

**Программируемые логические контроллеры
серии ТК-2**

Руководство по программированию

Самара, 2006

Содержание.

Содержание.....	2
1. Общие сведения.....	3
2. Основные технические характеристики	3
3. Описание языка.....	3
3.1 Программа.....	3
3.2 Структура сигналов.....	4
3.3 Знаки операций.....	4
3.4 Константы.....	5
3.5 Комментарии.....	5
3.6 Входы и выходы.....	5
3.7 Маркеры.....	5
3.8 Таймеры.....	5
3.9 Счетчики.....	6
4. Примеры программирования.....	8
4.1 Генератор импульсов.....	8
4.2 Единичный импульс.....	8
5. Работа с программой WinPLC.....	10
5.1 Назначение программы.....	10
5.2 Меню и внешний вид.....	10
5.3 Редактор рабочих программ.....	11
5.4 Монитор реального времени.....	12
5.5 Анализатор сигналов.....	13
5.6 Загрузка рабочей программы в контроллер.....	16

1. Общие сведения.

Программируемые логические контроллеры (ПЛК) серии ТК-2 предназначены для широкого спектра применений, в том числе для экономичного варианта замены устаревшего электрооборудования и релейной логики.

Описание схем релейной автоматики производится в виде совокупности логических выражений. Для преобразования описания системы в исполняемый код разработан компилятор входного языка.

2. Основные технические характеристики

В таблице 1 приведены основные технические характеристики ПЛК серии ТК-2.

	ТК-2-12	ТК-2-24	ТК-2-64
Количество входов	8	16	32
Количество выходов	4	8	32
Количество маркеров	32	64	128
Количество таймеров	8	16	16
Диапазон выдержек времени	от 10 мс до 255 с		
Количество счетчиков	4	4	4
Память программ	2 кб	3 кб	4 кб
Типовой цикл контроллера	10 мс		
Время выполнения 1К инструкций	~0.4 мс		

Таблица 1

3. Описание языка.

3.1 Программа.

Программа для контроллера состоит из последовательности строк, состоящих из операторов присваивания значений некоторым сигналам. При этом слева от знака равенства стоит наименование сигнала, а справа – некоторое логическое выражение. Программа линейная. Операторы перехода и циклов отсутствуют.

Пример программирования:

;Лист 13. Точная установка траверсы

Q1 = M7 * M26 * !Q2

Q2 = M9 * M26 * !Q1

M26 = I8 * M20

M27 = I8 * !M26

Q3 = !M26

;Лист 16. Шлифовальный двигатель. Суппорт 2

M11 = M9 * !M29 * Q4 * Q5 * !T14

$$M10 = M9 * (I11 * !Q8 + M5)$$

$$Q4 = (M10 + M11 * !M5) * !M7$$

$$Q5 = M10 * !M5 + M11$$

$$T12(10,1) = Q5 * I13$$

3.2 Структура сигналов.

Общий вид любого сигнала в программе логики (исключая таймеры и счетчики, которые встречаются слева от знака равенства – их формат описан отдельно):

S_{nn} ,

где символ S определяет тип сигнала и может принимать значения I, Q, M, C, T, E , а nn – номер сигнала.

Диапазон изменения номеров указан в таблице 2:

Сигнал	Назначение	TK-2-12	TK-2-24	TK-2-64
		Номера		
I	входы	0-7	0-15	0-31
Q	выходы	0-3	0-7	0-31
M	маркеры	0-31	0-64	0-127
T	таймеры	0-8	0-15	0-15
C	счетчики	0-3	0-3	0-3

Таблица 2

Примечание. Сигнал I (вход) не должен находиться слева от знака равенства, так как нельзя присвоить значение входу. Это вызовет ошибку компиляции.

3.3 Знаки операций.

Знак операции	Значение
*	логическое И
+	логическое ИЛИ
!	логическое НЕ (отрицание)
=	присваивание

Таблица 3

Для задания порядка операций в логических выражениях могут быть использованы скобки. Например:

$$M11 = (M11 * (I25 + M14)) * (!I27 * M16 + !M22) * !M12$$

3.4 Константы.

В программе могут присутствовать только две константы 0 и 1 (логический 0 и логическая 1). Например:

$C2(R) = 1$; безусловный сброс счетчика C2

3.5 Комментарии

Символ “;” используется для комментария. Он может стоять в начале строки или в середине строки. Все следующие за ним символы игнорируются компилятором. Например:

; Лист 7. Включение гидравлики
 $T9(5,2) = I21 * !Q19$
 $Q18 = I21 * !Q19 * !T9$
 $Q8 = M16$; лампочки подачи

3.6 Входы и выходы.

I и Q – сигналы, представляющие собой физические входы и выходы. Их количество зависит от поставляемой конфигурации контроллера, но не может быть больше чем 32/32 (см. таблицу 2).

Пример обозначения в программе:

I23 – двадцать третий вход.
Q5 – пятый выход.

3.7 Маркеры.

M – маркеры. Сигналы, представляющие собой рабочую память программы логики и использующиеся для промежуточных вычислений. Максимальное количество маркеров определяется моделью ПЛК (см. таблицу 2).

Пример обозначения в программе:

M30 – тридцатый маркер.

3.8 Таймеры.

Имеется максимум 16 таймеров (см. таблицу 2), которые могут быть запрограммированы на время от 10 миллисекунд до 255 секунд (4,25 минуты). Таймер начинает отсчет когда логическое выражение справа от знака равенства станет равно логической единице. По истечении заданного интервала он установится в единицу, которая будет присутствовать до тех пор пока выражение справа от знака равенства истинно. Если логическое выражение справа сбросится в ноль до истечения времени задержки, то таймер не сработает.

Таймер задается буквой T (timer) и имеет 2 формата.

1. Слева от знака равенства (определение таймера):
 $T_{nn}(mm, t)$, где

nn – номер таймера от 0 до 15,
 mm – задержка от 0 до 255 в единицах, указанных параметром t ,
 t – тип таймера от 0 до 2, определяемый так:

тип 0. 10 миллисекунд.
тип 1. 100 миллисекунд.
тип 2. 1 секунда.

2. Справа от знака равенства (использование выхода таймера):
 T_{nn} , где nn – номер таймера от 0 до 15.

$Q6 = M12 * !Q7 * !T13$

Здесь использовано значение таймера T13.

Пример использования в программе:

$T12(45,0)$ - таймер номер 12 с задержкой на 450 мс

$T12(45,1)$ - таймер номер 12 с задержкой на 4500 мс = 4,5 с

$T12(45,2)$ - таймер номер 12 с задержкой на 45 секунд

$T12(45,2) = M13 * !Q6$

В последней строчке таймер номер 12 сработает (установится в единицу) через 45 секунд после того как маркер M13 установится в единицу, а выходной сигнал Q6 в ноль одновременно.

3.9 Счетчики.

В программе можно использовать до 4 счетчиков, которые обозначаются буквой C (counter). Счетчик выполняет функцию делителя с заданным коэффициентом деления. По достижению заданного количества импульсов, на выходе счетчика формируется импульс по длительности равный последнему входному импульсу. См. ниже временную диаграмму для примера 3.

Формат команд для работы со счетчиком:

1. Слева от знака равенства (определение счетчика):

$C_n(\text{число})$ или $C_n(\text{команда R или P})$, где

n – номер счетчика от 0 до 3,

число – константа пересчета от 1 до 255,

R – сброс счетчика,

P – счет;

Имеется 3 команды работы со счетчиком, которые используются в левой части оператора присваивания:

$C2(20)$ = логическое выражение
 если выражение истинно, загрузить константу пересчета 20 для 2-го счетчика;
 $C1(R)$ = логическое выражение
 если выражение истинно, сбросить счетчик номер 1;
 $C1(P)$ = логическое выражение
 если значение выражения в прав. части изменяется из 0 в 1 (передний фронт), то увеличить значение счетчика на 1.

Пример 1:

$C2(20) = Q1$

$C2(R) = 1$; сброс счетчика C2

2. Справа от знака равенства (использование значения счетчика):
 Cn , где n – номер счетчика от 0 до 3.

Пример 2:

$M28 = I8 * !C3 * (M16 + M18)$; используется выход счетчика C3

Пример 3:

$C1(5)=1$; Безусловная установка счетчика C1.
 ; Константа пересчета равна 5. Будет считать по 5 импульсов.

$C1(P)=I0$; импульсы будут считаться на входе I0

$C1(R)=I3$; вход I3 будет использоваться для сброса счетчика C1

На рисунке 1 видно, что счетчик C1 отсчитывает 5 импульсов на входе I0, после чего выдает на выходе импульс, по длительности равный последнему импульсу на входе I0. Если во время счета приходит сигнал сброса на I3, то счет начинается заново.

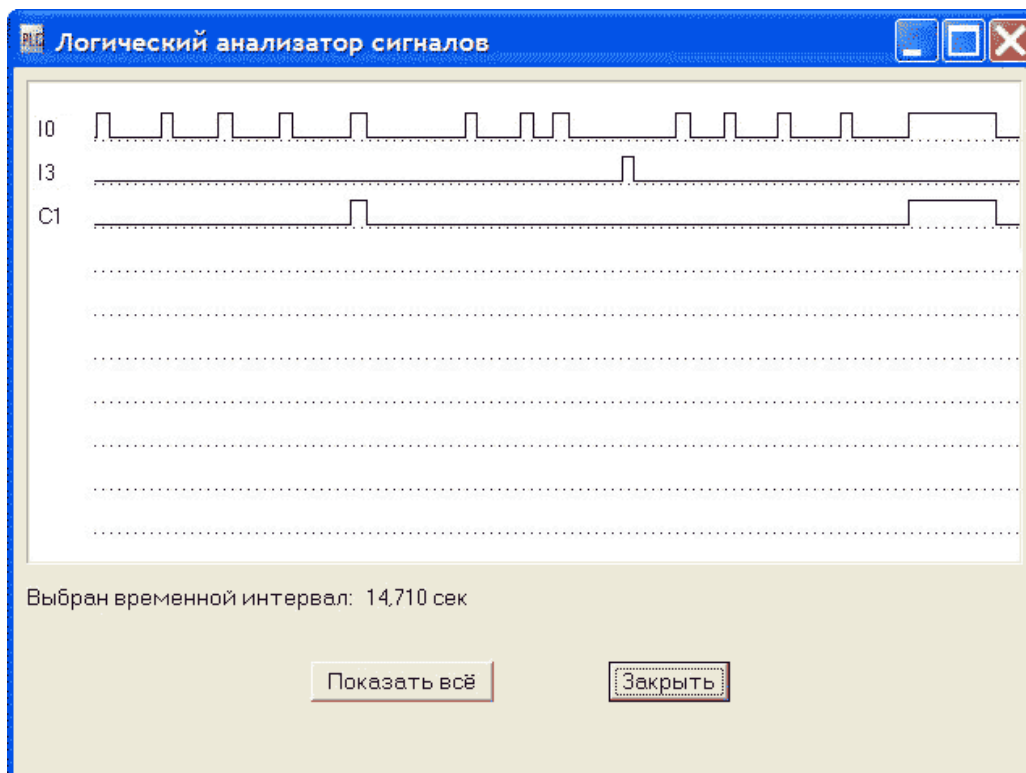


Рисунок 1

3.10 Ошибки.

В рабочей программе можно использовать до 127 ошибок (от 1 до 127), определенных пользователем для различных ситуаций, где необходимо отслеживать возникновение каких-то ошибочных условий в исполнительных механизмах. Ошибки также можно использовать для отладки рабочих программ.

Формат командной строки:

$E_{nn} =$ логическое выражение.

При возникновении ошибки (когда логическое выражение станет истинно) начнет мигать красный светодиод на контроллере и индикатор «Ошибка» в программе WinPLC.

Пример.

$$E20 = I0 * Q8$$

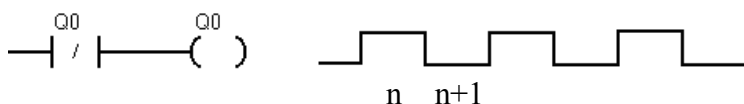
Если на входе I0 и на выходе Q8 будут единицы, то установится 20-ая ошибка.

4. Примеры программирования.

4.1 Генератор импульсов.

Генератор импульсов с периодом, равным двум циклам контроллера (20 миллисекунд):

$$Q0 = !Q0$$

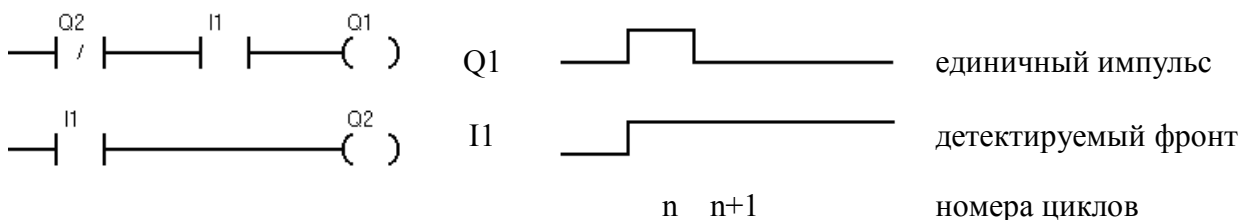


n и $n+1$ – два последовательных цикла контроллера.

4.2 Единичный импульс.

Работа многих устройств основана на считывании перепадов потенциала, а не уровней. Различие в скорости работы объекта управления и контроллера приводит к необходимости того, чтобы фронт или спад перепада считывался и обрабатывался в течение цикла, в то время как само физическое явление может продолжаться. *Единичный импульс* – это логический сигнал, создаваемый фронтом или спадом импульса и принимающий значение 1 в течение только одного цикла.

Детектирование фронта сигнала I1

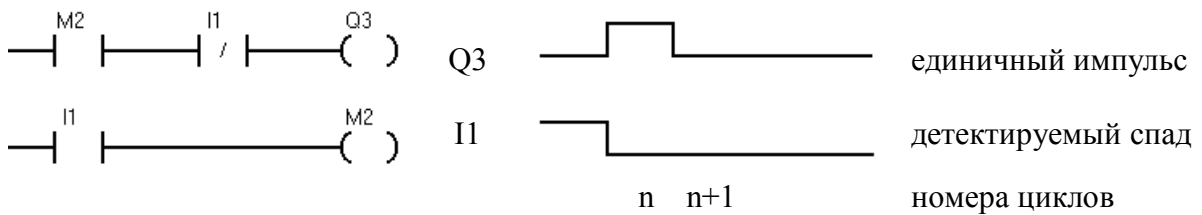


$$Q1 = !Q2 * I1$$

$$Q2 = I1$$

- Цикл n На входе I1 появляется единица. Выход Q1 устанавливается в единицу, т.к. $Q2 = 0$ и $!Q2 * I1$ равно единице.
В следующей команде (того же цикла n) выход Q2 установится в единицу.
- Цикл n+1 Выход Q1 установится в 0, т.к. Q2 теперь равен единице, а его отрицание – нулю и $!Q2 * I1$ будет равно нулю. В результате на Q1 сгенерируется единичный импульс, длительность которого равна одному циклу контроллера.

Детектирование спада сигнала I1

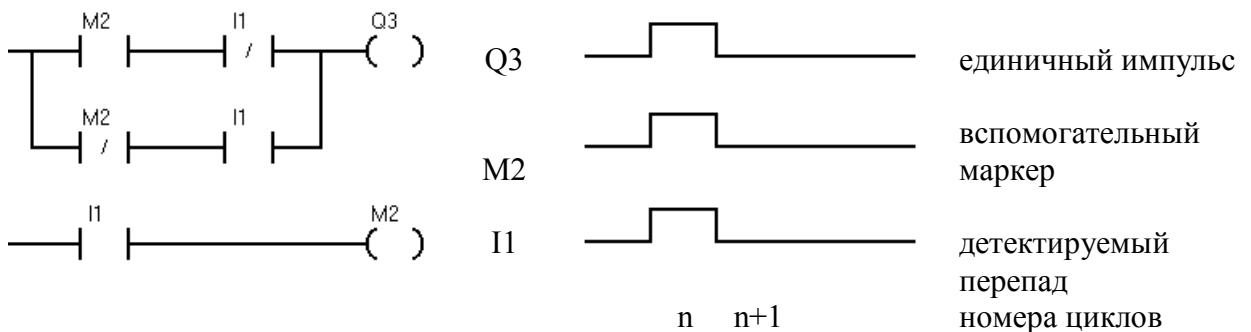


$$Q3 = M2 * !I1$$

$$M2 = I1$$

- Цикл n *Предыдущее состояние.* $I1 = 1, M2 = 1, Q3 = 0$
- 1 команда.* $Q3 = M2 * !I1$
На входе I1 появляется ноль. Выход Q3 устанавливается в единицу, т.к. $M2 = 1, I1 = 1$ и $M2 * !I1$ равно единице.
- 2 команда.* $M2 = I1$
M2 становится равным нулю.
- Цикл n+1 *1 команда.* $Q3 = M2 * !I1$
Выход Q3 установится в 0, т.к. M2 теперь равен нулю и $M2 * !I1$ будет равно нулю. В результате на Q3 сгенерируется единичный импульс, длительность которого равна одному циклу контроллера.
- 2 команда.* $M2 = I1$
I1 по-прежнему равно нулю, M2 – тоже.

Считывание перепада сигнала I1



5. Работа с программой WinPLC.

5.1 Назначение программы.

Программа WinPLC предоставляет единую среду для программирования и отладки контроллера ТК-2-64. Основные функции программы:

- создание и редактирование рабочих программ;
- компиляция рабочих программ и создание выполняемого модуля;
- загрузка выполняемого модуля в контроллер;
- монитор реального времени сигналов, поступающих из контроллера;
- анализатор сигналов и просмотр временных диаграмм.

5.2 Меню и внешний вид.

После запуска WinPLC экран будет выглядеть как показано на рис. 2:

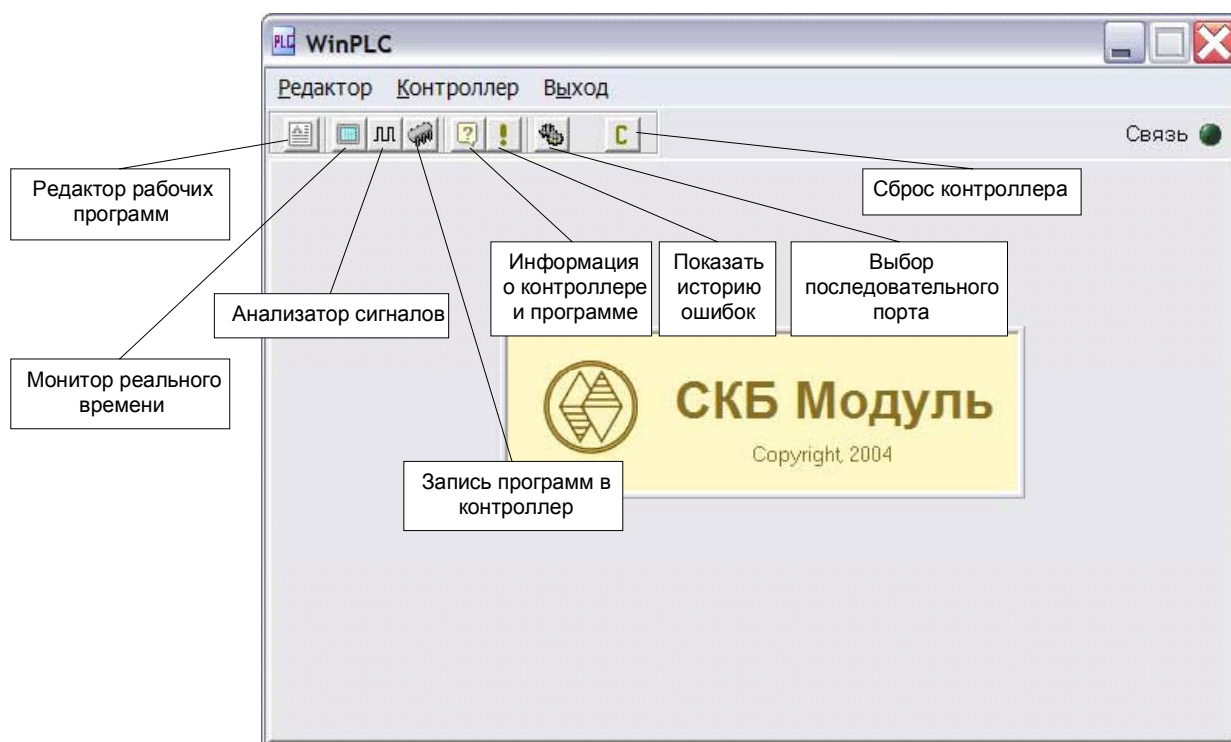


Рисунок 2

Все функции программы доступны как через кнопки на панели инструментов, так и через меню. Соответствие пунктов меню кнопкам приведено в таблице 5.

Пункт меню	Кнопка
Редактор	
Контроллер – монитор	
Контроллер – анализатор	
Контроллер – программатор	
Контроллер – информация	





Контроллер – ошибки	
Контроллер – сброс	
Контроллер – настройки	

Таблица 4


5.3 Редактор рабочих программ.

Редактор рабочих программ вызывается через меню «Редактор» или нажатием на кнопку . Редактор является простой программой редактирования файлов, аналогичной по функциям редактору Notepad в операционной системе Windows, но имеет чуть больше возможностей.

Список основных команд редактора:

Пункт меню	Пункт подменю	Клавиши быстрого вызова	Описание
Файл	Создать	Ctrl + N	Создает новый файл
Файл	Открыть	Ctrl + O	Открывает существующий файл
Файл	Сохранить	Ctrl + S	Сохранить текущий файл
Файл	Сохранить как		Сохранить текущий файл с другим именем
Файл	Печать	Ctrl + P	Печать текущего файла
Файл	Выход		Закончить работу с редактором
Правка	Откат	Ctrl + Z	Отменить последнее действие (на один шаг)
Правка	Вырезать	Ctrl + X	Вырезать выделенный текст и поместить его в буфер
Правка	Копировать	Ctrl + C	Скопировать выделенный текст и поместить его в буфер
Правка	Вставить	Ctrl + V	Вставить содержимое буфера
Правка	Поиск	Ctrl + F	Поиск заданного текста
Правка	Замена	Ctrl + H	Поиск и замена заданного текста
Правка	Шрифт		Сменить шрифт для всего текста
Контроллер	Компиляция	F7	Скомпилировать программу

Таблица 5

При нажатии на кнопку  или на клавишу F7 программа будет скомпилирована. Если есть ошибки, то в нижней статусной строке редактора появится сообщение о первой найденной ошибке и курсор покажет это место в программе. Сообщение об ошибке будет присутствовать в течение 5 секунд, после чего исчезнет с экрана. При необходимости снова посмотреть на ошибку можно еще раз нажать клавишу компиляции.

Если программа скомпилировалась без ошибок, то в текущей папке будет создан файл с тем же именем, что и исходный текст рабочей программы, но с расширением .hex. Этот файл содержит код рабочей программы в Intel Hex формате для загрузки во flash память микропроцессора (см. пункт 5.6 «Загрузка рабочей программы в контроллер»).

5.4 Монитор реального времени.

Вызывается нажатием на кнопку  или через меню Контроллер → Монитор.

В этом режиме программа в реальном времени принимает от контроллера все сигналы через последовательный порт на скорости 57600 бит/сек. Лампочка «Связь» горит если контроллер соединен с любым последовательным портом на компьютере, включен и отвечает на команды программы WinPLC.

Пользователь может наблюдать состояния входов, выходов, маркеров, таймеров и счетчиков внутри контроллера. Зажженная лампочка на экране отображает состояние логической единицы для данного сигнала (вход I2, выходные сигналы Q0, Q1 и Q2 на рис. 3), погашенная – логического нуля.

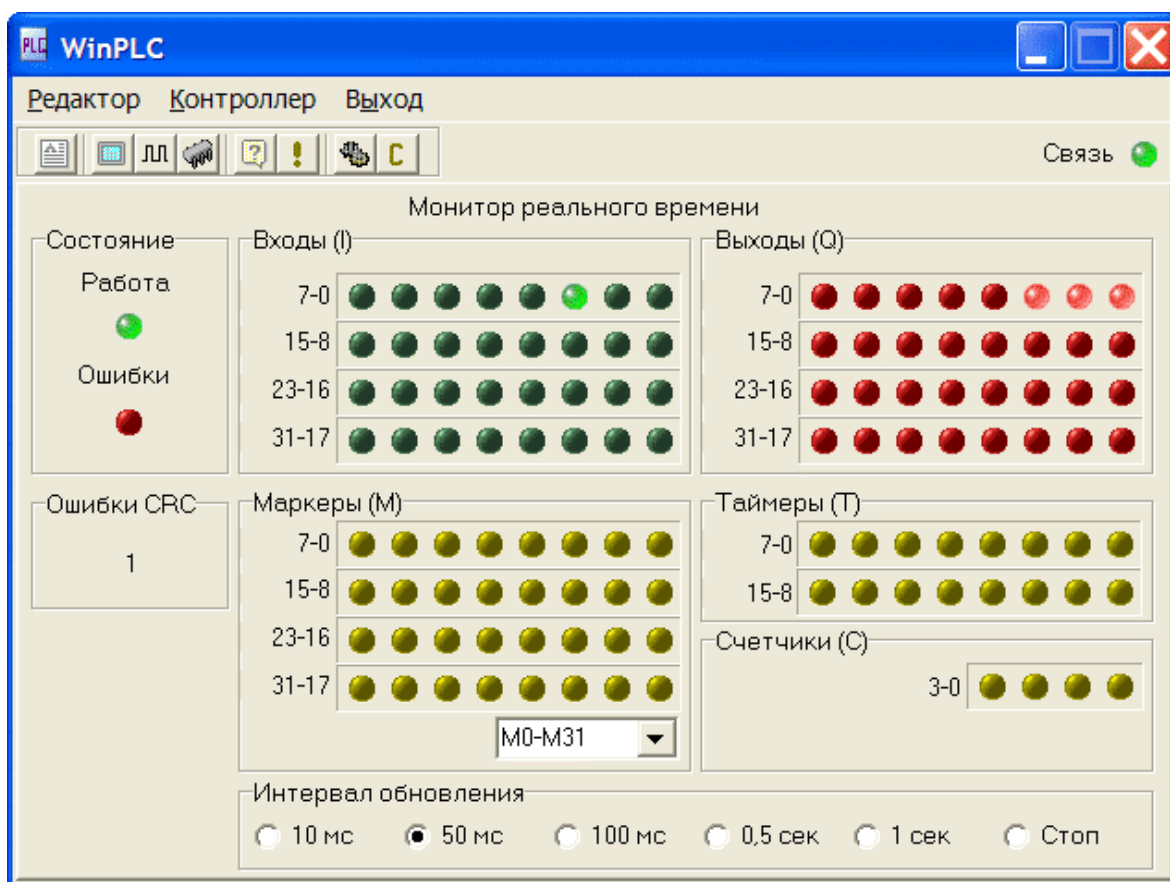


Рисунок 3

Поле «Ошибки CRC» отображает количество пакетов с ошибкой контрольной суммы, прошедших с момента включения монитора реального времени. Эти пакеты игнорируются и принимаются заново.

Поле «Интервал обновления» устанавливает частоту обновления данных на экране. Частота приема данных не меняется и составляет 0,01 сек, так как цикл контроллера выполняется за 10 миллисекунд и данные посылаются компьютеру через последовательный порт в конце каждого цикла.

Замечание. Не устанавливайте слишком маленький интервал (10 мс, 50 мс) на старых компьютерах с низкой частотой процессора. При этом будут задействованы 100% ресурсов процессора на прием данных и обновление экрана.

Значения индикаторов «Работа» и «Ошибки» описаны в таблице 7:


	Режим		
	«Работа»	«Ожидание»	«Программирование»
Рабочая программа	Загружена	Не загружена	Загружается
Индикатор «Работа» (соответствует светодиоду «Работа/Ожидание» на контроллере)	Горит	Мигает	Не горит
Индикатор «Ошибка» (соответствует светодиоду «Ошибка/Программирование» на контроллере)	Мигает при наличии ошибки	Мигает при наличии ошибки	Горит

Таблица 3

Индикатор «Работа» горит если загружена и выполняется рабочая программа в контроллере и мигает, если рабочая программа не загружена.

Индикатор «Ошибка» начинает мигать на контроллере и в программе WinPLC если были системные или пользовательские (определенные командой Enn =) ошибки.

5.5 Анализатор сигналов.

Вызывается нажатием на кнопку  или через меню Контроллер → Анализатор.

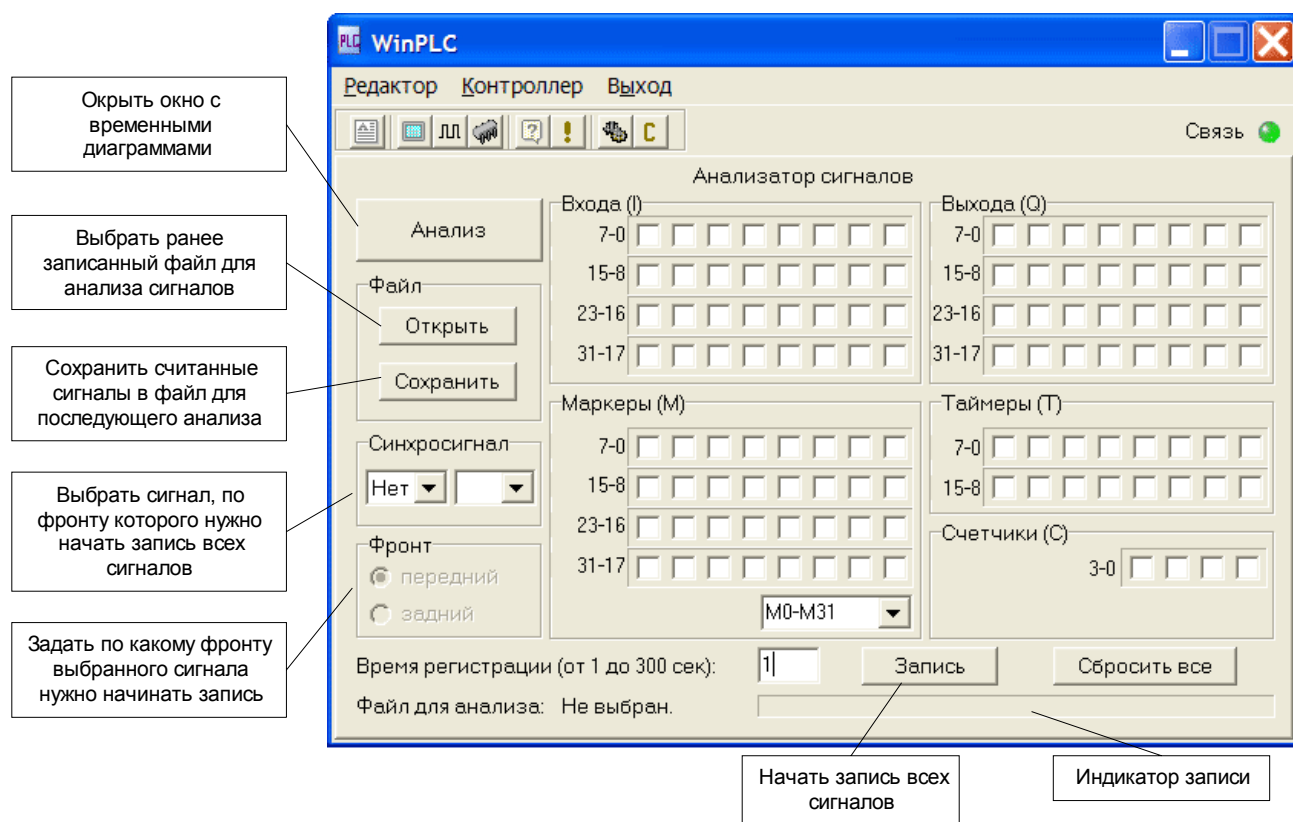


Рисунок 4

Для получения временных диаграмм сигналов, считанных из контроллера необходимо выполнить следующие действия:

1. Если запись сигналов должна начинаться по переднему или заднему фронту какого-либо сигнала, то нужно выбрать этот сигнал в пункте «Синхронизация по» (см. рисунок 1), а затем выбрать по какому фронту нужно синхронизироваться.
2. Задать время записи от 1 до 300 сек.
3. Нажать кнопку «Запись».

После нажатия кнопки «Запись» программа WinPLC начнет записывать все сигналы, поступающие с контроллера через последовательный порт или будет ожидать заданный фронт выбранного сигнала синхронизации, после прихода которого начнется запись. Сигналы записываются во внутреннем формате в файл winplc.log в текущей папке. При необходимости, их можно сохранить для последующего использования, нажав на кнопку сохранить и указав другое имя файла (так как при повторном нажатии на «Запись» файл winplc.log перезапишется).

Для просмотра временных диаграмм необходимо выбрать 14 любых сигналов с помощью левой кнопки мыши и нажать на кнопку «Анализ». Если анализ нажать сразу после записи, то данные будут браться из файла winplc.log. Для выбора другого, ранее записанного файла, нажмите «Открыть», выберите нужный файл и нажмите кнопку «Анализ».

При просмотре временных диаграмм можно выделить нужную область, нажав левую кнопку мыши и потянуть ее вправо, удерживая нажатой кнопку мыши (см. рис. 5).

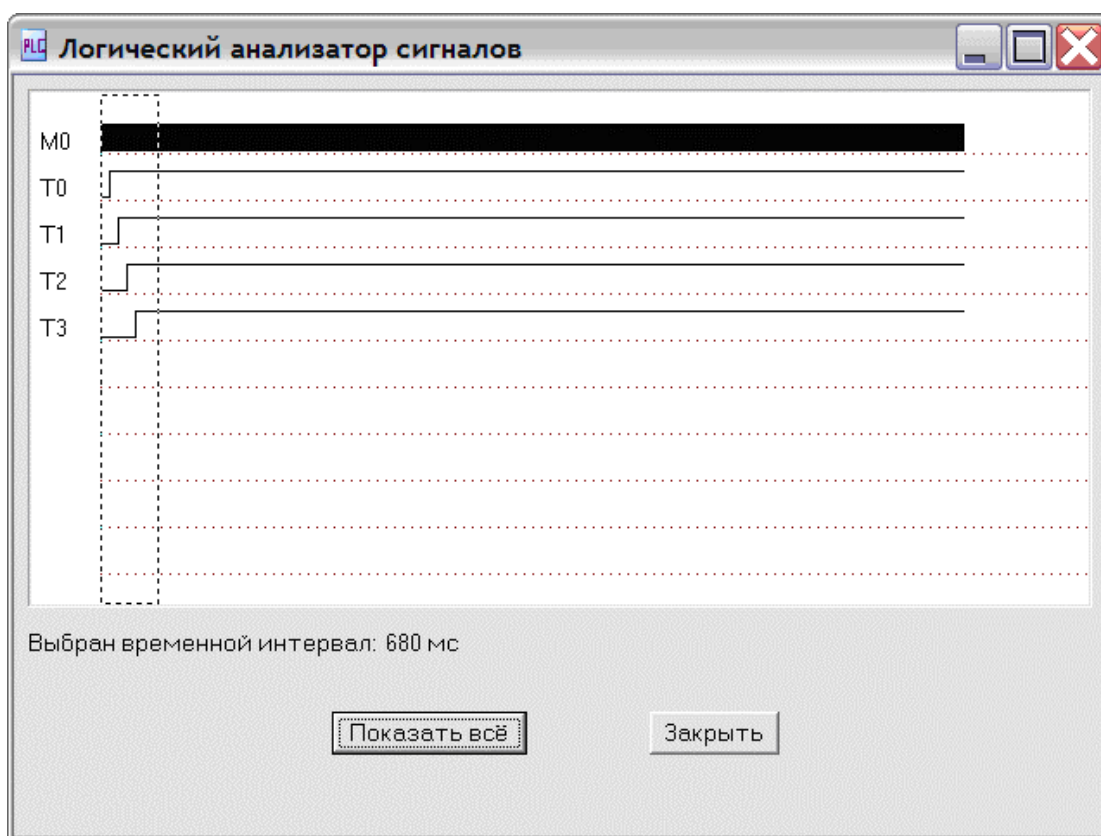


Рисунок 5

При этом область, попавшая в прямоугольник выбора, будет показана в укрупненном масштабе (см. рис. 6). Вернуться к прежнему масштабу (работает только на один шаг) можно, нажав правую кнопку мыши. Нажатие кнопки «Показать все» возвращает самый первый экран.

Замечание. Область можно выделить только движением мышки слева направо (особенность реализации).

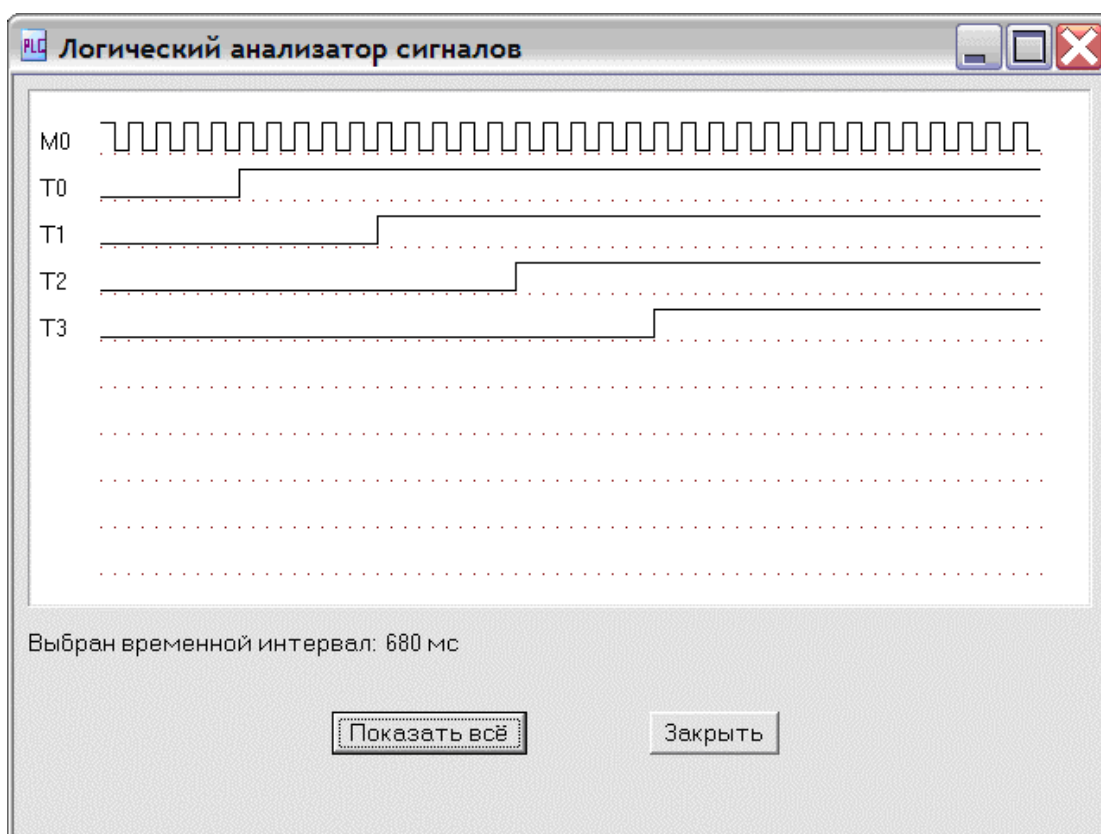

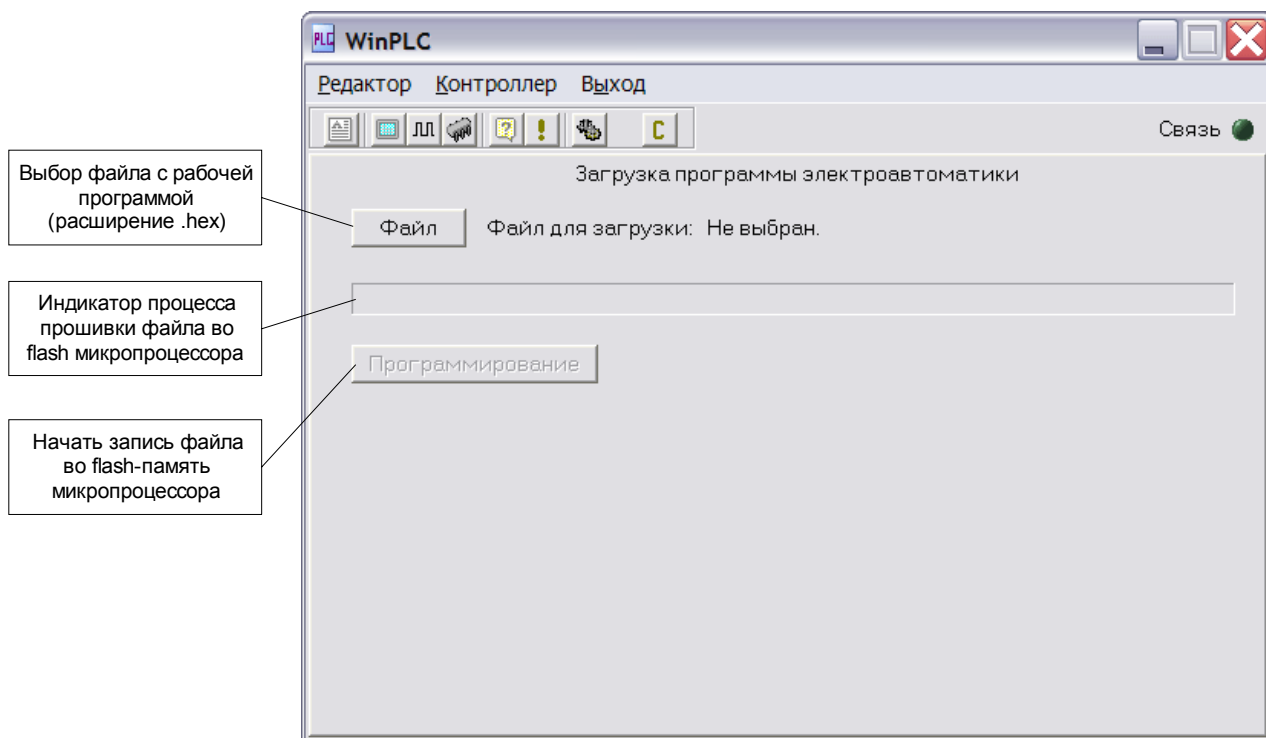


Рисунок 6

5.6 Загрузка рабочей программы в контроллер.

Экран загрузки (рис. 7) вызывается нажатием на кнопку  панели инструментов или через меню Контроллер → Программатор.



Для загрузки рабочей программы в контроллер необходимо:

1. Нажать кнопку «Файл» и выбрать файл с рабочей программой (который имеет расширение .hex);
2. Нажать кнопку «Программирование» и дождаться сообщения «Программирование закончилось успешно».

После этого контроллер автоматически перезагрузится и начнется выполнение рабочей программы, о чем будет сигнализировать горящий светодиод «Работа/Ожидание» на плате контроллера.