

# МОДУЛЬ ОТЛАДКИ МЕХАНИЗМОВ ЛОГИЧЕСКОЙ ЗАДАЧИ В ЯДРЕ СИСТЕМЫ ЧПУ

Доцент, к.т.н. Н.В. Козак  
Преп., магистр Р.А. Абдуллаев

## МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СТАНКИН»

Как правило, логическая задача в системах ЧПУ реализуется с помощью программируемых контроллеров [1]. Однако не всегда у разработчиков есть возможность работы с реальным ПЛК. Выполнение некоторых вспомогательных М-функций требует наличия подключенного контроллера электроавтоматики. В противном случае его отсутствие может привести к неполной или неправильной работе программных модулей ядра системы ЧПУ<sup>1</sup>.

В другом аспекте, при обучении операторов системы ЧПУ использование для каждого учебного места программируемого контроллера является дорогостоящим и не всегда целесообразным. Для выхода из этой ситуации можно использовать программный эмулятор контроллера электроавтоматики.

Можно выделить следующие направления использования эмулятора ПЛК:

1. Отладка механизмов работы с ПЛК в ядре системы ЧПУ.
2. Эмуляция работы функций электроавтоматики системы ЧПУ в учебных целях.
3. Демонстрация возможностей системы ЧПУ без аппаратных средств электроавтоматики.

Целью работы было поставлено разработка концепции и программного модуля – эмулятора контроллера электроавтоматики, реализующего необходимые функции для отладки механизмов логической задачи ядра системы ЧПУ.

Для того чтобы ядро системы ЧПУ не зависело от работы эмулятора, оно реализуется в виде отдельного процесса. По этой причине возникает задача организации межпроцессорного взаимодействия. Для этого требуется работа с механизмом разделяемой памяти, что в свою очередь влечет за собой неизбежное применение именованных системных объектов синхронизации (мьютексов или семафоров).

---

<sup>1</sup> Работа выполнена по Госконтракту № 16.740.11.0267 на проведение НИР в рамках ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009-2013 гг.



Рис. 1. Общее представление механизма взаимодействия с эмулятором ПЛК

Для реализации обмена данными между программируемым контроллером и ядром системы ЧПУ используется последовательный порт (Рис. 1). В случае работы ядра с эмулятором ПЛК в качестве последовательного порта выступает разделяемая память. Запросы от ядра системы ЧПУ передаются в эмулятор программируемого контроллера через разделяемую память, которая является связующим звеном между двумя различными процессами.

Эмулятор контроллера реализован в виде отдельной библиотеки, которая может быть подключена как к консольному, так и к графическому интерфейсу. Для реализации именованной разделяемой памяти используется оболочка CSharedMemoryWrapper, в которой осуществляется непосредственное создание и открытие системных объектов (Рис. 3).

Так же в объекте этого класса происходит распределение разделяемой памяти на участки запроса и ответа эмулятора контроллера электроавтоматики (Рис. 2). Первые четыре байта отведены под состояние запроса и ответа эмулятора программируемого контроллера. И область запроса, и область ответа начинаются с размера в байтах содержимого соответствующих полей.

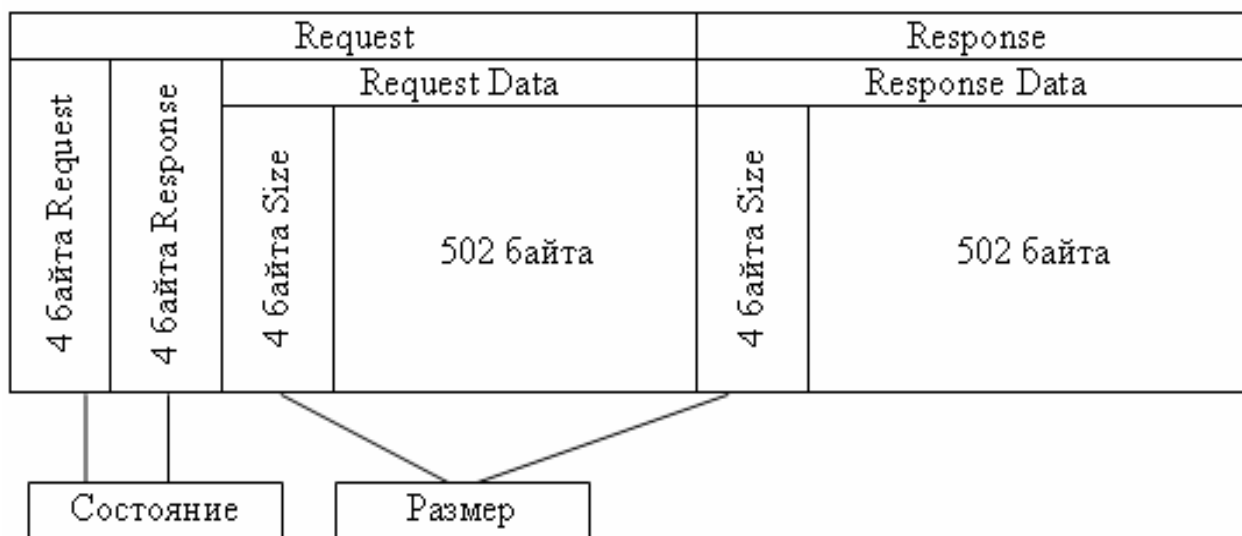


Рис. 2 Распределение разделяемой памяти

От базовой оболочки именуемой разделяемой памяти унаследованы два класса. Один из них представляет собой объект master-устройства, то есть он посылает запросы и считывает ответы. Исходя из этой особенности объект класса `CMasterSharedMemoryWrapper` обладает функциональной возможностью записывать данные в область запроса и считывать данные из области ответа. И напротив, объект класса `CSlaveSharedMemroyWrapper` считывает запросы, обрабатывает их и записывает ответ.

В ядре системы ЧПУ класс порта, который работает с именуемой разделяемой памятью, унаследован от класса последовательного порта `SerialPort` и переопределяет соответствующие методы чтения и записи данных. Внутри этих методов вызываются функции записи запроса и считывания ответа класса `CMasterSharedWrapper`. Класс `PortManagerExt` переопределен таким образом, что при возвращении указателя на класс последовательного порта возвращается объект унаследованного класса для работы с именованной разделяемой памятью `CSharedMemoryPort`.

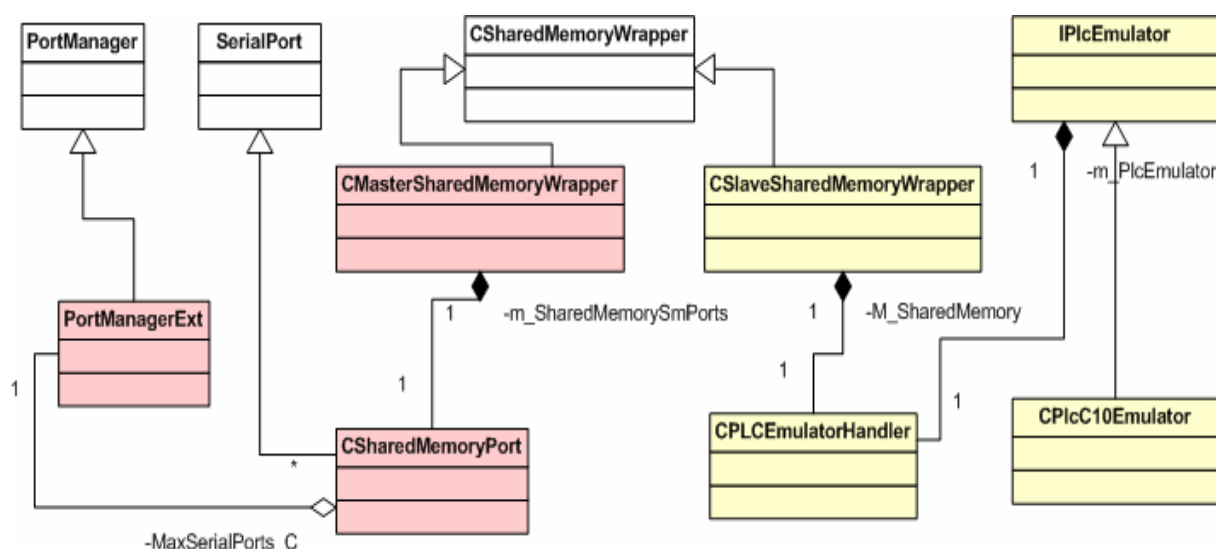


Рис. 3 Структура классов для работы с разделяемой памятью и объекты эмулятора ПЛК

Ниже показан пример обработки запроса от master-устройства (Рис. 4), в большинстве случаев которым является ядро системы ЧПУ. Переопределенный метод `Write` (вызов 1) записывает запрос в разделяемую память (вызов 2). Межпроцессорная синхронизация заключается в том, что после записи данных в разделяемую память (вызов 3) именованный межпроцессорный семафор, который является частью базового класса оболочки `CSharedMemroyWrapper`, а значит содержится и в master- и в slave-устройствах, освобождается (вызов 4).

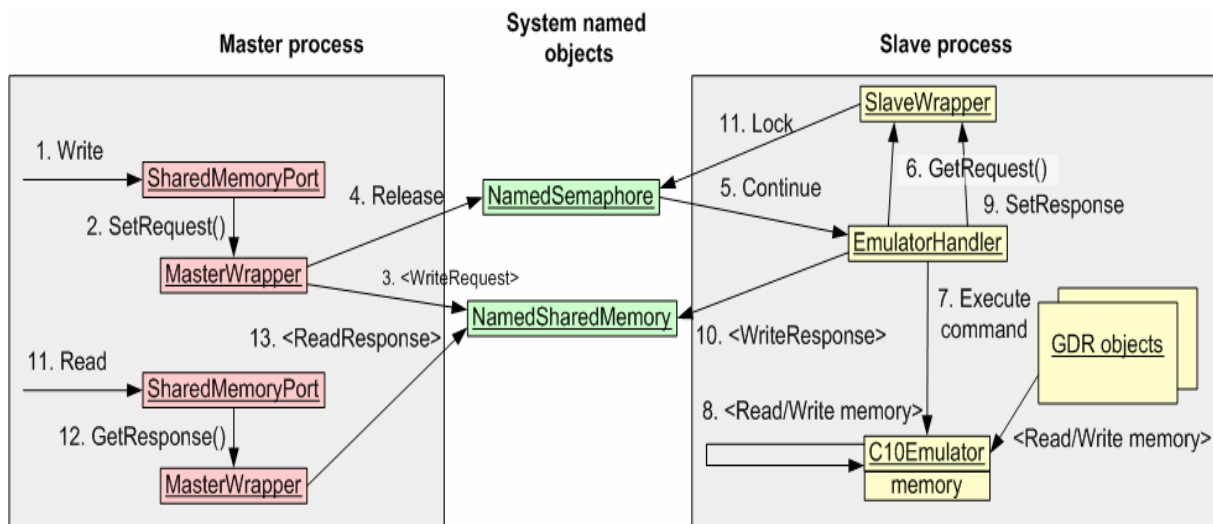


Рис. 4 Выполнение операции записи в память эмулятора ПЛК

Slave-устройство, которым является эмулятор программируемого контроллера, после освобождения именованного семафора возобновляет свою работу (вызов 5). Класс-обработчик запросов эмулятора EmulatorHandler считывает запрос (вызов 6) и передает его на исполнение (вызов 7) через интерфейс IPlcEmulator непосредственно эмулятору выбранного контроллера. При выполнении запроса происходит запись или чтение данных в или из памяти эмулируемого контроллера (вызов 8). Для удобства отладки работы вспомогательных M-команд, обрабатываемых программируемым контроллером, графический интерфейс показывает состояние памяти эмулятора ПЛК [2].

После обработки запроса данные от эмулятора контроллера электроавтоматики помещаются в разделяемую память в области ответа (вызов 9) через оболочку CSlaveSharedMemoryWrapper (вызов 8). После этого именованный семафор, используемый для межпроцессорной синхронизации, блокируется (вызов 11) до поступления нового запроса. Записанный ответ от эмулятора ПЛК считывается в переопределенном методе Read с помощью вызова метода GetResponse.

Эмулятор ПЛК является отдельной динамически подключаемой библиотекой, реализованной на высокоуровневом языке C++. На уровне программных компонентов эмулятор ПЛК взаимодействует с библиотекой KernelPortManager. В ней определены классы для работы с последовательными портами и их менеджером. Для переориентации работы библиотеки KernelPortManager на взаимодействие с эмулятором программируемого контроллера реализованы классы CSharedMemoryPort и PortManagerExt (Рис. 3), унаследованные от соответствующих базовых классов.

В результате проведенных исследований и разработки были получены следующие результаты:

1. Разработана концепция построения программного модуля – эмулятора ПЛК - для логической задачи ядра системы ЧПУ в виде отдельного приложения.
2. Предложен оригинальный механизм взаимодействия компонентов ядра системы ЧПУ с эмулятором ПЛК посредством разделяемой памяти и ее представления в виде виртуального порта (SharedMemoryPort).
3. Разработаны структура и алгоритмы взаимодействия компонентов логической задачи системы ЧПУ, эмулятора ПЛК через механизм виртуального порта.

#### **Библиографический список**

1. Сосонкин В.Л. Мартинов Г.М. Системы числового программного управления. - М. : Логос, 2005. - стр. 296 с..
2. Мартинов Г. М., Козак Н. В. Декомпозиция и синтез программных компонентов электроавтоматики // Приборы и системы. Управление, контроль, диагностика. 2006. №12. С. 4-11.
3. Козак Н.В. Основы интеграции компонентов электроавтоматики на примере графического редактора 3D-сцены визуализации // Труды Международной научно-технической конференции « Информационные средства и технологии». 16-18 октября 2007 г. в 3-х т. Т.3. – М.: МЭИ, 2007. 220 с. стр. 132-135.