

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	2
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ .....	3
1.1 Описание пневмоцилиндра .....	3
1.2 Описание модели пневмоцилиндра в среде SimInTech .....	5
2 ПРАКТИЧЕСКАЯ ЧАСТЬ .....	9
2.1 Описание задания .....	9
2.2 Исходные данные .....	10
2.3 Построение диаграммы Мура автоматной модели .....	10
2.4 Кодирование автоматной модели на языке Си .....	11
2.5 Моделирование автомата в среде SimInTech .....	14
ЗАКЛЮЧЕНИЕ .....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	20

## **ВВЕДЕНИЕ**

Целью курсовой работы является построение автоматной модели управления пневмоцилиндрами по заданной управляющей последовательности. Для выполнения необходимо по исходным данным построить диаграмму Мура. Написать на языке Си автомат. В программной среде SimInTech сгенерировать DLL, и построить модель, содержащую 8 пневмоцилиндров, после чего протестировать работу автомата: полный цикл работы, с ошибкой без перехода из состояния в ошибочное состояние, и проверить, что происходит переход из состояния в другое состояние, в случае ошибки, везде, где присутствует переход.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Описание пневмоцилиндра

Пневматический цилиндр — это один из элементов пневмосистемы. Он предназначен для преобразования энергии сжатого воздуха в механическое линейное перемещение в целях подъема или сдвига с силой, пропорциональной диаметру рабочего цилиндра и давлению подведенного сжатого воздуха. Можно сказать, что пневмоцилиндры — основной элемент всех устройств пневмоавтоматики.

В основе работы пневматического цилиндра лежит воздействие силы сжатого воздуха на поршень пневмоцилиндра. Здесь следует отметить, что это воздействие может иметь как одностороннее, так и двустороннее направления. В зависимости от этого цилиндры делят на пневматические цилиндры одностороннего действия и двустороннего действия.

При одностороннем действии воздух оказывает давление на поршень только в одной из рабочих полостей цилиндра, поэтому движение под воздействием силы сжатого воздуха происходит только в одном направлении. Движение в обратном направлении происходит под воздействием пружины, установленной на шток цилиндра внутри его второй рабочей полости.

Пневмоцилиндры двустороннего действия, осуществляют перемещение штока в обе стороны за счёт воздействия силы сжатого воздуха, подаваемого в одну из рабочих полостей цилиндра. Распределение воздуха между рабочими полостями пневматического цилиндра осуществляется пневмораспределителями.

Пневмоцилиндры — конструктивные особенности:

Основными деталями пневмоцилиндров являются гильза, шток, поршень штока и флянцы.

Каждый из этих элементов имеет свои конструктивные особенности, влияющие на работу пневмоцилиндра.

При изготовлении пневмоцилиндров применяется гладкая труба (пневмоцилиндры серии AF) и профилированная труба из алюминиевых сплавов (серия CF, DF). Отличие заключается в наличии специальных пазов в профилированной трубе, предназначенных для установки герконовых датчиков.

Для воздействия на герконовые датчики на поршни пневматических цилиндров устанавливается магнитное кольцо.

Особенностью конструкции фланцев пневматических цилиндров является регулируемый демпфер.

Для предотвращения возможных ударов поршня и поверхности фланца в конце рабочего хода пневмоцилиндры оборудованы тормозным механизмом — демпфером. Регулировка скорости торможения осуществляется дросселем, смонтированным во фланцы цилиндра.

Обычно при выборе пневматического цилиндра пользуются расчетным методом. Существуют для этой цели и специализированные компьютерные программы, а также графические методы и таблицы.

В основе расчетного метода стоит определение усилия, которое развивается на штоке. Данное усилие напрямую зависит от диаметра поршня, рабочего давления или сил трения. Когда определяют теоретическое усилие, то рассматривают осевое усилие на неподвижном штоке, а силы трения не учитывают. Различают усилие на штоке для цилиндра двустороннего действия при прямом ходе штока (выдвигании) и обратном (втягивание) и для цилиндра одностороннего действия (с пружинным возвратом). Можно решить и обратную задачу, и при заданной нагрузке на штоке определить диаметр цилиндра.

## 1.2 Описание модели пневмоцилиндра в среде SimInTech

Общий вид модели пневмоцилиндра представлен на Рисунке 1.1:

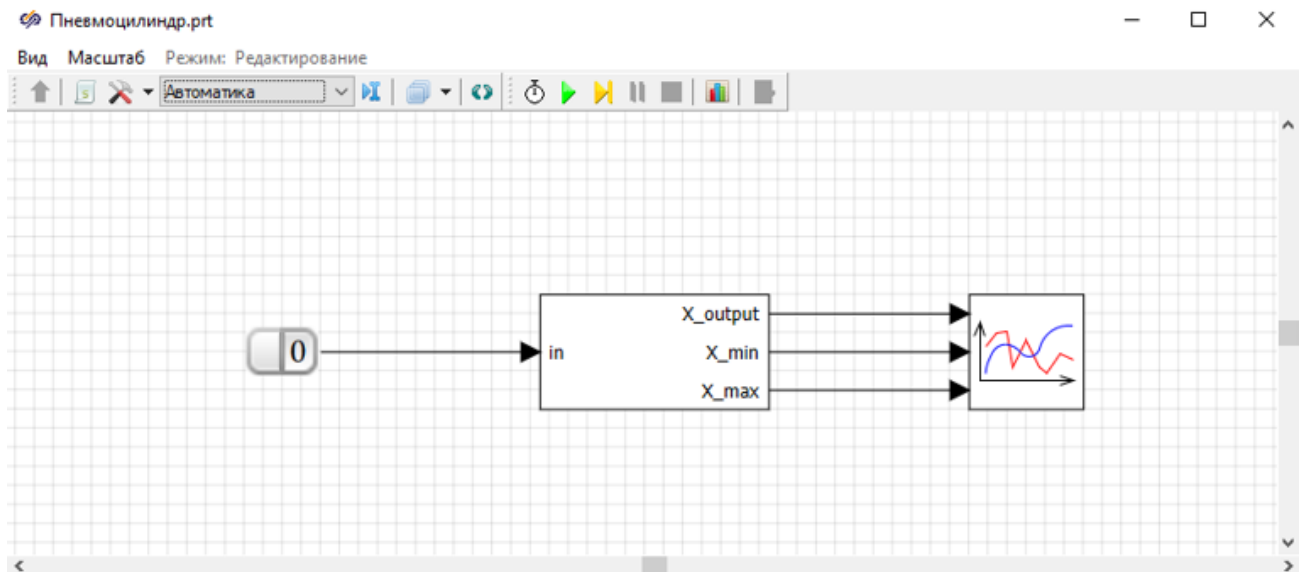


Рисунок 01 — Общий вид модели пневмоцилиндра

Заходим в субмодель и описываем следующую модель (Рисунок 1.2).

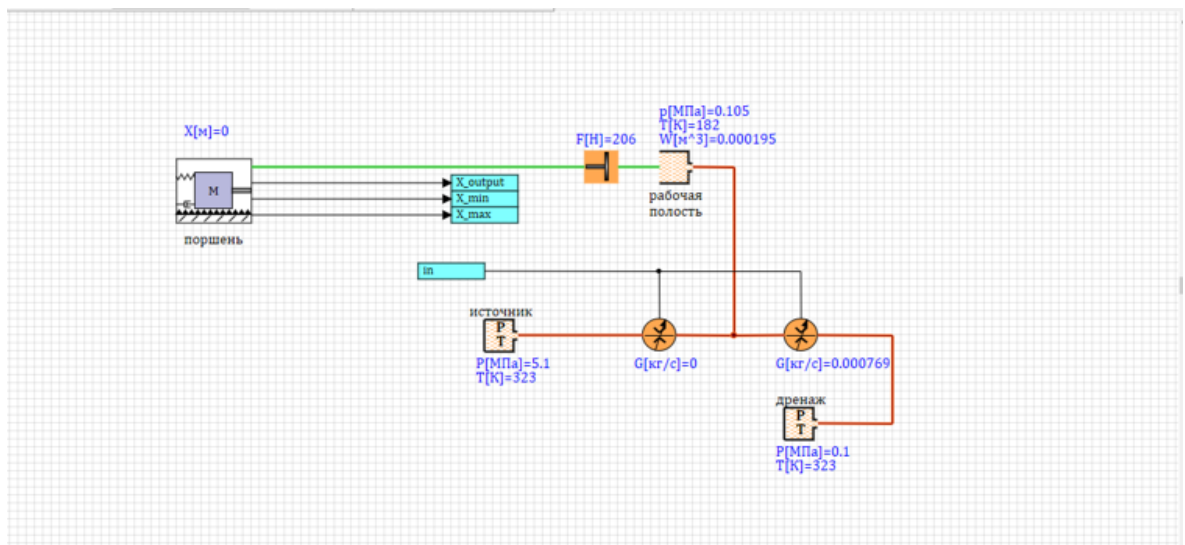


Рисунок 102 — Описание модели пневмоцилиндра

При этом, на модели используются следующие элементы: ГПС — Пневмомеханический преобразователь поступательного типа, ГПС — Пневматическая полость переменного объема (изотерма или адиабата), ГПС — Давление и температура газа, ГПС — Пневматический турбулентный дроссель с регулированием по произвольному параметру, ГПС — Механический элемент поступательного движения.

Параметры элементов представлены на Рисунках 1.3-1.8.

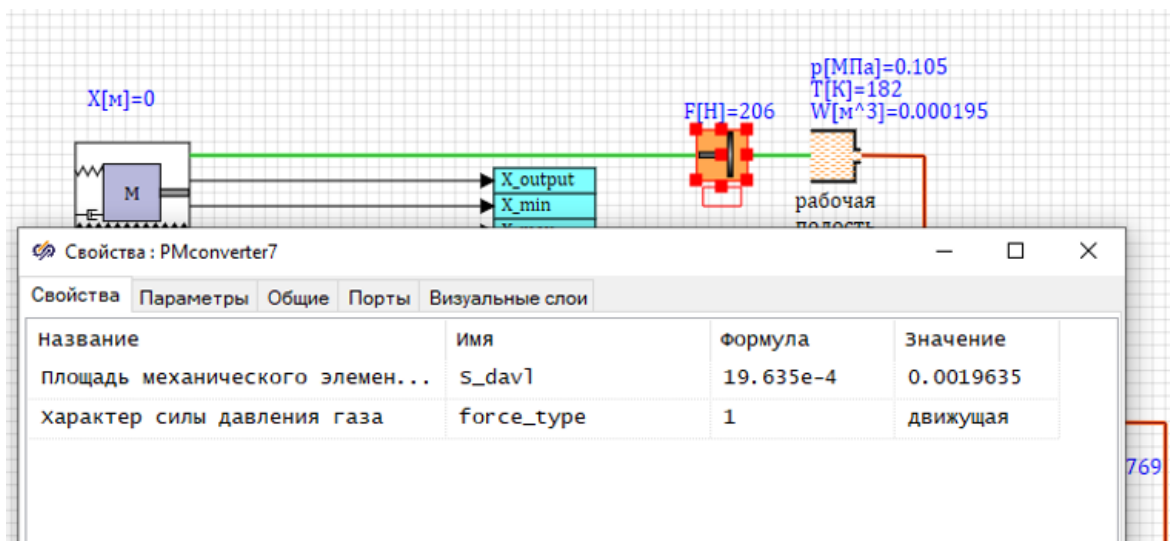


Рисунок 1.3 — Параметры пневматического преобразователя поступательного типа

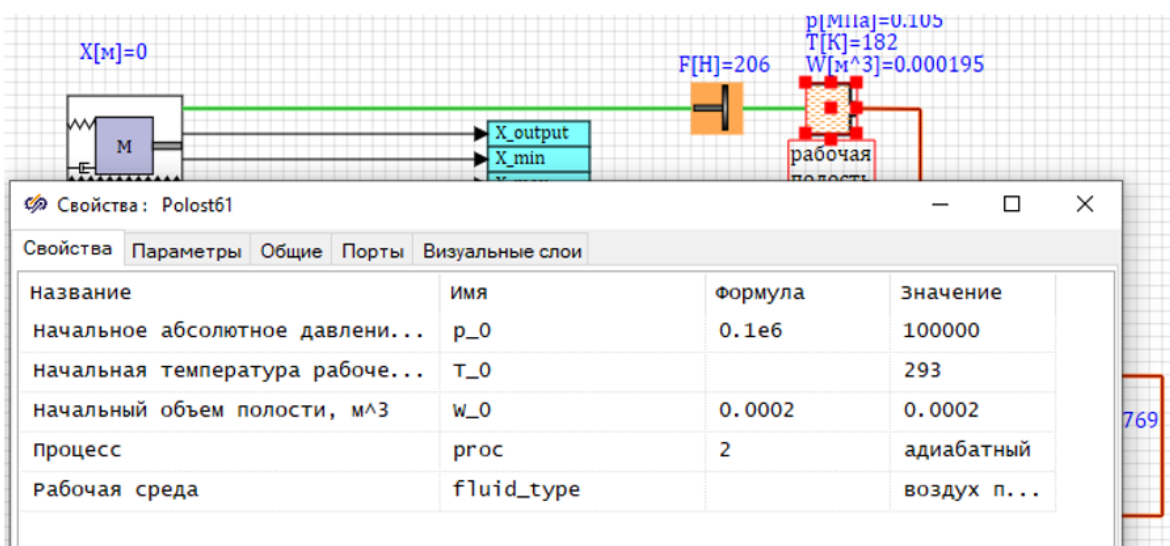


Рисунок 1.4 — Параметры пневматической полости переменного объем

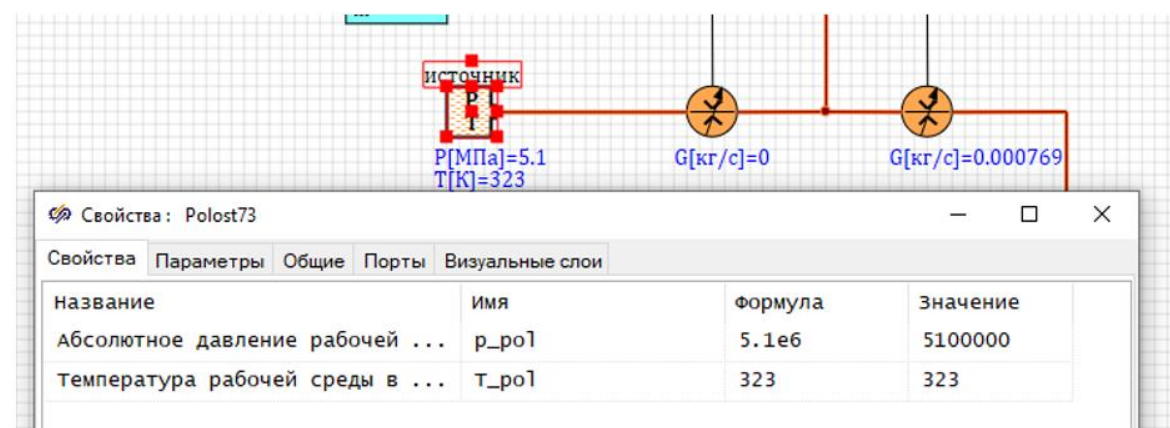
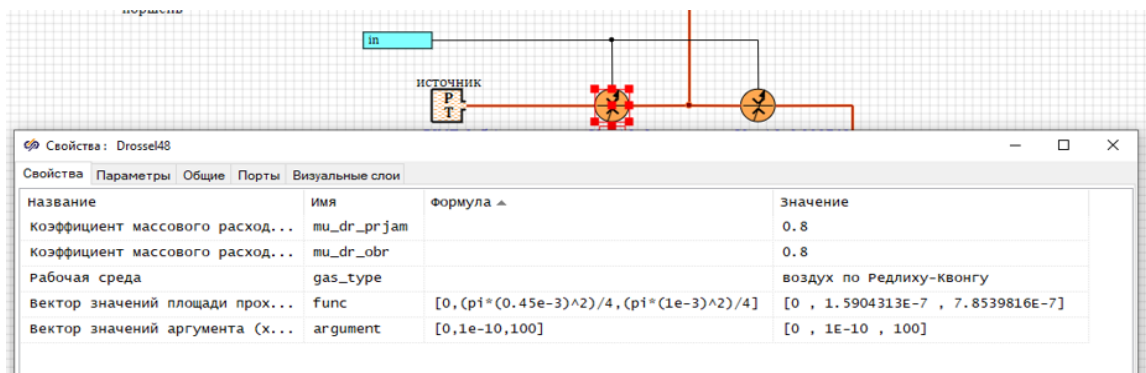
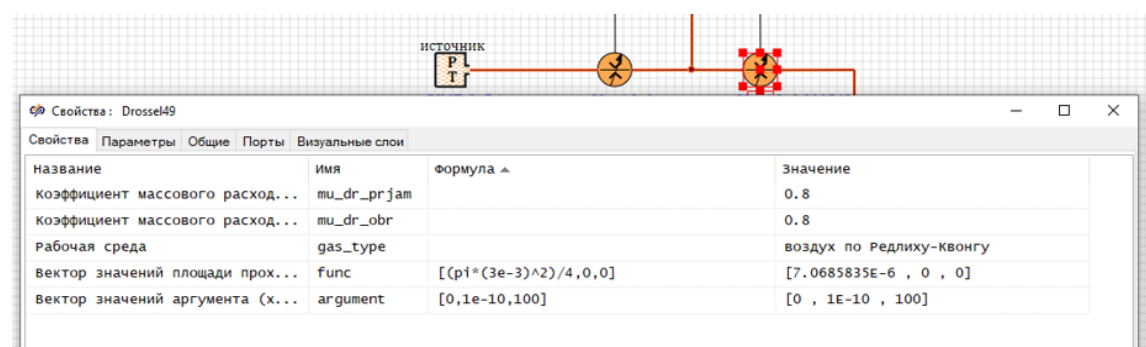


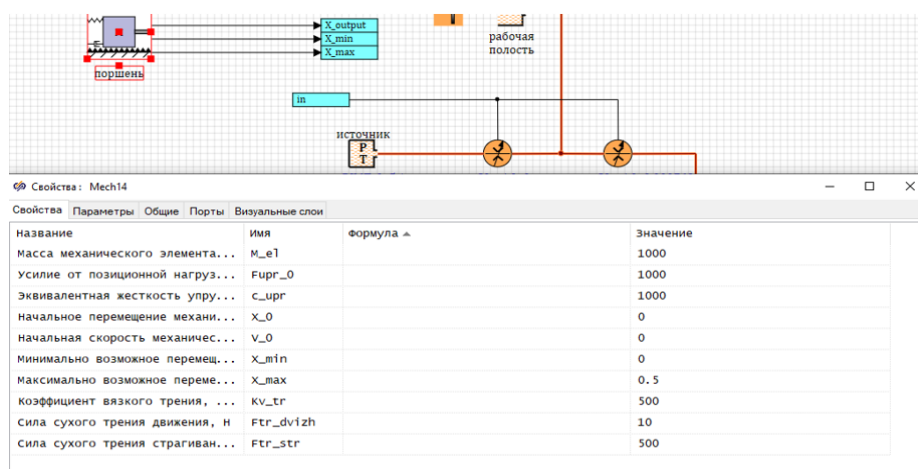
Рисунок 1.5 — Параметры давления и температуры газа в верхнем положении



**Рисунок 1.6 — Параметры пневматического турбулентного дросселя с регулированием по произвольному параметру**



**Рисунок 1.7 — Параметры пневматического турбулентного дросселя с регулированием по произвольному параметру**



**Рисунок 1.8 — Параметры механического элемента поступательного движения**

Для пневмоцилиндра, опишем дополнительно три выходных параметра: датчик положения поршня, концевой датчик положения поршня в крайнем верхнем положении и концевой датчик положения поршня в крайнем нижнем положении. «ГПС — Механический элемент поступательного движения», в базовой поставке, не имеет подобных выходных датчиков поэтому придется их сделать самостоятельно. SimInTech позволяет, при определенных условиях, модифицировать интересные в процессе моделирования базовые элементы.

Для изменения элемента необходимо поставить его в проект и вызвав правой кнопкой мыши выпадающее меню элемента выбрать пункт «Действия» далее «Войти в субмодель».

Войдя в субмодель необходимо добавить выходные сигналы (Рисунок 1.9).

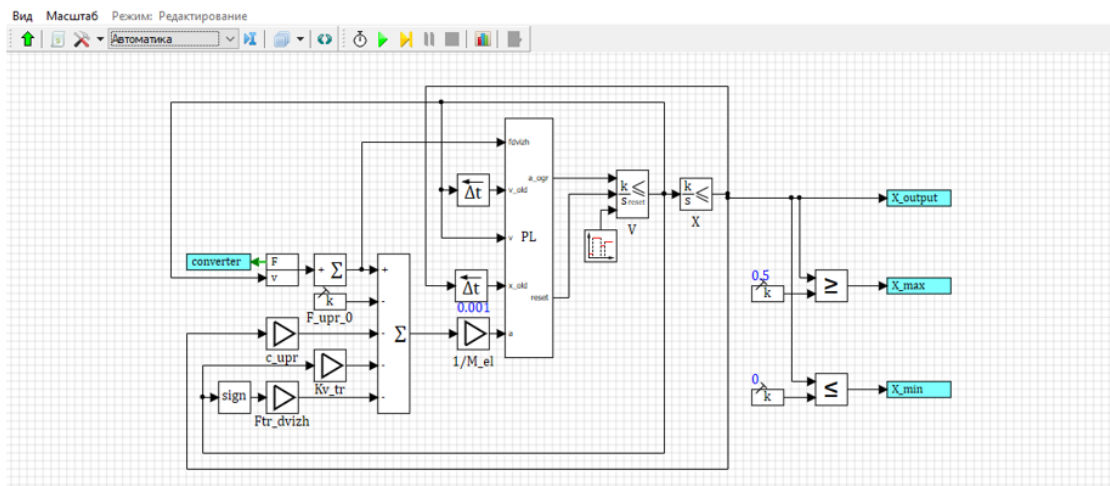


Рисунок 1.9 — Внутренняя модель механического элемента поступательного движения

При этом формулы констант должны соответствовать параметрам субмодели  $X_{\min}$  и  $X_{\max}$  (Рисунок 1.10, Рисунок 1.11).

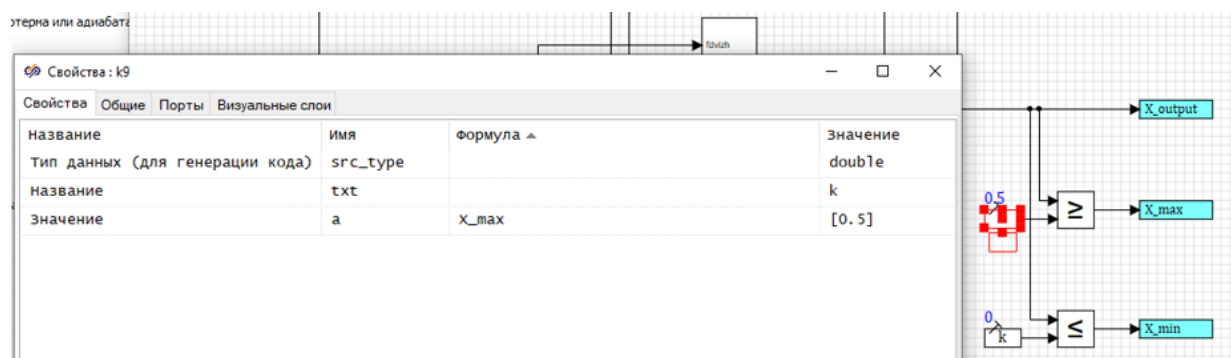


Рисунок 1.10 — Параметры константы

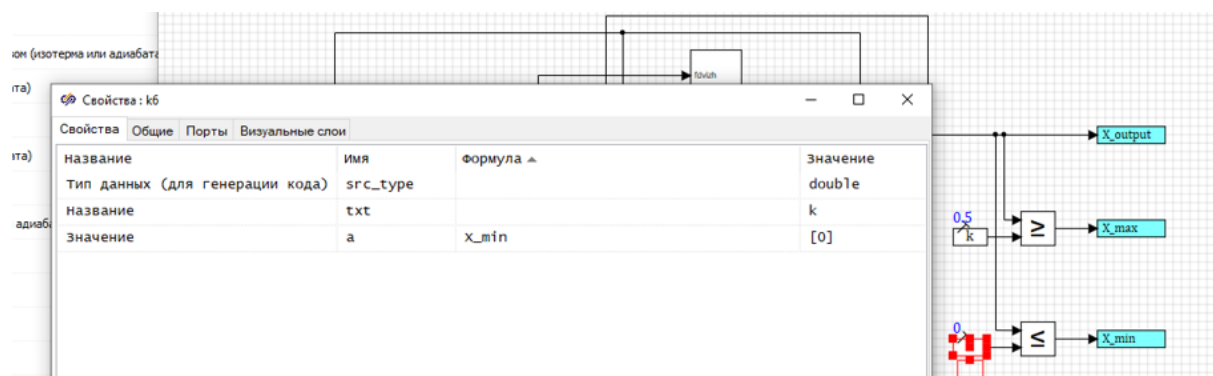


Рисунок 1.11 — Параметры константы



## 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1 Описание задания

Моделируемый процесс задается пятеркой  $(Y, T, D, E, P)$ . Где  $Y$  — множество пневмоцилиндров;  $T$  — множество отрезков времени за которые набор пневмоцилиндров на заданном шаге должны изменить свое положение;  $D$  — множество отрезков времени в течении которых на заданном шаге процесса должно удерживаться положение пневмоцилиндров после их стабилизации;  $E$  — множество пар  $(p_i p_j)$  переходов с шага процесса  $p_i$ , в котором за время  $t_i$  не удалось сменить положение пневмоцилиндров, на шаг  $p_j$ ;  $P$  — последовательность наборов пневмоцилиндров изменяемых на заданном шаге процесса.

Последовательность наборов цилиндров представляет из себя строку, в которой задана последовательность включения и отключения пневмоцилиндров, при этом выключение (цилиндр находится в крайнем нижнем положении) обозначается  $\overline{y1}$ , а включение (цилиндр находится в крайнем верхнем положении) обозначено  $y1$ . В одну единицу шага процесса в скобках указывается в какое состояние должны перейти пневмоцилиндры. Например,  $P = (y1y2)$  говорит о том, что по завершении этого шага пневмоцилиндр  $y1$  и  $y2$  должны перейти в крайнее нижнее положение, при этом предыдущее положение этих цилиндров должно быть крайним верхним ( $y1y2$ ).

Отдельно задается время, за которое пневмоцилиндры должны переместиться за одну единицу шага процесса ( $T = \{t1, t2, t3, \dots, tn\}$ ) и время нахождения на данном шаге процесса ( $D = \{d1, d2, d3, \dots, dn\}$ ). Например, запись  $T = \{t1 = t2 = t5 = 50, t3 = t4 = 40\}$ ,  $D = \{d4 = d5 = 120, d1 = d2 = d3 = 180\}$  говорит о том, что на шагах 1, 2, 5 пневмоцилиндр должен переместиться за время равное или меньше 50 дискретам, а на шагах 3, 4 за 40; время нахождения на шагах 4 и 5, после выставления требуемого положения пневмоцилиндров, равно 120 дискретам, а на 1-3 шагах равное 180 дискретам.

В случаях, когда на каком-либо из шагов, цилиндрам не удалось за

заданное время сменить свое положение, процесс должен завершиться с ошибкой и ожидать сброса или, если есть дополнительное описание, произвести переход на необходимый шаг. Переход задается парой, где в скобках описывается шаг, с которого осуществляется переход, в случае описанном выше и далее после запятой шаг, на который осуществляется переход. Например,  $E = (p1, p5)$  — говорит о том, что, если на шаге 1 процесса, не удалось за заданное время привести цилиндры в необходимое положение надо перейти на шаг исполнения 5.

## 2.2 Исходные данные

$$Y = y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8$$

$$P = (\overline{y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8}), (y_8 y_1 y_2 y_4 y_5), \\ (\overline{y_1 y_2 y_3 y_4 y_5 y_6 y_7}), (y_1 y_2 y_4 y_5 \overline{y_7 y_8}), \\ (\overline{y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8}), (y_3 y_4 y_6 \overline{y_7 y_8}), \\ (y_2 \overline{y_3 y_4 y_6 y_8}), (y_1 \overline{y_2 y_3 y_4 y_5 y_6 y_7 y_8}), \\ (\overline{y_1 y_3 y_4 y_6 y_8}), (y_1 y_2 y_3 y_4 \overline{y_5 y_6 y_7 y_8}), \\ (\overline{y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8}), (y_1 y_3 y_4 y_6 \overline{y_7 y_8}), \\ (\overline{y_1 y_3 y_4 y_6 y_7 y_8}), (y_1 y_2 y_3 y_4 \overline{y_5 y_6 y_7 y_8}), \\ (\overline{y_1 y_3 y_4 y_5 y_8}), (\overline{y_2 y_3 y_4 y_5 y_6 y_7 y_8})$$

$$T = \begin{cases} t_0 = t_1 = t_{12} = t_{14} = 20 \\ t_{10} = 120 \\ t_6 = 90 \\ t_4 = t_5 = t_9 = t_{11} = 60 \\ t_2 = t_3 = t_7 = t_{13} = 45 \\ t_8 = 30 \end{cases} \quad D = \begin{cases} d_0 = d_1 = 33 \\ d_3 = d_7 = d_9 = 50 \\ d_5 = 70 \\ d_2 = d_{10} = d_{13} = d_{14} = 120 \\ d_6 = 60 \\ d_4 = d_8 = d_{11} = d_{12} = 76 \end{cases}$$

$$E = (p_1, p_{11}), (p_2, p_1), (p_7, p_9).$$

## 2.3 Построение диаграммы Мура автоматной модели

Мы построили диаграмму Мура (Рисунок 2.1) для исходных данных с помощью ориентированного графа, вершины которого взаимно однозначно соответствуют состояниям автомата, а дуги – входным условиям. Данный автомат включает в себя 16 состояний, а также состояние ошибки, в случае невыполненного перехода в следующее состояние в заданный промежуток времени.

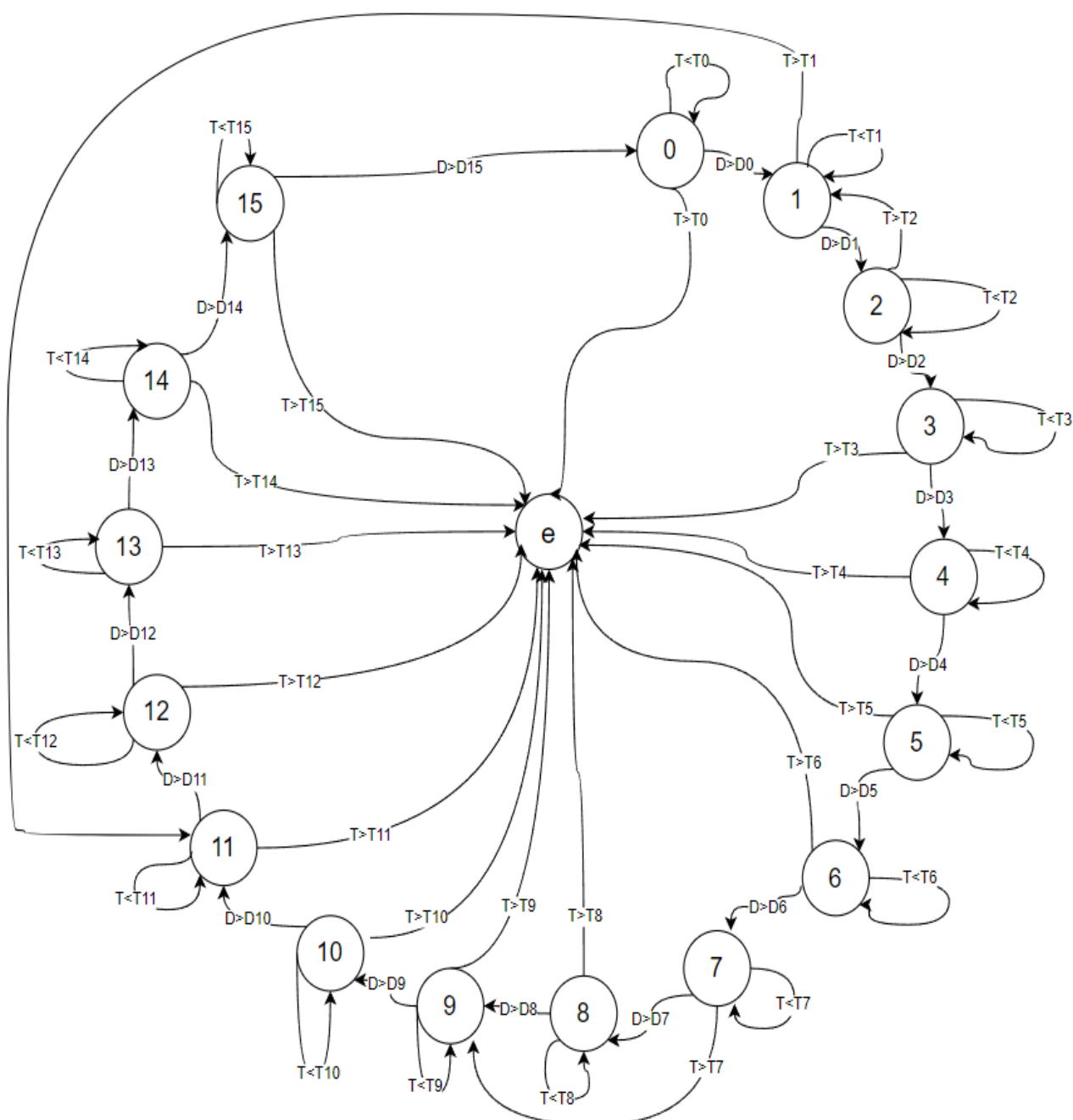


Рисунок 2.1 — Диаграмма Мура

## 2.4 Кодирование автоматной модели на языке Си

На Листинге 2.1 представлены шаги (PneumoState\_\*), на которых изменяется заданная последовательность пневмоцилиндров, структура пневмоцилиндра (struct PneumoCylinder), вход (output\_signal) и выходы (input\_signal) пневмоцилиндра, и структура (PneumoEngine), которая включает в себя шаг, на котором находится процесс симуляции, временные промежутки (timeouts, delays) и сами цилиндры.

*Листинг 2.1 — Функция инициализации автомата*

```
enum PneumoState {
    PneumoState_0, PneumoState_1, PneumoState_2,
    PneumoState_3, PneumoState_4, PneumoState_5,
    PneumoState_6, PneumoState_7, PneumoState_8,
    PneumoState_9, PneumoState_10, PneumoState_11,
    PneumoState_12, PneumoState_13, PneumoState_14,
    PneumoState_15,
    PneumoState_FatalException
};

#define PNEUMO_CYLINDER_SIGNAL_UP 0
#define PNEUMO_CYLINDER_SIGNAL_DOWN 1

struct PneumoCylinder {
    int input_signal[2];
    int output_signal;
};

#define PNEUMO_CYLINDER_Y1 0
#define PNEUMO_CYLINDER_Y2 1
#define PNEUMO_CYLINDER_Y3 2
#define PNEUMO_CYLINDER_Y4 3
#define PNEUMO_CYLINDER_Y5 4
#define PNEUMO_CYLINDER_Y6 5
#define PNEUMO_CYLINDER_Y7 6
#define PNEUMO_CYLINDER_Y8 7

struct PneumoEngine {
    enum PneumoState state;
    int timeout;
    int delay;
    int timeouts[PneumoState_FatalException];
    int delays[PneumoState_FatalException];
    struct PneumoCylinder cylinders[8];
};
```

На Листинге 2.2 представлена функция инициализации автомата принимает на вход структурную переменную engine, которая содержит информацию о входных и выходных портах пневмоцилиндра. Внутри функции происходит присвоение значения нуля всем входным и выходным портам пневмоцилиндра с помощью оператора присваивания.

Далее в функции задается значение временных промежутков на каждом

шаге симуляции. Эти временные промежутки могут быть заданы как константы или переменные, содержащие информацию о времени, необходимом для выполнения определенных операций в автомате.

В качестве аргумента этой функции на Листинге 2.2 подается структурная переменная(engine), рассмотренная на Листинге 2.1.

*Листинг 2.2 — Функция инициализации автомата*

```
__attribute__((unused)) void pneumoEngineInit(struct PneumoEngine* engine) {
    if (engine != 0) {
        for (int i = 0; i < 8; i++)
        {
            engine->cylinders[i].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = 0;
            engine->cylinders[i].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = 0;
            engine->cylinders[i].output_signal = 0;
        }

        engine->state = PneumoState_0;
        engine->delay = 0;
        engine->timeout = 0;

        delayTimeoutInit(engine, 20, 33, PneumoState_0);
        delayTimeoutInit(engine, 20, 33, PneumoState_1);
        delayTimeoutInit(engine, 45, 120, PneumoState_2);
        ...
    }
}
```

Рассмотрим Листинг 2.3, на котором представлена функция перехода автомата из одного состояния в другое, здесь задается булевая переменная (ret), имеющая только два значения: истина и ложь. Это переменная, которая меняет свое значение на ложь в том случае, если происходит переход процесса симуляции на шаг исключения (PneumoState\_FatalException).

Переход автомата из одного состояния в другое осуществляется с помощью оператора принятия решения SWITCH, в качестве выражения подается переменная шага(state), выполняется case, константа которого равна выражению в switch. В качестве примера рассмотрим представленный ниже Листинг 2.3. Если в выражение switch будет подано состояние PneumoState\_0, то выполнится case PneumoState\_0, в котором мы, по условию, подаем нули на входные порты пневматических цилиндров, для того чтобы пневматические цилиндры перешли в крайнее нижнее положение. Затем если условие перехода будет соблюдено, то через заданный промежуток времени автомат перейдет на следующий шаг

симуляции, в нашем случае PneumoState\_1, иначе автомат должен перейти в состояние ошибки PneumoState\_FatalException.

*Листинг 2.3 — Функция перехода автомата из одного состояния в другое*

```
__attribute__((unused)) bool pneumoEngineTick(struct PneumoEngine* engine) {
    bool ret = true;
    if (0 == engine)
        return false;
    switch (engine->state) {
        case PneumoState_0: {
            int cylinderSignals[] = {0, 0, 0, 0, 0, 0, 0, 0};
            pneumoStateBody(engine, cylinderSignals, PneumoState_1,
PneumoState_FatalException);
            break;
        }
        case PneumoState_1: {
            int cylinderSignals[] = {1, 1, 0, 1, 1, 0, 0, 1};
            pneumoStateBody(engine, cylinderSignals, PneumoState_2,
PneumoState_11);
            break;
        }
        case PneumoState_2: {
            int cylinderSignals[] = {0, 0, 1, 0, 0, 1, 1, 1};
            pneumoStateBody(engine, cylinderSignals, PneumoState_3,
PneumoState_1);
            break;
        }
        case PneumoState_3: {
            int cylinderSignals[] = {1, 1, 1, 1, 1, 1, 0, 0};
            pneumoStateBody(engine, cylinderSignals, PneumoState_4,
PneumoState_FatalException);
            break;
        }
        case PneumoState_4: {
            int cylinderSignals[] = {0, 0, 0, 0, 0, 0, 1, 1};
            pneumoStateBody(engine, cylinderSignals, PneumoState_5,
PneumoState_FatalException);
            break;
        }
        ...
    }
    case PneumoState_FatalException: {
        ret = false;
        break;
    }
}

engine->timeout++;
return ret;
}
```

## 2.5 Моделирование автомата в среде SimInTech

Построим модель SimInTech для генерации dll файла. В блоке PL 16 входов для сигнала верхнего и нижнего положения каждого пневмоцилиндра. Также в нем 9 выходов. 8 из них для выходного сигнала состояния каждого

пневмоцилиндра и последний выход для определения состояния ошибки (Рисунок 2.2).

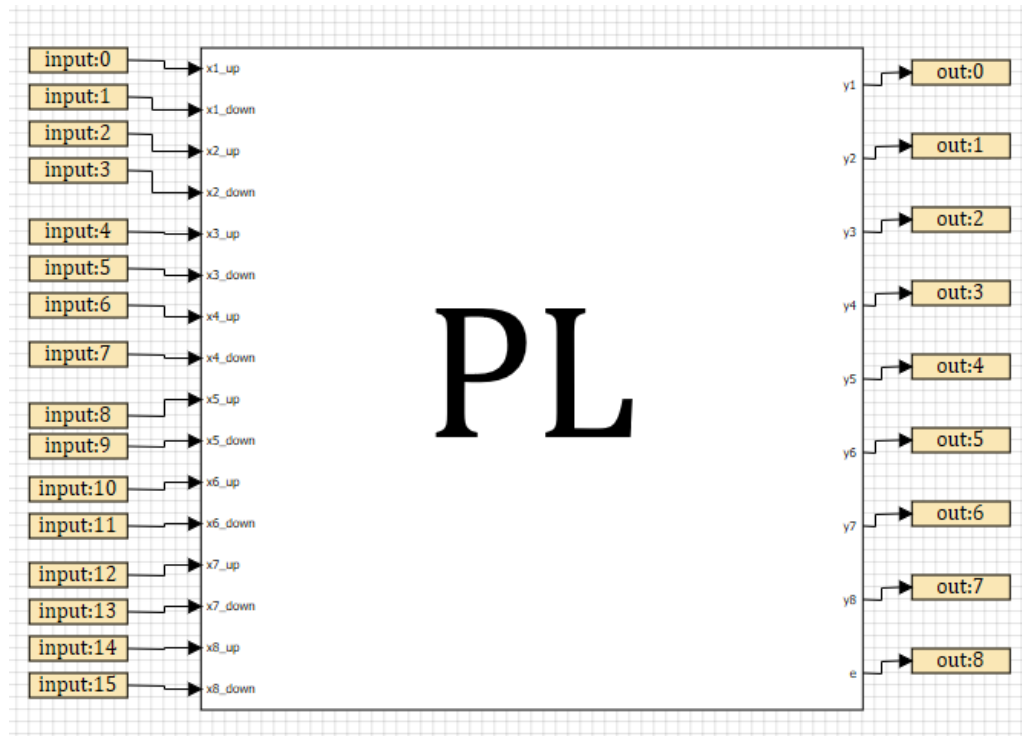


Рисунок 2.2 — Модель среды SimInTech для генерации dll

Мы назначили наши переменные выходов и входов пневмоцилиндра на входные и выходные контакты модели (Рисунок 2.3).

```
engine.cylinders[PNEUMO_CYLINDER_Y1].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:0%;
engine.cylinders[PNEUMO_CYLINDER_Y1].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:1%;
engine.cylinders[PNEUMO_CYLINDER_Y2].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:2%;
engine.cylinders[PNEUMO_CYLINDER_Y2].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:3%;
engine.cylinders[PNEUMO_CYLINDER_Y3].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:4%;
engine.cylinders[PNEUMO_CYLINDER_Y3].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:5%;
engine.cylinders[PNEUMO_CYLINDER_Y4].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:6%;
engine.cylinders[PNEUMO_CYLINDER_Y4].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:7%;
engine.cylinders[PNEUMO_CYLINDER_Y5].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:8%;
engine.cylinders[PNEUMO_CYLINDER_Y5].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:9%;
engine.cylinders[PNEUMO_CYLINDER_Y6].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:10%;
engine.cylinders[PNEUMO_CYLINDER_Y6].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:11%;
engine.cylinders[PNEUMO_CYLINDER_Y7].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:12%;
engine.cylinders[PNEUMO_CYLINDER_Y7].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:13%;
engine.cylinders[PNEUMO_CYLINDER_Y8].input_signal[PNEUMO_CYLINDER_SIGNAL_UP] = %input:14%;
engine.cylinders[PNEUMO_CYLINDER_Y8].input_signal[PNEUMO_CYLINDER_SIGNAL_DOWN] = %input:15%;

%out:8% = pneumo_engine_tick(&engine);
%out:0% = engine.cylinders[PNEUMO_CYLINDER_Y1].output_signal;
%out:1% = engine.cylinders[PNEUMO_CYLINDER_Y2].output_signal;
%out:2% = engine.cylinders[PNEUMO_CYLINDER_Y3].output_signal;
%out:3% = engine.cylinders[PNEUMO_CYLINDER_Y4].output_signal;
%out:4% = engine.cylinders[PNEUMO_CYLINDER_Y5].output_signal;
%out:5% = engine.cylinders[PNEUMO_CYLINDER_Y6].output_signal;
%out:6% = engine.cylinders[PNEUMO_CYLINDER_Y7].output_signal;
%out:7% = engine.cylinders[PNEUMO_CYLINDER_Y8].output_signal;
```

Рисунок 2.3 — Основная секция кода

Основная модель автомата выглядит следующим образом (Рисунок 2.4):

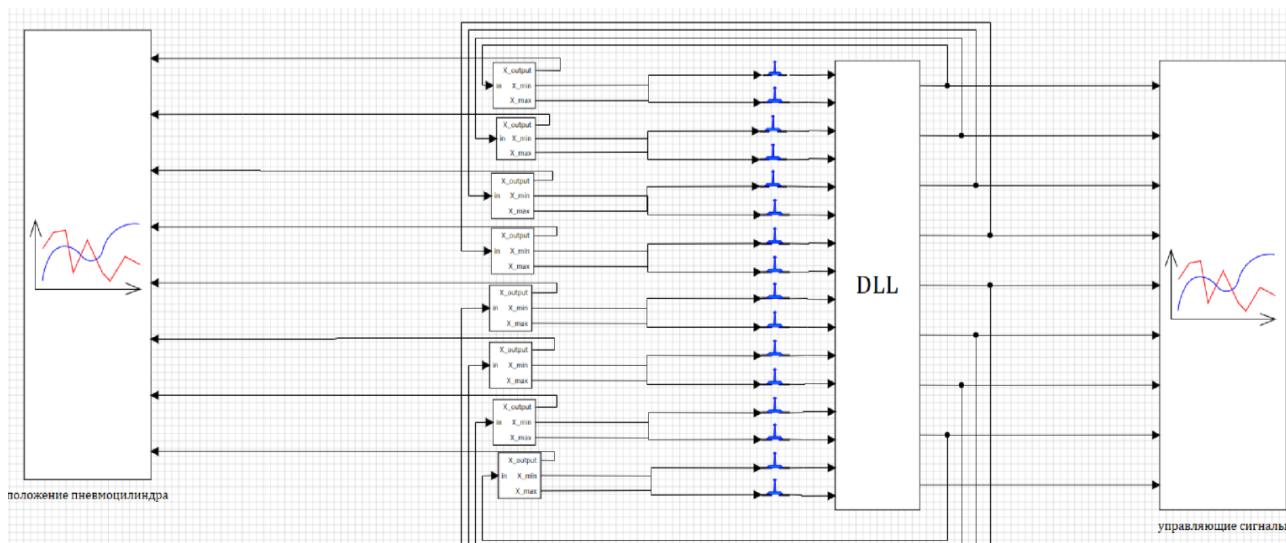


Рисунок 2.4 — Основная модель автомата

Результат симуляции процесса управления автоматной модели представлен в виде графиков движения и положения всех пневмоцилиндров относительно времени на Рисунке 2.5.

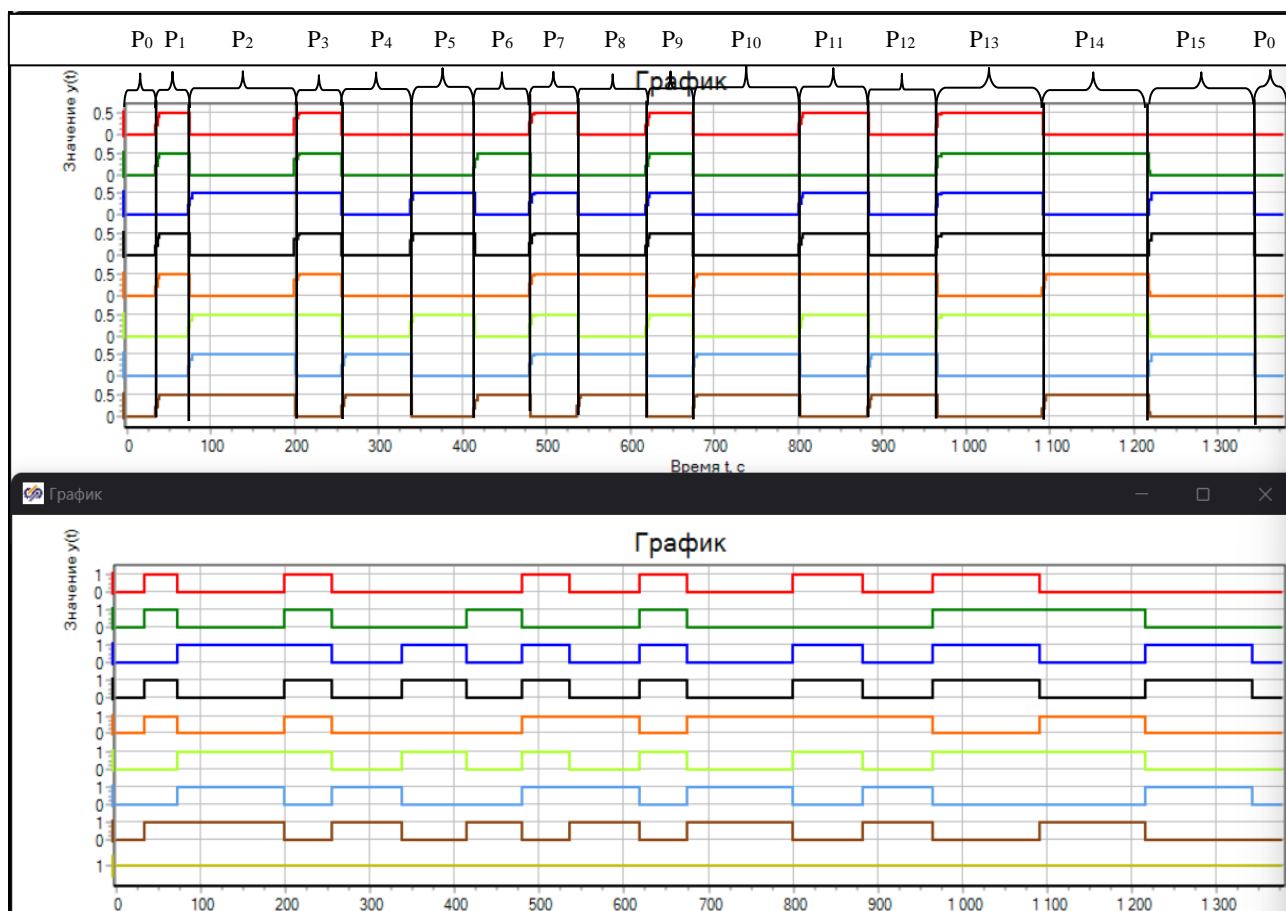
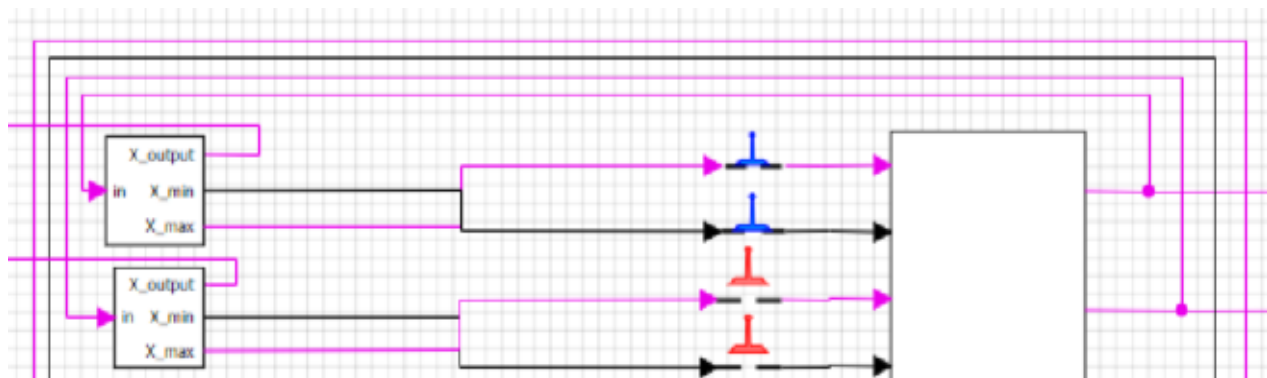


Рисунок 2.5 — Результат симуляции

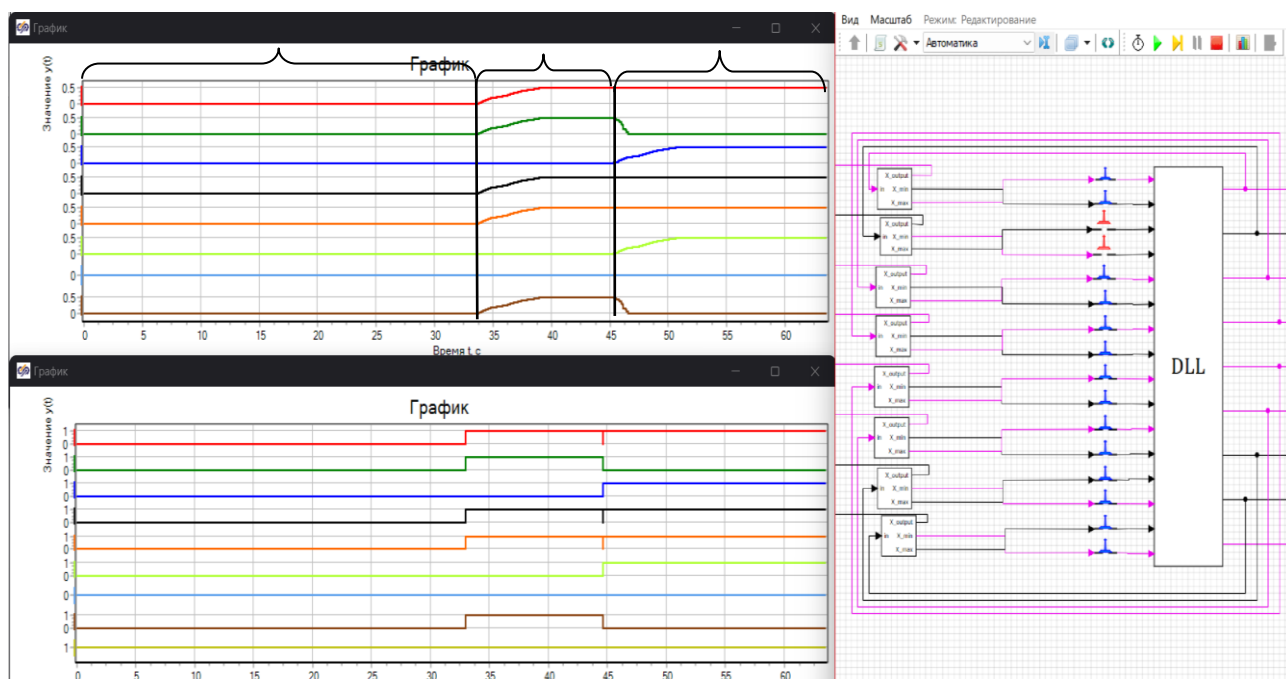


Смоделируем отказ второго пневмоцилиндра и переход  $E = (p_1, p_{11})$ , для этого мы используем управляемые ключи на выходах второго пневмоцилиндра (Рисунок 2.6).



**Рисунок 2.6 — Ключи на выходах второго пневмоцилиндра**

Перейдя на шаг  $p_1$  по условию все цилиндры должны перейти в установленное для них положение, если условие соблюдено автомат должен был перейти на шаг  $p_2$ , но так как мы закрыли доступ сигнала на выходах второго пневмоцилиндра, он перешел на шаг  $p_{11}$ . (Рисунок 2.7).

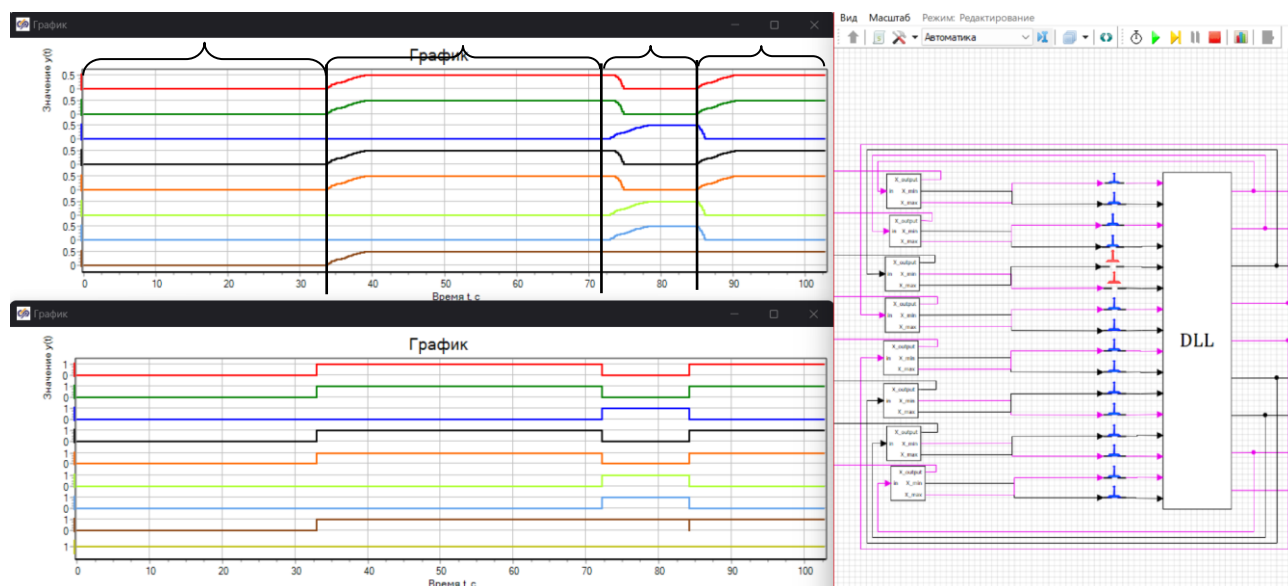


**Рисунок 2.7 — Результат отказа второго пневмоцилиндра и перехода процесса симуляции с первого шага на одиннадцатый**

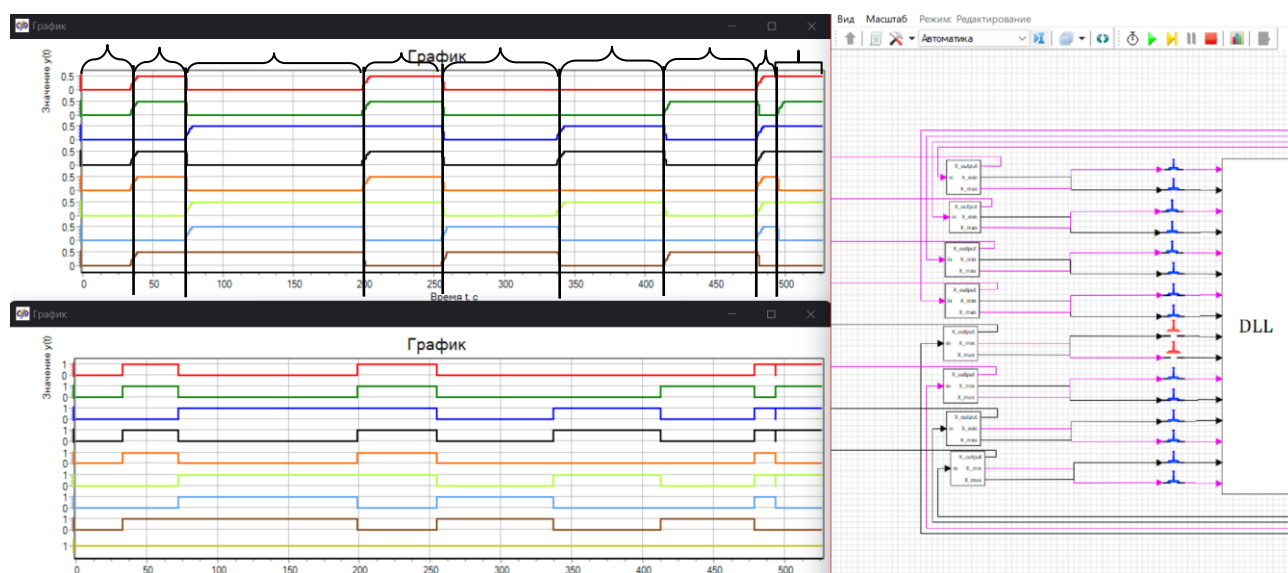
В результате симуляции получается, что автоматная модель не успевает изменить положения поршней за этот промежуток времени и переходит, согласно условию, в одиннадцатое состояние.

По аналогии смоделируем отказ пневмоцилиндров для переходов  $E = (p_2,$

$p_1$ ) и  $E = (p_7, p_9)$ . (Рисунки 2.8 и 2.9).



**Рисунок 2.8 — Результат отказа третьего пневмоцилиндра и перехода процесса симуляции со второго шага на первый**



**Рисунок 2.9 — Результат отказа пятого пневмоцилиндра и перехода процесса симуляции с седьмого шага на девятый**

## ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы я приобрел навыки работы в среде динамического моделирования SimInTech, создал диаграмму Мура и усовершенствовал свои навыки программирования на языке Си. Мы изучили методы построения автоматных моделей на основе заданных управляющих последовательностей и успешно применили их на практике. Мы погрузились в прикладные аспекты парадигмы автоматного программирования и использовали автомат, созданный по этой парадигме. Кроме того, мы улучшили свои навыки проектирования архитектуры конечных автоматов.

Мы рассмотрели возможные процессы работы пневмоцилиндров, создали модель и провели испытания автомата в различных условиях — как в идеальных, так и приближенных к реальным. Мы тщательно изучили каждый переход между состояниями, чтобы выявить возможные недостатки автомата. Также мы научились предсказывать ошибки в работе автомата и разработали методы их предотвращения.

По итогам курсовой работы мы освоили проектирование автоматов для контроля различных технологических процессов с возможностью прогнозирования потенциальных ошибок.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Принцип работы пневмоцилиндра, его параметры и характеристики – URL: <http://pnevmodetal.ru/index.php/produktsiya/pnevmocelidr> (Дата обращения: 24.05.2024)
2. Пневматические цилиндры: для чего они нужны — URL: <https://izhpnevmo.ru/novosti/pnevmaticheskie-czilindryi-dlya-chego-oni-nuzhnyi.html> (Дата обращения: 24.05.2024)
3. Хлебников А.А. Курсовая работа по дисциплине АПСУ, 2020. – С. 6-9, 112. (Дата обращения: 24.05.2024)
4. Автоматное программирование систем управления: практическая работа №9 / Под ред. А. А. Хлебникова. — М.: Построение автоматной модели управления пневмоцилиндрами по заданной управляющей последовательности, 2021. — 17 с. (Дата обращения: 24.05.2024)
5. Автоматное программирование систем управления: практическая работа №7 / Под ред. А. А. Хлебникова. — М.: ПОСТРОЕНИЕ ПРОСТОЙ МОДЕЛИ «ЗАДЕРЖКА» С УПРАВЛЯЮЩИМ АВТОМАТОМ НА ЯЗЫКЕ СИ, 2020. — 10 с. (Дата обращения: 24.05.2024)
6. Справочная система SimInTech — URL: [https://help.simintech.ru/#o\\_simintech/browsers.html](https://help.simintech.ru/#o_simintech/browsers.html) (Дата обращения: 24.05.2024)
7. Руководство по языку программирования Си — URL: <https://metanit.com/c/tutorial/> (Дата обращения: 24.05.2024)