

Алгоритм таймера пользовательской библиотеки.

Б.Г. Севастьянов, И.А. Жолобов

Программируемые логические контроллеры (далее ПЛК) получили широкое применение в промышленности. Важной частью современных контроллеров являются алгоритмы пользователя, реализуемые в виде библиотек. Качество создаваемой автоматической системы управления технологическим производством (далее АСУ ТП) во многом определяют эти библиотеки.

Отдельные алгоритмы в стандартных библиотеках не удовлетворяют пользователя или вовсе отсутствуют. К счастью, на сегодняшний момент практически в любой среде программирования промышленных контроллеров имеется возможность добавления пользователем своих алгоритмов. Разработка новых алгоритмов расширяет возможности и улучшает качество программ.

Рассмотрим недостатки алгоритма таймер, реализованного в библиотеке алгоритмов контроллера Ремиконт Р-130, контроллера Simatic S7-300 фирмы Siemens (German, Berlin), контроллера ПЛК-150 и контроллера фирмы Phoenix Contact (Germany, Blomberg).

Возможности, которые предоставляется программисту благодаря применению расширяемой пользовательской библиотеки можно продемонстрировать на примере модернизации алгоритма таймер.

ТАЙМЕР

Алгоритм таймера является одним из самых распространённых и используемых алгоритмов. Таймеры могут использоваться в программах управления запуском и остановкой двигателей, турбин, конвейеров и т.д.

Существующие таймеры из стандартных библиотек, без дополнительных надстроек, не могут управлять запуском и остановкой какого-либо двигателя в силу отсутствия тех или иных функций.

В каждой из рассматриваемых сред есть свой алгоритм таймера, в некоторых даже не один.

Библиотека алгоритмов контроллера Ремиконт Р-130 имеет алгоритм таймер (ТМР), структура которого приведена на рис. 1.

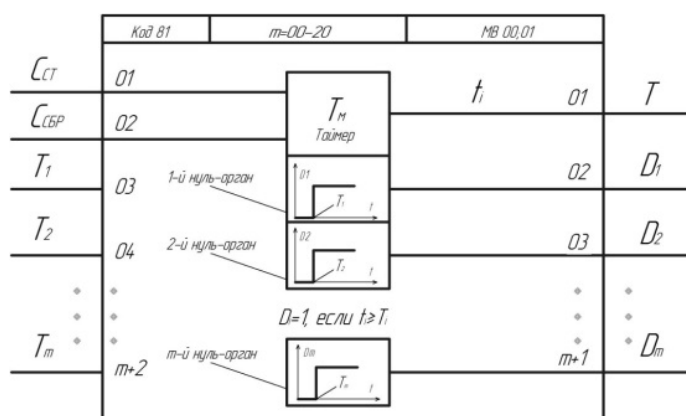


Рис. 1. Структура алгоритма таймер ТМР

Данный алгоритм имеет входы для остановки и сброса таймера, а также в его состав входит несколько нуль-органов, количество которых определяется модификатором.

Подробно данный алгоритм описан в [1,2]. Описание недостатков и пути их исправления описывается в [2]. Основным и главным недостатком данного таймера является то, что он начинает работу сразу, как только программа загружается в ОЗУ контроллера, т.е. данный таймер не нужно запускать.

В среде Step 7 существует несколько видов таймеров: – импульсный таймер (рис. 2), таймер с продлённым импульсом, таймер задержки включения, таймер задержки включения с запоминанием, таймер задержки выключения.

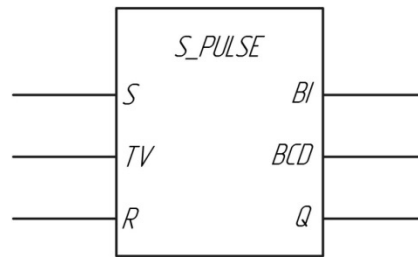


Рис. 2. Внешний вид FBD блока «Импульсный таймер» в среде Step 7,

где S – вход запуска, TV – спецификация длительности времени, R – вход сброса, BI – текущее значение времени в двоичном представлении, BCD – текущее значение времени в BCD-представлении, Q – состояние таймера.

Все эти таймеры имеют одинаковый принцип работы, и их различие заключается лишь в дополнительных функциях. Их подробное описание изложено в [3,4,5], а так же в файле-справке, поставляемом вместе с программным обеспечением. Из описания и временных диаграмм, представленных в [3,4,5] видно, что данный алгоритм является одновибратором, и его главным недостатком является невозможность приостановить работу таймера, а лишь остановить, поэтому, например, нельзя будет приостановить работу двигателя, а лишь остановить его.

В среде Codesys существует три вида таймеров: таймер TP (рис. 3), таймер с задержкой включения TON, таймер с задержкой выключения TOF.

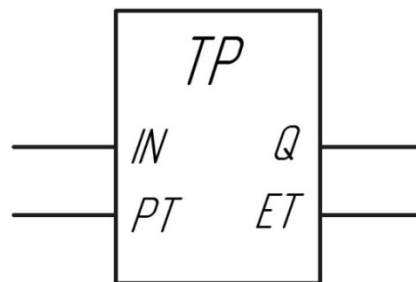


Рис. 3. Таймер TP в среде Codesys,

где IN – вход запуска таймера, PT – задание длительности работы, Q – состояние таймера, ET – время работы.

Подробное описание блоков TP, TON и TOF представлено в [6,7], и справочном файле среды Codesys. Все эти таймеры являются одновибраторами и имеют ограниченный функционал, в результате чего так же могут вызывать проблемы при использовании их в программах.

В среде PC Worx так же присутствуют три таймера их подробное описание представлено в справочном файле, поставляемом вместе с программным обеспечением. Поскольку PCWorx и Codesys полностью придерживаются стандарта МЭК 61131-3 (IEC 61131-3)[8], то и таймеры в средстве PC Worx имеет те же недостатки. Возможным путём решение вышеописанных проблем может быть реализация модернизированного универсального алгоритма таймера, входящего в состав пользовательской библиотеки. Этот таймер имеет расширенный функционал, в частности функцию памяти, поэтому его работу можно приостановить и запустить с момента остановки, а так же сбросить значения, например, для процессов которые повторяются по циклу. Предлагаемый вариант таймера представлен на рис. 4.

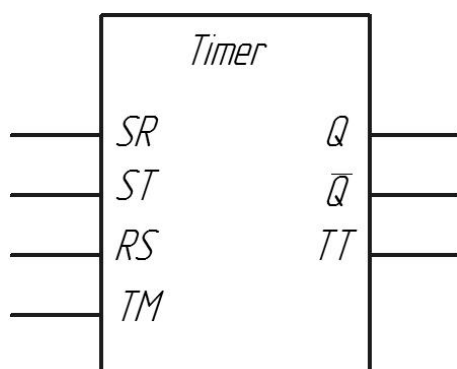


Рис. 4. Пример пользовательского алгоритма таймер в виде функционального блока.

Данный алгоблок имеет три булевых входа и один вход для переменной типа TIME. Булевые входы соответственно SR: START – переменная запускающая и останавливающая действие таймера, ST: STOP – переменная приостанавливающая действия таймера, ставящая таймер на паузу, но не останавливающая его полностью, и RS: RESET – переменная отвечающая за сброс таймера. Вход TM – задание времени работы таймера в секундах.

Таймер имеет три выхода: Q – сигнализирующий о его работе в виде переменной типа BOOL, nQ – НЕ Q, который является инверсией выхода Q для использования таймера в качестве таймера с задержкой включения и TT – демонстрирующий, текущее время работы таймера. Для наглядности алгоритм «Таймер» реализован на языке CFC и представлен на рис. 5.

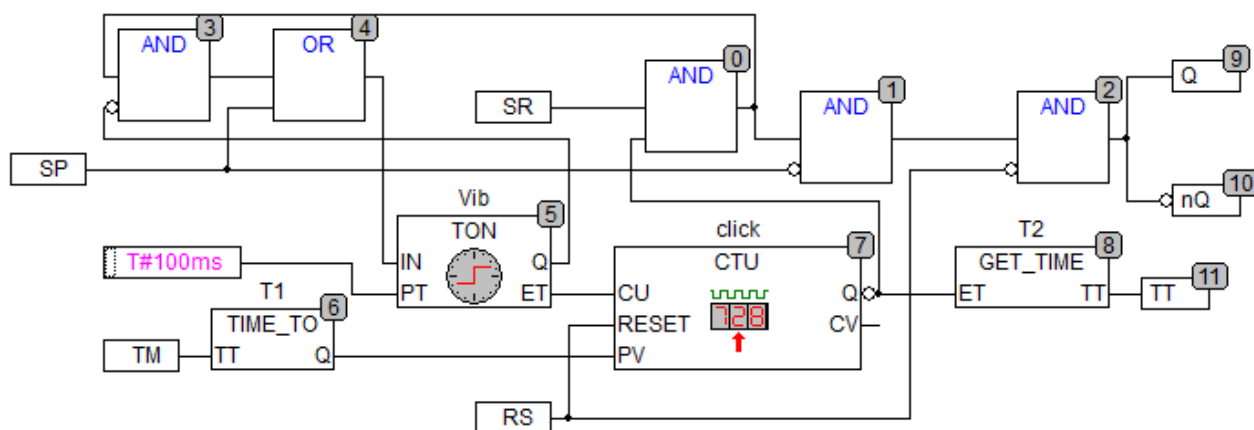


Рис. 5. Внутренняя реализация алгоритма таймера на языке CFC.

Для запуска таймера, необходимо, чтобы переменная SR (запуск таймера) приняла значение TRUE. Сигнал от неё поступает на алгоблок 0 (логическая операция AND), на этот блок так же поступает инверсно сигнал с выхода Q счетчика click, алгоблок 7. Если этот сигнал имеет значение FALSE, то алгоблок 0 на выходе имеет сигнал TRUE, и этот сигнал поступает на блок 1 (логическая операция AND). Так же этот сигнал поступает на блок 3 (логическая операция AND), на который помимо этого поступает инверсированный сигнал с выхода Q таймера (TON) стандартной библиотеки, который используется здесь как одновибратор для управления блоком счётчика 7. Если этот сигнал имеет значение FALSE, то вход блока 3 принимает значение TRUE и поступает на вход блока 4 (логическая операция OR), сигнал с которого примет значение TRUE, если переменная SP имеет значение FALSE, т.е. таймер не остановлен. Этот сигнал запускает работу блока TON (алгоблок 5). Выход Q блока TON соединён с входом CU блока

STU, это запускает и приостанавливает работу счетчика 7. На вход PT этого блока регулирует длительность его работы, его можно регулировать и настраивать в соответствии с длительностью цикла используемой модели ПЛК. На инверсированный вход блока 1 поступает сигнал от переменной SP (приостановка таймера), если этот сигнал принимает значение FALSE, то сигнал с блока 1 принимает значение TRUE и поступает на следующий блок под номером 3 (логическая операция AND). На последнем блоке проверяется значением переменной RS (сброс таймера), если она принимает значение FALSE, то сигнал от этого блока передается в переменную Q, которая является выходом таймера и сигнализирует о том, что таймера запущен. Одновременно инверсированный сигнал поступает в переменную НЕ Q. Сигнал от переменной RS так же поступает на вход RESET счетчика 7, это реализует функцию сброса таймера. Для управления временем работы таймера используется переменная TM, сигнал с которой поступает на блок 6, который является блоком преобразования переменной типа TIME в переменную типа UINT, это преобразование необходимо для работы счетчика 7. После преобразования значения поступают на вход PV счётчика 7. Результат работы счётчика, с выхода CV поступает в блок 8, который является обратным блоку 6, в нём переменная типа UINT преобразуется в переменную типа TIME и уже с него значения поступают в выходную переменную TT, с помощью которой можно следить за тем, сколько времени проработал таймер. Поскольку в программном средстве Codesys не существует функции непосредственного преобразования переменной времени TIME в переменную типа INT или UINT для реализации алгоритма таймера так же были составлены блоки TIME_TO (листинг 1) и GET_TIME (листинг 2) для прямого и обратного преобразования. Эти блоки так же являются составными частями пользовательской библиотеки и могут быть использованы отдельно в других программах. Реализация функционального блока TIME_TO на языке ST:

Заголовок и объявление переменных:

```
FUNCTION_BLOCK TIME_TO
VAR_INPUT
    TT: TIME;
END_VAR
VAR_OUTPUT
    Q: UINT;
END_VAR
VAR
    F: UINT;
END_VAR
```

Тело функции:

```
F:=TIME_TO_UINT(TT);
Q:=F/1000;
```

Реализация функционального блока GET_TIME на языке ST:

Заголовок и объявление переменных:

```
FUNCTION_BLOCK GET_TIME
VAR_INPUT
    ET: UINT;
END_VAR
VAR_OUTPUT
    TT: TIME;
END_VAR
VAR
    F: UINT;
END_VAR
```

Тело функции:

```
F:=ET*1000;
TT:=UINT_TO_TIME(F);
```

К примеру, такой модифицированный алгоритм таймера может быть использован при создании систем автоматического управления описанных в [9,10].

Пользовательская библиотека может распространяться в свободном доступе, и дополняться любым пользователем, в этом и ценность данной библиотеки, она может впитывать опыт большого числа пользователей.

Заключение

Представленный таймер имеет расширенные функции, поэтому может использоваться в системах контроля и управления реальными объектами, что повысит надёжность разрабатываемых АСУ ТП. Данный алгоритм целесообразно включать в пользовательские библиотеки промышленных контроллеров. Это позволит создавать более гибкие программно-аппаратные системы контроля и управления и повысить эффективность существующих.

Список использованной литературы:

1. Микропроцессорный контроллер Ремиконт Р – 130. – М.: НИИТЕПЛОПРИБОР, 1990. – 330с.
2. Севастьянов Б.Г. Реализация дискретных систем управления на контроллерах. – Учебное пособие. Волгоград, 2011.-230с.
3. Berger Hans, Automating with STEP 7 in LAD and FBD, 2012 г – 451с.
4. Berger Hans, Automating with STEP 7 in STL and SCL, 2012г – 553с.
5. Романов В.П. Основы языка программирования STEP7 и базового программного обеспечения промышленных контроллеров Siemens: Учебно-методическое пособие. - Новокузнецк: ФГОУ СПО "Кузнецкий индустриальный техникум", 2009. - 45 с.
6. Петров И.В., "Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования" Под ред. проф. В.П. Дьяконова. – М.:СОЛОН-Пресс, 2004. - 256с.
7. Руководство пользователя по программированию ПЛК в CoDeSys 2.3, 2006г. – 453с. Режим доступа: http://www.kipshop.ru/CoDeSys/steps/codesys_v23_ru.pdf (доступ свободный) – Яз. Рус.
8. Karl-Heinz John, Michael Tiegelkamp., IEC 61131-3: Programming Industrial Automation Systems. – Springer, 2001г – 376с.
9. С.А. Панкратов. Анализ эффективности автоматизации процесса заготовки вторсырья на производственном предприятии. «Инженерный вестник Дона», 2012. №4. – Режим доступа: <http://ivdon.ru/magazine/archive/n4p1y2012/1150> (доступ свободный) – Загл. с экрана. – Яз. Рус.
10. А.И. Исмагулова. Системы автоматического регулирования динамического уровня жидкости в скважине в штанговых глубинно-

насосных установках. «Инженерный вестник Дона», 2012. №4. – Режим доступа: <http://ivdon.ru/magazine/archive/n4p1y2012/1325> (доступ свободный) – Загл. с экрана. – Яз. Рус.