

# Разработка приложений для ПЛК на основе принципов стандарта ARINC 653

А.Н. Онин<sup>1</sup>

<sup>1</sup>НИЦ «Курчатовский институт» – НИИСИ, Москва, Россия, alexii@niisi.ras.ru.

**Аннотация.** В статье рассматривается использование принципов стандарта ARINC 653 как основы для разработки приложений на языках стандарта МЭК 61131-3 для исполнения на программируемом логическом контроллере (ПЛК), функционирующем под управлением ОСРВ Багет 3.6. Рассматривается схема использования многопоточного подхода реализации прикладных задач, когда каждая задача функционирует со своим периодом и приоритетом, независимо от других прикладных задач и программ-драйверов обслуживающих устройства ввода-вывода. В статье также рассматриваются принципы формирования диагностических сообщений.

**Ключевые слова:** ПЛК, ОС РВ Багет 3.6, МЭК 61131-3, ARINC 653, система диагностики

## 1. Введение

В статье рассматриваются методы разработки приложений для программируемых логических контроллеров (ПЛК) [1], написанных на языках стандарта МЭК 61131-3 [2, 6] и предназначенных для исполнения в среде, поддерживающей стандарт ARINC 653 [3]. В статье приведено описание принципов функционирования задач на ПЛК, поддерживающих стандарт ARINC 653, приведены основные характеристики и отличия от аналогов.

Рассматриваемые в статье принципы функционирования задач на ПЛК (далее будем называть это схемой функционирования) отличается от типовой схемы, в которой ПЛК имеет один процессор и выполняет несколько задач псевдопараллельно, последовательными порциями [4].

Типовая схема функционирования задач на ПЛК основана на следующих принципах:

- организуется один поток управления (диспетчер);
- задаётся длительность основного цикла (такт). Такт представляет собой целое количество миллисекунд (мс);
- каждая задача имеет период. Период задачи должен быть кратен такту;
- длительность такта зависит от периодов функционирования прикладных задач, и равен наибольшему кратному делителю периодов всех задач (ниже будет приведён пример);
- диспетчер в начале цикла получает входные данные от устройств ввода и записывает их во

входные переменные задач;

- диспетчер определяет, какие задачи должны выполняться на текущем такте, и запускает их;
- диспетчер в конце цикла извлекает данные из выходных переменных задач и отправляет их в устройства вывода.

К особенностям типовой схемы можно отнести следующие:

- длительность такта зависит от периодов функционирования прикладных задач, и может оказаться намного меньше минимального периода задач. Приведём пример: на ПЛК должны функционировать три задачи с периодами 10, 15 и 32 мс. При этом наибольший кратный делитель периодов всех задач будет равен 1 мс, таким образом длительность такта будет равна 1 мс, что в 10 раз меньше периода задачи с минимальным периодом. В результате диспетчер будет в 10 раз чаще обращаться к устройствам ввода-вывода, чем это требуется для решения задач. Кроме того, задача может обрабатывать данные более 1 мс, а это больше периода такта, что приведёт к общему замедлению обработки данных на ПЛК;
- поскольку все задачи, а также программы обращения к устройствам ввода-вывода, выполняются в одном потоке, то сбой в одной из задач, или зависание на приёме данных от входного устройства, может привести к неработоспособности всех задач на ПЛК.

Предлагаемая схема функционирования задач на ПЛК с использованием стандарта ARINC 653 имеет преимущество по сравнению с типовой схемой.

## 2. Описание схемы функционирования

ПЛК функционирует циклически, выполняя в основном цикле предписанные для цикла действия. Цикл начинается со сбора данных с модулей ввода, затем исполняется прикладная программа ПЛК и заканчивается цикл выводом данных в устройства вывода. Во время сбора данных с модулей ввода, и отправки данных в устройства вывода, прикладная программа находится в состоянии ожидания завершения операций ввода-вывода. Прикладная программа обращается только к своим и глобальным переменным, и не взаимодействует с внешним оборудованием, устройствами ввода и вывода. В соответствии с перечисленными выше положениями прикладная программа может функционировать в операционной системе, построенной на принципах стандарта ARINC 653.

### 2.1. Использование принципов стандарта ARINC 653

В качестве операционной системы, функционирующей на ПЛК, рассмотрим ОСРВ Baget 3.6 [5], которая поддерживает выполнение прикладных программ в изолированных ARINC-partition (ARINC-процессах) [3, 5]. Связь прикладных программ (задач), функционирующих в ARINC-процессах, с внешним оборудованием происходит через системный POSIX-partition (POSIX-процесс, ядро операционной системы) [5]. Для передачи значений каждой переменной задачи, связанной с входными или выходными данными устройства, между ARINC-процессом и POSIX-процессом организуется Sampling-канал [3, 5]. Если переменная задачи связана с типом данных MEMORY [6], то есть является как входной, так и выходной, тогда для такой переменной организуются два разнонаправленных Sampling-канала (входной и выходной). Далее по тексту множество Sampling-каналов будем называть буферной зоной.

Каждая задача, выполняющая прикладные программы, написанные на языках стандарта МЭК 61131-3 [6], представляет собой отдельный ARINC-process (поток управления) [3, 5], под управлением которого задача выполняется циклически, с заданным периодом и приоритетом. При этом периоды выполнения задач не обязаны быть кратны друг другу.

В соответствии со стандартом ARINC 653 в функционирующей системе должен быть установлен основной цикл выполнения всех потоков управления. Этот цикл называется «Major Frame» [3, 5]. Длительность периода Major Frame может быть любой, при этом необходимо учитывать следующие обстоятельства:

- длительность Major Frame должна быть больше времени решения самой длительной задачи;

- длительность Major Frame не должна быть слишком большой, поскольку необходимо обеспечить в ПЛК поддержку возможности безударного переключения:

- а) - на резервный модуль, в случае обнаружения сбоя программы или оборудования на основном модуле;

- б) - на обновлённую программу, при «горячем» обновлении прикладного программного обеспечения на ПЛК.

Периоды циклических задач не обязаны быть кратными длительности периода Major Frame. При этом, по окончании периода Major Frame все задачи должны завершить свои циклы обработки данных. В случае, если задача не успеет завершить обработку данных по окончании периода Major Frame, то произойдёт исключительная ситуация, обработчик которой отправит соответствующее сообщение в систему диагностики, и задача продолжит своё выполнение в следующем периоде Major Frame, в соответствии со своим приоритетом [5].

### 2.2. Синхронизация управления

Функционирование ПЛК состоит в выполнении циклических и разовых задач. Циклические задачи выполняются периодически. Разовые задачи выполняются по событию. Событие – изменение по переднему фронту заданной пользователем переменной типа BOOL. Переменная должна устанавливаться в значение TRUE в контексте другой прикладной задачи.

Каждая задача обладает приоритетом. Приоритеты задач устанавливаются в соответствии со стандартом МЭК 61131-3, в диапазоне от 0 до 31. Задачи, у которых приоритет меньше, выполняются раньше.

Циклическая задача обладает периодом. Период — это промежуток времени, через который задача должна быть запущена снова. Операционная система планирует выполнение периодических задач в соответствии с заданной конфигурацией периодов потоков управления [5].

Прикладная программа, согласно МЭК 61131-3, не взаимодействует с внешними устройствами, все входные и выходные данные сохраняются в переменных программы. В начале цикла входные данные, полученные от входных устройств, загружаются из POSIX-процесса во входные переменные программы. В конце цикла значения выходных переменных программы отправляются в POSIX-процесс для формирования выходных данных, которые отправляются в выходные устройства. Загрузка и выгрузка данных между ARINC-процессом и

POSIX-процессом осуществляется через буферную зону.

В POSIX-процессе для каждого обслуживаемого устройства создаётся свой циклический поток управления с заданными периодом и приоритетом. Поток управления на каждом цикле выполняет следующие действия:

- выполняет инициализацию устройства в случае, если устройство не было инициализировано;
- получает от устройства входные данные;
- отправляет полученные входные данные через буферную зону задачам в раздел ARINC;
- читает выходные данные задач, сформированные в буферной зоне;
- отправляет выходные данные в устройство.

Потоки управления обслуживающие устройства работают независимо и не синхронизированы между собой.

В случае обнаружения ошибки при обращении к устройству ввода-вывода будет установлен признак, что устройство неработоспособно, и на следующем цикле потока управления обслуживания устройства будет выполнена попытка инициализировать устройство заново. Таким образом, осуществляется программная поддержка резервирования модулей ввода-вывода.

## 2.3 Глобальные переменные МЭК-программ

Задачи вызывают программы, которые могут обращаться к глобальным переменным. Каждая задача выполняется в своём потоке управления, таким образом может образоваться конкурентный доступ к глобальным переменным. Чтобы избежать одновременного обращения к глобальной переменной из двух или более задач, в задачах используется преимущественная блокировка [3]. Это означает, что задача выполняет свой цикл непрерывно, за счёт того, что в начале цикла устанавливается блокировка, и в конце цикла блокировка снимается. Для разовых задач блокировка устанавливается на всё время выполнения обработки данных задачи.

## 2.4 Принципы формирования диагностических сообщений

Система диагностики может быть реализована в виде библиотеки функций, которые будут вызывать прикладные программы, драйверы обслуживающие устройства ввода-вывода, программы синхронизации и системные компоненты.

Система диагностики поддерживает несколько типов диагностических сообщений:

- неисправимый сбой. При этом система диагностики автоматически запускает процедуру перехода функционирования задач на резервный

модуль;

- критическая ошибка. При этом требуется немедленное исправление ошибки или необходимо выполнить переход на резервный модуль;

- ошибка. При этом считается, что программа самостоятельно исправила ошибку и возможно продолжение функционирования задачи;

- предупреждение. Программа сообщает о нештатной ситуации, которая не приводит к сбою функционирования;

- уведомление. Программа сообщает о важном событии;

- информирование. Такие сообщения формируются программой при штатном функционировании задач;

- статистика. Программа сообщает о статистических показателях, накопленных во время функционирования задач;

- отладка. Этот тип предназначен для вывода отладочных сообщений.

В результате вызова функций системы диагностики функции формируют текстовые сообщения, которые сохраняются в буфере в оперативной памяти. Кроме сохранения сообщений в буфере система диагностики отправляет сообщения в консоль операционной системы [5].

Система диагностики обеспечивает передачу сообщений из ARINC-процесса в POSIX-процесс по каналам с очередью сообщений [3, 5]. В POSIX-процессе сообщения, полученные из ARINC-процессов и POSIX-процесса, накапливаются в кольцевом буфере.

С целью долговременного хранения сообщений в системе диагностики функционирует поток управления, который копирует сообщения из кольцевого буфера в файловую систему. Файлы сообщений организованы следующим образом:

- название файла содержит дату и время первого сообщения, которое сохраняется в файле;

- название файла содержит тип диагностического сообщения;

- в файле сохраняются множество сообщений одного типа. Таким образом, происходит естественная фильтрация сообщений по типам, что упрощает обнаружение критически важных сообщений;

- сообщения накапливаются в файле пока размер файла не превышает 1 Мбайт. Небольшой размер файла позволяет быстро загружать его из ПЛК на инженерную станцию;

Новый файл с диагностическими сообщениями создаётся в следующих случаях:

- произошло превышение размера файла в 1 Мбайт;

- наступили новые сутки. Это способствует быстрому поиску сообщений по дате;

- произошёл перезапуск задач на ПЛК. Это

способствует быстрому поиску сообщений, которые возникали в определённый запуск ПЛК.

### 3. Заключение

Рассмотренный подход к разработке приложений для ПЛК на основе принципов стандарта ARINC 653 соответствует принципам стандарта МЭК 61131-3, и обладает следующими положительными свойствами:

- многопоточная реализация прикладных задач и программ обслуживания устройств ввода-вывода повышает надёжность функционирования ПЛК за счёт независимого выполнения прикладных задач от программ обслуживающих устройства. В случае сбоя одного или нескольких компонент (задачи, устройства), другие компоненты продолжают своё функционирование, а система диагностики сообщит о произошедших сбоях;

- использование системы диагностики позволяет автоматизировать сбор диагностических и статистических данных с сохранением их в файлах файловой системы, а также автоматизировать реакцию на критически важные события.

- в прикладных задачах используется преимущественная блокировка, которая обладает положительными и отрицательными свой-

ствами. Отрицательным свойством преимущественной блокировки является следующее: в то время, пока одна задача не завершит свой цикл обработки данных, другие задачи, в том числе и более приоритетные, будут находиться в состоянии ожидания. Положительным свойством преимущественной блокировки является то, что не будет создаваться конкурентный доступ к глобальным переменным, а также будет исключена возможность прерывания процедуры обработки данных выполняемой задачи. На практике преимущественная блокировка реализована как опциональная возможность.

Рассмотренный подход к разработке приложений для ПЛК на основе принципов стандарта ARINC 653 был успешно реализован в решении задач на одном из образцов ПЛК-2.

Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме «Создание и реализация доверенных систем искусственного интеллекта, основанных на новых математических и алгоритмических методах, моделях быстрых вычислений, реализуемых на отечественных вычислительных системах» (FNEF-2024-0001)».

## Programming IEC 61131-3 Tasks for PLC Running under RTOS Baget 3.6 Based on ARINC 653 Standard

A. Onin

**Abstract.** The article discusses the use of the ARINC 653 standard principles as a basis for developing applications in the languages of the IEC 61131-3 standard for execution on a programmable logic controller (PLC) running under the RTOS Baget 3.6. The article discusses the scheme for using a multi-threaded approach to implementing application tasks, when each task operates with its own period and priority, independently of other application tasks and driver programs servicing input-output devices. The article also discusses the principles of generating diagnostic messages.

**Keywords:** PLC, OS RV Baget 3.6, IEC 61131-3, ARINC 653, diagnostic system

### Литература

1. Шишов О.В. Программируемые логические контроллеры: учебник / О.В. Шишов. — Москва : ИНФРА-М, 2024. — 461 с. — (Высшее образование). — DOI 10.12737/2030899. - ISBN 978-5-16-018581-1.
2. ГОСТ Р МЭК 61131-3-2016.
3. ARINC 653 API and its application – An insight into Avionics System Case Study. Ananda C.M., Sabitha Nair, and Mainak G.H. CSIR-National Aerospace Laboratories, Bangalore–560 017, India. Defence Science Journal, Vol. 63, No. 2, March 2013, pp. 223-229.

4. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / – М.: СОЛОН-Пресс, 2004.
5. Годунов А.Н., Солдатов В.А. Спецификация ARINC 653 и её реализация в операционной системе реального времени Багет 3, НИИСИ РАН г. Москва, 2015 № 5, с. 3-17.
6. Романов С. Изучаем Structured Text МЭК 61131-3. ISBN: 978-1-64199-106-3 2020.