Министерства науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

Институт двигателей и энергетических установок

Кафедра автоматических систем энергетических установок

Отчёт по лабораторным работам

по дисциплине «Программное обеспечение систем автоматического управления»

Выполнили студенты группы 2414

Кузьмин М. А.

Корпейкин В.

Макаров А. С.

Анисова С. А.

Проверил: проф. Матюнин С. А.

Самара 2018

# РЕФЕРАТ

Лабораторные работы.

Пояснительная записка: 25 страниц, 27 рисунков, 3 таблицы, 3 источника.

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР, ПИД-РЕГУЛЯТОР, RSLOGIXCPU 5/05, САМОПОДХВАТ, РЕГУЛЯТОР ПЕЧИ, ТАЙМЕР, СЧЕТЧИК, СТЕК

Выполнены лабораторные работы по темам: реле с самоподхватом, формирование сигналов с задержкой. таймеры-счетчики, ПИД-регулятор.

Написана программа управления температурой печи с последующим моделированием поведения печи при помощи программного обеспечения RSLogixCPU 5/05.

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc532554273)

[ВВЕДЕНИЕ 4](#_Toc532554274)

[1 Лабораторная работа №1 5](#_Toc532554275)

[2 Лабораторная работа №2 7](#_Toc532554276)

[3 Лабораторная работа №3 9](#_Toc532554277)

[4 Лабораторная работа №4 13](#_Toc532554278)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc532554279)

# ВВЕДЕНИЕ

Современное производство представляет собой совокупность сложных технологический операций, эффективность и качество которых сильно зависит от степени автоматизации. Даже самый лучший комплект оборудования не сможет эксплуатироваться на необходимом уровне, без качественно продуманной и реализованной системы управления. Одним из вариантов построения системы управления являются промышленные контроллеры. Благодаря быстрому развитию различных семейств ПЛК и постоянной поддержки разработчиками, промышленные контроллеры с достоинством занимают свое место в системах автоматизации.

# 1 Лабораторная работа №1

Задание: написать программу, в которой реализуется замкнутая электрическая цепь с помощью реле.

Принцип работы:

1. При нажатии на кнопку «START» срабатывает реле самоподхвата и замыкает кнопку «START». Поэтому при отпускании кнопки «START» электрическая цепь остается замкнутой.

2. При нажатии на кнопку «STOP» электрическая цепь размыкается и реле самоподхвата выключается.

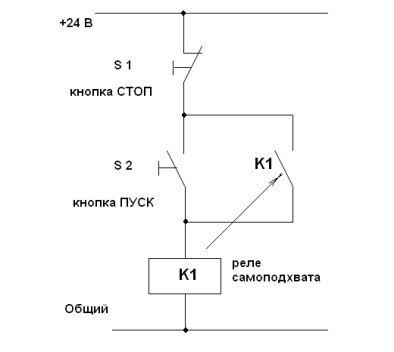


Рисунок 1 - Схема работы электрической цепи

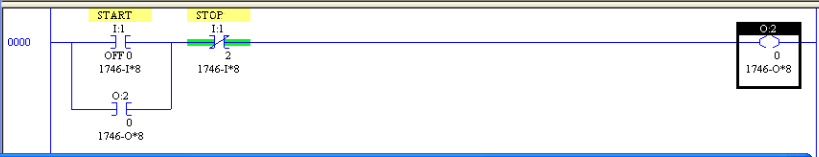
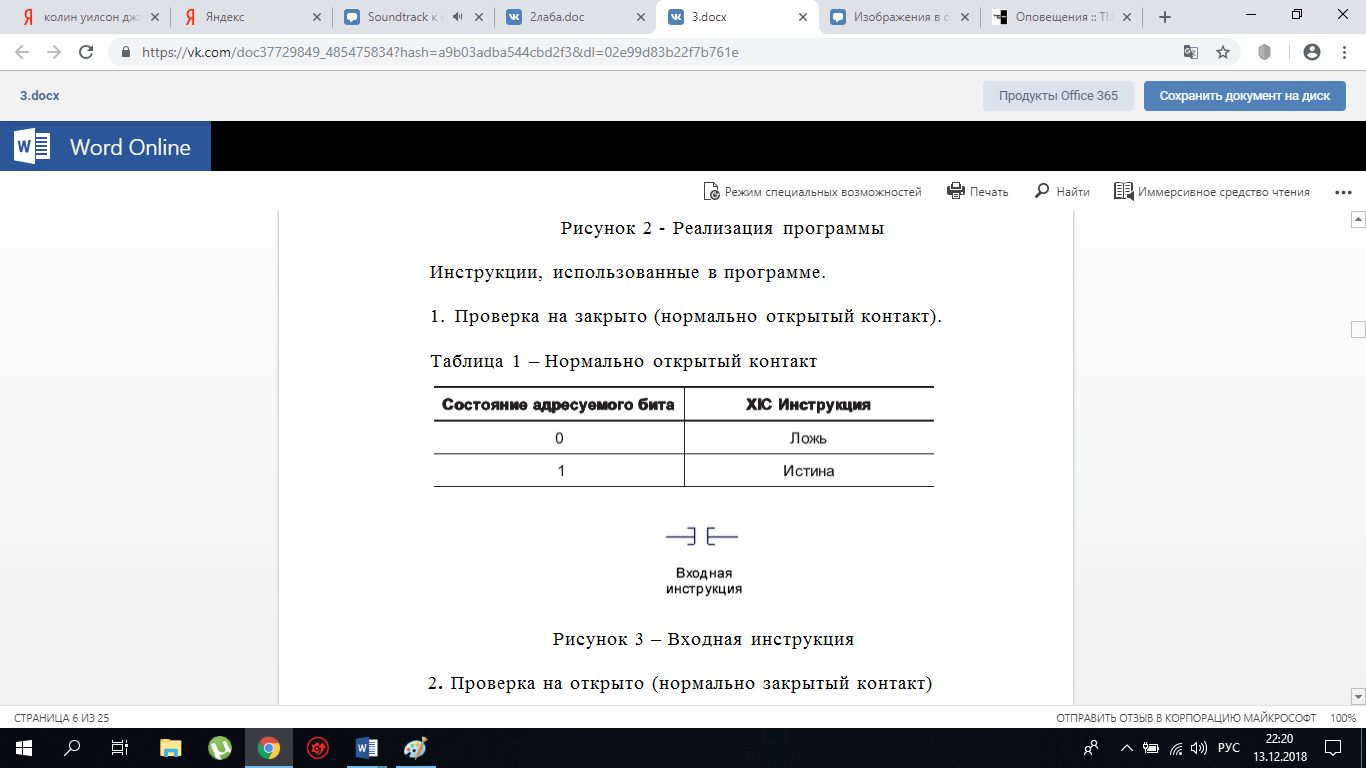


Рисунок 2 - Реализация программы

Инструкции, использованные в программе.

1. Проверка на закрыто (нормально открытый контакт).

Таблица 1 – Нормально открытый контакт



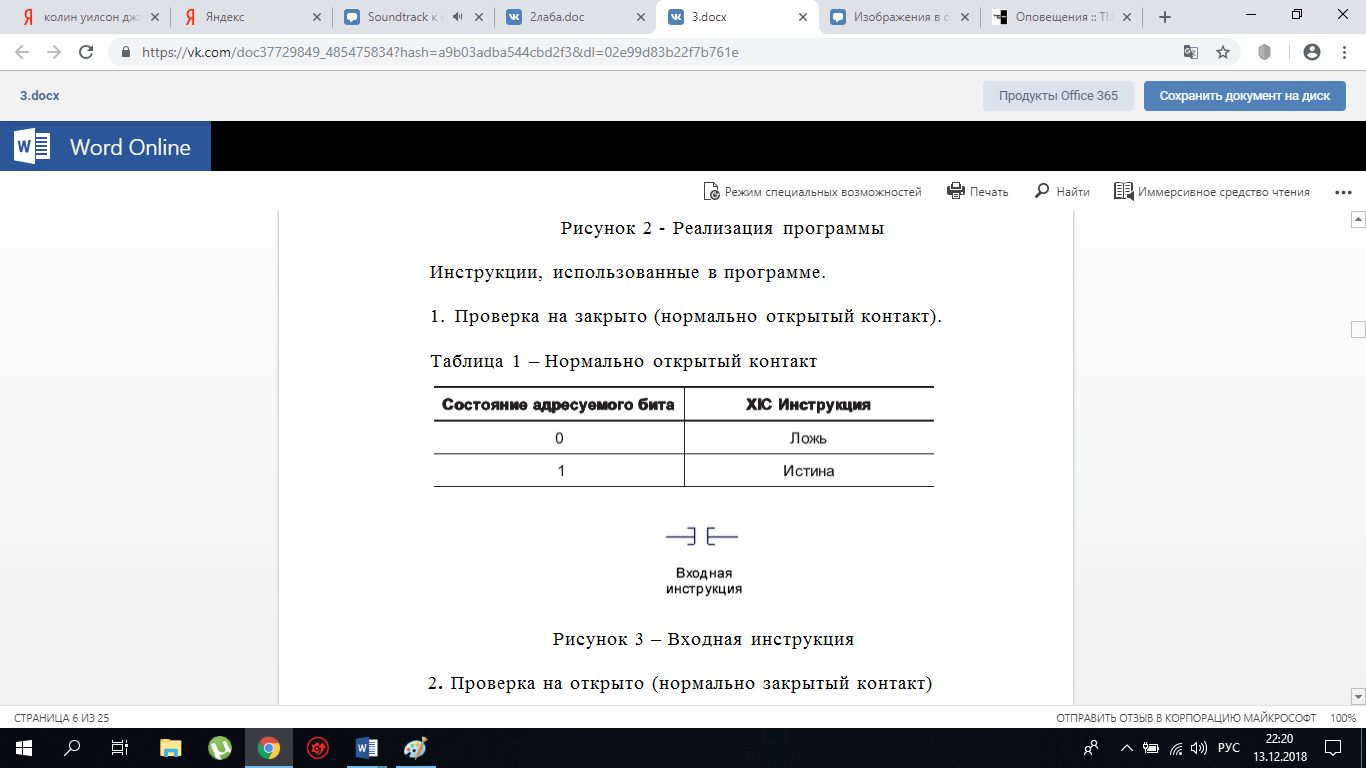
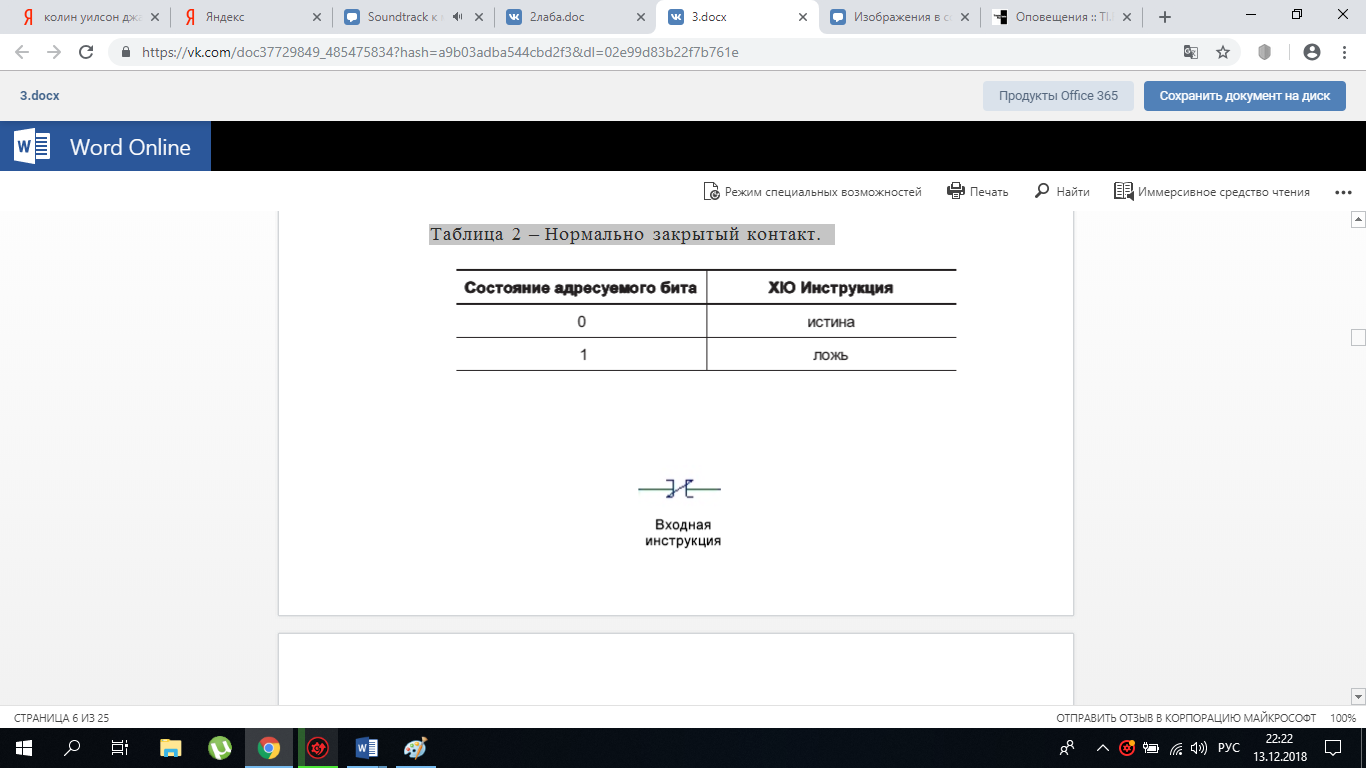


Рисунок 3 - Входная инструкция нормально открытая

2. Проверка на открыто (нормально закрытый контакт)

Таблица 2 – Нормально закрытый контакт



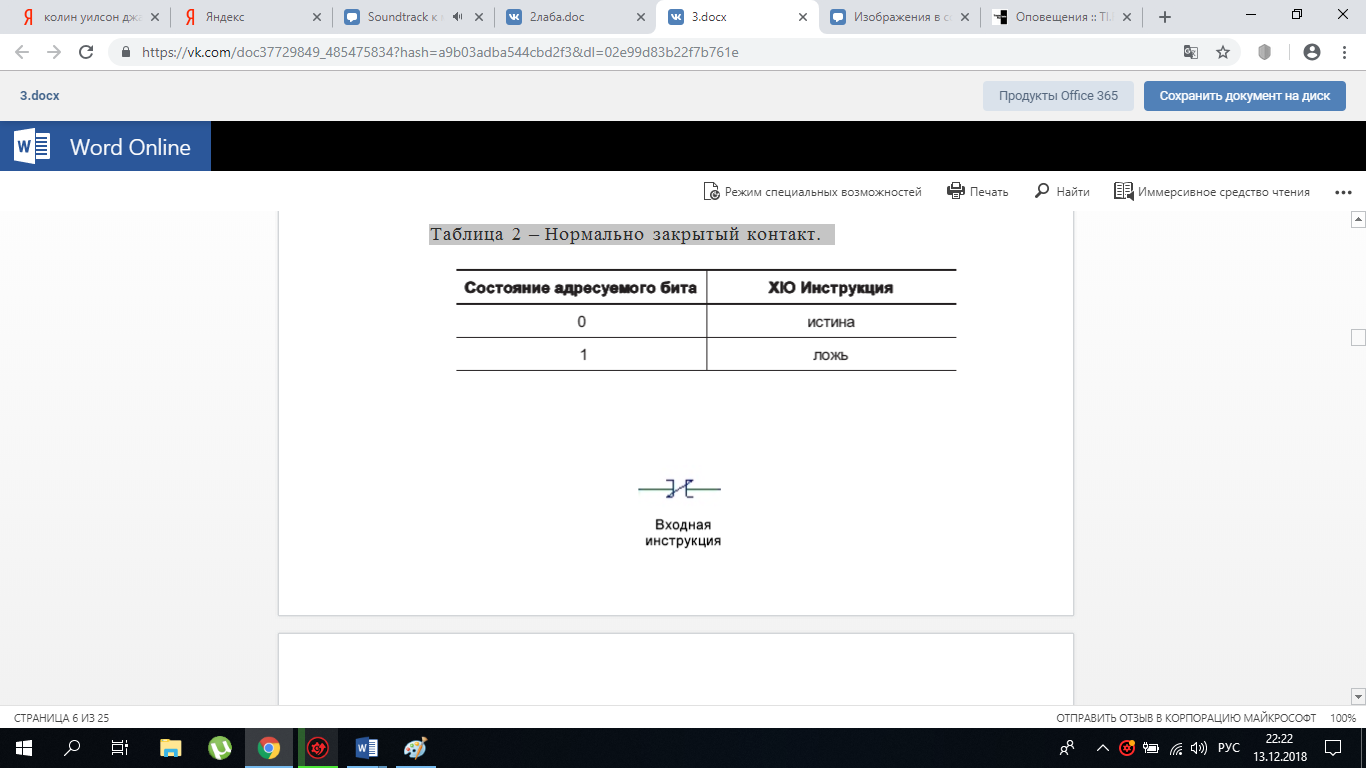


Рисунок 4 – Входная инструкция нормально закрытая

3. Включение бита

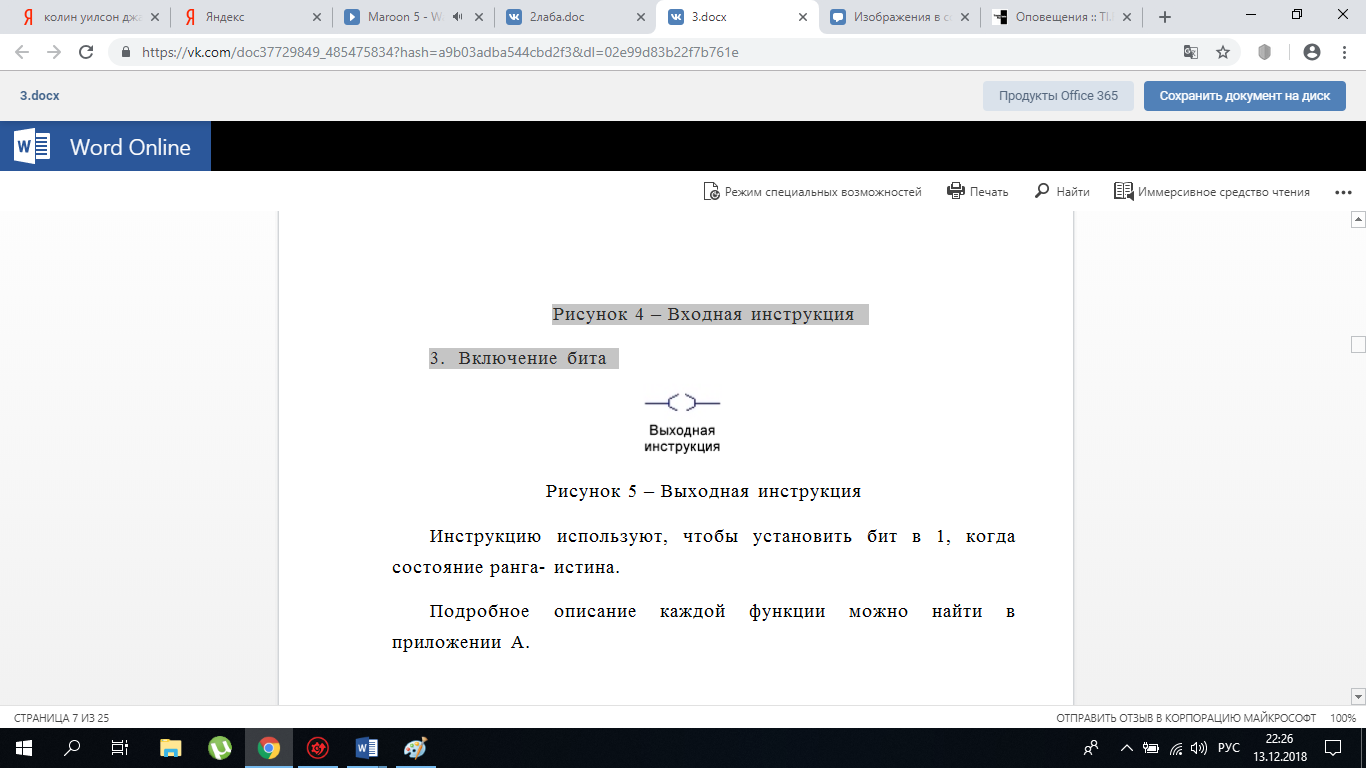


Рисунок 5 – Выходная инструкция

Инструкцию используют, чтобы установить бит в 1, когда состояние ранга- истина.

# 2 Лабораторная работа №2

Задание: Написать программу для формирования сигналов с задержкой по включению.

Принцип работы:

Когда входной сигнал F1 (например, кнопка ПУСК) изменяет свое состояние от «нуля» к «единице» активируется таймер 1.

Дискретный выход OUTPUT 2 должен установиться в состоянии «единица» и оставаться в этом состоянии на все время работы таймера.

Через 7,5 секунд от активации таймера дискретный выход OUTPUT 3 устанавливается в «ноль»

Дискретный выход OUTPUT 4 должен быть установлен в состояние «единица» по прошествии этого времени и сбросится в «ноль» по окончанию работы таймера.

При выполнении этого задания в качестве входного сигнала используйте битовую переменную.

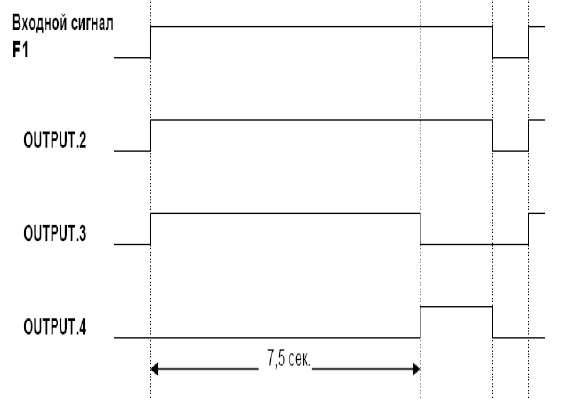


Рисунок 6 - Схема формирования сигналов

Для написания программы были использованы следующие функция таймера с задержкой включения:

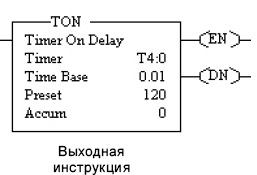


Рисунок 7 – Таймер

Реализация программы представлена на рисунке 8.

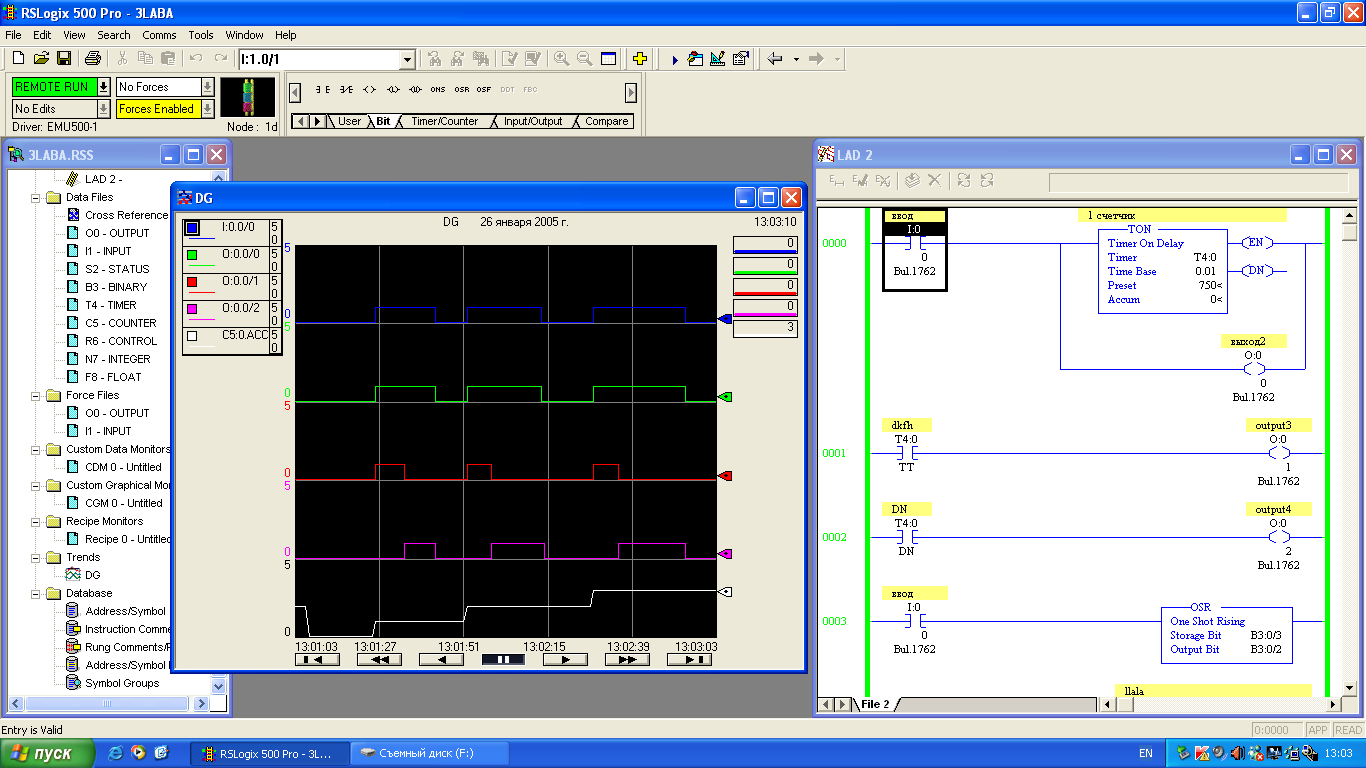


Рисунок 8 – Реализация формирования сигналов

# 3 Лабораторная работа №3

Задание: Создать генератор бесконечного сигнала, который, каждые Т секунд, увеличивает частоту сигнала N раз подряд, а потом уменьшает частоту N раз подряд. Создать бесконечный цикл выполняемых действий.

Принцип работы:

1. При нажатии на кнопку «Пуск» включается таймер 1.

2. Когда таймер 1 заканчивает отсчёт, включается таймер 2, который, в свою очередь, после окончания отсчёта перезапустит таймер 1.

3. «Счётчик вверх» отсчитывает N перезапусков таймеров, причём с каждым перезапуском задержка таймера 1 увеличивается на 0,25 с.

4. Когда «Счётчик вверх» переполнится, запустится «Счётчик вниз» и теперь задержка таймера будет уменьшаться на 0,25 с после перезапуска.

5. Когда «Счётчик вниз» отсчитает N итераций, «Счётчик вверх» перезапустится, а в аккумулятор «Счётчика вниз» запишется начальное значение, и процесс повторится.

Используемые элементы:

1. Таймер с задержкой включения

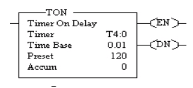




Рисунок 9 – Таймер

1. Счетчик вверх



Рисунок 10 – Счетчик вверх

3. Счетчик вниз

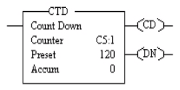


Рисунок 11 – Счетчик вниз

4. Равно

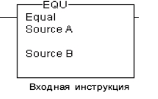


Рисунок 12 – Равно

5. Сложение

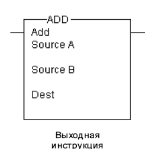


Рисунок 13 –Сложение

6. Вычитание

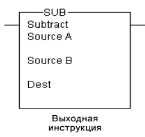


Рисунок 14 – Вычитание

7. Меньше чем

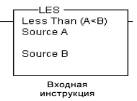


Рисунок 15 – Сравнение

8. Умножение

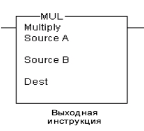


Рисунок 16 – Умножение

9. Деление



Рисунок 17 – Деление

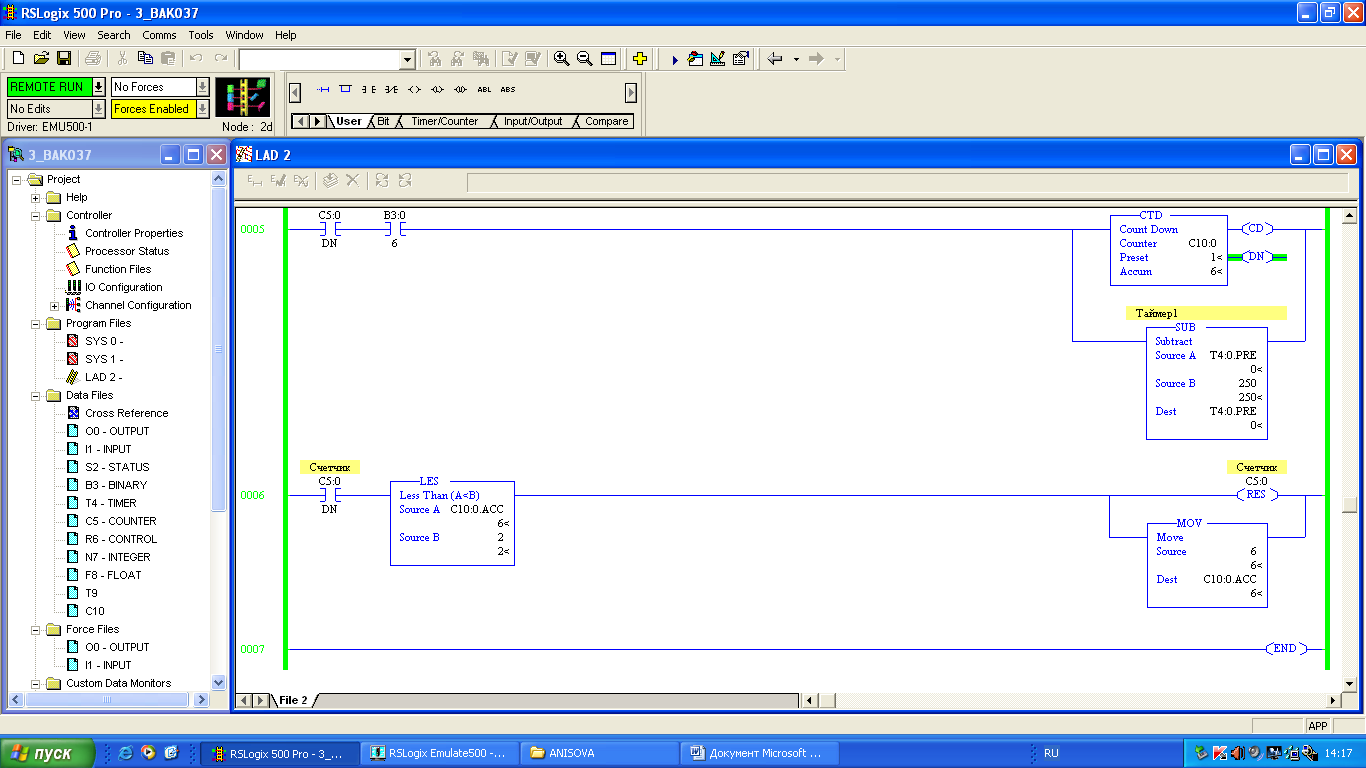
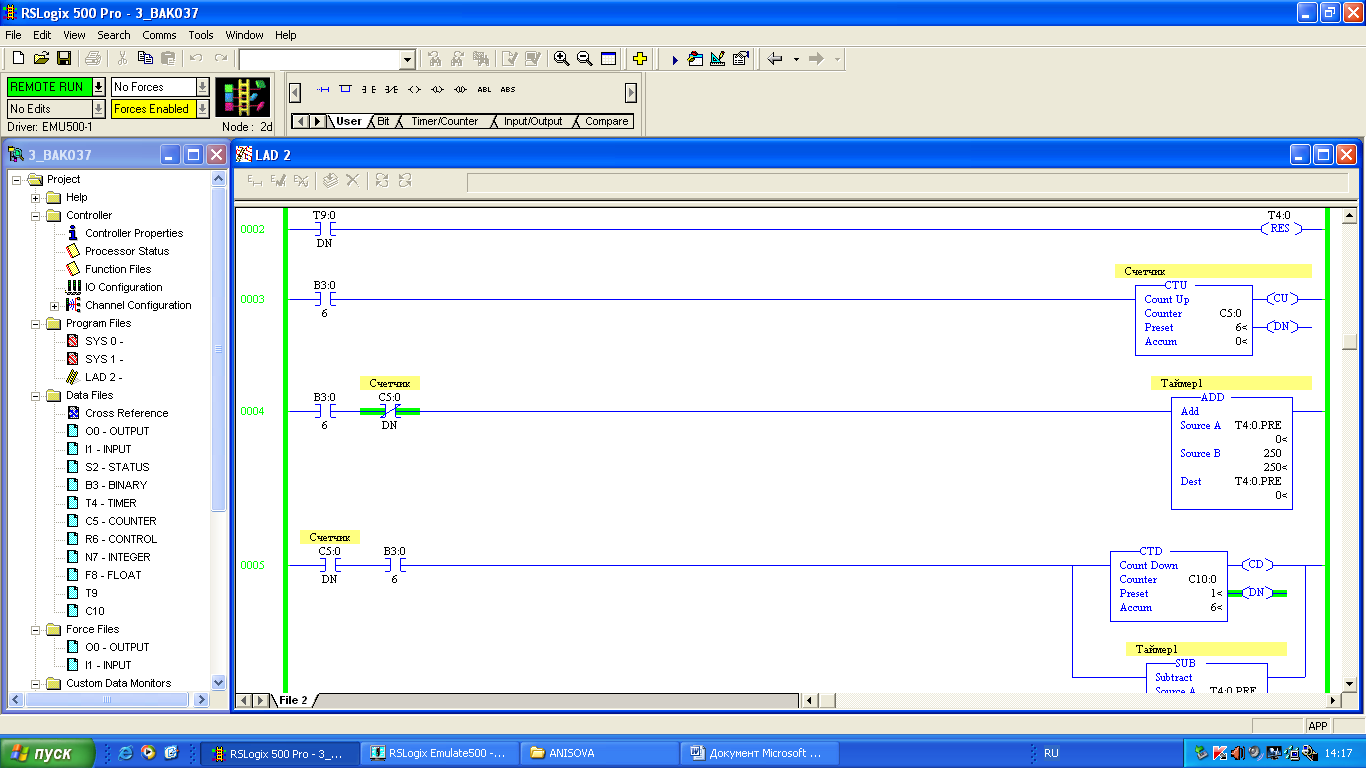
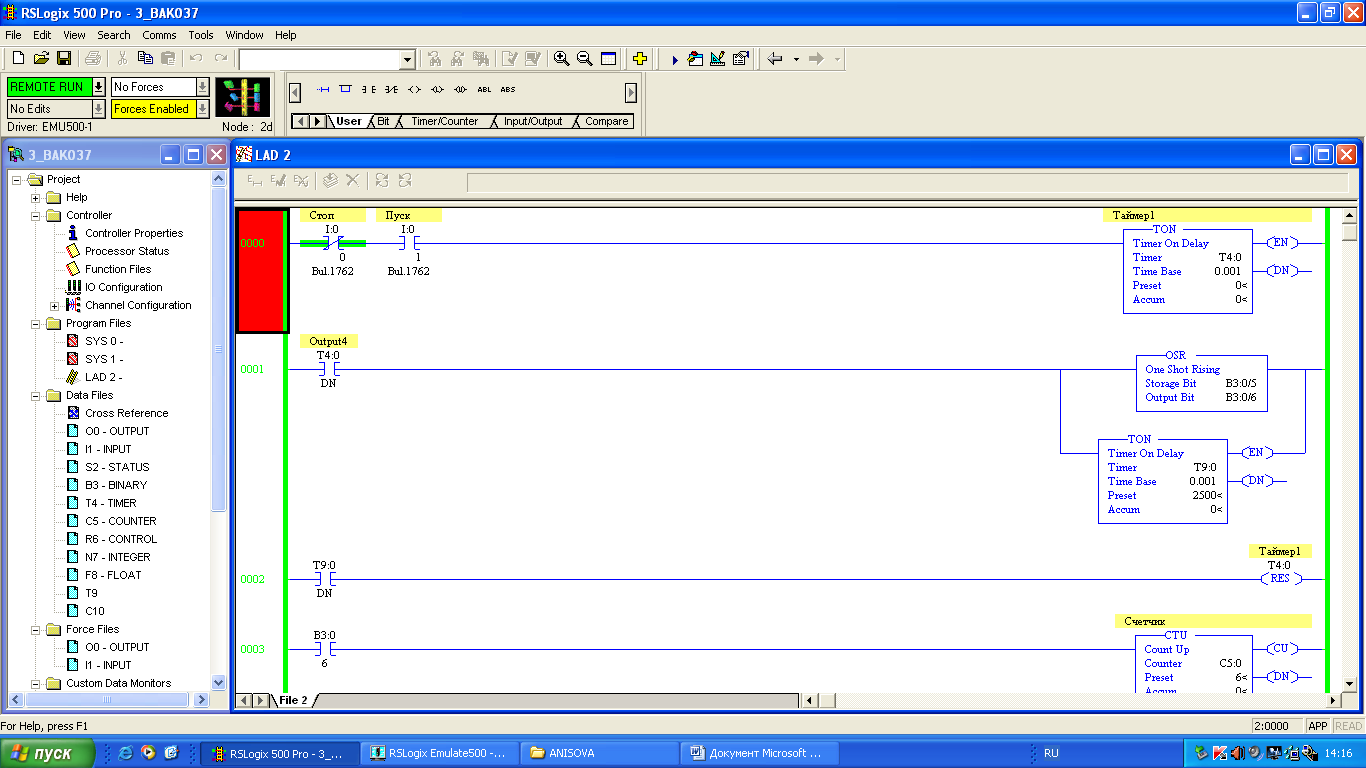


Рисунок 18 - Реализация генератора импульсов с переменной скважностью

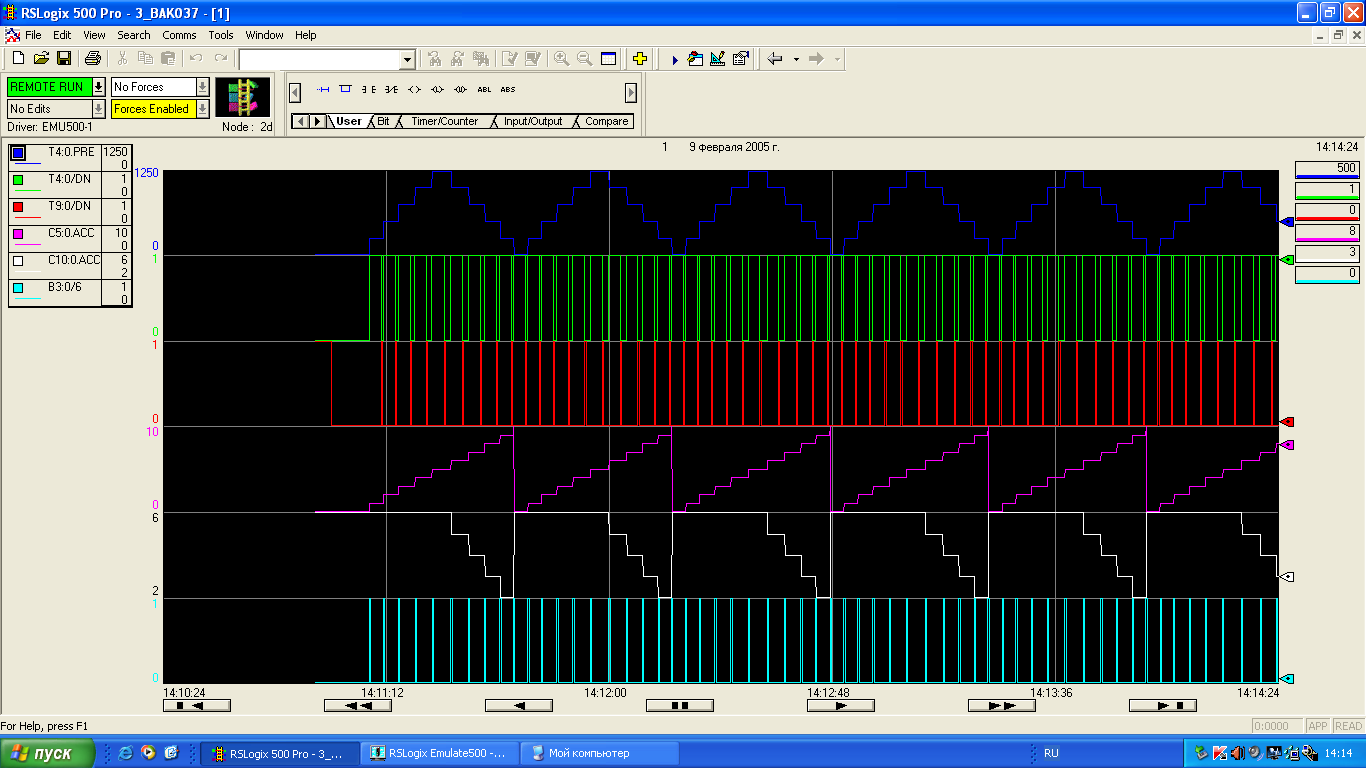


Рисунок 19 - График импульсов с переменной скважностью

# 4 Лабораторная работа №4

Задание: создать симуляцию работы системы согласно структурной схеме (рис.20) с заданными параметрами.

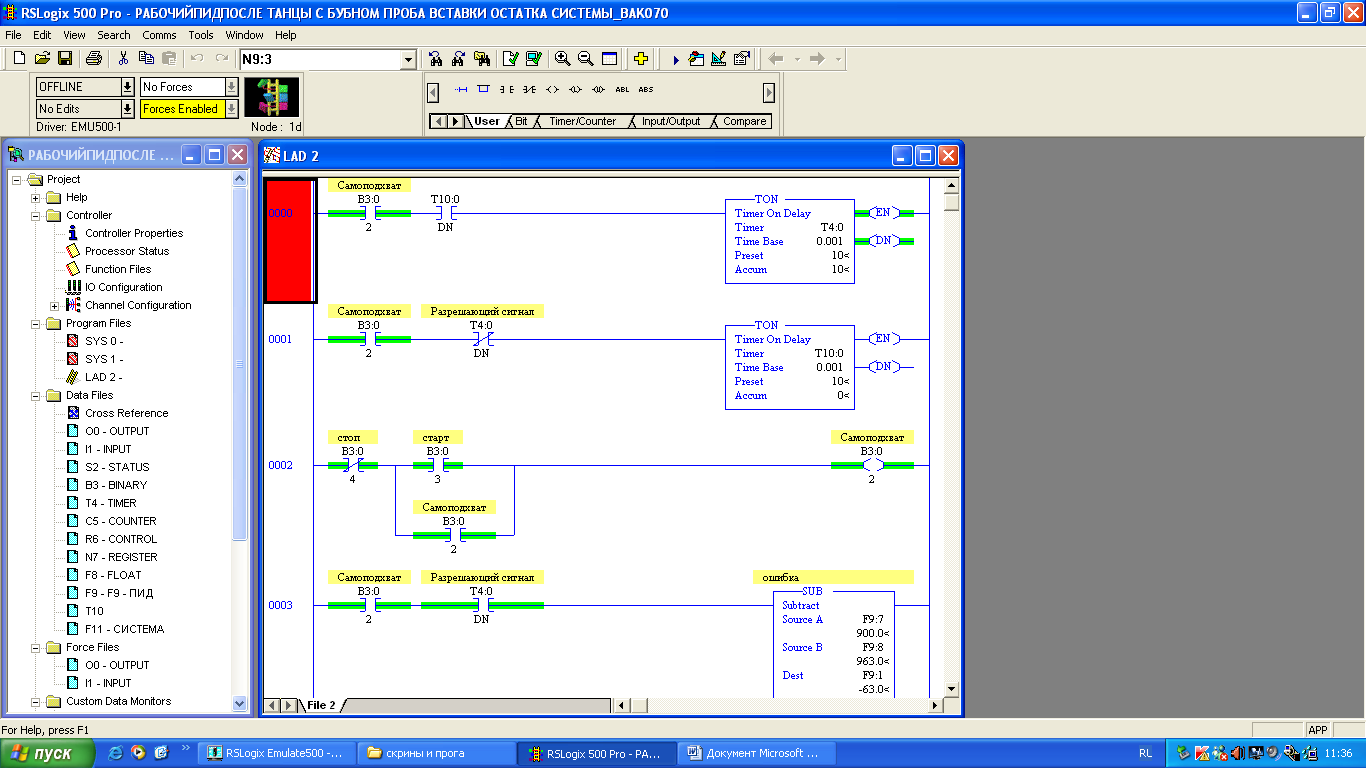


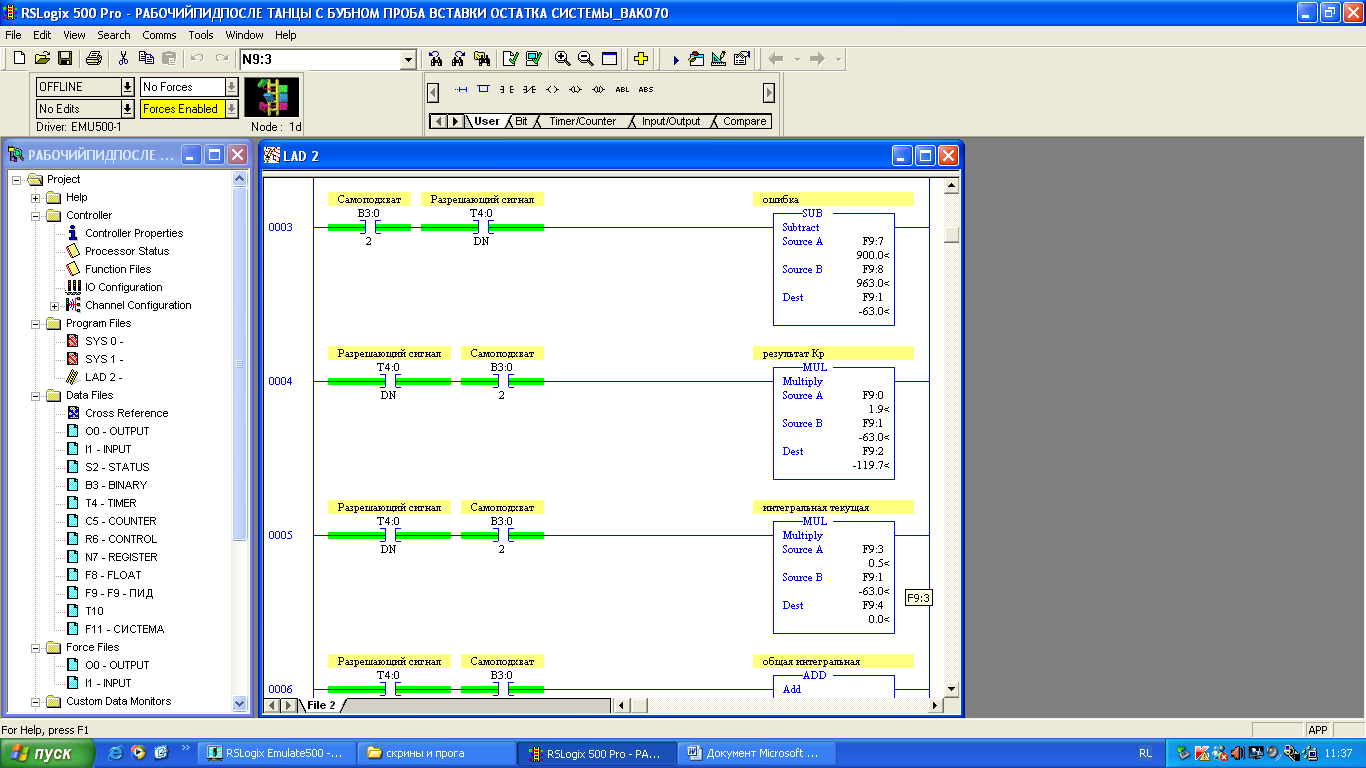
Рисунок 20 – Структурная схема системы

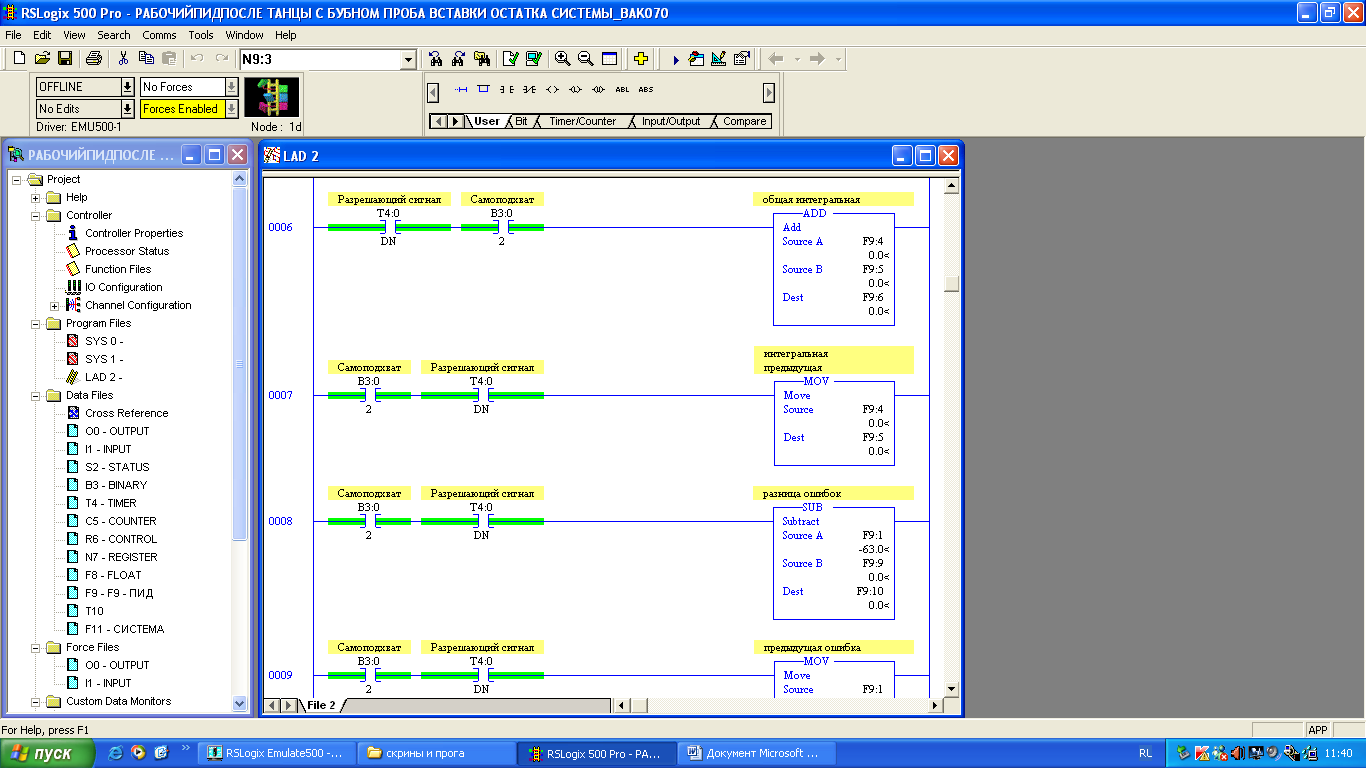
Параметры:

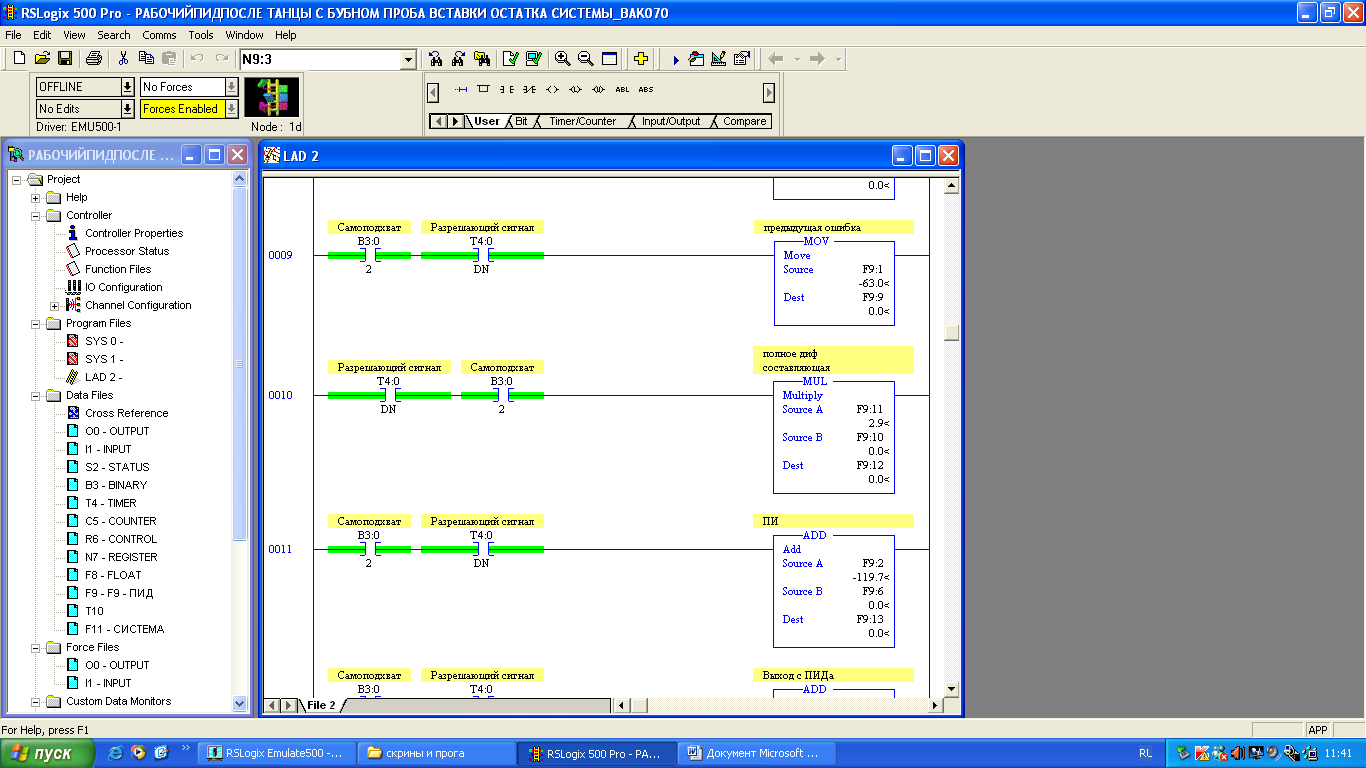
1. ПЛК – Allen-Bradley Micrologix 1200 серия C.
2. Используемое ПО: Allen-Bradley RSLogix 500, RSLinx 2.42.00, RSLogix Emulate 500.
3. Транспортная задержка Tз = 0…500 мс.
4. Допустимая величина перерегулирования не более 10%.
5. Время выхода печи на режим не более 100 секунд.
6. Постоянная печи Tп=20 с. Температура печи от 0 до 900 градусов Цельсия. Коэффициент передачи печи Kп = 4 гр./ватт.
7. Датчик температуры – термопара. Чувствительность термопары Kд = 20 мкВ/гр.
8. Постоянная датчика Tд = 0.3 с.
9. Коэффициент передачи усилителя Kу = 0.5 ватт/В.

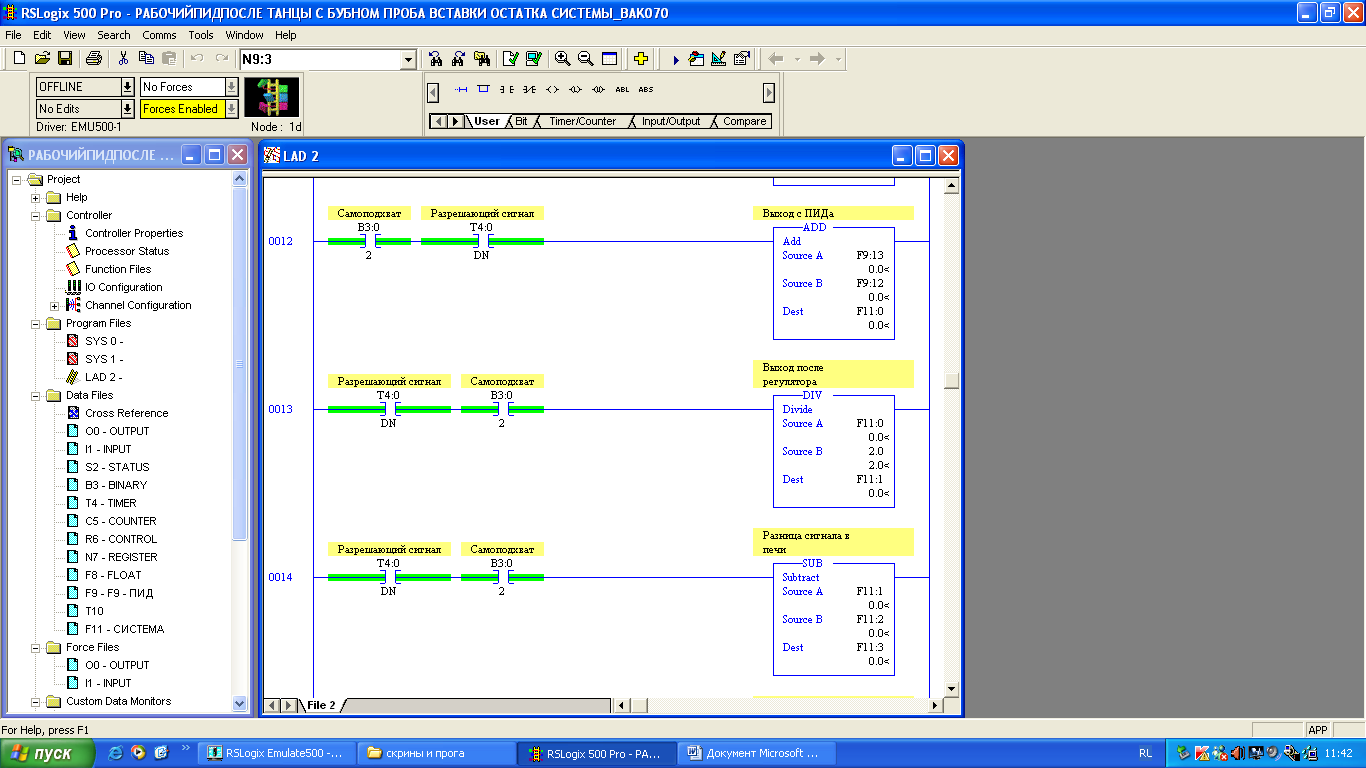
Программа представлена на рисунке 21.

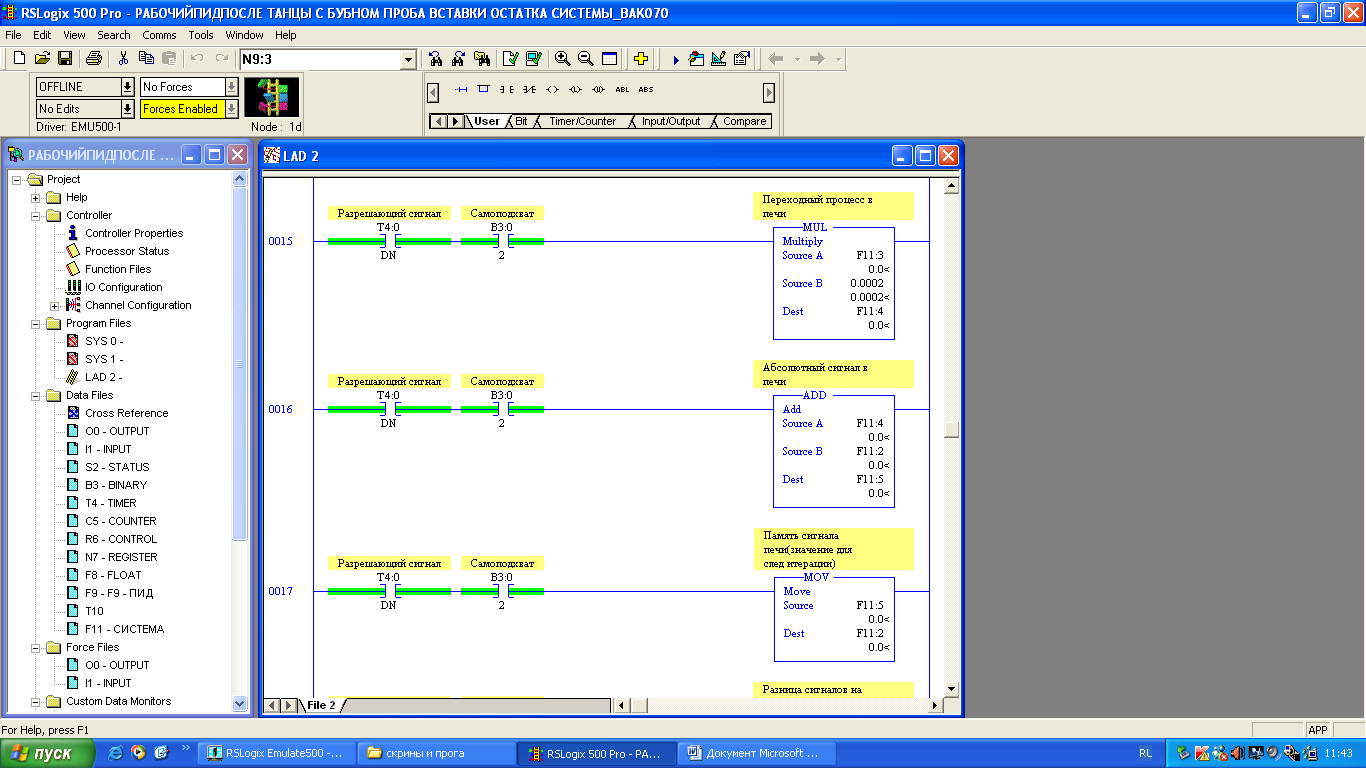


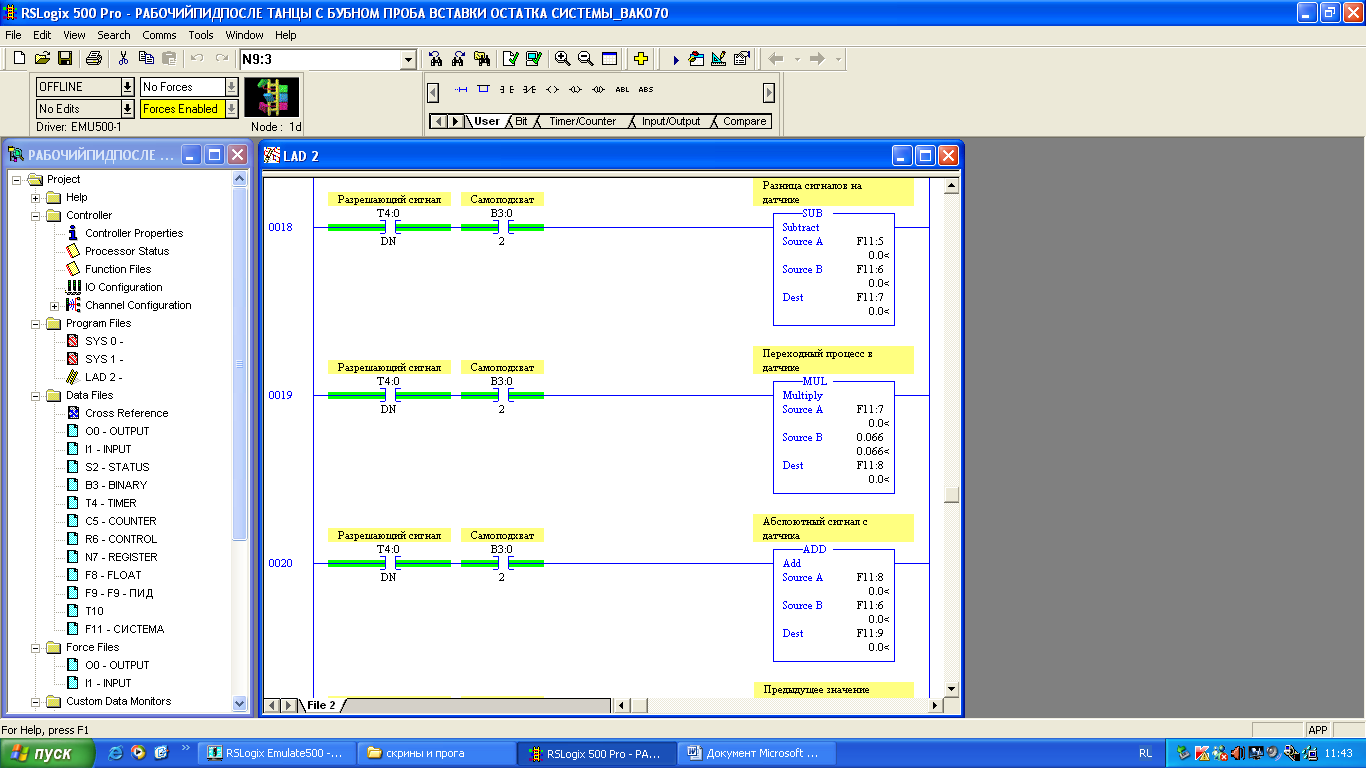


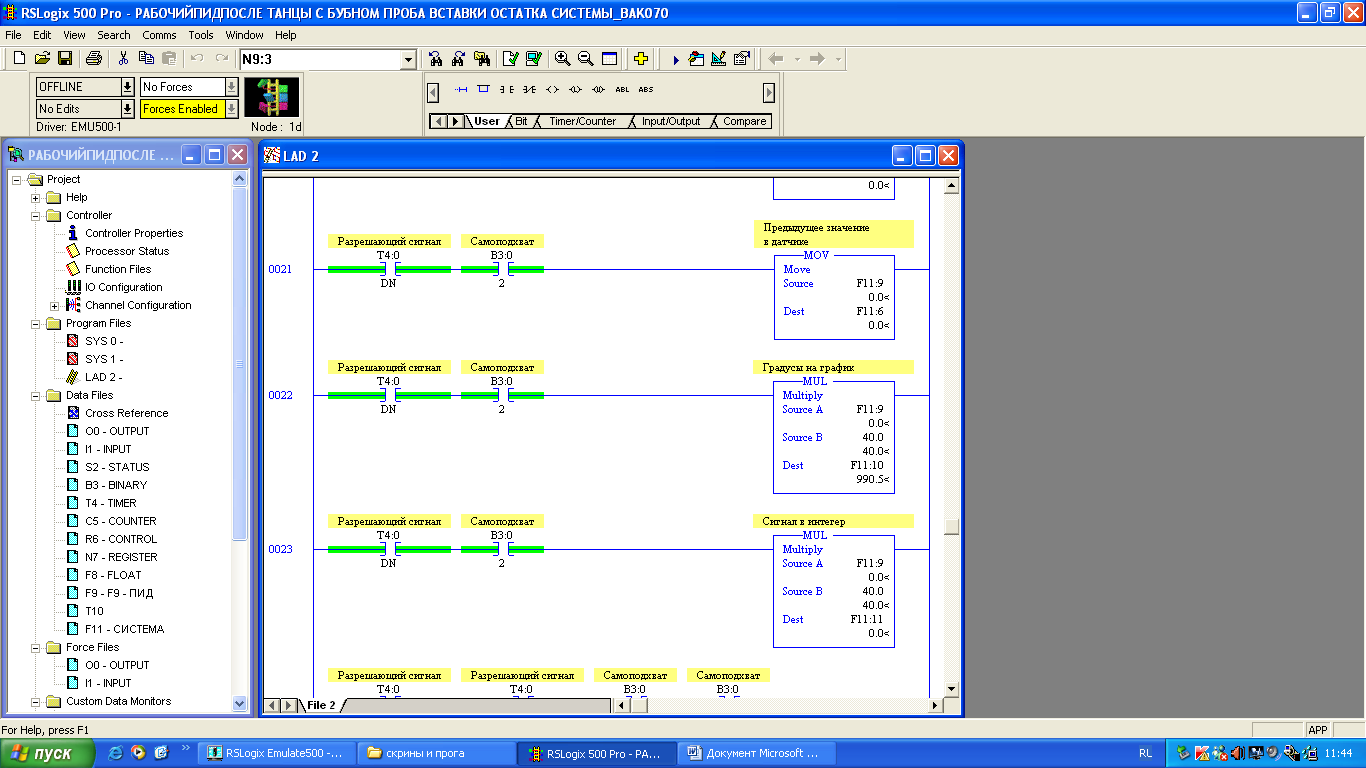


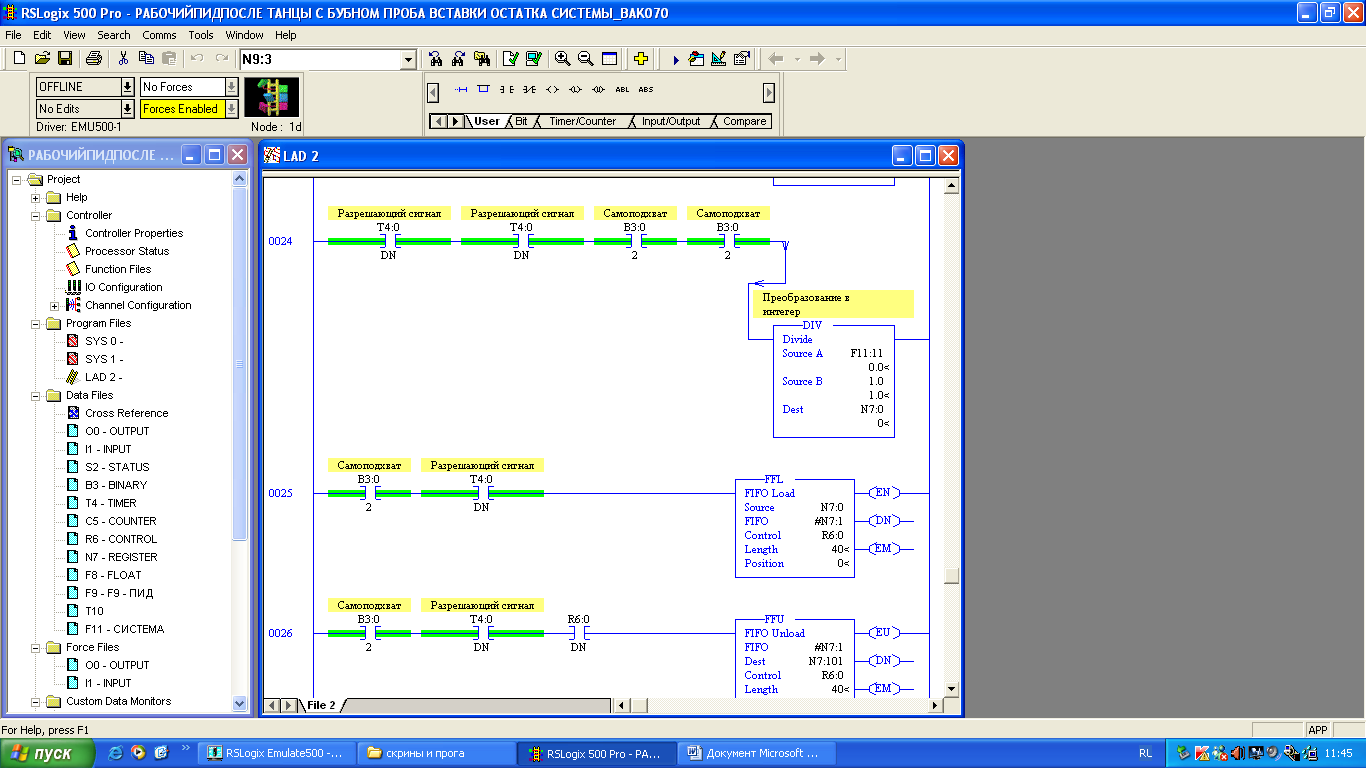












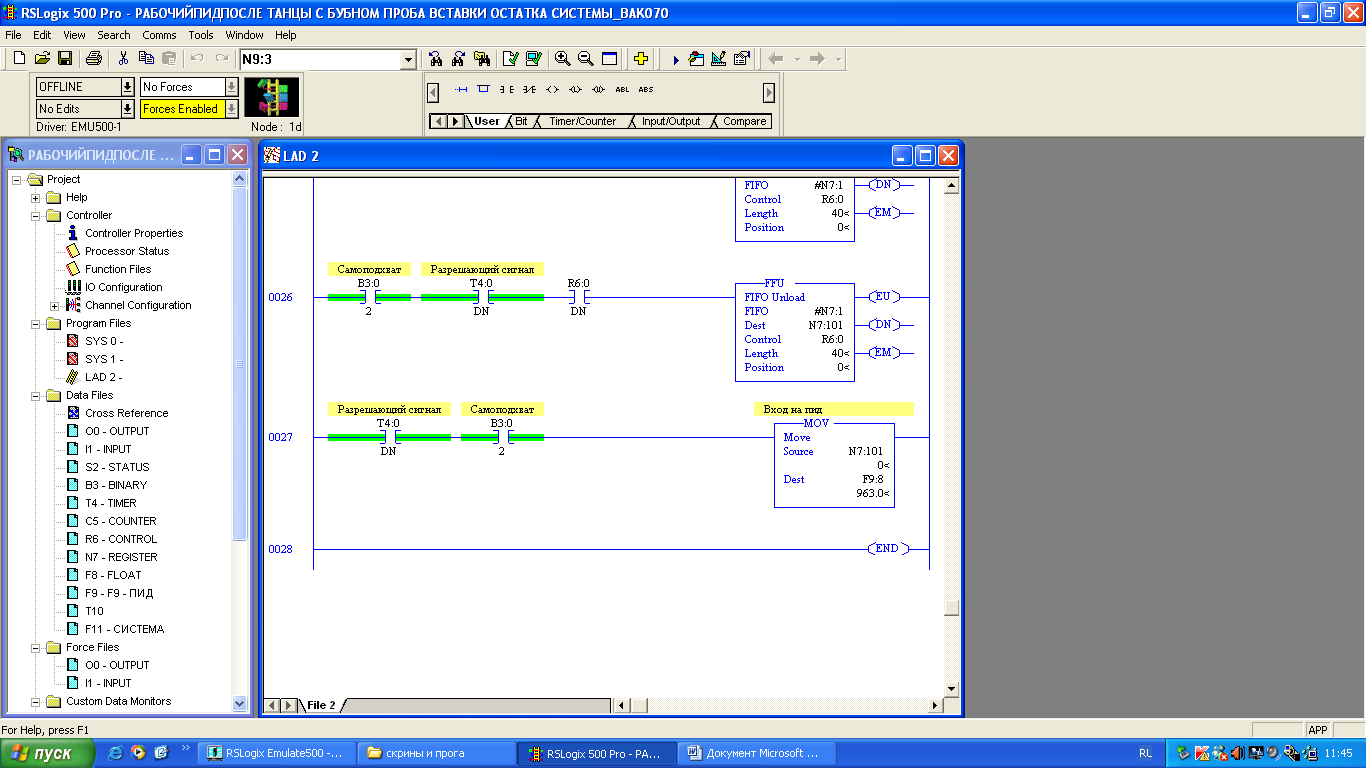


Рисунок 21 – программа регулятора печи

Используемые переменные:

* Блок ПИД- регулятора:

1. F9:0 – Кп – пропорциональный коэффициент ПИД-регулятора;
2. F9:1 – текущая ошибка *e* (*t*);
3. F9:2 – Пропорциональная составляющая регулятора П;
4. F9:3 - Ки – интегральный коэффициент ПИД-регулятора;
5. F9:4 - текущая интегральная ошибка;
6. F9:5 - общая интегральная ошибка I;
7. F9:6 - память интегральной ошибки для следующей итерации;
8. F9:7 - уставка;
9. F9:8 - текущее значение температуры в печи;
10. F9:9 - предыдущее значение дифференциальной составляющей;
11. F9:10 - разница ошибок дифференциальной составляющей;
12. F9:11 - Кд – дифференциальный коэффициент ПИД-регулятора;
13. F9:12 - дифференциальная составляющая регулятора Д;
14. F9:13 - сумма П + I регулятора;
15. F11:0 - П+Д+ I;

* Система:

1. F11:1 – сигнал после прохождения через усилитель;
2. F11:2 – сигнал в печи на предыдущей итерации;
3. F11:3 – разница текущего и предыдущего сигнала в печи;
4. F11:4 – сигнал после переходного процесса в печи;
5. F11:5 – сигнал с печи суммарный;
6. F11:6 – сигнал в датчике на предыдущей итерации;
7. F11:7 – разница текущего и предыдущего сигнала в датчике;
8. F11:8 – сигнал после переходного процесса в датчике;
9. F11:9 – сигнал с датчика суммарный;
10. F11:10 – масштабирование сигнала с датчика в градусы;
11. F11:11 – масштабирование сигнала с датчика в тип Integer для ПИД-регулятора;
12. N7:0 – преобразование типа Float в тип Integer;
13. N7 – массив ячеек, реализующих сдвиговый регистр заданной длины
14. N7:101 – ячейка массива, из которой информация записывается в ПИД-регулятор.

Принцип работы:

Синтез ПИД регулятора: формула ПИД-регулятора приведена на рисунке 2.

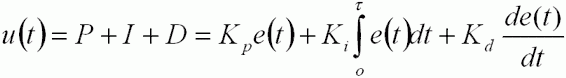


Рисунок 22 – Формула ПИД-регулятора

Для синтеза регулятора использовалась дискретная реализация, соответствующая формулам нахождения трех составляющих согласно:

*u*(*t*) = *P* (*t*) + *I* (*t*) + *D* (*t*);  
*P* (*t*) = *Kp* \* *e* (*t*);  
*I* (*t*) = *I* (*t* — 1) + *Ki* \* *e* (*t*);  
*D* (*t*) = *Kd* \* {*e* (*t*) — *e* (*t* — 1)}, где

* *u* (*t*) — наша функция;
* *P* — пропорциональная составляющая;
* *I* — интегральная составляющая;
* *D* — дифференциальная составляющая;
* *e* (*t*) – текущая ошибка;
* *Kp* — пропорциональный коэффициент;
* *Ki* — интегральный коэффициент;
* *Kd* — дифференциальный коэффициент;

Все эти преобразования происходили с помощью переменных из блока F9, указанных выше.

Дальнейшее прохождение сигнала через систему:

Усилитель имеет передаточную функцию Wу = 0,5. Сигнал, проходя через него ослабевает в два раза.

Печь и датчик имеют одинаковую структуру аппериодических звеньев с разными коэффициентами усиления и постоянной времени. Для симуляции прохождения сигнала через эти два звена удобно использовать метод линеаризации: зная значение сигнала в предыдущий (Х0) и текущий (Х1) момент времени мы можем найти разницу значений этих сигналов (Δy(t)) в промежуток времени **Δ**X = Х1 – Х0. Имея коэффициенты передаточной функции печи и датчика мы можем представить ее в виде:  . Далее мы умножаем разницу сигналов на эту функцию, которая является константой и прибавляем полученное значение к текущему значению сигнала, и переходим к следующей итерации, запоминая данное значение в отдельную переменную.

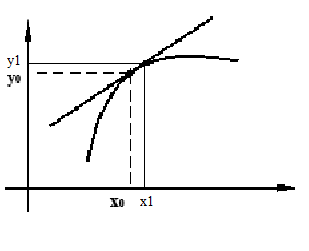


Рисунок 23 – Линеаризация нелинейной функции

Далее необходимо реализовать транспортную задержку. Для этого мы будем использовать массив элементов типа Integer. Саму задержку мы можем менять, изменяя количество элементов в массиве. Мы используем данный массив по аналогии со сдвиговым регистром с помощью функций FIFO load, FIFO unload, которые загружают информацию в массив, сдвигая его на 1, и выгружают из определенной ячейки соответственно. В данном случае эта ячейка N7:101. Далее этот сигнал снова приходит в ПИД-регулятор и происходит следующая итерация.

Полученные сигналы приведены на рисунках 4, 5, 6, 7 для транспортных задержек 0, 100, 200, 300 миллисекунд соответственно.

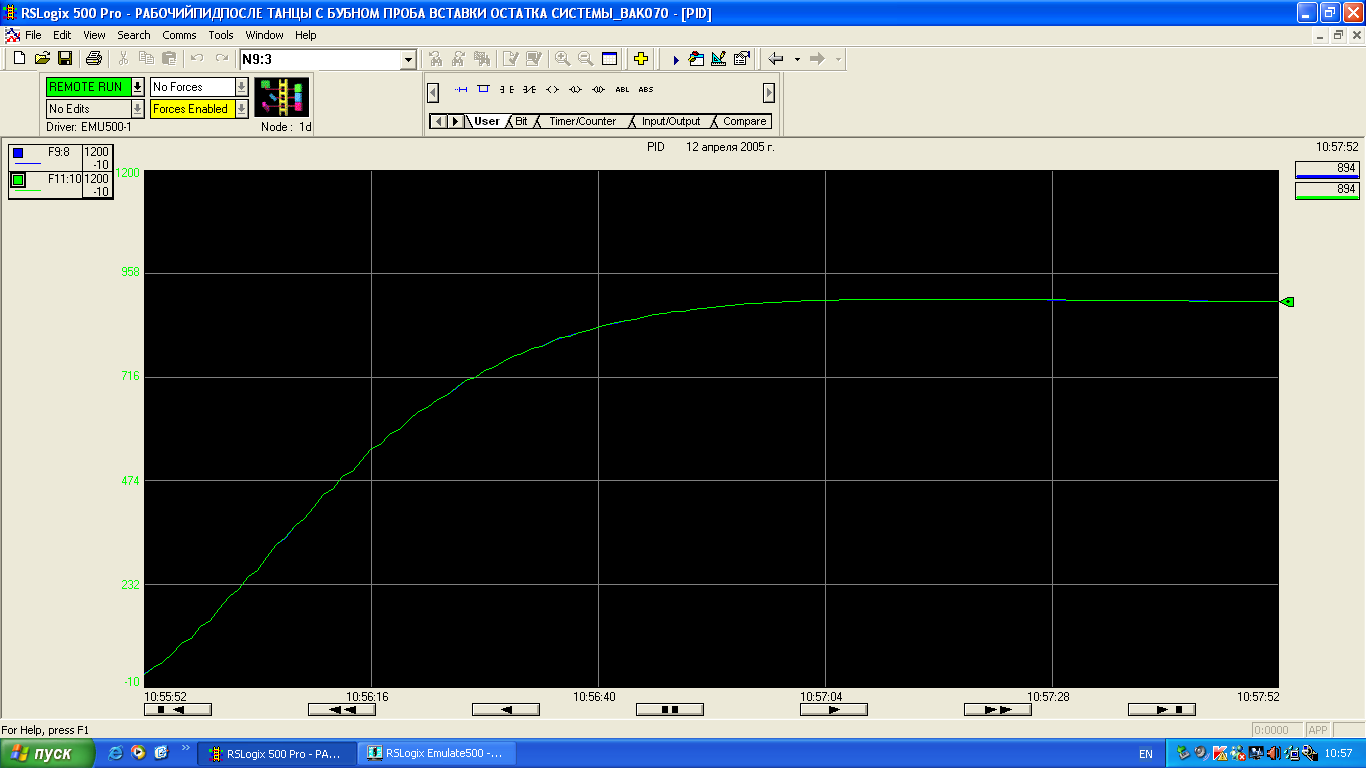


Рисунок 24 – Процесс для транспортной задержки 0 миллисекунд

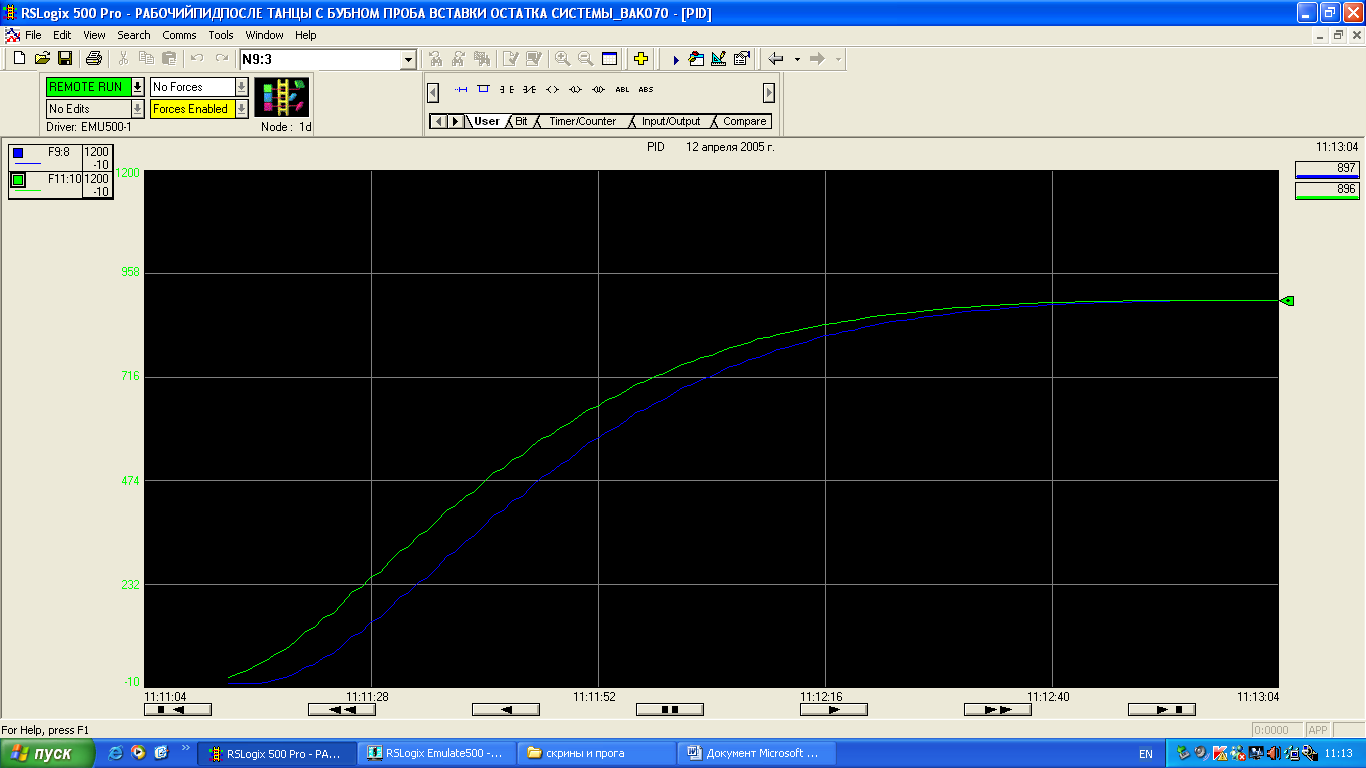


Рисунок 25 – Процесс для транспортной задержки 100 миллисекунд

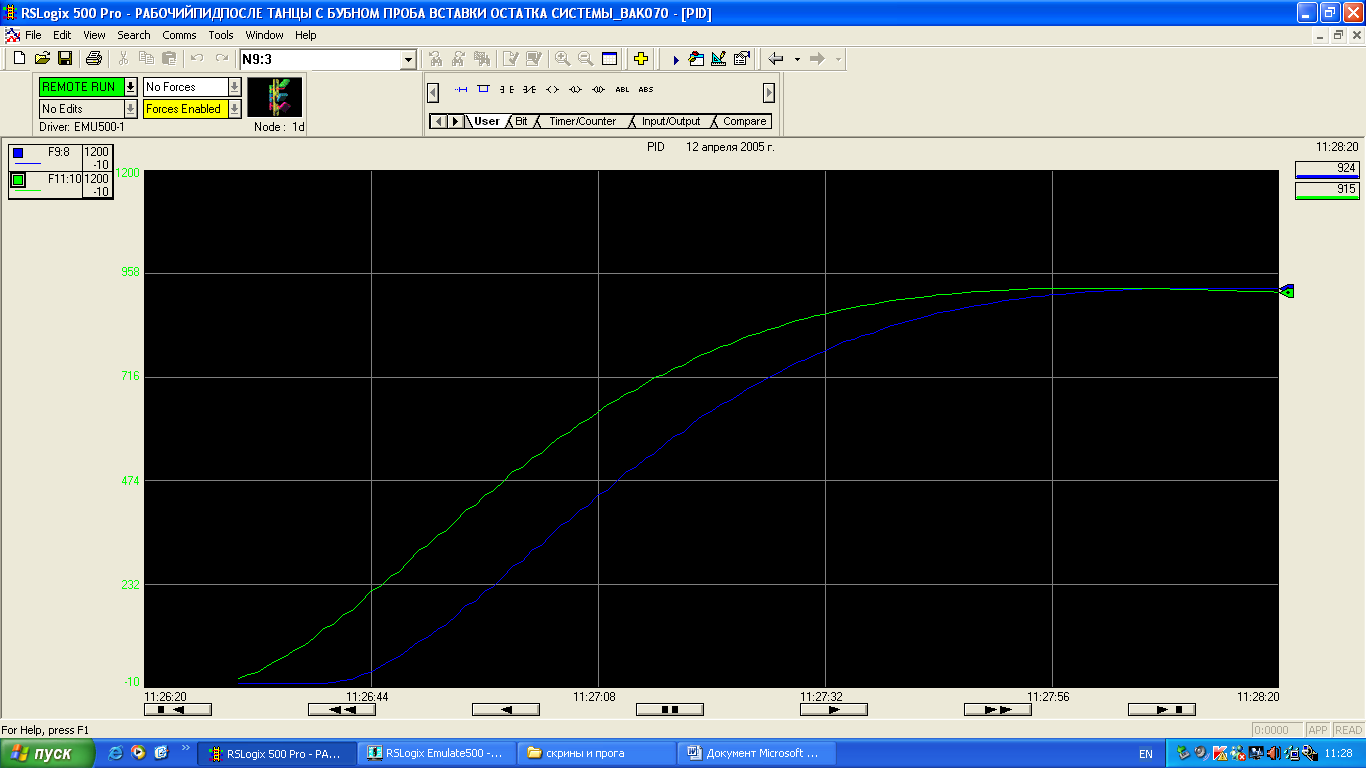


Рисунок 26 – Процесс для транспортной задержки 200 миллисекунд

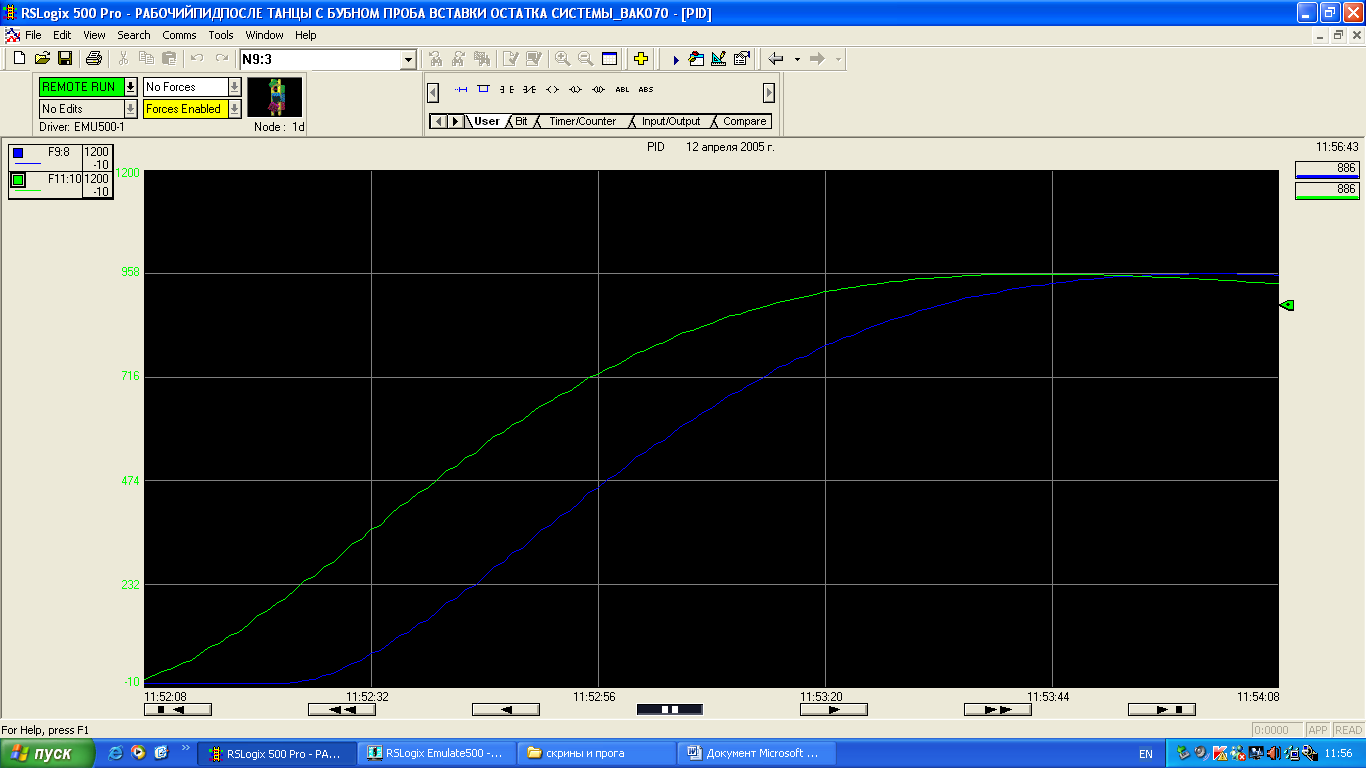


Рисунок 27 – Процесс для транспортной задержки 300 миллисекунд

Для каждой транспортной задержки необходимо перенастраивать ПИД-регулятор, так как она сильно влияет на переходный процесс. Коэффициенты приведены в таблице 1.

Таблица 3 – Коэффициенты при различных задержках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Коэффициент | T=0 мс | T=100 мс | T=200 мс | T=300 мс |
| Кп | 5 | 3 | 2.8 | 2.1 |
| Ки | 3 | 0.8 | 4 | 3.9 |
| Кд | 0.6 | 0.6 | 0.5 | 0.6 |

Из таблицы видно, что изменение коэффициентов ПИД-регулятора довольно велико, и при любом изменении транспортной задержку нужно проводить автоматическую настройку регулятора.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 MicroLogix 1200 ProgrammableControllersUserManual. Руководство

пользователя. –RockwellAutomation. –2010. –172c.

2 СТО СГАУ 02068410-004-2007. Общие требования к оформлению учебных текстовых документов: Методические указания. Самара: ИПО СГАУ, 2007. – 30с.

3 HELP MicroLogix 500 Pro.