

ГУАП

КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

старший преподаватель  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

А.Н.Долидзе  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ ОВЕН ПЛК110 НА ЯЗЫКЕ ST

по курсу: ЦИФРОВЫЕ СИСТЕМЫ АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4143

\_\_\_\_\_  
подпись, дата

Е.Д.Тегай  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

Схема технологического процесса

Искомая схема технологического прогресса согласно варианту №19 продемонстрирована на рисунке 1.

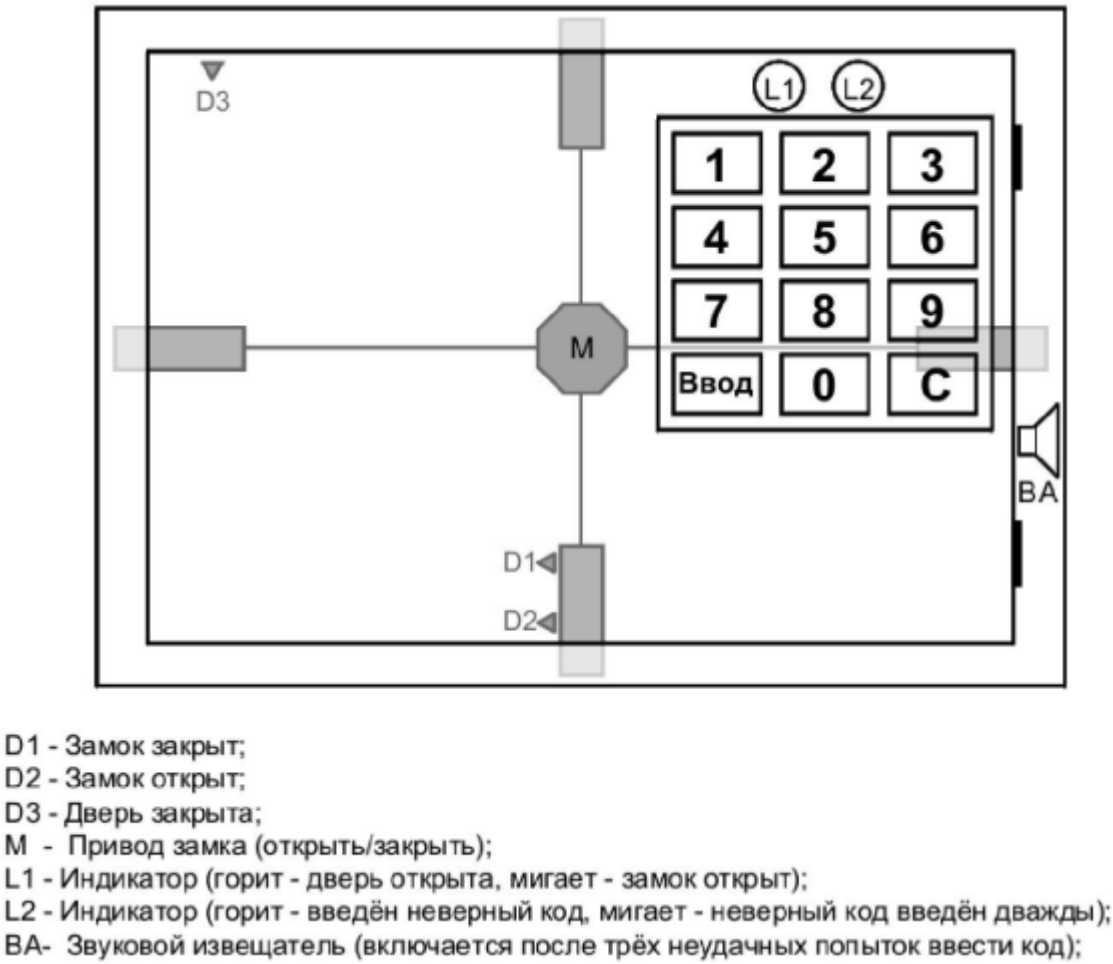


Рисунок 1 – Схема ТП

Помимо этого, на рисунке 2 приведены уточнения, которые помогают при составлении схемы и при выборе контроллера.

Тегай Екатерина Дмитриевна	19	Релейный	220ACV	1. Привод замка: 24DCV; 2. Индикаторы: 24DCV; 3. Оповещатель: 24DCV; 4. Длина кода: 3; 5. Тип ввода: кодированный (используется двоичный код чисел); 6. Числовой диапазон (сколько кнопок на клавиатуре): 0-5; 7. Аварийные ситуации типа «невозможная комбинация сигналов»: 2; 8. Аварийные ситуации типа «отсутствие ожидаемой реакции»: 2
-------------------------------	----	----------	--------	---

Рисунок 2 - Уточнения

Описание технологического процесса и оборудования

Данный процесс поддерживается работой нескольких устройств, а именно:

- Датчики замка D1 и D2 – служат для мониторинга состояния замка. Первый фиксирует факт закрытия, а второй – открытия.
- Датчик двери D3 – необходим для контроля состояния двери сейфа. Он фиксирует момент, когда дверь открыта или закрыта.
- Привод замка М – нужен для управления механизмом открытия и закрытия замка сейфа. Он преобразует электрические сигналы, поступающие от системы управления, в механическое действие, которое фактически и открывает или закрывает замок.
- Индикаторы L1 и L2 – являются визуальными сигнализирующими устройствами в отношении различных ситуаций в целях подтверждения или предупреждения. Например, первый индикатор может оповещать о состоянии двери или замка, а второй – о неверности кода и об оставшихся попытках (если мигает, то при следующем неправильно введённом коде прозвучит аварийный сигнал)
- Звуковой извещатель ВА – служит для аудиосигнализации в случае неудачного многоповторного ввода кода.

Как видно из рисунка 1, технологический процесс представляет собой сейф. Если пользователь хочет ввести двоичный код как вариацию своей кодовой комбинации из 3 цифр, то он должен нажимать кнопку «Ввод» после каждого ввода цифры. Если же он хочет ввести код в обычном десятичном виде, то после его ввода так же нажимается соответствующая кнопка. Помимо кнопок с цифрами от 1 до 9 и кнопки «Ввод» есть также кнопка сброса «С». Она служит для сброса введённой комбинации.

Изначально в этом процессе замок сейфа открыт. Также, уникальной кодовой комбинацией является три нуля, что открывает сейф в любом случае. (так называемый мастер код). Исходя из этого состояния, пользователь должен придумать новый код, чтобы сейф успешно закрылся. Но до этого, при открытой дверце, ему нужно нажать кнопку сброса, чтобы сейф сбросил старый код. Визуальной индикацией данного результата будет мигание обоих

индикаторов. После ввода пользователем нового кода индикаторы снова зажгутся на 5 секунд - подтверждение ввода.

Если пользователю необходимо закрыть сейф, то после закрытия самой дверцы ему нужно будет нажать кнопку ввода.

Бывает так, что пользователь не с первого раза вводит код. При обычном корректном вводе кода с первого раза дверца сейфа, собственно, открывается. Но если ввести неверно код один раз, то об этом сигнализирует индикатор L2, а если второй – начинает мигать. Если же при последней, третьей, попытке пользователь всё так же неверно вводит комбинацию, то в таком случае включается звуковой оповещатель ВА.

### **Разработка программы**

В качестве языка программирования был выбран язык ST. Ниже продемонстрирован текст программы.

<b>Сигнал ТП</b>	<b>Канал ПЛК</b>	<b>Комментарий</b>
<b>Дискретные входы</b>		
BTN_0	IX1.0.0	Кнопка для ввода 0
BTN_1	IX1.0.1	Кнопка для ввода 1
BTN_2	IX1.0.2	Кнопка для ввода 2
BTN_3	IX1.0.3	Кнопка для ввода 3
BTN_4	IX1.0.4	Кнопка для ввода 4
BTN_5	IX1.0.5	Кнопка для ввода 5
BTN_6	IX1.0.6	Кнопка для ввода 6
BTN_7	IX1.0.7	Кнопка для ввода 7
BTN_8	IX1.1.0	Кнопка для ввода 8
BTN_9	IX1.1.1	Кнопка для ввода 9
BTN_ENTER	IX1.1.2	Кнопка «Ввод»
BTN_RESET	IX1.1.3	Кнопка «Сброс»
<b>Дискретные выходы</b>		
D1	QX3.0.0	Замок закрыт

D2	QX3.0.1	Замок открыт
D3	QX3.0.2	Дверь закрыта
L1	QX3.0.3	Индикатор L1
L2	QX3.0.4	Индикатор L2
BA	QX3.0.5	Звуковой извещатель

### Код программы

PROGRAM PLC\_PRG

VAR

MixingTimerStart: BOOL := FALSE;

MixingTimerElapsed: BOOL := FALSE;

MixingTimerCount: TIME := T#0S;

MixingTimerPreset: TIME := T#1S;

LastTime: TIME := T#0S;

MixingTimerStart\_ON: BOOL := FALSE;

MixingTimerCount\_ON: TIME := T#0S;

MixingTimerPreset\_ON: TIME := T#5S;

LastTime\_ON: TIME := T#0S;

MixingTimerStart\_L2: BOOL := FALSE;

MixingTimerCount\_L2: TIME := T#0S;

MixingTimerPreset\_L2: TIME := T#1S;

LastTime\_L2: TIME := T#0S;

D1 : BOOL;  
D2 : BOOL := TRUE;  
D3 : BOOL;  
BTN\_0 : BOOL;  
BTN\_1 : BOOL;  
BTN\_2 : BOOL;  
BTN\_3 : BOOL;  
BTN\_4 : BOOL;  
BTN\_5 : BOOL;  
BTN\_6 : BOOL;  
BTN\_7 : BOOL;  
BTN\_8 : BOOL;  
BTN\_9 : BOOL;  
BTN\_ENTER : BOOL;  
BTN\_RESET : BOOL;

M : BOOL;  
L1 : BOOL := FALSE;  
L2 : BOOL := FALSE;  
BA : BOOL;

Code: ARRAY [0..2] OF INT;  
MasterCode : ARRAY[0..2] OF INT;  
InputCode : ARRAY[1..3] OF INT;  
CheckCode : ARRAY[1..3] OF INT;

```
AttemptCount : INT := 0;  
CodeIndex : INT := 1;  
IsCodeSet : BOOL := FALSE;
```

```
i: INT := 0;
```

```
END_VAR
```

```
(*ÒÀÉÌÅÐ ÄËß ÌÈÃÀÍÈß*)
```

```
IF MixingTimerStart THEN
```

```
    MixingTimerCount := MixingTimerCount + (TIME() - LastTime);
```

```
    L1 := TRUE;
```

```
    L2 := TRUE;
```

```
IF MixingTimerCount >= MixingTimerPreset THEN
```

```
    L1 := FALSE;
```

```
    L2 := FALSE;
```

```
    MixingTimerCount := T#0S;
```

```
END_IF;
```

```
END_IF;
```

```
(*ÒÀÉÌÅÐ ÄËß ÌÈÃÀÍÈß L2*)
```

```
IF MixingTimerStart_L2 THEN
```

```
    MixingTimerCount_L2 := MixingTimerCount_L2 + (TIME() - LastTime);
```

```
    L2 := TRUE;
```

```
IF MixingTimerCount_L2 >= MixingTimerPreset_L2 THEN
```

```
    L2 := FALSE;
```

```
    MixingTimerCount_L2 := T#0S;
```

```
END_IF;
```

```
END_IF;
```

(\*ÒÀÉÌÅÐ ÄËß ÃÎÐÅÍÈß 5Ñ\*)

```
IF MixingTimerStart_ON THEN
    MixingTimerCount_ON := MixingTimerCount_ON + (TIME() -
LastTime);
    L1 := TRUE;
    L2 := TRUE;
    IF MixingTimerCount_ON >= MixingTimerPreset_ON THEN
        L1 := FALSE;
        L2 := FALSE;
        MixingTimerCount_ON := T#0S;
        MixingTimerStart_ON := FALSE;
    END_IF;
END_IF;
LastTime := TIME();
```

(\*ÑÁÐÎÑ ÊÎÄÀ È ÂÂÎÄ ÎÂÎÎÎ, ËÀÎÎÎ×ÊÈ ÌÈÃÀÐÒ\*)

```
IF NOT D3 AND BTN_RESET THEN
    MixingTimerStart := TRUE;
    IsCodeSet := FALSE;

    FOR i := 1 TO 3 DO
        CheckCode[i] := 0;
    END_FOR;
END_IF;
```

(\*ÂÂÎÄ ×ÅÐÅÇ ÊÎÎÎÈÈ ÊÎÄÀ\*)

```
IF BTN_0 THEN

    InputCode[CodeIndex] := 0;
```



```
        CodeIndex := CodeIndex + 1;  
        BTN_0 := FALSE;  
    END_IF;
```

```
IF BTN_1 THEN
```

```
    InputCode[CodeIndex] := 1;  
    CodeIndex := CodeIndex + 1;  
    BTN_1 := FALSE;  
END_IF;
```

```
IF BTN_2 THEN
```

```
    InputCode[CodeIndex] := 2;  
    CodeIndex := CodeIndex + 1;  
    BTN_2 := FALSE;  
END_IF;
```

```
IF BTN_3 THEN
```

```
    InputCode[CodeIndex] := 3;  
    CodeIndex := CodeIndex + 1;  
    BTN_3 := FALSE;  
END_IF;
```

```
IF BTN_4 THEN
```

```
    InputCode[CodeIndex] := 4;  
    CodeIndex := CodeIndex + 1;
```

```
        BTN_4 := FALSE;  
    END_IF;
```

```
IF BTN_5 THEN
```

```
    InputCode[CodeIndex] := 5;  
    CodeIndex := CodeIndex + 1;  
    BTN_5 := FALSE;  
END_IF;
```

```
IF BTN_6 THEN
```

```
    InputCode[CodeIndex] := 6;  
    CodeIndex := CodeIndex + 1;  
    BTN_6 := FALSE;  
END_IF;
```

```
IF BTN_7 THEN
```

```
    InputCode[CodeIndex] := 7;  
    CodeIndex := CodeIndex + 1;  
    BTN_7 := FALSE;  
END_IF;
```

```
IF BTN_8 THEN
```

```
    InputCode[CodeIndex] := 8;  
    CodeIndex := CodeIndex + 1;  
    BTN_8 := FALSE;  
END_IF;
```

IF BTN\_9 THEN

    InputCode[CodeIndex] := 9;

    CodeIndex := CodeIndex + 1;

    BTN\_9 := FALSE;

END\_IF;

(\* ОТКРЫТИЕ ЗАМКА МАСТЕР-КОДОМ\*)

IF InputCode[1] = 0 AND InputCode[2] = 0 AND InputCode[3] = 0 AND  
BTN\_ENTER AND D1 THEN

    D1 := FALSE;

    D2 := TRUE;

    D3 := TRUE;

    BTN\_ENTER:= FALSE;

END\_IF;

(\*ЗАКРЫТИЕ ЗАМКА\*)

IF D3 AND BTN\_ENTER AND D2 THEN

    D1:= TRUE;

    D2 := FALSE;

    D3 := TRUE;

    BTN\_ENTER := FALSE;

    CodeIndex := 1;

END\_IF;

(\*ОТКРЫТИЕ ЗАМКА, 3 ПОПЫТКИ\*)

IF D1 AND BTN\_ENTER AND IsCodeSet THEN

IF InputCode[1] = CheckCode[1] AND InputCode[2] = CheckCode[2] AND  
InputCode[3] = CheckCode[3] THEN

D1 := FALSE;

D2 := TRUE;

D3 := TRUE;

(\*CodeIndex := 1;\*)

BTN\_ENTER:= FALSE;

ELSE IF InputCode[1] <> CheckCode[1] AND InputCode[2] <>  
CheckCode[2] AND InputCode[3] <> CheckCode[3] AND AttemptCount = 0  
THEN

AttemptCount := AttemptCount + 1;

BTN\_ENTER:= FALSE;

L2 := TRUE;

(\*CodeIndex := 1;\*)

BTN\_ENTER:= FALSE;

ELSE IF InputCode[1] <> CheckCode[1] AND InputCode[2] <>  
CheckCode[2] AND InputCode[3] <> CheckCode[3] AND AttemptCount = 1  
THEN

L2 := FALSE;

MixingTimerStart\_L2 := TRUE;

AttemptCount := AttemptCount + 1;

(\*CodeIndex := 1;\*)

BTN\_ENTER:= FALSE;

ELSE IF AttemptCount = 2 THEN

MixingTimerStart\_L2 := FALSE;

L2 := FALSE;

BA := TRUE;

BTN\_ENTER:= FALSE;

```

    END_IF;
    END_IF;
    END_IF;
    END_IF;
    CodeIndex := 1;
END_IF

(*ПРИНЯТИЕ КОДА*)
IF BTN_ENTER AND CodeIndex = 4 AND NOT IsCodeSet THEN
    MixingTimerStart := FALSE;
    IsCodeSet := TRUE;
    MixingTimerStart_ON := TRUE;
    BTN_ENTER := FALSE;
    BTN_RESET := FALSE;
    CodeIndex := 1;

    FOR i := 1 TO 3 DO
        CheckCode[i] := InputCode[i];
        InputCode[i] := 0;
    END_FOR;
END_IF;

```

### **Результаты работы программы**

На рисунках 1 – продемонстрированы результаты выполнения программы.

0001	MixingTimerStart = <b>TRUE</b>
0002	MixingTimerElapsed = <b>FALSE</b>
0003	MixingTimerCount = T#138ms
0004	MixingTimerPreset = T#1s0ms
0005	LastTime = T#71073m13s813ms
0006	MixingTimerStart_ON = <b>FALSE</b>
0007	MixingTimerCount_ON = T#0ms
0008	MixingTimerPreset_ON = T#5s0ms
0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = <b>FALSE</b>
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = <b>FALSE</b>
0015	D2 = <b>TRUE</b>
0016	D3 = <b>FALSE</b>
0017	BTN_0 = <b>FALSE</b>
0018	BTN_1 = <b>FALSE</b>
0019	BTN_2 = <b>FALSE</b>
0020	BTN_3 = <b>FALSE</b>
0021	BTN_4 = <b>FALSE</b>
0022	BTN_5 = <b>FALSE</b>
0023	BTN_6 = <b>FALSE</b>
0024	BTN_7 = <b>FALSE</b>
0025	BTN_8 = <b>FALSE</b>
0026	BTN_9 = <b>FALSE</b>
0027	BTN_ENTER = <b>FALSE</b>
0028	BTN_RESET = <b>TRUE</b>
0029	M = <b>FALSE</b>
0030	L1 = <b>TRUE</b>
0031	L2 = <b>TRUE</b>
0032	BA = <b>FALSE</b>
0033	▣---Code
0034	▣---MasterCode
0035	▣---InputCode
0036	▣---CheckCode
0037	AttemptCount = 0
0038	CodeIndex = 1
0039	IsCodeSet = <b>FALSE</b>
0040	i = 4
0041	
0042	

Рисунок 1 – Сброс кода, ввод нового – мигание индикаторов (начало)

0001	MixingTimerStart = TRUE
0002	MixingTimerElapsed = FALSE
0003	MixingTimerCount = T#262ms
0004	MixingTimerPreset = T#1s0ms
0005	LastTime = T#71074m3s372ms
0006	MixingTimerStart_ON = FALSE
0007	MixingTimerCount_ON = T#0ms
0008	MixingTimerPreset_ON = T#5s0ms
0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = FALSE
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = FALSE
0015	D2 = TRUE
0016	D3 = FALSE
0017	BTN_0 = FALSE
0018	BTN_1 = FALSE
0019	BTN_2 = FALSE
0020	BTN_3 = FALSE
0021	BTN_4 = FALSE
0022	BTN_5 = FALSE
0023	BTN_6 = FALSE
0024	BTN_7 = FALSE
0025	BTN_8 = FALSE
0026	BTN_9 = FALSE
0027	BTN_ENTER = FALSE
0028	BTN_RESET = TRUE
0029	M = FALSE
0030	L1 = TRUE
0031	L2 = TRUE
0032	BA = FALSE
0033	▣--Code
0034	▣--MasterCode
0035	▣--InputCode
0036	InputCode[1] = 1
0037	InputCode[2] = 0
0038	InputCode[3] = 0
0039	▣--CheckCode
0040	AttemptCount = 0
0041	CodeIndex = 2
0042	IsCodeSet = FALSE
0043	i -- 4

Рисунок 2 – Ввод нового кода (первая цифра)

0007	MixingTimerCount_ON = T#0ms
0008	MixingTimerPreset_ON = T#5s0ms
0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = FALSE
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = FALSE
0015	D2 = TRUE
0016	D3 = FALSE
0017	BTN_0 = FALSE
0018	BTN_1 = FALSE
0019	BTN_2 = FALSE
0020	BTN_3 = FALSE
0021	BTN_4 = FALSE
0022	BTN_5 = FALSE
0023	BTN_6 = FALSE
0024	BTN_7 = FALSE
0025	BTN_8 = FALSE
0026	BTN_9 = FALSE
0027	BTN_ENTER = FALSE
0028	BTN_RESET = FALSE
0029	M = FALSE
0030	L1 = FALSE
0031	L2 = FALSE
0032	BA = FALSE
0033	⊕---Code
0034	⊕---MasterCode
0035	⊖---InputCode
0036	---InputCode[1] = 0
0037	---InputCode[2] = 0
0038	---InputCode[3] = 0
0039	⊖---CheckCode
0040	---CheckCode[1] = 1
0041	---CheckCode[2] = 2
0042	---CheckCode[3] = 3
0043	AttemptCount = 0
0044	CodeIndex = 1
0045	IsCodeSet = TRUE
0046	i = 4

Рисунок 3 – Новый код принят (код передаётся в память)



0007	MixingTimerCount_ON = T#0ms
0008	MixingTimerPreset_ON = T#5s0ms
0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = FALSE
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = TRUE
0015	D2 = FALSE
0016	D3 = TRUE
0017	BTN_0 = FALSE
0018	BTN_1 = FALSE
0019	BTN_2 = FALSE
0020	BTN_3 = FALSE
0021	BTN_4 = FALSE
0022	BTN_5 = FALSE
0023	BTN_6 = FALSE
0024	BTN_7 = FALSE
0025	BTN_8 = FALSE
0026	BTN_9 = FALSE
0027	BTN_ENTER = FALSE
0028	BTN_RESET = FALSE
0029	M = FALSE
0030	L1 = FALSE
0031	L2 = TRUE
0032	BA = FALSE
0033	Code
0034	MasterCode
0035	InputCode
0036	InputCode[1] = 0
0037	InputCode[2] = 0
0038	InputCode[3] = 0
0039	CheckCode
0040	CheckCode[1] = 1
0041	CheckCode[2] = 2
0042	CheckCode[3] = 3
0043	AttemptCount = 1
0044	CodeIndex = 1
0045	IsCodeSet = TRUE
0046	i = 4

Рисунок 4 – Выполнение условия закрытия замка

0006	MixingTimerStart_ON = FALSE
0007	MixingTimerCount_ON = T#0ms
0008	MixingTimerPreset_ON = T#5s0ms
0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = FALSE
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = TRUE
0015	D2 = FALSE
0016	D3 = TRUE
0017	BTN_0 = FALSE
0018	BTN_1 = FALSE
0019	BTN_2 = FALSE
0020	BTN_3 = FALSE
0021	BTN_4 = FALSE
0022	BTN_5 = FALSE
0023	BTN_6 = FALSE
0024	BTN_7 = FALSE
0025	BTN_8 = FALSE
0026	BTN_9 = FALSE
0027	BTN_ENTER = FALSE
0028	BTN_RESET = FALSE
0029	M = FALSE
0030	L1 = FALSE
0031	L2 = TRUE
0032	BA = FALSE
0033	⊞---Code
0034	⊞---MasterCode
0035	⊞---InputCode
0036	InputCode[1] = 2
0037	InputCode[2] = 3
0038	InputCode[3] = 4
0039	⊞---CheckCode
0040	CheckCode[1] = 1
0041	CheckCode[2] = 2
0042	CheckCode[3] = 3
0043	AttemptCount = 1
0044	CodeIndex = 4
0045	IsCodeSet = TRUE
0046	i = 4
0047	
0048	

Рисунок 5 – Ввод неверного кода

0009	LastTime_ON = T#0ms
0010	MixingTimerStart_L2 = FALSE
0011	MixingTimerCount_L2 = T#0ms
0012	MixingTimerPreset_L2 = T#1s0ms
0013	LastTime_L2 = T#0ms
0014	D1 = TRUE
0015	D2 = FALSE
0016	D3 = TRUE
0017	BTN_0 = FALSE
0018	BTN_1 = FALSE
0019	BTN_2 = FALSE
0020	BTN_3 = FALSE
0021	BTN_4 = FALSE
0022	BTN_5 = FALSE
0023	BTN_6 = FALSE
0024	BTN_7 = FALSE
0025	BTN_8 = FALSE
0026	BTN_9 = FALSE
0027	BTN_ENTER = FALSE
0028	BTN_RESET = FALSE
0029	M = FALSE
0030	L1 = FALSE
0031	L2 = TRUE
0032	BA = FALSE
0033	⊞ Code
0034	⊞ MasterCode
0035	⊞ InputCode
0036	InputCode[1] = 2
0037	InputCode[2] = 3
0038	InputCode[3] = 4
0039	⊞ CheckCode
0040	CheckCode[1] = 1
0041	CheckCode[2] = 2
0042	CheckCode[3] = 3
0043	AttemptCount = 1
0044	CodeIndex = 4
0045	IsCodeSet = TRUE
0046	i = 4
0047	

Рисунок 6 – Результат неверного ввода (лампочка горит)

## Вывод

В ходе данной лабораторной работе была разработана программа на языке ST для ОВЕН ПЛК110, реализующая заданный технологический процесс.