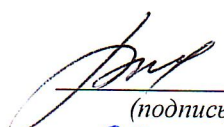


Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

Работа выполнена в СКБ «Промышленная робототехника»

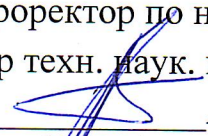
СОГЛАСОВАНО

Начальник отдела ОНиПКРС


  
(подпись) Е.М. Димитриади  
« 06 » 06 2024 г.

УТВЕРЖДАЮ

Проректор по научной работе,  
д-р техн. наук, профессор

  
(подпись) А.В. Космынин  
« 06 » 06 2024 г.


Декан ФЭУ

  
(подпись) А.С. Гудим  
« 06 » 06 2024 г.

«Система дискретно-событийного управления технологическими операциями»


Комплект документации на управляющую программу для автоматизированной системы

Руководитель СКБ

  
(подпись, дата) 06.06.2024


С.И. Сухоруков

Руководитель проекта

  
(подпись, дата) 06.06.24


А.В. Бузикаева

Наставник проекта

  
(подпись, дата) 06.06.24

А.С. Савельева

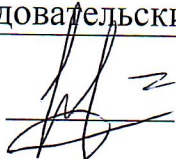

Исполнитель проекта

  
(подпись, дата) 06.06.24

Т.Е. Безнаев

Комсомольск-на-Амуре 2024

### Карточка проекта

Название	Система дискретно-событийного управления технологическими операциями
Тип проекта	Научно-исследовательский проект
Исполнители	Наставник: Студент  А.С. Савельева – 3АУМ-1 Исполнитель:  Т.Е. Безнаев Студент лицея при ФГБОУ ВО «КНАГУ»
Срок реализации	02.2024-05.2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

ЗАДАНИЕ  
на разработку

Название проекта: Система дискретно-событийного управления технологическими операциями

Назначение: Повышение эффективности предприятия

Область использования: Крупные промышленный заводы, металлургическая промышленность

Функциональное описание проекта: Алгоритм работы программы должен реализовать процедуры управления технологическими операциями производственных циклов

Техническое описание программы: Программа должна реализовать процедуры управления технологическими операциями производственных циклов.

Разрабатываемый программный комплекс выступает в роли экспертной системы управления, а его интеграция в систему автоматического управления позволит увеличить количество изготавливаемой продукции за счет уменьшения времени задержек между различными операциями производственного цикла.

Требования: Программа должна соответствовать заданию на разработку, быть надежной, практичной

План работ:

Наименование работ	Срок
Разработка инновационного подхода к решению задачи регулирования технологическими операциями литейного производства	02.2024
Проектирование системы управления посредством ее описания дискретно-событийным моделированием	03.2024
Описание подхода дискретно-событийного моделирования для управления технологическими операциями для тезисов статьи Всероссийской национальной научной конференции молодых учёных «Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований»	04.2024
Разработка алгоритма работы объекта	05.2024

Комментарии:

---

---

---

---

---

Перечень графического материала:

1. Код программы
2. Имитационная модель объекта управления
3. Сравнительные графики переходных процессов

---

---

---

---

Руководитель проекта


 06.06.24  
(подпись, дата)

А.В. Бузикаева

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

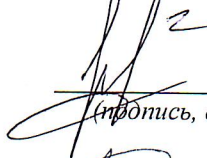
ПАСПОРТ  
Управляющей программы для  
автоматизированной/роботизированной системы  
«Система дискретно-событийного управления технологическими  
операциями»

Руководитель проекта

  
06.06.24  
(подпись, дата)


А.В. Бузикаева

Наставник проекта

  
06.06.24  
(подпись, дата)

А.С. Савельева

Исполнитель проекта

  
06.06.24  
(подпись, дата)

Т.Е. Безнаев

Комсомольск-на-Амуре 2024

## Содержание

1	Общие положения.....	7
1.1	Наименование изделия.....	7
1.2	Наименования документов, на основании которых ведется проектирование изделия.....	7
1.3	Перечень организаций, участвующих в разработке изделия.....	7
1.4	Сведения об использованных при проектировании нормативно-технических документах.....	8
2	Описание программы.....	9
2.1	Назначение изделия.....	9
2.2	Области использования изделия.....	9
2.3	Принцип действия изделия.....	9
2.1	Используемые технические средства.....	12
2.2	Вызов и загрузка.....	12
2.3	Входные данные (при наличии).....	12
2.4	Выходные данные (при наличии).....	12
3	Руководство оператора.....	13
3.1	Назначение программы;.....	13
3.2	Условия выполнения программы;.....	13
3.3	Сообщения оператору.....	13
	ПРИЛОЖЕНИЕ А.....	14

					СКБФЭУ.1.ИП.01000000	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		6

## 1 Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Система дискретно-событийного управления технологическими операциями» (далее «программа»).

Паспорт входит в комплект поставки программы. Перед запуском программы внимательно изучите правила ее эксплуатации.

### 1.1 Наименование изделия

Полное наименование программы – «Система дискретно-событийного управления технологическими операциями».

### 1.2 Наименования документов, на основании которых ведется проектирование изделия

Создание программы «Система дискретно-событийного управления технологическими операциями» осуществляется на основании требований и положений следующих документов:

- задание на разработку.

### 1.3 Перечень организаций, участвующих в разработке изделия

Заказчиком создания программы «Система дискретно-событийного управления технологическими операциями» является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 27.

Исполнителями работ по созданию программы «Система дискретно-событийного управления технологическими операциями» являются Конструкторы студенческого конструкторского бюро (далее СКБ), студент группы ЗАУм-1, Савелева Алена Сергеевна.

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		7

#### 1.4 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 19.001-77. Единая система программной документации (ЕСПД). Общие положения.

ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения

ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов.

ГОСТ 19.102-77. ЕСПД. Стадии разработки.

ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению.

ГОСТ 19.402-78. ЕСПД. Описание программы.

ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.

ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению.

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		8



## 2 Описание программы

### 2.1 Назначение изделия

Имитационная модель создана на основе дискретного моделирования и логике StateFlow.

### 2.2 Области использования изделия

Система может применяться на предприятиях где недопустим прямой контакт оператора с рабочей средой.

### 2.3 Принцип действия изделия

Программа позволяет реализовать процедуры управления технологическими операциями производственных циклов. Предложенный программный комплекс выступает в роли экспертной системы управления, а его интеграция в систему автоматического управления позволит увеличить количество изготавливаемой продукции за счет уменьшения времени задержек между различными операциями производственного цикла.

Типовую имитационную модель технологического процесса литейного производства (далее ТПЛП) можно представить следующим образом, показанном на рисунке 1. Дискретно-событийная модель технологического процесса мелкосерийного литейного производства отображает производственные этапы, соответствующие базовым процессам металлургических предприятий: плавку, отливку, охлаждение и постобработку. В имитационной дискретно-событийной модели каждый этап представляет собой подсистему со своими параметрами [4]. Каждая из имеющихся подсистем может быть представлена типовым набором элементов операций (рисунок 2).

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		9

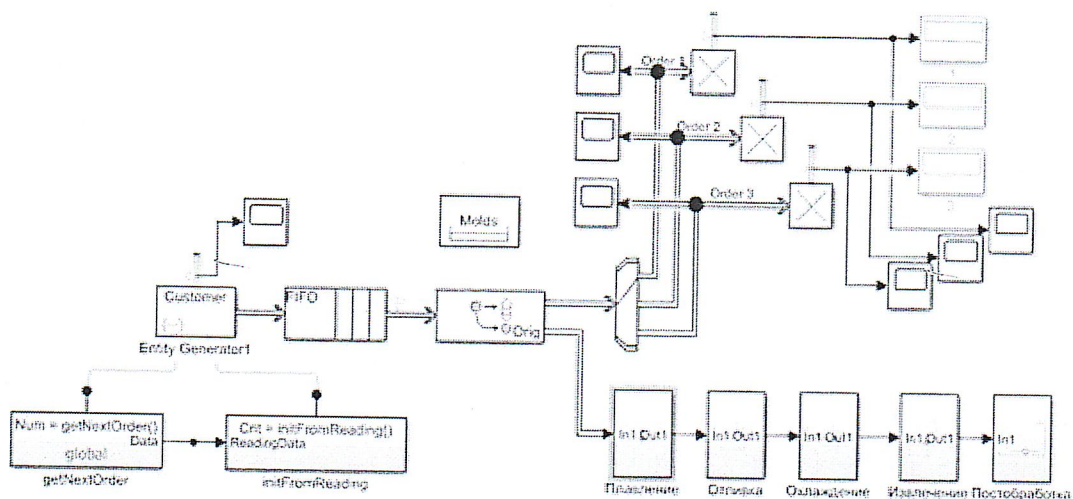


Рисунок 1 – Дискретно-событийная модель типового ТПЛП

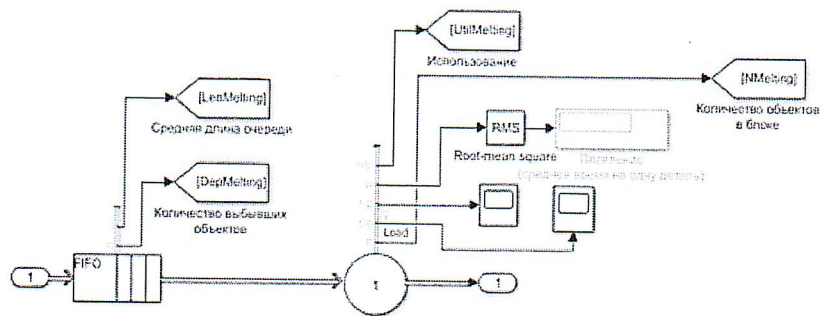


Рисунок 2 – Имитационная модель подсистемы плавления

Для сравнения и сопоставления результатов исследования, разработана и представлена (рисунок 3) дискретно-событийная модель автоматизированного ТПЛП.

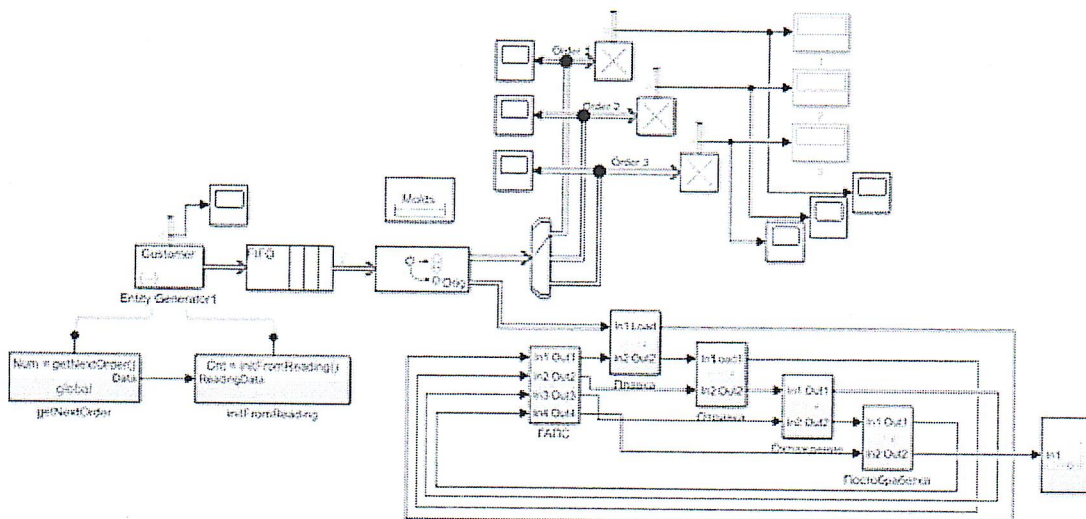


Рисунок 3 – Дискретно-событийная модель автоматизированного ТПЛП

Сравнительные характеристики производительности производства трех различных типов изделий за неделю для ТПЛП с автоматизированным и

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

ручным трудом представлены на рисунках 4-6.

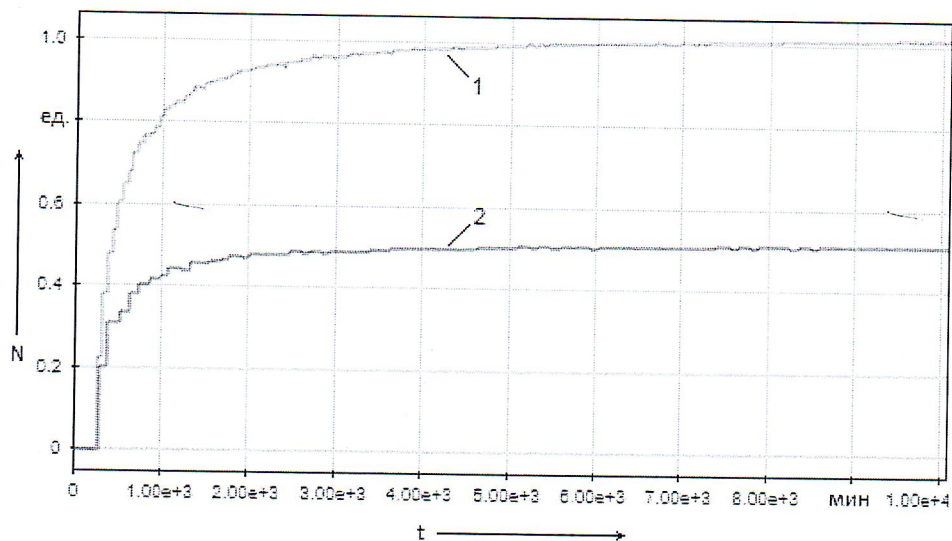


Рисунок 4 – Производительность изделий первого типа  
1 – автоматизированный ТПЛП; 2 – неавтоматизированный ТПЛП

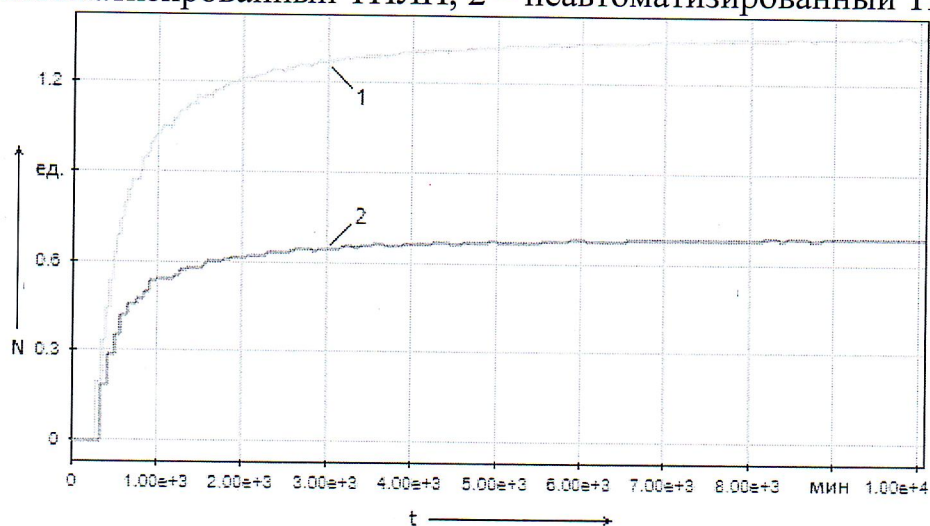


Рисунок 5 – Производительность изделий второго типа  
1 – автоматизированный ТПЛП; 2 – неавтоматизированный ТПЛП

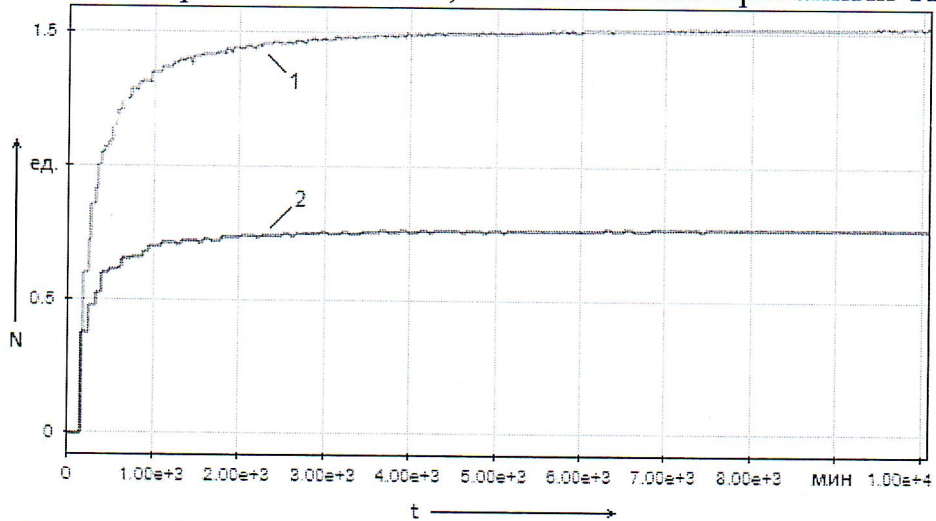


Рисунок 6 – Производительность изделий третьего типа  
1 – автоматизированный ТПЛП; 2 – неавтоматизированный ТПЛП

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

СКБФЭУ.1.ИП.01000000

Лист

11

Исходя из представленных переходных процессов можно сделать вывод, что автоматизация ТПЛП позволяет минимизировать задержки между технологическими операциями, регулируя изготовление того или иного типа деталей на каждом этапе жизненного цикла изделий, тем самым повышая количество производимых деталей более чем в два раза каждого типа.

На основе информации ранее сказанной, можно сделать вывод, что при использовании средств автоматизации в ТПЛП, появляется возможность его оптимизации по быстродействию, а дальнейшее применение интеллектуальных технологий в качестве экспертных систем, позволит значительно улучшить распределение задач между технологическими операциями. Таким образом, существенный прирост производительности позволит сократить время изготовления каждого изделия и, как следствие, повысить производительность и прибыль предприятия.

## **2.1 Используемые технические средства**

Используется программирование логического контроллера.

Программный комплекс, реализованный на языке программирования «С», предназначен для устройств Atmel на архитектуре AVR (32-bit).

## **2.2 Вызов и загрузка**

На логический контроллер переносится управляющая модель, после чего контроллер интегрируется в ТПЛП.

## **2.3 Входные данные (при наличии)**

Модель подразумевает под собой полностью автономную систему позволяющую реализовать экспертное управление технологическими операциями для уменьшения времени простоя технологических операций.

## **2.4 Выходные данные (при наличии)**

В качестве выходных данных модель позволяет получить данные об использовании отдельных операций в процентном соотношении.

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		12

### 3 Руководство оператора

Оператору не требуется принимать активное участие в работе модели, а лишь следить за работоспособностью оборудования.

#### 3.1 Назначение программы;

Модель позволяет использовать возможности автоматизации технологических операций с целью обеспечения уменьшения временных задержек. Необходимо реализовать дискретно-событийную модель системы, работающую не хуже, чем типовая система регулирования.

Код программы приведен в Приложении А.

#### 3.2 Условия выполнения программы;

Для стабильного и исправного выполнения программы заданной в модели не требуется калибровка.

#### 3.3 Сообщения оператору

Программа является полностью автономной и не требует вмешательства оператора.

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		13

## ПРИЛОЖЕНИЕ А

(обязательное)

```
/*
 * File: ert_main.c
 *
 * Code generated for Simulink model 'FLGateController'.
 *
 * Model version           : 1.2
 * Simulink Coder version  : 9.0 (R2018b) 24-May-2018
 * C/C++ source code generated on : Thu Nov  9 15:12:23 2023
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Atmel->AVR (32-bit)
 * Code generation objectives:
 *   1. Execution efficiency
 *   2. RAM efficiency
 * Validation result: Not run
 */

#include <stddef.h>
#include <stdio.h>           /* This ert_main.c example uses
printf/fflush */
#include "FLGateController.h" /* Model's header file */
#include "rtwtypes.h"

/*
 * Associating rt_OneStep with a real-time clock or interrupt service routine
 * is what makes the generated code "real-time". The function rt_OneStep is
 * always associated with the base rate of the model. Subrates are managed
 * by the base rate from inside the generated code. Enabling/disabling
 * interrupts and floating point context switches are target specific. This
 * example code indicates where these should take place relative to executing
 * the generated code step function. Overrun behavior should be tailored to
 * your application needs. This example simply sets an error status in the
 * real-time model and returns from rt_OneStep.
 */
void rt_OneStep(void);
void rt_OneStep(void)
{
    static boolean_T OverrunFlag = false;

    /* Disable interrupts here */

    /* Check for overrun */
    if (OverrunFlag) {
        rtmSetErrorStatus(rtm, "Overrun");
        return;
    }

    OverrunFlag = true;

    /* Save FPU context here (if necessary) */
    /* Re-enable timer or interrupt here */
    /* Set model inputs here */

    /* Step the model for base rate */
    FLGateController_step();

    /* Get model outputs here */
}
```

Изм.	Лист	№ документа	Подп.	Дата

СКБФЭУ.1.ИП.01000000

Лист

14

```

/* Indicate task complete */
OverrunFlag = false;

/* Disable interrupts here */
/* Restore FPU context here (if necessary) */
/* Enable interrupts here */
}

/*
 * The example "main" function illustrates what is required by your
 * application code to initialize, execute, and terminate the generated code.
 * Attaching rt_OneStep to a real-time clock is target specific. This exam-
 * ple
 * illustrates how you do this relative to initializing the model.
 */
int_T main(int_T argc, const char *argv[])
{
    /* Unused arguments */
    (void) (argc);
    (void) (argv);

    /* Initialize model */
    FLGateController_initialize();

    /* Attach rt_OneStep to a timer or interrupt service routine with
     * period 0.042016806722689079 seconds (the model's base sample time) here.
     The
     * call syntax for rt_OneStep is
     *
     * rt_OneStep();
     */
    printf("Warning: The simulation will run forever. "
           "Generated ERT main won't simulate model step behavior. "
           "To change this behavior select the 'MAT-file logging' option.\n");
    fflush((NULL));
    while (rtmGetErrorStatus(rtM) == (NULL)) {
        /* Perform other application tasks here */
    }

    /* Disable rt_OneStep() here */
    return 0;
}

/*
 * File trailer for generated code.
 *
 * [EOF]
 */

/*
 * File: FLGateController.c
 *
 * Code generated for Simulink model 'FLGateController'.
 *
 * Model version                : 1.2
 * Simulink Coder version        : 9.0 (R2018b) 24-May-2018
 * C/C++ source code generated on : Thu Nov 9 15:12:23 2023
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Atmel->AVR (32-bit)

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		15

```

* Code generation objectives:
*   1. Execution efficiency
*   2. RAM efficiency
* Validation result: Not run
*/

#include "FLGateController.h"

/* Private macros used by the generated code to access rtModel */
#ifndef rtmIsMajorTimeStep
#define rtmIsMajorTimeStep(rtm)      (((rtm)->Timing.simTimeStep) == MA-
JOR_TIME_STEP)
#endif

#ifndef rtmIsMinorTimeStep
#define rtmIsMinorTimeStep(rtm)     (((rtm)->Timing.simTimeStep) == MI-
NOR_TIME_STEP)
#endif

#ifndef rtmSetTPtr
#define rtmSetTPtr(rtm, val)        ((rtm)->Timing.t = (val))
#endif

/* Real-time model */
RT_MODEL rtM_;
RT_MODEL *const rtM = &rtM_;

/* Model step function */
void FLGateController_step(void)
{
/* Update absolute time for base rate */
/* The "clockTick0" counts the number of times the code of this task has
* been executed. The absolute time is the multiplication of "clockTick0"
* and "Timing.stepSize0". Size of "clockTick0" ensures timer will not
* overflow during the application lifespan selected.
*/
rtM->Timing.t[0] =
    (++rtM->Timing.clockTick0) * rtM->Timing.stepSize0;

{
/* Update absolute timer for sample time: [0.042016806722689079s, 0.0s]
*/
/* The "clockTick1" counts the number of times the code of this task has
* been executed. The resolution of this integer timer is
0.042016806722689079, which is the step size
* of the task. Size of "clockTick1" ensures timer will not overflow dur-
ing the
* application lifespan selected.
*/
rtM->Timing.clockTick1++;
}
}

/* Model initialize function */
void FLGateController_initialize(void)
{
/* Registration code */
{
/* Setup solver object */
rtsiSetSimTimeStepPtr(&rtM->solverInfo, &rtM->Timing.simTimeStep);
rtsiSetTPtr(&rtM->solverInfo, &rtM->Timing.t);
}
}

```

СКБФЭУ.1.ИП.01000000

Лист

Изм. Лист. № документа Подп. Дата.

16



```

    rtsiSetStepSizePtr(&rtM->solverInfo, &rtM->Timing.stepSize0);
    rtsiSetErrorStatusPtr(&rtM->solverInfo, ((const char_T **)
        (&rtMGetErrorStatus(rtM))));
    rtsiSetRTModelPtr(&rtM->solverInfo, rtM);
}

rtsiSetSimTimeStep(&rtM->solverInfo, MAJOR_TIME_STEP);
rtsiSetSolverName(&rtM->solverInfo, "FixedStepDiscrete");
rtmSetTPtr(rtM, &rtM->Timing.tArray[0]);
rtM->Timing.stepSize0 = 0.042016806722689079;
}

/*
 * File trailer for generated code.
 *
 * [EOF]
 */

/*
 * File: FLGateController.h
 *
 * Code generated for Simulink model 'FLGateController'.
 *
 * Model version                : 1.2
 * Simulink Coder version       : 9.0 (R2018b) 24-May-2018
 * C/C++ source code generated on : Thu Nov 9 15:12:23 2023
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Atmel->AVR (32-bit)
 * Code generation objectives:
 *   1. Execution efficiency
 *   2. RAM efficiency
 * Validation result: Not run
 */

#ifndef RTW_HEADER_FLGateController_h_
#define RTW_HEADER_FLGateController_h_
#include "rtwtypes.h"
#ifndef FLGateController_COMMON_INCLUDES_
#define FLGateController_COMMON_INCLUDES_
#include "rtwtypes.h"
#include "rtw_continuous.h"
#include "rtw_solver.h"
#endif /* FLGateController_COMMON_INCLUDES_
*/

/* Macros for accessing real-time model data structure */
#ifndef rtmGetErrorStatus
#define rtmGetErrorStatus(rtm) ((rtm)->errorStatus)
#endif

#ifndef rtmSetErrorStatus
#define rtmSetErrorStatus(rtm, val) ((rtm)->errorStatus = (val))
#endif

#ifndef rtmGetT
#define rtmGetT(rtm) (rtmGetTPtr((rtm))[0])
#endif

#ifndef rtmGetTPtr
#define rtmGetTPtr(rtm) ((rtm)->Timing.t)

```

Изм.	Лист.	№ документа	Подп.	Дата.

СКБФЭУ.1.ИП.01000000

Лист

17

```

#endif

/* Forward declaration for rtModel */
typedef struct tag_RTM RT_MODEL;

#ifndef DEFINED_TYPEDEF_FOR_struct_D0u0d7bkNR3XquPeJ1bGjB_
#define DEFINED_TYPEDEF_FOR_struct_D0u0d7bkNR3XquPeJ1bGjB_

typedef struct {
    uint8_T SimulinkDiagnostic;
    uint8_T Model[16];
    uint8_T Block[40];
    uint8_T OutOfRangeInputValue;
    uint8_T NoRuleFired;
    uint8_T EmptyOutputFuzzySet;
} struct_D0u0d7bkNR3XquPeJ1bGjB;

#endif

#ifndef DEFINED_TYPEDEF_FOR_struct_RCPloovUDF2zF6C9vcJmtG_
#define DEFINED_TYPEDEF_FOR_struct_RCPloovUDF2zF6C9vcJmtG_

typedef struct {
    uint8_T type[5];
    int32_T origTypeLength;
    real_T params[3];
    int32_T origParamLength;
} struct_RCPloovUDF2zF6C9vcJmtG;

#endif

#ifndef DEFINED_TYPEDEF_FOR_struct_UC9pSe5wBJPvWoG496e2WC_
#define DEFINED_TYPEDEF_FOR_struct_UC9pSe5wBJPvWoG496e2WC_

typedef struct {
    struct_RCPloovUDF2zF6C9vcJmtG mf[2];
    int32_T origNumMF;
} struct_UC9pSe5wBJPvWoG496e2WC;

#endif

#ifndef DEFINED_TYPEDEF_FOR_struct_hlXSzWrKgWOpSQjHCTSmS_
#define DEFINED_TYPEDEF_FOR_struct_hlXSzWrKgWOpSQjHCTSmS_

typedef struct {
    uint8_T type[7];
    uint8_T andMethod[3];
    uint8_T orMethod[3];
    uint8_T defuzzMethod[3];
    uint8_T impMethod[3];
    uint8_T aggMethod[3];
    real_T inputRange[6];
    real_T outputRange[2];
    struct_UC9pSe5wBJPvWoG496e2WC inputMF[3];
    struct_UC9pSe5wBJPvWoG496e2WC outputMF;
    real_T antecedent[24];
    real_T consequent[8];
    real_T connection[8];
    real_T weight[8];
    int32_T numSamples;
}

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		18

```

int32_T numInputs;
int32_T numOutputs;
int32_T numRules;
int32_T numInputMFs[3];
int32_T numCumInputMFs[3];
int32_T numOutputMFs;
int32_T numCumOutputMFs;
real_T outputSamplePoints[101];
int32_T orrSize[2];
int32_T aggSize[2];
} struct_hlXSzWrKgWOpSQjHCTSmS;

#endif

/* Custom Type definition for MATLAB Function: '<S1>/Evaluate Rule Conse-
quents' */
#ifndef struct_tag_sDfM9XCdU9DLkphJPFDDd11C
#define struct_tag_sDfM9XCdU9DLkphJPFDDd11C

struct tag_sDfM9XCdU9DLkphJPFDDd11C
{
    uint8_T type[5];
    int32_T origTypeLength;
    real_T params[3];
    int32_T origParamLength;
};

#endif /*struct_tag_sDfM9XCdU9DLkphJPFDDd11C*/

#ifndef typedef_sDfM9XCdU9DLkphJPFDDd11C
#define typedef_sDfM9XCdU9DLkphJPFDDd11C

typedef struct tag_sDfM9XCdU9DLkphJPFDDd11C sDfM9XCdU9DLkphJPFDDd11C;

#endif /*typedef_sDfM9XCdU9DLkphJPFDDd11C*/

#ifndef struct_tag_sMFrsvRoXd0x6pLc3r0utC
#define struct_tag_sMFrsvRoXd0x6pLc3r0utC

struct tag_sMFrsvRoXd0x6pLc3r0utC
{
    sDfM9XCdU9DLkphJPFDDd11C mf[2];
    int32_T origNumMF;
};

#endif /*struct_tag_sMFrsvRoXd0x6pLc3r0utC*/

#ifndef typedef_sMFrsvRoXd0x6pLc3r0utC
#define typedef_sMFrsvRoXd0x6pLc3r0utC

typedef struct tag_sMFrsvRoXd0x6pLc3r0utC sMFrsvRoXd0x6pLc3r0utC;

#endif /*typedef_sMFrsvRoXd0x6pLc3r0utC*/

#ifndef struct_tag_sLAKYRwLBORj81JHJQCKABC
#define struct_tag_sLAKYRwLBORj81JHJQCKABC

struct tag_sLAKYRwLBORj81JHJQCKABC
{
    uint8_T type[7];

```

Изм.	Лист.	№ документа	Подп.	Дата.

СКБФЭУ.1.ИП.01000000

Лист

19

```

uint8_T andMethod[3];
uint8_T orMethod[3];
uint8_T defuzzMethod[3];
uint8_T impMethod[3];
uint8_T aggMethod[3];
real_T inputRange[6];
real_T outputRange[2];
SMFRsvoRoXd0x6pLc3rOutC inputMF[3];
SMFRsvoRoXd0x6pLc3rOutC outputMF;
real_T antecedent[24];
real_T consequent[8];
real_T connection[8];
real_T weight[8];
int32_T numSamples;
int32_T numInputs;
int32_T numOutputs;
int32_T numRules;
int32_T numInputMFs[3];
int32_T numCumInputMFs[3];
int32_T numOutputMFs;
int32_T numCumOutputMFs;
real_T outputSamplePoints[101];
int32_T orrSize[2];
int32_T aggSize[2];
};

#endif /*struct_tag_sLAKYrWLBORj81JHJQCKABC*/

#ifndef typedef_sLAKYrWLBORj81JHJQCKABC
#define typedef_sLAKYrWLBORj81JHJQCKABC

typedef struct tag_sLAKYrWLBORj81JHJQCKABC sLAKYrWLBORj81JHJQCKABC;

#endif /*typedef_sLAKYrWLBORj81JHJQCKABC*/

/* Custom Type definition for MATLAB Function: '<S1>/Evaluate Rule Anteced-
ents' */
#ifndef struct_tag_szswZam3rGKiBZJCIP3eaqB
#define struct_tag_szswZam3rGKiBZJCIP3eaqB

struct tag_szswZam3rGKiBZJCIP3eaqB
{
    uint8_T SimulinkDiagnostic;
    uint8_T Model[16];
    uint8_T Block[40];
    uint8_T OutOfRangeInputValue;
    uint8_T NoRuleFired;
    uint8_T EmptyOutputFuzzySet;
};

#endif /*struct_tag_szswZam3rGKiBZJCIP3eaqB*/

#ifndef typedef_szswZam3rGKiBZJCIP3eaqB
#define typedef_szswZam3rGKiBZJCIP3eaqB

typedef struct tag_szswZam3rGKiBZJCIP3eaqB szswZam3rGKiBZJCIP3eaqB;

#endif /*typedef_szswZam3rGKiBZJCIP3eaqB*/

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		20

```

/* Custom Type definition for MATLAB Function: '<S1>/Evaluate Rule Conse-
quents' */
#ifndef struct_tag_sXsSeVB4E0ntMvFGNG3D1eC
#define struct_tag_sXsSeVB4E0ntMvFGNG3D1eC

struct tag_sXsSeVB4E0ntMvFGNG3D1eC
{
    uint8_T type[5];
    real_T params[3];
};

#endif /*struct_tag_sXsSeVB4E0ntMvFGNG3D1eC*/

#ifndef typedef_sXsSeVB4E0ntMvFGNG3D1eC
#define typedef_sXsSeVB4E0ntMvFGNG3D1eC

typedef struct tag_sXsSeVB4E0ntMvFGNG3D1eC sXsSeVB4E0ntMvFGNG3D1eC;

#endif /*typedef_sXsSeVB4E0ntMvFGNG3D1eC*/

/* Real-time Model Data Structure */
struct tag_RTM {
    const char_T * volatile errorStatus;
    RTWSolverInfo solverInfo;

    /*
     * Timing:
     * The following substructure contains information regarding
     * the timing information for the model.
     */
    struct {
        uint32_T clockTick0;
        time_T stepSize0;
        uint32_T clockTick1;
        SimTimeStep simTimeStep;
        time_T *t;
        time_T tArray[2];
    } Timing;
};

/* Model entry point functions */
extern void FLGateController_initialize(void);
extern void FLGateController_step(void);

/* Real-time Model object */
extern RT_MODEL *const rtM;

/*-
 * These blocks were eliminated from the model due to optimizations:
 *
 * Block '<Root>/Scope' : Unused code path elimination
 * Block '<S1>/InputConversion' : Eliminate redundant data type conversion
 */

/*-
 * The generated code includes comments that allow you to trace directly
 * back to the appropriate location in the model. The basic format
 * is <system>/block_name, where system is the system number (uniquely
 * assigned by Simulink) and block_name is the name of the block.
 */

```

Изм.	Лист.	№ документа	Подп.	Дата.

СКБФЭУ.1.ИП.01000000

Лист

21

```

* Use the MATLAB hilite_system command to trace the generated code back
* to the model. For example,
*
* hilite_system('<S3>') - opens system 3
* hilite_system('<S3>/Kp') - opens and selects block Kp which resides in S3
*
* Here is the system hierarchy for this model
*
* '<Rest>' : 'FLGateController'
* '<S1>' : 'FLGateController/Fuzzy Logic Controller'
* '<S2>' : 'FLGateController/Fuzzy Logic Controller/Defuzzify Outputs'
* '<S3>' : 'FLGateController/Fuzzy Logic Controller/Evaluate Rule Ante-
cedents'
* '<S4>' : 'FLGateController/Fuzzy Logic Controller/Evaluate Rule Conse-
quents'
*/
#endif
/* RTW_HEADER_FLGateController_h_ */

/*
* File trailer for generated code.
*
* [EOF]
*/

/*
* File: rtwtypes.h
*
* Code generated for Simulink model 'FLGateController'.
*
* Model version : 1.2
* Simulink Coder version : 9.0 (R2018b) 24-May-2018
* C/C++ source code generated on : Thu Nov 9 15:12:23 2023
*
* Target selection: ert.tlc
* Embedded hardware selection: Atmel->AVR (32-bit)
* Code generation objectives:
* 1. Execution efficiency
* 2. RAM efficiency
* Validation result: Not run
*/

#ifndef RTWTYPES_H
#define RTWTYPES_H

/* Logical type definitions */
#if (!defined(__cplusplus))
# ifndef false
# define false (0U)
# endif

# ifndef true
# define true (1U)
# endif
#endif

/*=====
* Target hardware information
* Device type: Atmel->AVR (32-bit)
* Number of bits: char: 8 short: 16 int: 32
* long: 32 long long: 64
* native word size: 32
*/

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		22

```

* Byte ordering: LittleEndian
* Signed integer division rounds to: Zero
* Shift right on a signed integer as arithmetic shift: on
*=====*/

/*=====
* Fixed width word size data types:
* int8_T, int16_T, int32_T - signed 8, 16, or 32 bit integers
* uint8_T, uint16_T, uint32_T - unsigned 8, 16, or 32 bit integers
* real32_T, real64_T - 32 and 64 bit floating point numbers
*=====*/

typedef signed char int8_T;
typedef unsigned char uint8_T;
typedef short int16_T;
typedef unsigned short uint16_T;
typedef int int32_T;
typedef unsigned int uint32_T;
typedef long long int64_T;
typedef unsigned long long uint64_T;
typedef float real32_T;
typedef double real64_T;

/*=====
*
* Generic type definitions: boolean_T, char_T, byte_T, int_T, uint_T,
*
* real_T, time_T, ulong_T, ulonglong_T.
*=====
*/

typedef double real_T;
typedef double time_T;
typedef unsigned char boolean_T;
typedef int int_T;
typedef unsigned int uint_T;
typedef unsigned long ulong_T;
typedef unsigned long long ulonglong_T;
typedef char char_T;
typedef unsigned char uchar_T;
typedef char_T byte_T;

/*=====
* Min and Max:
* int8_T, int16_T, int32_T - signed 8, 16, or 32 bit integers
* uint8_T, uint16_T, uint32_T - unsigned 8, 16, or 32 bit integers
*=====*/

#define MAX_int8_T ((int8_T) (127))
#define MIN_int8_T ((int8_T) (-128))
#define MAX_uint8_T ((uint8_T) (255U))
#define MAX_int16_T ((int16_T) (32767))
#define MIN_int16_T ((int16_T) (-32768))
#define MAX_uint16_T ((uint16_T) (65535U))
#define MAX_int32_T ((int32_T) (2147483647))
#define MIN_int32_T ((int32_T) (-2147483647-1))
#define MAX_uint32_T ((uint32_T) (0xFFFFFFFFU))
#define MAX_int64_T ((int64_T) (9223372036854775807LL))
#define MIN_int64_T ((int64_T) (-9223372036854775807LL-1LL))
#define MAX_uint64_T ((uint64_T) (0xFFFFFFFFFFFFFFFFULL))

/* Block D-Work pointer type */

```

Изм.	Лист.	№ документа	Подп.	Дата.

СКБФЭУ.1.ИП.01000000

Лист

23

```
typedef void * pointer_T;
```

```
#endif
```

```
/* RTWTYPES_H */
```

```
/*
```

```
 * File trailer for generated code.
```

```
 *
```

```
 * [EOF]
```

```
 */
```

Изм.	Лист.	№ документа	Подп.	Дата.

СКБФЭУ.1.ИП.01000000

Лист

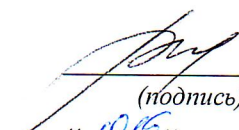
24



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

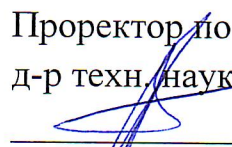
СОГЛАСОВАНО

Начальник отдела ОНиПКРС


  
(подпись) Е.М. Димитриади  
« 06 » 06 2024 г.

УТВЕРЖДАЮ

Проректор по научной работе,  
д-р техн. наук, профессор

  
(подпись) А.В. Космынин  
« 06 » 06 2024 г.

Декан ФЭУ

  
(подпись) А.С. Гудим  
« 06 » 06 2024 г.

АКТ

о приемке в эксплуатацию проекта  
«Система дискретно-событийного управления технологическими  
операциями»

г. Комсомольск-на-Амуре

« 06 » 06 2024 г.

Комиссия в составе представителей:

со стороны заказчика

- С.И. Сухоруков – руководитель СКБ,
- А.С. Гудим – декан ФЭУ

со стороны исполнителя

- А.В. Бузикаева – руководителя проекта,
- А.С. Савельева ЗАУМ-1 – наставник проекта,
- Т.Е. Безнаев – студент лицея при ФГБОУ ВО «КНАГУ» – исполнитель проекта

составила акт о нижеследующем:

«Исполнитель» передает проект «Система дискретно-событийного управления технологическими операциями», в составе:

«Исполнитель» передает управляющую программу для автоматизированной/роботизированной системы «Система дискретно-событийного управления технологическими операциями», в составе:


Программное обеспечение:

- Текст управляющей программы

Эксплуатационная документация:

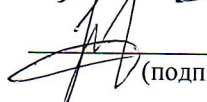
- Паспорт управляющей программы для автоматизированной/роботизированной системы

Руководитель проекта

  
06.06.24  
(подпись, дата)

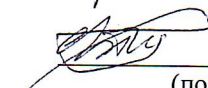
А.В. Бузикаева

Наставник проекта

  
06.06.24  
(подпись, дата)

А.С. Савельева

Исполнитель проекта

  
06.06.24  
(подпись, дата)

Т.Е. Безнаев