-- инициализация  
function OnInit()
```

```

old_time = os.clock() - 2.0;
old_cntr4 = server.ReadCurrentTag();
old_freq = 300;
Error = false;
end
-- деинициализация
function OnClose()
end
-- обработка
function OnRead()
  crnt_cntr4,qual = server.ReadCurrentTag();
  if qual==OPC_QUALITY_BAD then
    Error = true;
    server.Message("Freq(",crnt_cntr4,") = ",9999.999);
    server.WriteCurrentTag(false, OPC_QUALITY_BAD)
    if Error==true and qual==OPC_QUALITY_GOOD then
      server.UndoCurrentTag( ); -- предыдущее значение
      Error=false;
    end;
  else
    inc_cntr4 = crnt_cntr4 - old_cntr4; -- counter's increment
    old_cntr4 = crnt_cntr4;

    crnt_time = os.clock();
    dt = (crnt_time - old_time); -- time increment
    old_time = crnt_time;

    freq = math.abs(inc_cntr4/dt);
    if math.abs(freq - old_freq) > 20 then
      freq = old_freq;
    end;
    server.Message("Freq(",crnt_cntr4,") = ",freq);
    server.WriteCurrentTag(freq, OPC_QUALITY_GOOD)
    old_freq = freq;
  end;
end
end

```

4. Запустите OPC сервер. Убедитесь, что сервер обеспечивает связь с устройством ввода-вывода МК110.
5. Рассмотрите структуру данных закладок “Сообщение” сервера, “Запросы” и “Сообщения скриптов”.
6. Запустите MasterSCADA.
7. Создайте новый проект, например, Demo_Thermostat_system.
8. Добавьте в дерево “Система” компьютер. Если вы в первый раз используете данный OPC, необходимо произвести его поиск. Для этого необходимо в контекстном меню “Компьютер 1” выбрать “Поиск OPC DA серверов”.
9. Отметьте галочкой “InSAT Modbus OPC server DA”.
10. Добавьте OPC сервер в “компьютер”.

11. Добавьте в OPC все переменные и группы. Обратите внимание, что все созданные в OPC переменные перенеслись в MasterSCADA.
12. Запустите режим исполнения, выполнив  затем  и далее, имя оператора: “sa”, “OK”, “Закреть”.
13. Измените выход Output4 - дважды щелкнув по нему введите значение 500. Услышав характерный звук работы, выключите его, записав значение 0.
14. Проверьте работу счетчика. Подайте импульсы датчика термостатирования на 4-ой вход (input4).
15. Сбросьте счетчик 4-го входа, введя ноль в соответствующую ячейку колонки “Значение”.
16. Остановите работу MasterSCADA.
17. Выделите узел дерева Система. Перейдите на закладку “Шкалы”.
18. Добавьте новую шкалу, дайте ей имя Temperature (Температура). Эта шкала предназначена для отображения температуры пропорциональной приращением показаниям 4-го счетчика МК110. Диапазон шкалы 10..60 град.С.
19. При помощи параметра “Формат” задайте параметр числа f2. Буква f задает формат с плавающей запятой, e – экспоненциальный формат. Цифра означает количество знаков после запятой.
20. Выберите “Ед. измерения” – “Градус С. ”

Общие | Опрос | Архив | Сфера | Окна | Режим исполнения | Операторы | Библиотеки | Сообщения | Шкалы | Журналы

Аналоговые значения :

| Интервал | Начало | Конец | Формат | Ед. измерения |
|--------------------|--------|--------|--------|---------------|
| Проценты | 0.00 | 100.00 | f2 | Процент |
| Градусы | 0.0 | 100.0 | f1 | Градус |
| Килопаскаль | 0.00 | 100.00 | f2 | Килопаскаль |
| Температура | 10.00 | 60.00 | f2 | Градус |

"Мертвая зона" % Скорость % в сек Мигание

Аварийные границы: Предупредительные границы:

нижняя %  нижняя % 

верхняя %  верхняя % 

| Единицы измерения | |
|-------------------|-------------|
| Название | Обозначение |
| Ампер | A |
| Атмосфера | атм |
| Бар | бар |
| Ватт | Вт |
| Вольт | В |

Аналоговые / Дискретные / Перечислимые / Интервальные / Метрология /

21. Нажмите клавишу “Применить”.

22. Используя демонстрационные проекты MasterSCADA, например ..MasterSCADA Projects\Projects\Tutorials\Demo\ Demo.vav, соберите проект системы термостатирования.

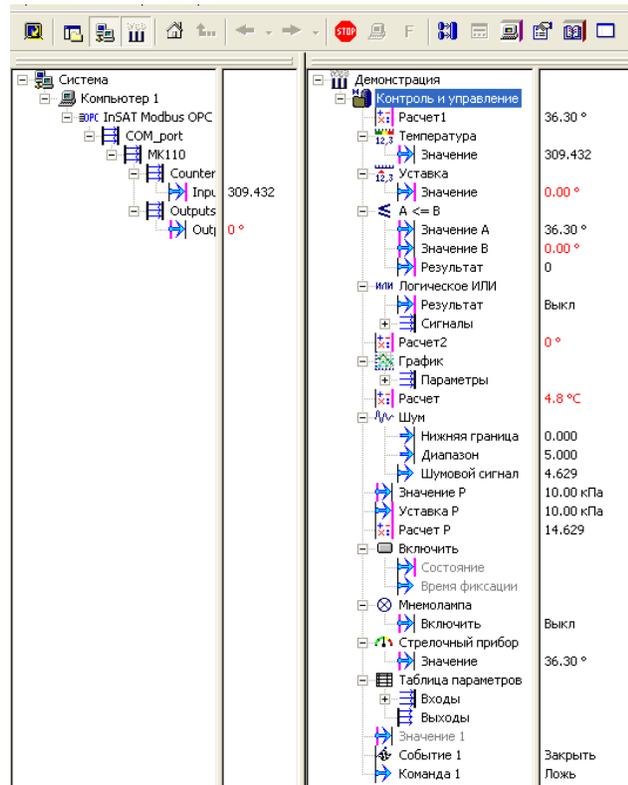
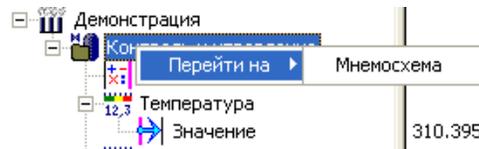


Рис. 15. Структура системы термостатирования.

23. Запустите на выполнение систему термостатирования нажав на кнопку “Пуск” с изображением ракеты .



24. Откройте мнемосхему.

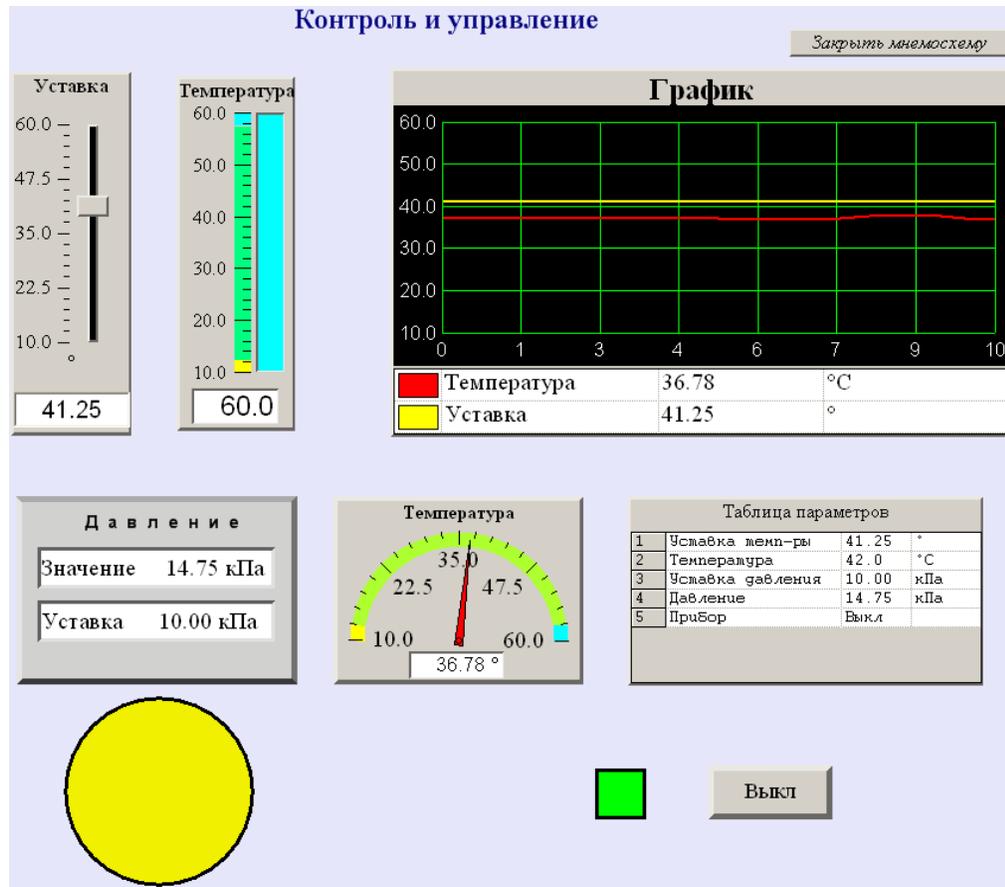


Рис. 16. Мнемосхема системы термостатирования.

25. Проверьте работоспособность проекта системы термостатирования на базе OPC сервера и MasterSCADA.
26. Рассмотрите связи параметров OPC сервера, дерева “Объектов” и элементов мнемосхемы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В последнем задании найдите причины скачков температуры до 55 градусов.
2. За счет чего система контроля на базе кода m-file без использования таймера (Задание 2) работает как система реального времени?
3. Сравните датчики, например, температуры с аналоговым и частотным выходами.
4. Как по переходному процессу изменения температуры определить тепловое сопротивление среды, которая поглощает тепло.
5. Назовите пути повышения точности термостатирования.
6. Используя результаты тестирования системы постройте модель тепловых процессов – зависимость температуры от поступления и потерь тепла.
7. Какой максимальной мощностью можно управлять используя реле 4A/240В?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Help MatLAB.
2. Руководство пользователя программой «Конфигуратор М110»
3. Модули дискретного ввода/вывода МК110-220.8Д(ДН).4Р, МК110-24.8Д(ДН).4Р, МК110-224.8Д(ДН).4Р. Руководство по эксплуатации
4. Сайт дилера ПО "ОВЕН" в СПб системный интегратор ПО "ОВЕН" <http://ovenspb.ru/>
5. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>.