

## ОТЗЫВ

официального оппонента на диссертацию Розова Андрея Сергеевича

«РАЗРАБОТКА МОДЕЛЕЙ И МЕТОДОВ  
ПРОЦЕСС-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ ДЛЯ  
ОТКРЫТЫХ МИКРОКОНТРОЛЛЕРНЫХ ПЛАТФОРМ»,

представленную на соискание ученой степени

кандидата технических наук, специальность 05.13.18 -

«Математическое моделирование, численные методы и комплексы программ»

### **Актуальность темы:**

В конце 2000-х/начале 2010-х на рынке появился широкий ассортимент компактных недорогих микроконтроллеров (самым распространённым из которых сейчас является семейство Arduino). Эти микроконтроллеры снабжены разнообразными интерфейсами ввода/вывода (ЦАП, АЦП, GPIO) и коммуникационными (Ethernet, USB, RS232/485, I2C), расположенными либо на основной плате, либо на дочерних (в случае Arduino именуемых «шилдами»). Стоимость такого комплекта оборудования начинается от нескольких десятков долларов, что позволяет использовать его для автоматизации даже небольших и недорогих установок (ранее комплект оборудования для автоматизации обошёлся бы во многие тысячи долларов).

В то же время на первый план выходят вопросы создания управляющего программного обеспечения: снижение трудозатрат на его разработку (становящихся особо заметными на фоне низкой цены управляющей электроники) и повышение надёжности с одновременным снижением «порога вхождения» для разработчиков.

Таким образом, рассмотренная в диссертационной работе А.С. Розова проблема создания новых методов разработки управляющего ПО для микроконтроллеров является весьма *актуальной* и имеет большое практическое значение для развития методологии применения встраиваемых систем.

### **Новизна результатов:**

В диссертации предложен новый метод разработки программного обеспечения для встраиваемых систем, заключающийся в представлении программы в виде набора гиперпроцессов (расширяющих концепцию конечных автоматов), с представлением внешних прерываний в виде т. н. «источников активации». Создана математическая модель работы такого

ПО, разработан синтаксис процесс-ориентированного языка IndustrialC, совмещающего эту модель с элементами синтаксиса широко распространённого языка C, и реализован компилятор языка IndustrialC для семейства микроконтроллеров семейства AVR.

**Достоверность и обоснованность результатов:**

Полученные А.С. Розовым результаты подтверждаются практическим внедрением (описанным в диссертации и имеющим акты внедрения), публикациями в научных рецензируемых изданиях и докладами на конференциях.

Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы из 74 наименований и 2 приложений, и изложена на 128 страницах.

Во **Введении** (озаглавленном «Общая характеристика работы») обосновывается актуальность диссертации, описываются объект и предмет исследования, формулируются задачи, выдвигается гипотеза о рациональности использования процесс-ориентированного подхода, рассматриваются вопросы новизны и практической значимости полученных результатов.

**Первая глава** содержит описание особенностей применения микроконтроллеров по сравнению с компьютерами общего назначения, а также обзор широко распространённых в настоящее время подходов к программированию микроконтроллеров.

Во **второй главе** предлагается методика построения управляющего ПО встраиваемых систем на основе процесс-ориентированной модели, являющейся глубокой переработкой модели, принятой в языке Reflex, под специфику микроконтроллеров, имеющих ограниченные аппаратные ресурсы. Важной особенностью предложенной модели является возможность отработки реакции на внешние события – такие, как прерывания и таймауты – в той же парадигме, что и основная программа, в виде интегральной её части.

В **третьей главе** описывается созданный в качестве реализации процесс-ориентированной модели язык IndustrialC, сочетающий базовый синтаксис языка C с конструкциями описания процессов (включая состояния и таймауты), заимствованными из языка Reflex.

**Четвёртая глава** посвящена описанию разработанному транслятору с языка IndustrialC. Транслятор реализован в виде транскомпилятора с языка IndustrialC в язык C, поэтому в данной главе приводятся формальные правила перевода специфических конструкций IndustrialC в код на языке C, а также правила модификации имён (необходимой для возможности взаимодействия кода на IndustrialC с фрагментами на C). Во втором разделе данной главы обосновывается использование фреймворка GNU Flex/Bison для создания лексического и синтаксического

анализаторов. В третьем разделе кратко описывается интегрированная среда разработки, реализованная в виде плагина к редактору Notepad++.

В пятой главе приводятся результаты апробации работы на примере трёх практических применений. Первые два сравнительно простые (т. н. «метеосервер» для получения данных о погоде с электронного табло и автоматизация установки термовакuumного напыления). Третий пример – станция пробоподготовки SorbiPrep – представляет собой более сложную установку (сочетающую многоканальную аппаратуру, требующую управления с обратной связью, локальный пульт оператора, а также возможность подключения к ПК при настройке и отладке), требующую параллельного исполнения разнообразных действий, в т.ч. использующих общие ресурсы. Т.е., третий пример и является представителем того класса задач, для решения которых создаются специализированные системы, являющиеся предметом диссертационной работы.

В **Заключении** (озаглавленном «Выводы и результаты») сформулированы основные результаты диссертационной работы.

По представленной диссертации можно высказать следующие **замечания**:

1. Основным недостатком текста работы является то, что во введении и Главе 1 автор не даёт чёткого определения того, о каких классах встраиваемых систем и микроконтроллеров идёт речь. Между тем, термин «встраиваемая система» очень широкий: таковой является как контроллер термостата (где хватит мощности процессора от банковской карты), так и автопилот самоуправяемого автомобиля. Аналогично, понятие «микроконтроллер» тоже очень широкое: от простейшего с 4-битной архитектурой до SoM с несколькими разнородными ARM-ядрами и гигабайтами ОЗУ.

В зависимости от класса встраиваемой системы и микроконтроллера подход к созданию ПО может радикально различаться, но в постановке задачи отсутствует однозначная формулировка, и лишь далее по тексту можно понять, что целевой платформой является Arduino.

2. Используемая модель работы с прерываниями неявно подразумевает такую организацию прерываний, как в процессорах AVR. Однако у разных архитектур процессоров работа с прерываниями существенно различается, поэтому в предлагаемом виде реализация жёстко привязана к процессорам семейства AVR (то есть даже с Arduino Due на ARM Cortex M уже несовместима).

3. Также стоило бы в тексте чётко сказать, что данная работа является продолжением проводимого в ИАиЭ цикла, в результате которого и был создан язык Reflex и введено понятие «гиперпроцесс».

4. В реализации выбрано (раздел 2.3, стр. 39) использование двух разных представлений времени: в миллисекундах и в микросекундах, в зависимости от требования конкретного применения; однако оба варианта имеют свои ограничения. Неясно, почему не был выбран вариант с совместным раздельным хранением секунд и микросекунд, широко применяемый в ОС семейства UNIX (тип `struct timeval`), обеспечивающий широкий диапазон (136 лет) с микросекундной точностью.

5. В разделе 2.4 обсуждается семантика исполнения реакций на события в случаях, когда реакция состоит из нескольких разных операций, поскольку при этом корректность выполнения в случае использования в реакциях разделяемых ресурсов зависит от того, параллельно ли выполняются реакции или последовательно, а также от конкретной последовательности исполнения. Неясно, почему не был выбран вариант с явным указанием режима исполнения (как это делается, например, в языке OCCAM при помощи директив SEQ и PAR), что устранило бы саму причину проблемы. (Видимо, была сделана попытка сохранить методологическую совместимость с языком Reflex; но она всё равно нарушена кардинально отличающейся семантикой.)

6. В разделе 3.1.2 на стр. 46 перепутаны местами определения строковых констант и строковых литералов.

7. Избранный подход к набору типов – *«Семантика остальных типов переменных полностью соответствует соответствующей семантике типов языка C и зависит от конкретной реализации используемого C-компилятора и целевой платформы»* (раздел 3.2.1, стр. 48) – весьма спорен. При работе с электроникой обычно требуется конкретный размер в битах, поэтому очень полезны были бы типы с явным указанием размера – такие, как `int32`, `uint8` и т.п.

8. В разделе 4.1.2 на стр. 69 в определении правил трансляции описания гиперпроцесса в макроопределении символ «точка с запятой» в конце директив `#define` выглядит явно лишним.

9. Ничего не сказано о платформе, на которой велась реализация компилятора и среды разработки. Лишь из упоминания Notepad++ можно предположить, что это Windows.

Указанные недостатки не снижают важности и полезности полученных в работе результатов, которые обладают научной новизной, обоснованы, достоверны, имеют теоретическую и практическую значимость и достаточно полно опубликованы. Диссертация является

законченной научно-исследовательской работой и удовлетворяет всем требованиям ВАК, предъявляемым к кандидатским диссертациям. Автореферат правильно отражает содержание диссертации. Считаю, что А.С. Розов заслуживает присуждения ему ученой степени кандидата технических наук по специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ».

Согласен на включение своих персональных данных в документы, связанные с работой диссертационного совета, и дальнейшую их обработку.

Официальный оппонент,  
Старший научный сотрудник  
сектора 5-12 ИЯФ СО РАН  
кандидат технических наук,



Д.Ю. Болховитянов

Подпись Д.Ю. Болховитянова заверяю,  
Ученый секретарь ИЯФ СО РАН  
к. ф.-м. н. А.С. Аракчеев  
30.04.2021

