«Национальный исследовательский ядерный университет «МИФИ»

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ (012)

Отчёт

по лабораторной работе №2

учебной дисциплины

«Автоматизированные системы специального назначения»

выполнил: студент Кротов Е.Г.

группа: С21-501

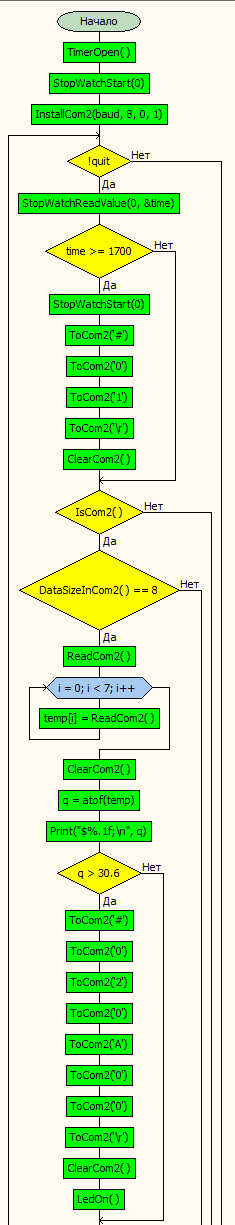
проверил: Сапрыкина В.С.

Москва, 2024 г.

**Цель работы:** освоение процесса программирования контроллера ICP CON I-7188.

**Задача:** Удержание температуры исследуемой среды в диапазоне от 27 tco до 30.6 tco(частота опроса 1.7 сек).

**Алгоритм работы программы:**

****

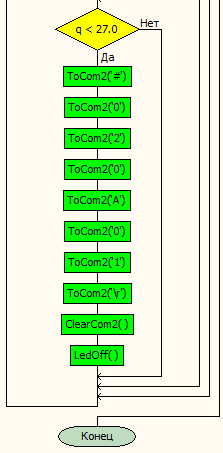
****

Рис. 1

**Список использованных функций:**

· LedOn(); - включение индикатора

· LedOff(); - выключение индикатора

· TimerOpen(); - инициализация таймера.

· StopWatchReadValue(); - чтение значения таймера.

· StopWatchStart(0); - обнуление таймера.

· InstallCom2; - Инициализация COM соединения

· ToCom2; - Отправить данные на Com2.

· ClearCom2(); - Очистить буфер Com2.

· IsCom2(); - Проверка, если на вход контроллера поступили данные

· DataSizeInCom2(); - длина данных в Com2.

· ReadCom2(); - считывание данных из Com2.

· atof(); - преобразование считанной температуры **к типу float.**

· Print(); - отобразить данные на ПК

**Код программы:**

#include "i7188.h"

#include <iostream>

// Объявление переменных

int quit = 0;

int LedMode = 0;

int data;

long baud = 9600;

int i = 0;

float q;

char temp[80];

unsigned long time;

// Начало главной функции

int main(void) {

TimerOpen();

StopWatchStart(0);

// Инициализация COM соединения

InstallCom2(baud, 8, 0, 1);

// Запуск цикла

while (!quit){

// Обнуление таймера

StopWatchReadValue(0, &time);

// На отметке таймера в 1700мс выполняется запрос к ПНК

if (time >= 1700){

StopWatchStart(0);

ToCom2('#');

ToCom2('0');

ToCom2('1');

ToCom2('\r');

ClearCom2();

} // Конец команды запроса

// Проверка, если на вход контроллера поступили данные

if (IsCom2()) {

// Если данные длиной 8 бит то считываем. В противном случае ждем

if (DataSizeInCom2() == 8) {

ReadCom2();

for (i = 0; i < 7; i++) {

temp[i] = ReadCom2();

}

// Очищаем буфер порта

ClearCom2();

// Преобразовываем температуру в float формат в переменной q

q = atof(temp);

/\* Примечание: в данном коде программы, команда Print выводит переменную по COM порту в ПК(если он подключен). Это стоит делать для того, чтобы было возможно наблюдать за текущей температурой на мониторе компьютера.\*/

Print("$%.1f;\n", q);

// Сравнение температуры с границами диапазона

if (q > 30.6){

ToCom2('#');

ToCom2('0');

ToCom2('2');

ToCom2('0');

ToCom2('A');

ToCom2('0');

ToCom2('0');

ToCom2('\r');

ClearCom2();

LedOn();

}

if (q < 27.0){

ToCom2('#');

ToCom2('0');

ToCom2('2');

ToCom2('0');

ToCom2('A');

ToCom2('0');

ToCom2('1');

ToCom2('\r');

ClearCom2();

LedOff();

}

}

}

}

}

**Скриншот успешной компиляции программы в среде программирования:**

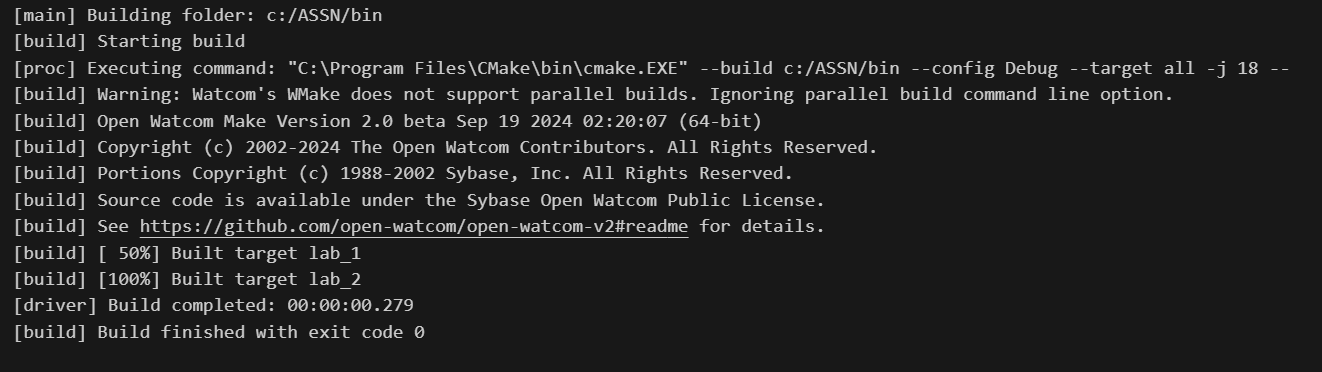
****

Рис. 2

**Визуализация тестирования:**

**Полученные результаты:**

В ходе выполнения задания удалось создать программу, которая управляет нагревательным элементом. Контроллер I-7188 успешно передавал команду на модуль I-7011, получая от него данные о температуре. Сравнив их с заданными значениями, контроллер автоматически управлял нагревательным элементом, включая или отключая его при нарушении температурных границ.

**Заключение**

Работа выполнена в полном объеме и в срок. Было подтверждено корректное взаимодействие контроллера I-7188 с модулями I-7011 и реле для управления температурным режимом. Система успешно поддерживает температуру исследуемой среды в заданном диапазоне от 29.0 tco до 30.6 tco(частота опроса 1.7 сек), включая и отключая нагревательный элемент в зависимости от поступающих данных о температуре.