

*На правах рукописи*

**Розов Андрей Сергеевич**

**РАЗРАБОТКА МОДЕЛЕЙ И МЕТОДОВ  
ПРОЦЕСС-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ ДЛЯ  
ОТКРЫТЫХ МИКРОКОНТРОЛЛЕРНЫХ ПЛАТФОРМ**

Специальность: 05.13.18 “Математическое моделирование, численные методы и  
комплексы программ”

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени

кандидата технических наук

Новосибирск, 2020

Работа выполнена в Федеральном государственном бюджетном учреждении  
науки «Институт автоматики и электрометрии» Сибирского отделения  
Российской академии наук

Научный **Зюбин Владимир Евгеньевич**, д-р техн. наук, доцент  
руководитель:

Официальные **Шидловский Станислав Викторович**, д-р техн. наук,  
оппоненты: Федеральное государственное бюджетное  
образовательное учреждение высшего профессионального  
образования «Национальный исследовательский Томский  
государственный университет», декан факультета  
инновационных технологий.

**Болховитянов Дмитрий Юрьевич**, канд. техн. наук,  
Федеральное государственное бюджетное учреждение  
науки Институт ядерной физики им. Г.И. Будкера  
Сибирского отделения Российской академии наук, сектор  
5-12, старший научный сотрудник.

Ведущая Федеральное государственное бюджетное  
организация: образовательное учреждение высшего образования  
«Новосибирский государственный технический  
университет».

Защита состоится “\_\_\_”\_\_\_\_\_202\_ г. в \_\_\_ часов на заседании  
диссертационного совета К 003.005.01 в Институте автоматики и электрометрии  
СО РАН по адресу 630090, г. Новосибирск, пр. ак. Коптюга, 1.

С диссертацией можно ознакомиться в библиотеке Института автоматики и  
электрометрии СО РАН.

Автореферат разослан “\_\_\_”\_\_\_\_\_202\_ г.

Ученый секретарь диссертационного совета

# ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

## Актуальность

Снижение стоимости микропроцессоров и тенденция к совмещению нескольких устройств в одной микросхеме привели к активному развитию встраиваемых систем (ВС). Замещение специализированных цифровых схем универсальными микроконтроллерами позволяет значительно удешевить разработку и производство ВС. Затраты на создание программного обеспечения (ПО) встраиваемых систем при этом многократно превышают расходы на проектирование и производство аппаратной составляющей. Этот эффект значительно усиливается активным развитием микроконтроллерных платформ с открытой архитектурой (МПОА), таких как Arduino и Itead Maple. Использование готовых аппаратных решений и широкий выбор совместимых периферийных устройств делают возможным быстрое прототипирование аппаратной составляющей встраиваемых систем. При этом исключается необходимость в дорогостоящих и затратных по времени операциях по изготовлению специализированных печатных плат. Таким образом, стоимость результирующей системы определяется в первую очередь стоимостью разработки ПО микроконтроллера.

Трудоемкость разработки ПО в значительной мере зависит от выбранных методик и языков программирования и их соответствия решаемой задаче. В настоящее время для программирования микроконтроллеров преимущественно используются языки C/C++. Понятийный аппарат этих языков изначально ориентирован на вычислительные задачи и обработку сложно структурированных данных. Разработка ПО ВС при помощи объектно-ориентированных технологий приводит к быстрому росту сложности программ по мере увеличения числа компонентов внешней среды. Обеспечение надежности работы системы в этом случае требует значительных временных затрат, что увеличивает стоимость разработки и сопровождения ПО.

Встраиваемые системы обладают многими свойствами алгоритмов управления в промышленной автоматизации – сложное поведение системы,

одновременное взаимодействие с множеством внешних для системы устройств. Для описания алгоритмов управления разработаны специализированные методы, средства и языки программирования. Методики событийного программирования на основе конечных автоматов, такие как switch-технология (А. А. Шалыто), иерархические автоматы (D. Harel), язык QuickStep (К. Crater), фреймворк StateWORKS (F. Wagner), позволяют эффективно описывать взаимодействие системы с внешней средой. Описание системы в виде набора взаимодействующих автоматов позволяет обеспечить логический параллелизм ПО ВС. Этот подход используется в технологии процесс-ориентированного программирования.

Использование наработок в области промышленной автоматизации может снизить трудоемкость разработки и отладки ПО ВС. Программирование микроконтроллеров предполагает активную работу с аппаратными прерываниями и минимизацию накладных расходов. Современные методы и языки описания алгоритмов управления ориентированы на аппаратные платформы промышленной автоматизации, существенно отличающиеся от микроконтроллерных платформ. Эффективное применение таких подходов в области встраиваемых систем требует их доработки с учетом специфики программирования микроконтроллеров.

Таким образом, задача разработки методик и языков программирования, учитывающих особенности алгоритмов управления и специфику встраиваемых систем, является актуальной.

**Объект исследования** – программное обеспечение встраиваемых систем на базе микроконтроллерных платформ с открытой архитектурой.

**Предмет исследования** – модели, методы и инструментальные средства спецификации процесс-ориентированного программного обеспечения открытых микроконтроллерных платформ.

**Цель работы** – разработка моделей, методов и инструментальных средств процесс-ориентированного программирования открытых микроконтроллерных платформ во встраиваемых системах.

**Задачи:**

1. Проанализировать специфику программирования микроконтроллеров и существующие подходы к описанию управляющих алгоритмов;
2. Сформулировать требования к разрабатываемым средствам;
3. Разработать математическую модель программного обеспечения встраиваемой системы;
4. Разработать метод вычисления реакций системы на внешние события на микроконтроллерах;
5. Определить синтаксис специализированного языка программирования, реализующего разработанный метод;
6. Создать среду разработки для языка в виде комплекса инструментальных программ;
7. Опробовать разработанные формальные, языковые и инструментальные средства на задачах разработки встраиваемых систем.

**Основная гипотеза:**

Встраиваемые системы на микроконтроллерных платформах имеют ряд общих свойств с алгоритмами управления в промышленной автоматизации и могут эффективно описываться с применением процесс-ориентированных технологий.

На базе процесс-ориентированного подхода и с учетом специфики программирования микроконтроллеров могут быть созданы понятийные, языковые

и инструментальные средства, позволяющие существенно снизить затраты на разработку ПО встраиваемых систем.

### **Научная новизна:**

1. Предложена математическая модель ПО ВС в виде набора гиперпроцессов с различными источниками активации и взаимодействием гиперпроцессов через запуск и остановку прерываний, совмещающая кооперативную и вытесняющую модели многозадачности;
2. Разработан синтаксис процесс-ориентированного языка программирования IndustrialC, совмещающий высокоуровневые понятийные элементы программирования микроконтроллеров и процесс-ориентированного программирования;
3. Разработан метод вычисления внешних реакций систем на микроконтроллерах семейства AVR с одновременным использованием аппаратных прерываний и фоновой программы и представлением гиперпроцессов в виде процедур обработки прерываний.

### **Теоретическая и практическая значимость:**

Созданный язык и средства разработки, расширяющие процесс-ориентированный подход на область программирования микроконтроллеров, обеспечивают снижение временных затрат на разработку, модификацию и сопровождение встраиваемых систем.

Возможность работы с прерываниями в процесс-ориентированном ПО повышает скорость реакции на критические события при сохранении преимущественно кооперативной модели многозадачности. Активация следующего цикла гиперпроцесса по окончании предыдущего цикла устраняет непроизводительные простои и повышает производительность системы в целом, в первую очередь снижает время реакции на внешнее событие.

Использование специализированного языка IndustrialC уменьшает размер исходного кода, повышает его читаемость и простоту изучения для разработчика.

Реализованные в трансляторе методы статического анализа кода позволяют выявлять семантические ошибки на этапе трансляции, снижают вероятность ошибки программиста, а также снижают объем накладных расходов на синхронизацию служебных операций. Понятийный аппарат, ориентированный на предметную область встраиваемых систем, повышает читаемость и сопровождаемость создаваемых программ.

Результаты работы внедрены в учебный процесс ФИТ НГУ, что позволило повысить качество подготовки. Разработанные средства использовались в инициативных проектах при создании системы метеосервера, системы автоматизации установки термовакуумного напыления, ПО адаптера электронного блока весоизмерительной системы, при выполнении хоздоговоров СОРБИ-16, СОРБИ-17. Практическая ценность полученных результатов подтверждается актами о внедрении.

#### **Методология и методы исследования:**

Исследование осуществлялось с использованием общенаучных теоретико-эмпирических методов и специальных методов из областей автоматного программирования, теории управления, математического моделирования и итеративных методов разработки.

Анализ предметной области выполнялся с использованием содержательного и логико-исторического подходов. При разработке модели понятийного аппарата и синтаксиса языка, а также при создании программных средств, использовался формальный подход. Исследование свойств полученных средств разработки на практических задачах выполнялось с использованием эмпирического подхода в виде экспериментов с системами управления физическими процессами и численного эксперимента с программной имитацией объектов управления.

### **Область исследования:**

Содержание диссертации соответствует паспорту специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ» (технические науки) по следующим областям исследований: п.1 «Разработка новых математических методов моделирования объектов и явлений», п.3 «Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий», п. 4 «Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента» и п. 8 «Разработка систем компьютерного и имитационного моделирования».

### **Положения, выносимые на защиту:**

1. Математическая модель ПО ВС в виде набора гиперпроцессов с различными источниками активации;
2. Метод вычисления реакции системы на внешние события с совмещением кооперативной и вытесняющей многопоточности, с выделенным фоновым гиперпроцессом и гиперпроцессами, активируемыми аппаратными прерываниями;
3. Синтаксис и семантика языка программирования IndustrialC, реализующего разработанный метод вычисления реакций на внешние события;
4. Среда разработки языка IndustrialC в виде программного комплекса, включающего редактор, транслятор языка, средства загрузки исполняемого кода и средства интеграции компонентов комплекса.

### **Апробация работы:**

Результаты работы докладывались на: Международной научной студенческой конференции "Студент и научно-технический прогресс" (Новосибирск, 2013, 2014, 2015), XIII Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям (Новосибирск, 2012), Международной IEEE-Сибирской конференции по



управлению и связи SIBCON-2013 (Красноярск, Томск, 2013, 2019), международной конференции "International Conference on Advanced Technology & Sciences" ICAT'15 (Анталия, 2015), Международной мультikonференции по инженерным, компьютерным и информационным наукам IEEE SIBIRCON (Новосибирск, 2019), Международной научно-технической конференция "Автоматизация" (RusAutoCon, Новосибирск, 2019).

### **Публикации**

По теме диссертации опубликовано 25 печатных работы, из них семь статей в журналах, рекомендованных ВАК, семь статей в изданиях, индексируемых Scopus, и два свидетельства официальной регистрации программ.

### **Структура и объем диссертации**

Диссертация состоит из введения, пяти глав и заключения. Объем работы 127 страниц, количество рисунков 12.

В первой главе приводится обзор существующих подходов к созданию ПО встраиваемых систем, анализируются особенности систем управления и специфика микроконтроллерных платформ, формулируются требования к разрабатываемым средствам. Во второй главе описывается математическая модель ПО встраиваемых систем, задающая понятийный аппарат для программирования микроконтроллеров с использованием процесс-ориентированной технологии. В третьей главе описывается грамматика специализированного языка IndustrialC. В четвертой главе описывается трансляционная семантика языка IndustrialC, задающая метод вычисления реакций встраиваемых систем на внешние события, а также разработанный программный комплекс среды разработки IndustrialC. В пятой главе приводятся результаты практической апробации разработанных понятийного аппарата, языка и инструментальных средств в проектах по созданию ВС МПОА.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **ВВЕДЕНИИ** обоснована актуальность работы, определены цель и задачи исследования, научная новизна, практическая ценность и основные положения, выносимые на защиту.

**ПЕРВАЯ ГЛАВА** посвящена исследованию специфики управляющего ПО и микроконтроллерных платформ и анализу существующих методов разработки ПО микроконтроллеров.

Микроконтроллерные платформы имеют относительно ограниченный объем вычислительных ресурсов — низкие тактовые частоты, малый объем ОЗУ и ПЗУ, доступных для работы и хранения программ. Для снижения энергопотребления многие микроконтроллеры предоставляют возможность перехода в спящий режим, выход из которого осуществляется по прерыванию.

Ограниченность общих вычислительных ресурсов микроконтроллеров компенсируется наличием большого количества интегрированных периферийных устройств, таких как таймеры, АЦП, программируемые входы-выходы и аппаратные интерфейсы передачи данных. Работа с этими устройствами предполагает использование аппаратных прерываний.

Эти особенности программирования микроконтроллеров — ограниченность ресурсов, наличие спящего режима процессора и активная работа с внутренней периферией через аппаратные прерывания — должны учитываться при выборе подходов к разработке систем на микроконтроллерных платформах.

Управляющие алгоритмы имеют ряд свойств, отличающих их от так называемых трансформационных или вычислительных систем: открытость, событийность, параллелизм, синхронизм, надежность и устойчивость. Эти свойства определяют требования к методикам и языкам программирования, используемым для описания алгоритмов управления, и справедливы также и при разработке встраиваемых систем на базе микроконтроллеров.

Были рассмотрены существующие подходы к разработке ПО встраиваемых микроконтроллерных систем: использование процесс-ориентированных языков

общего назначения, использование языков стандарта МЭК 61131-3, автоматическая генерация Си-кода из среды Matlab/Simulink, применение операционных систем, комбинированный подход с реализацией операционной системы в специализированном языке программирования, подходы на основе автоматного и процесс-ориентированного программирования.

Разработка ПО со сложным поведением в объектно-ориентированном стиле приводит к запутанному, плохо читаемому коду и высоким накладным расходам. Операционные системы реального времени используют модели вытесняющей многозадачности, приводящие к возникновению гонок. Специализированные методы и подходы на основе конечных автоматов демонстрируют наибольшую эффективность при описании систем со сложным поведением. Наиболее ярко эти методы представлены в работах Д. Харела (D. Harel), Ф. Вагнера (F. Wagner) и А. Шалыто.

Попытки описать весь алгоритм управления одним автоматом приводят к комбинаторному взрыву сложности при описании систем со множеством параллельных процессов. Модель иерархического автомата позволяет устранить повторяющиеся реакции на события за счет использования вложенных состояний. Другой способ решения проблем комбинаторного взрыва сложности и облегчения восприятия системы – представление системы в виде набора отдельных параллельно исполняющихся автоматов – гиперпроцесса. В методике процесс-ориентированного программирования, реализованной в специализированном языке программирования Reflex, эта идея используется для описания алгоритмов управления промышленной автоматизации на базе ПЛК.

Понятийный аппарат процесс-ориентированного программирования обеспечивает событийность, логический параллелизм и синхронизм алгоритмов управления, удовлетворяя всем требованиям управляющих систем. Специфика программирования микроконтроллеров дополнительно требует поддержки работы с аппаратными прерываниями. Существующие реализации языка Reflex и модели гиперпроцесса предполагают абстракцию обработки прерываний в отдельном программном слое. Таким образом, средствами Reflex реализуется только

высокоуровневая часть алгоритма управления. Обработка прерываний реализуется средствами языка Си. Разработка и поддержание такой системы требуют работы с двумя языками программирования, увеличивая требования к квалификации разработчика.

На основании проведенного анализа специфики программирования микроконтроллеров и существующих подходов к разработке управляющих алгоритмов были сформулированы требования к разрабатываемым средствам:

1. Возможность представления системы в виде набора независимо исполняющихся взаимодействующих автоматов.
2. Встроенная поддержка работы с временными интервалами.
3. Встроенная поддержка работы с прерываниями.
4. Минимизация накладных расходов на обеспечение работы и взаимодействия автоматов.
5. Реализация в специализированном языке программирования с трансляцией в язык Си.

Во **ВТОРОЙ ГЛАВЕ** представляется разработанный формализм описания ПО ВС МПОА.

За основу был взят формализм гиперпроцесса, используемый в процессориентированном языке Reflex. Программа микроконтроллера описывается набором асинхронно исполняющихся гиперпроцессов с отдельными источниками активации. На микроконтроллерах в качестве источников активации выступают аппаратные прерывания и основной цикл программы. При этом процедуры обработки прерываний оформляются в виде гиперпроцессов. В связи с асинхронным исполнением гиперпроцессов изменены механизмы отслеживания времени. Вместо дискретного времени, измеряемого тактами гиперпроцесса, введен счетчик физического времени в миллисекундах, обновляемый одним из таймеров микроконтроллера. Переопределены операции смены состояния, запуска и остановки процессов и сброса тайм-аута, а также события тайм-аута и проверки активности процессов. Предложена схема независимого тактирования операций

обработки тайм-аутов, обеспечивающая сохранение корректной работы тайм-аутов в гиперпроцессах с нерегулярной активацией. Определены механизмы взаимодействия между гиперпроцессами посредством операций запуска/останова источников активации и механизмы синхронизации работы с разделяемой памятью при помощи критических секций.

Разработанный формализм предполагает представление ПО микроконтроллера в виде совокупности множеств:

$$(H, V, C, R, T), \text{ где}$$

$H$  – множество гиперпроцессов,  $H = \{h_0, \dots, h_n\}$ ;

$V = \{v_1, \dots, v_{nv}\}$  и  $C = \{c_1, \dots, c_{nc}\}$  – множества глобальных переменных и констант программы, типов  $t_1^v, \dots, t_{nv}^v$  и  $t_1^c, \dots, t_{nc}^c$  соответственно;

$R = \{r_1, \dots, r_{nr}\}$  – множество используемых в программе входных и выходных регистров;

$T$  – счетчик системного времени.

Гиперпроцесс  $h_i \in H$  определяется парой источник активации – множество процессов:

$$h_i \equiv (a_i, P_i).$$

Процесс  $p_j^i \in P_i$  представляется множеством функций-состояний  $S_l^i$ , выделенными начальным состоянием  $s_j^{i,l}$ , текущим состоянием  $cs_j^i$  и временем  $t_j^i$  последней смены состояния:

$$p_j^i \equiv (S_l^i, s_j^{i,l}, cs_j^i, t_j^i).$$

Состояние  $s_k^{i,j} \in S_j^i$  процесса  $p_j^i$  представляется упорядоченным множеством операций  $sbody_k^{i,j}$ , составляющих тело состояния, и выделенной операцией обработки тайм-аута  $ts_k^{i,j}$ :

$$s_k^{i,j} \equiv (sbody_k^{i,j}, ts_k^{i,j}), ts_k^{i,j} \in sbody_k^{i,j}.$$

Разработанный формализм отражает специфику программирования микроконтроллеров при сохранении принципов процесс-ориентированного программирования и выполняет следующие функции:

- задает понятийный аппарат для разработанного языка IndustrialC;

- служит в качестве промежуточного представления между программой на IndustrialC и ее эквивалентным представлением в процессе трансляции;
- дает терминологическую базу для дальнейшего определения эквивалентного представления на языке C в привязке к синтаксису языка.

**В ТРЕТЬЕЙ ГЛАВЕ** рассмотрен разработанный синтаксис специализированного языка программирования IndustrialC, описаны основные лексические элементы и конструкции языка их семантика в привязке к разработанному формализму описания ПО МК. За основу взяты языки C и Reflex. При разработке синтаксиса специфических для IndustrialC конструкций использовался синтаксис из C-подобных языков, таких как C++.

Синтаксис IndustrialC практически полностью включает синтаксис C, за исключением конструкций работы с памятью через указатели. Семантика конструкций IndustrialC, заимствованных из C, полностью сохранена.

Из языка Reflex заимствованы конструкции описания процессов и состояний, операций тайм-аутов, а также операций и выражений для взаимодействия между процессами. При этом семантика этих конструкций существенно отличается от их синтаксических эквивалентов в Reflex.

Синтаксис языка расширен конструкциями, позволяющими описывать процессы, активируемые прерываниями, унифицированными операциями управления источниками активации и обеспечения синхронизации доступа к разделяемой памяти. В целях сохранения гибкости язык предусматривает механизмы вставки кода на языке C с возможностью совместного использования общих данных между участками кода на двух языках.

**В ЧЕТВЕРТОЙ ГЛАВЕ** представлена разработанная трансляционная семантика программ на языке IndustrialC, задающая эквивалентное представление программ на IndustrialC на языке C. Описаны принципы работы разработанного транслятора языка IndustrialC, методы реализации транслятора и интегрированной среды разработки.

Правила трансляции задаются для 8-битных микроконтроллеров семейства AVR. По причине общей высокой зависимости программ микроконтроллера от аппаратной платформы сохранение семантики конструкций языка в реализации для других семейств микроконтроллеров может потребовать изменения правил трансляции. При описании правил трансляции используется понятийный аппарат формализма описания ПО МК, заданный в Главе 2. Описанные правила применяются в реализованном трансляторе языка, а также задают трансляционную семантику языка, которая в совокупности с формальными семантиками языка С может быть в дальнейшем использована для статической верификации программ микроконтроллеров.

Транслятор языка реализован с использованием генераторов лексического и синтаксического анализаторов Flex/Bison, что значительно упрощает дальнейшие модификацию и развитие языка. Для обеспечения поддержки директив препроцессора разработан подход с предварительной обработкой программы С-препроцессором и анализом оставляемых им меток. Значительное пересечение по синтаксису и семантике с языком С позволило значительно сократить объем анализа кода, производимый транслятором. Разработанные алгоритмы и критерии семантического анализа кода обеспечивают возможность автоматической расстановки критических секций на этапе трансляции и выдачи предупреждающих сообщений.

Интегрированная среда разработки на базе редактора Notepad++ предоставляет удобный интерфейс для редактирования программ с подсветкой синтаксиса языка, возможностью организации файлов в проекты, трансляции, сборки и загрузки программ на контроллер непосредственно из редактора.

**В ПЯТОЙ ГЛАВЕ** описываются результаты практической апробации разработанных средств при решении задач разработки встраиваемых систем на микроконтроллерных платформах: разработка метеосервера Института автоматизации и электрометрии СО РАН, автоматизация установки термовакуумного напыления УВН-71 ПЗ и разработка системы управления станции пробоподготовки SorbiPrep.

В результате апробации выявлено, что разработанный язык IndustrialC в совокупности с созданными инструментальными средствами позволяет эффективно описывать программное обеспечение микроконтроллеров во встраиваемых системах. При этом обеспечиваются:

- простота модификации и расширяемость системы – выявлена высокая локальность правок при изменении и расширении функциональности системы, изменении используемых внешних периферийных устройств, добавление в систему новых процессов и гиперпроцессов для обеспечения дополнительной функциональности не влияет на работу существующих в системе процессов;

- масштабируемость системы – увеличение количества одинаковых контролируемых устройств в системе преимущественно сводится к копированию и переименованию соответствующего набора процессов и гиперпроцессов, как пример рассматривалось увеличение количества портов станции пробоподготовки SorbiPrep с трех до шести;

- простота переиспользования кода – процессы работы с USART, управления нагревателями и взаимодействия с АЦП переиспользовались между ПО SorbiPrep и системой управления установки термовакуумного напыления;

- надежность программы за счет преимущественного использования кооперативной многозадачности, что значительно снижает вероятность возникновения гонок по сравнению с вариантом вытесняющей многопоточности.

## **ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ**

В работе предложен оригинальный подход к процесс-ориентированной разработке управляющего ПО встраиваемых систем на базе открытых микроконтроллерных платформ с использованием разработанного специализированного языка IndustrialC в совокупности с инструментальными средствами – транслятором в язык C и интегрированной средой на базе Notepad++.

В ходе работы получены следующие основные результаты:

1. Проведен анализ специфики управляющего ПО ВС и программирования открытых микроконтроллерных платформ, сформулированы требования к



математическим, языковым и инструментальным средствам разработки ПО ВС МПОА.

2. Разработана математическая модель управляющего ПО на микроконтроллерных платформах, предусматривающая возможность работы с аппаратными прерываниями за счет представления системы набором отдельно активируемых гиперпроцессов, определения механизмов взаимодействия между гиперпроцессами и синхронизации доступа к разделяемым данным.

3. Разработан и формализован синтаксис процесс-ориентированного языка программирования IndustrialC, имеющий низкий порог вхождения за счет пересечения с языком C и согласованности синтаксиса специфических конструкций с C-подобными языками.

4. Задана трансляционная семантика языка IndustrialC, выраженная через правила трансляции IndustrialC-программ в эквивалентный код на языке C и задающая метод вычисления реакций встраиваемых систем на внешние события.

5. Разработан и реализован программный комплекс среды разработки языка IndustrialC, включающий препроцессор, транслятор и средства интеграции с редактором Notepad++. Транслятор IndustrialC в язык C обеспечивает поддержку и анализ значительной части конструкций языка C за счет механизма чтения и расстановки меток препроцессора. Выявлены критерии автоматической расстановки критических секций для служебных операций языка и выдачи предупреждающих сообщений о возможном асинхронном доступе к разделяемым данным.

6. Полученные средства использованы в практических задачах разработки ВС на МПОА и при организации учебного процесса.

7. Практическая ценность полученных результатов подтверждена актами о внедрении.

## СПИСОК ОСНОВНЫХ РАБОТ, ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Andrei S. Rozov, Vladimir E. Zyubin Process-Oriented Programming Language for MCU-Based Automation // 2013 International Siberian Conference on Control and Communications (SIBCON)
2. Andrey S. Rozov, Vladimir E. Zyubin A hyperprocess-based approach in Arduino programming // International Conference on Advanced Technology & Sciences (ICAT'15), Antalya, 4-7 August, 2015.
3. Розов А. С., Зюбин В. Е. Расширенная модель гиперпроцесса для программирования микроконтроллеров // Промышленные АСУ и контроллеры. 2016 г., № 9. С. 34-38.
4. Краснов Д. В., Нефедов Д. В., Санжиев Е. С., Лях Т. В., Розов А.С. Практическая апробация процесс-ориентированной технологии программирования на открытых микроконтроллерных платформах // Вестник ВСГУТУ. 2017, Т. 66, вып. 3. С.85–92.
5. Розов А. С., Лях Т. В., Краснов Д. В., Санжиев Е. С. Практическая апробация языка IndustrialC на примере автоматизации установки термовакuumного напыления // Вестник НГУ Серия: Информационные технологии, Том 15, Выпуск № 3 (2017). С. 90-99.
6. Розов А. С., Зюбин В. Е., Нефедов Д. В. Программирование встраиваемых микроконтроллерных систем на основе гиперпроцессов // Вестник НГУ, Серия: информационные технологии. 2017. Т. 15, № 4. С. 64–73.
7. Зюбин В. Е., Петухов А. С., Розов А. С. Автоматизация измерений мюонным скважинным плотномером // Промышленные АСУ и контроллеры, 2018, №11, С.35-41.
8. I. Anureev, N. Garanina, T. Liakh, A. Rozov, H. Schulte and V. Zyubin Towards Safe Cyber-Physical Systems: the Reflex Language and Its Transformational Semantics // 2019 International Siberian Conference on Control and Communications (SIBCON), Tomsk, Russia, 2019, pp. 1-6.

9. Розов А. С., Зюбин В. Е. Адаптация процесс-ориентированного подхода к разработке встраиваемых систем на микроконтроллерах // Автометрия 2019. Т. 55, № 2. С. 114-122.
10. Anureev I., Garanina N., Liakh T., Rozov A., Zyubin V., Gorchak S. Two-Step Deductive Verification of Control Software Using Reflex // Lecture Notes in Computer Science, vol 11964.
11. Rozov A., Anureev I., Garanina, N., Liakh T., Zyubin V. Towards Safe Embedded Systems: Industrial C Translational Semantics for AVR Microcontrollers. // SibirCon 2019, Novosibirsk, 21-22 October 2019.
12. V. Drobotun, A. Rozov Bias-Corrected Optical Measurement of Film Thickness for Vacuum Evaporation Coating. // 2020 International Russian Automation Conference (RusAutoCon), Sochi, Russia, September 2020, pp. 965-970.
13. S. E. Belenkaia, V. E. Zyubin and A. S. Rozov Generating Process Diagrams for Control Software in the Reflex Language // 2020 International Russian Automation Conference (RusAutoCon), Sochi, Russia, September 2020, pp. 1010-1014.