

# Использование рецептов

ock

# 13

STEP 7-Micro/Win предоставляет в ваше распоряжение Мастер рецептов, чтобы помочь вам организации рецептов и их определении. Рецепты хранятся не в ПЛК, а в модуле памяти.

## В этой главе

Обзор	358
Определение рецептов и терминология	359
Использование Мастера рецептов	359
Команды, создаваемые Мастером рецептов	363

## Обзор



Рецепт

Поддержка рецептов встроена в STEP 7-Micro/WIN и ПЛК S7-200. STEP 7-Micro/Win предоставляет в ваше распоряжение Мастер рецептов, чтобы помочь вам организации рецептов и их определении. Рецепты хранятся не в ПЛК, а в модуле памяти.

Все рецепты хранятся в модуле памяти. Поэтому для использования рецептов в ПЛК должен быть вставлен не обязательный модуль памяти на 64 или 256 Кбайт. Дополнительную информацию о модулях памяти вы найдете в Приложении А.

Все рецепты хранятся в модуле памяти. Однако, в память ПЛК считывается только один рецепт, когда программа пользователя обрабатывает это отдельный рецепт. Например, если вы выпекаете печенье, то имеются рецепты для печенья с шоколадной крошкой, сахарного печенья и овсяного печенья. В каждый данный момент времени может выпекаться только один вид печенья, так что должен быть выбран и считан в память ПЛК только один рецепт.

На рис. 13-1 показан процесс изготовления нескольких видов печенья с помощью рецептов. Рецепт для каждого вида печенья хранится в модуле памяти. С помощью текстового дисплея TD 200C оператор выбирает вид печенья, подлежащего изготовлению, а программа пользователя загружает этот рецепт в память.

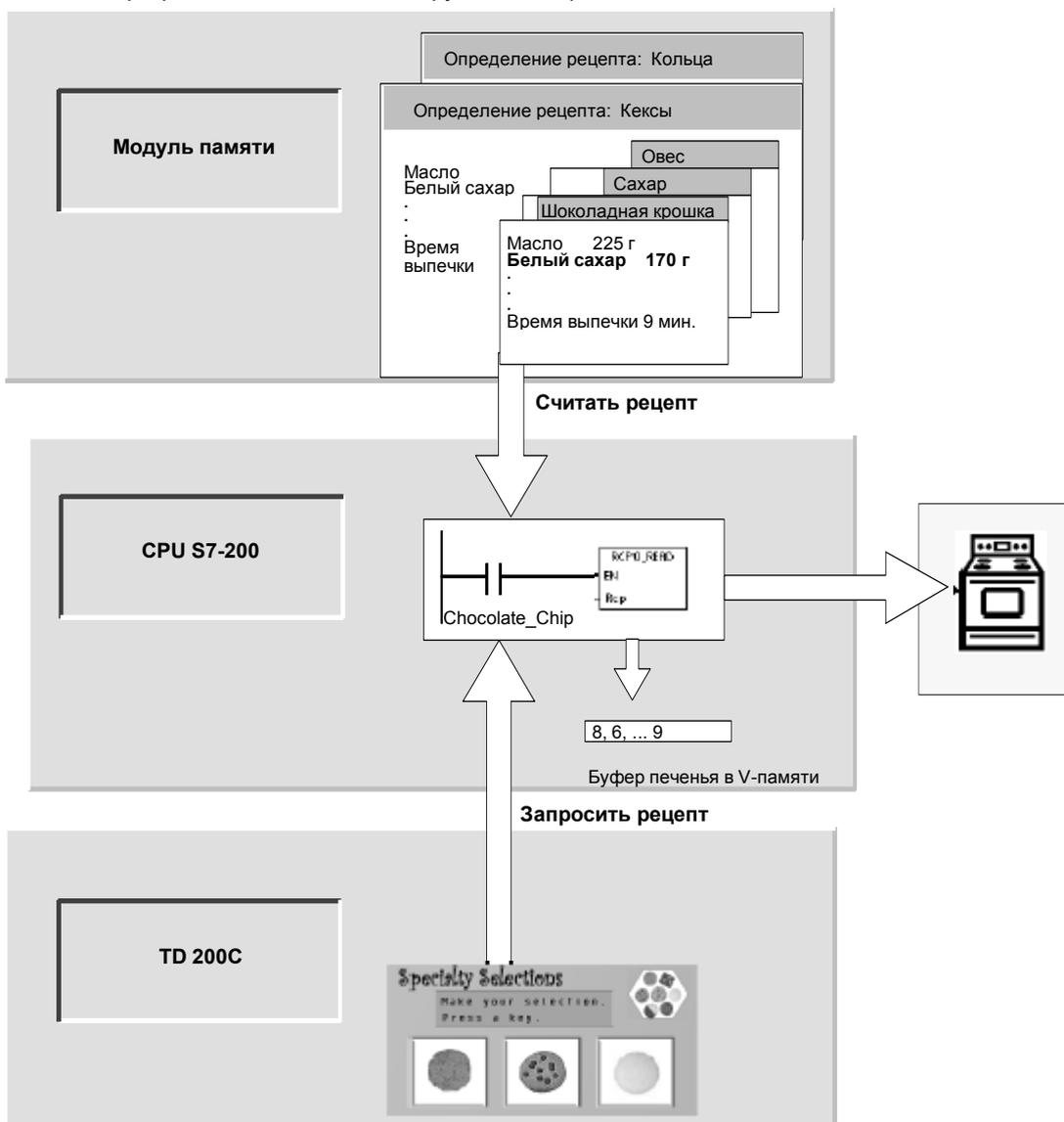


Рис. 13-1. Пример применения рецептов

## Определение рецептов и терминология

Для лучшего понимания Мастера рецептов ниже разъясняются следующие определения и термины.

- Конфигурация рецептов – это набор компонентов проекта, созданных Мастером рецептов. Эти компоненты включают в себя подпрограммы с командами, регистры блоков данных и таблицы символов.
- Определение рецептов - это собрание рецептов, обладающих одинаковым набором параметров. Однако значения параметров могут меняться в зависимости от рецепта.
- Рецепт – это набор параметров и их значений, дающий информацию, необходимую для производства продукта или управления процессом.

Например, могут быть созданы различные определения рецептов, например, колечек и печенья. Определение рецептов печенья может содержать много различных рецептов, например, печенье с шоколадной крошкой и сахарное печенье. Пример полей и значений приведен в таблице 13–1.

Таблица 13–1. Пример определения рецептов - печенье

Название поля	Тип данных	С шоколадной крошкой (рецепт 0)	Сахарное (рецепт 1)	Комментарий
Butter [Масло]	Byte	225	225	граммов
White_Sugar [Белый сахар]	Byte	170	340	граммов
Brown_Sugar [Желтый сахарный песок]	Byte	170	0	граммов
Eggs [Яйца]	Byte	2	1	штук
Vanilla [Ваниль]	Byte	1	1	чайных ложек
Flour [Мука]	Byte	500	900	граммов
Baking_Soda [Пищевая сода]	Real	1, 0	0, 5	чайных ложек
Baking_Powder [Разрыхлитель]	Real	0	1, 0	чайных ложек
Salt [Соль]	Real	1, 0	0, 5	чайных ложек
Chocolate_Chips [Шоколадная крошка]	Real	450	0, 0	граммов
Lemon_Peel [Цедра лимона]	Real	0, 0	1, 0	столовых ложек
Cook_Time [Время приготовления]	Real	9, 0	10, 0	минут

## Использование Мастера рецептов

Рецепты и их определения создаются с помощью Мастера рецептов. Рецепты хранятся в модуле памяти. Рецепты и их определения могут быть введены непосредственно в Мастер рецептов. В дальнейшем изменения отдельных рецептов могут быть сделаны повторным запуском Мастера рецептов или программированием подпрограммы-команды RCPx\_WRITE.

Мастер рецептов создает конфигурацию рецептов, в состав которой входят:

- Таблица символов для каждого определения рецептов. Каждая таблица включает в себя имена символов, совпадающие с названиями полей рецепта. Эти символы определяют адреса в памяти переменных (V), необходимые для доступа к значениям, загруженным в данный момент в память. Каждая таблица содержит также символическую константу для ссылки на отдельные рецепты.
- Регистр блока данных для каждого определения рецептов. Этот регистр определяет начальные значения для адресов в памяти переменных, представленных в таблице символов.
- Подпрограмма-команда RCPx\_READ. Эта команда используется для считывания определенного рецепта из модуля памяти в память переменных.
- Подпрограмма-команда RCPx\_WRITE. Эта команда используется для записи значений рецепта из памяти переменных в модуль памяти.

## Определение рецептов

Для создания рецепта с помощью Мастера рецептов выберите команду меню **Tools > Recipe Wizard [Инструментальные средства > Мастер рецептов]**. Первое диалоговое окно является вводным окном, определяющим основные операции Мастера рецептов. Чтобы начать конфигурирование своих рецептов, щелкните на кнопке Next [Дальше].

Для создания определения рецептов действуйте следующим образом. См. рис. 13–2.

1. Введите имена полей для определения рецептов. Каждое имя становится символом в вашем проекте, как это определено выше.
2. Выберите тип данных из разворачивающегося списка.
3. Введите значение по умолчанию и комментарий для каждого имени. Все новые рецепты, задаваемые в рамках этого определения, будут начинаться с этих значений.
4. Щелкните на Next [Дальше], чтобы создавать и редактировать рецепты для этого определения рецептов.

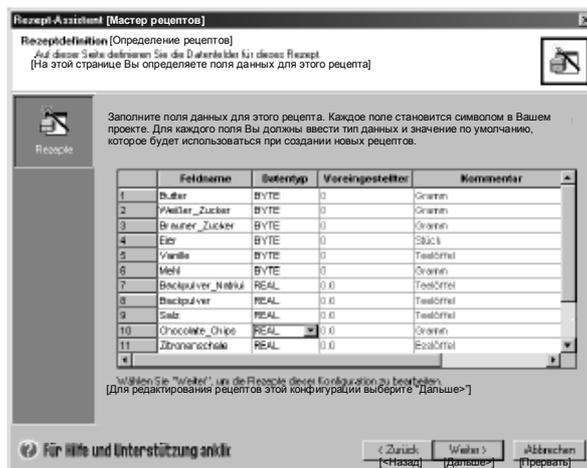


Рис. 13–2. Определение рецептов

**Примечание к рисунку:** текст таблицы на этом рисунке совпадает с текстом таблицы 13–1.

Для определения всех полей данных в рецепте используйте столько строк, сколько необходимо. Вы можете создать до четырех различных определений рецептов. Количество рецептов для каждого определения ограничено только доступным объемом памяти в вашем модуле памяти.

## Создание и редактирование рецептов

В диалоговом окне для создания и редактирования рецептов (Create and Edit Recipes) вы можете создавать отдельные рецепты и вводить для этих рецептов значения. Каждый редактируемый столбец представляет собой один рецепт.

Рецепты можно создавать нажатием кнопки New [Новый]. Каждый рецепт инициализируется значениями по умолчанию, указанными при создании определения рецептов.

Рецепты можно создавать с помощью контекстного меню, появляющегося при нажатии правой кнопки мыши, путем копирования и вставки существующих рецептов. Новые столбцы вставляются слева от текущего положения курсора, включая поле комментариев.

Каждому новому рецепту дается имя по умолчанию, которое включает в себя ссылку на определение рецептов и номер рецепта. Это имя имеет формат DEFx\_RCPy.

Для создания и редактирования рецептов действуйте следующим образом. См. рис. 13–3.

1. Щелкните на кнопке Next [Дальше], чтобы перейти в диалоговое окно Create and Edit Recipe [Создание и редактирование рецептов].
2. Выберите кнопку New [Новый], чтобы вставить при необходимости новый рецепт.
3. Переименуйте рецепт, если хотите, чтобы его имя отличалось от установленного по умолчанию.
4. Измените при необходимости значения набора данных каждого рецепта.
5. Щелкните на ОК.

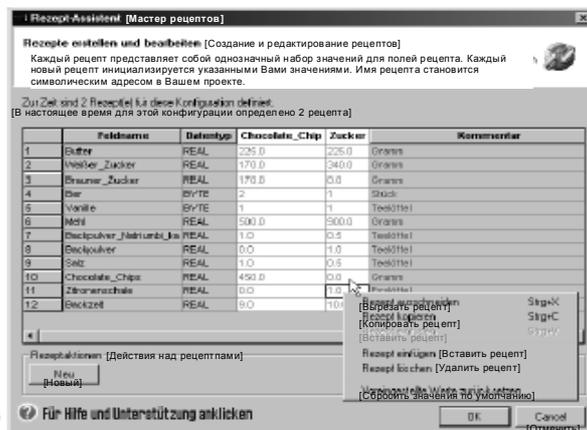


Рис. 13–3. Создание и редактирование рецептов

## Выделение памяти

В диалоговом окне для выделения памяти указывается начальный адрес области памяти переменных, где будет храниться рецепт, загруженный из модуля памяти. Вы можете ввести адрес в памяти переменных или предоставить Мастеру рецептов предложить адрес неиспользуемого блока памяти переменных нужного размера.

Для выделения памяти действуйте следующим образом. См. рис. 13–4.

1. Для выбора адреса в памяти переменных, где должен храниться рецепт, щелкните в окне и введите адрес.
2. Чтобы дать возможность Мастеру рецептов выбрать в памяти переменных неиспользуемый блок нужного размера, щелкните на кнопке Suggest Address [Предложить адрес].
3. Щелкните на кнопке Next [Дальше].



Рис. 13–4. Выделение памяти

## Компоненты проекта

Диалоговое окно компонентов проекта перечисляет различные компоненты, которые будут добавлены к вашему проекту. См. рис. 13–5.

Щелкните на Finish [Закончить], чтобы закончить работу с Мастером рецептов и добавить эти компоненты.

Каждой конфигурации рецепта может быть присвоено уникальное имя. Это имя будет отображаться в дереве проекта с отдельными конфигурациями, созданными Мастером. К концу этого имени добавляется определение рецепта (RCPx).

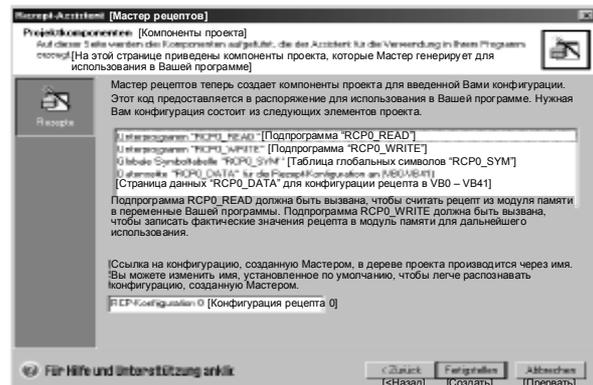


Рис. 13–5. Компоненты проекта

## Использование таблицы символов

Таблица символов создается для каждого определения рецептов. Каждая таблица определяет постоянные значения для отдельных рецептов. Эти символы могут использоваться в качестве параметров для команд RCPx\_READ и RCPx\_WRITE, чтобы задать желаемый рецепт. См. рис. 13–6.

Кроме того, в каждой таблице создаются символические имена для каждого поля рецепта. Вы можете использовать эти символы для доступа к значениям рецепта в памяти переменных.

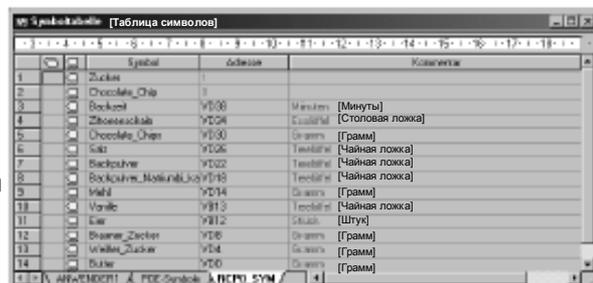


Рис. 13–6. Таблица символов

## Загрузка проектов с конфигурациями рецептов в ПЛК

Для загрузки проекта с конфигурацией рецептов в ПЛК действуйте следующим образом. См. рис. 13–7.

1. Выберите команду меню **File > Download [Файл > Загрузить]**.
2. В диалоговом окне в разделе Options [Возможные варианты] отметьте триггерные кнопки Program Block [Программный блок], Data Block [Блок данных] и [Рецепты].
3. Щелкните на кнопке Download [Загрузить].



Рис. 13–7. Загрузка проекта с конфигурацией рецептов

## Существующие конфигурации рецептов

Для редактирования существующих конфигураций рецептов действуйте следующим образом. См. рис. 13–8.

1. Откройте разворачивающееся поле списка конфигураций и выберите существующую конфигурацию рецепта.
2. Для удаления существующей конфигурации рецепта щелкните на кнопке Delete Configuration [Удалить конфигурацию].

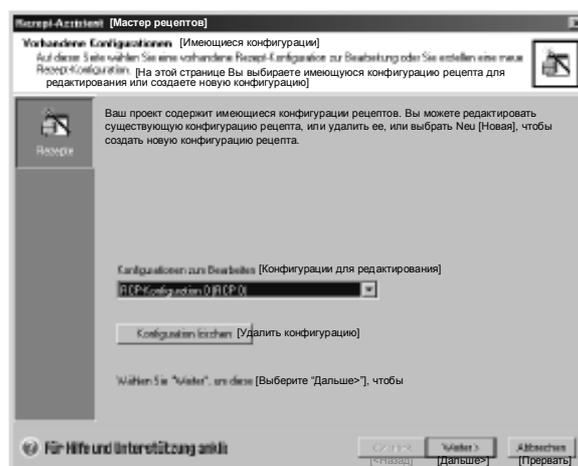


Рис. 13–8. Редактирование существующей конфигурации рецептов

## Команды, создаваемые Мастером рецептов

### Подпрограмма RCPx\_Read

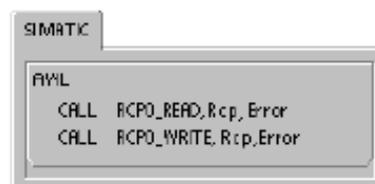
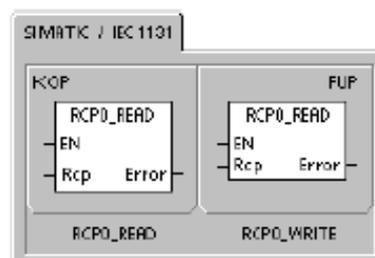
Подпрограмма RCPx\_READ создается Мастером рецептов и используется для считывания отдельного рецепта из блока памяти в указанную область в памяти переменных.

Символ x в команде RCPx\_READ соответствует определению рецептов, которое содержит рецепт, который вы хотите считать.

Вход EN активизирует исполнение команды при высоком уровне сигнала на этом входе.

Вход Rcp определяет рецепт, который будет загружен из модуля памяти

Выход Error возвращает результат исполнения этой команды. Определения кодов ошибок вы найдете в таблице 13–3.



### Подпрограмма RCPx\_Write

Подпрограмма RCPx\_WRITE создается Мастером рецептов и используется для замены рецепта в модуле памяти содержимым рецепта, содержащегося в памяти переменных.

Символ x в команде RCPx\_WRITE соответствует определению рецептов, которое содержит рецепт, который вы хотите заменить.

Вход EN активизирует исполнение команды при высоком уровне сигнала на этом входе.

Вход Rcp определяет рецепт, который будет заменен в модуле памяти.

Выход Error возвращает результат исполнения этой команды. Определения кодов ошибок вы найдете в таблице 13–3.

Таблица 13–2. Допустимые операнды для подпрограммы Recipe

Входы/выходы	Тип данных	Операнды
Rcp	Word	VW, IW, QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD, константа
Error	Byte	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Таблица 13–3. Коды ошибок для команд Recipe

Код ошибки	Описание
0	Нет ошибок
132	Не удался доступ к модулю памяти



#### Совет

ЭСППЗУ, используемый в модуле памяти, поддерживает ограниченное количество операций записи. Обычно это один миллион циклов записи. Когда это предел достигнут, ЭСПЗУ перестает работать надлежащим образом.

Обратите внимание на то, чтобы активизировать команду RCPx\_WRITE не в каждом цикле. Активизация этой команды в каждом цикле приведет к изнашиванию модуля памяти за относительно короткий интервал времени.

