



УТВЕРЖДАЮ

Генеральный директор
ООО «КОМИТАС»

_____/ Л.А. Григорян

СИСТЕМА ЕДИНОВРЕМЕННОГО УПРАВЛЕНИЯ ГРУППОЙ
АВТОНОМНЫХ МОБИЛЬНЫХ РОБОТОВ
RMSCOMITAS (ROBOT MANAGEMENT SYSTEM COMITAS)

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК

ЛИСТ УТВЕРЖДЕНИЯ

РОФ.ЛТПГ.00001-01 90 ЛУ

СОГЛАСОВАНО

Руководитель группы разработки
робототехнических систем

П.В. Икомасов

«___» _____ 20__ г.

Директор по инновациям,
научный руководитель НИОКР

М.Н. Кочетков

«___» _____ 20__ г.

Москва, 2024г.

Име. №подл.	Подпись и дата	Взам. име. №	Име. №дубл.	Подпись и дата

УТВЕРЖДЕНО

РОФ.ЛТПГ.00001-01 90 ЛУ

СИСТЕМА ЕДИНОВРЕМЕННОГО УПРАВЛЕНИЯ ГРУППОЙ
АВТОНОМНЫХ МОБИЛЬНЫХ РОБОТОВ
RMSCOMITAS (ROBOT MANAGEMENT SYSTEM COMITAS)

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК

РОФ.ЛТПГ.00001-01 90

Инв. №подл.	Подпись и дата	Взам. инв. №	Инв. №дубл.	Подпись и дата

2024

АННОТАЦИЯ

Документ «Описание функциональных характеристик» предназначен для сотрудников эксплуатирующей организации и содержит перечень функциональных возможностей Программного обеспечения для единовременного управления группой автономных мобильных роботов.

Документ актуален для указанной версии и для последующих версий вплоть до выпуска обновления описания функциональных характеристик.

Выход новой версии программного обеспечения сопровождается обновлением данного документа только в случае наличия в версии значительных изменений в перечне функциональных возможностей Программного обеспечения. Если таких изменений версия не содержит, то остается актуальным описание функциональных характеристик от предыдущей версии с учетом изменений, содержащихся в новой версии.

СОДЕРЖАНИЕ

АННОТАЦИЯ	3
СОДЕРЖАНИЕ	4
1 ВВЕДЕНИЕ	5
2 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ	6
2.1 Цели и назначение	6
2.2 Ключевые функции.....	6
3 ИНФОРМАЦИЯ НЕОБХОДИМАЯ ДЛЯ УСТАНОВКИ И ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	9
3.1 Системные требования.....	9
3.2 Требования к квалификации персонала	9
4 ПОДГОТОВКА К РАБОТЕ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ	10
4.1 Предварительные условия для установки.....	10
4.2 Подготовка рабочей станции разработчика к установке ПО.....	10
4.2.1 Установка Ubuntu 22.04	10
4.2.2 Установка python3	11
4.2.3 Установка необходимых пакетов.....	12
4.2.4 Предварительная настройка рабочей станции.....	13
4.3 Подготовка целевого сервера к установке ПО	14
4.3.1 Установка Ubuntu 22.04	14
4.3.2 Установка необходимых пакетов.....	14
4.3.3 Предварительная настройка целевого сервера	14
4.4 Настройка сервера RMS с рабочей станции разработчика	15
4.5 Деплой системы RMS на сервер	16
5 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА	18

1 ВВЕДЕНИЕ

Настоящий документ предназначен для сотрудников эксплуатирующей организации, в должностные обязанности которых входят функции использованию или администрированию программы единовременного управления группой автономных мобильных роботов.

2 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

2.1 Цели и назначение

Программное обеспечение для единовременного управления группой складских автономных мобильных роботов предназначено для автоматизации и оптимизации процессов на складе, таких как:

- распределение складских единиц по заданным параметрам;
- мониторинг и отслеживание местоположения и статуса автономных мобильных роботов;
- сбор отчетности о работе роя роботов в рамках поставленных задач.

Цели программного обеспечения:

- координация и управление работой множества роботов для обеспечения эффективного и безопасного выполнения задач на складе;
- оптимизация маршрутов и задач для каждого робота для уменьшения времени и расстояния, необходимых для выполнения задач;
- отслеживание состояния автономных мобильных роботов и зарядных устройств для обеспечения бесперебойной работы;
- отслеживание местоположения и статус каждого робота, а также мониторинг их работы в реальном времени;
- обеспечение безопасности в рамках предотвращения столкновения и других опасных ситуаций;
- интеграция с системами верхнего уровня (опционально).

2.2 Ключевые функции

Внутреннее программное обеспечение робота позволяет обеспечить следующие основные функции, направленные на работоспособность робота:

- Управление движением и навигацией:

Встроенное ПО АМР отвечает за управление движением робота, включая его скорость, ускорение, направление движения и маневрирование в пространстве. Оно обеспечивает внутреннюю навигацию робота с использованием различных технологий, таких как различные датчики и устройства локализации на

местности, позволяя роботу перемещаться по заданной траектории и оптимально следовать заложенному пути.

– Оперативное реагирование на внешние факторы:

Встроенное ПО АМР, обеспечивающее обработку данных сенсоров, установленных на роботе, позволяет осуществлять распознавание и интерпретацию внешних воздействий и окружающей среды. Это позволяет роботу обнаруживать препятствия, определять условия работы и принимать решения на основе полученных данных.

– Алгоритмы обеспечения безопасности:

Внутреннее программное обеспечение робота включает в себя алгоритмы, позволяющие ему более оперативно принимать различные решения на основе текущего состояния и заданных целей. К примеру, осуществлять экстренное торможение при определении препятствия, не дожидаясь отклика от сервера. Данные алгоритмы обеспечивают безопасность и эффективность работы автономного мобильного робота.

– Внутренние регуляторы движения:

Внутренние регуляторы движения позволяют роботу придерживаться заложенных в него траекторий и путей следования даже при возможных сбоях в физике робота. К примеру, если робот по тем или иным причинам, осуществил сход с траектории, данные внутренние алгоритмы способствуют оперативному возвращению робота на нее.

– Взаимодействие со смежными системами:

Встроенное ПО АМР обеспечивает возможность коммуникации и взаимодействия мобильного робота со смежными системами. К примеру, данное взаимодействие может осуществляться с системой единовременного управления группой автономных мобильных роботов (RMS), а также с внутренним ПО зарядной станции.

– Диагностические утилиты:

Встроенное ПО АМР предоставляет инструменты для диагностики текущего состояния автономного мобильного робота. К примеру, при возникновении

проблем или неисправностей, робот может дать световой либо звуковой сигнал, а также отправить информацию о проблеме оператору либо на сервер управления.

– Сбор, запись и хранение информации (логирование) о процессе функционирования:

Встроенное ПО АМР осуществляет постоянное логирование действий автономного мобильного робота, что позволяет осуществлять отладку внутренних процессов, а также оперативно детектировать возможные проблемы. Помимо этого, благодаря логированию может быть осуществлен сбор внутренних метрик, позволяющий определить эффективность бизнес-процессов.

– Удаленный доступ и управление:

Встроенное ПО АМР предоставляет пользователю и оператору различные средства удаленного доступа и контроля. Это может быть применено для обновления программного обеспечения, настройки параметров работы робота, управления каждым отдельным роботом вручную.

3 ИНФОРМАЦИЯ НЕОБХОДИМАЯ ДЛЯ УСТАНОВКИ И ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Системные требования

Специализированное программное обеспечение «RMSCOMITAS», созданное для координации функционирования роя автономных мобильных складских роботов рекомендуется разворачивать на серверном оборудовании, имеющем следующие параметры:

- поддерживаемые ОС - Ubuntu 22.04 LTS и старше;
- объем жесткого диска: 1тб или более;
- процессор: Intel(R) Core (TM) i7-12700H 2.30 GHz и выше;
- количество CUDA ядер: от 6000;
- оперативная память: 32gb или более;
- сетевая карта: 10gb+ ethernet x2 (in, out);
- бесперебойность работы: 10 минут без питания.

3.2 Требования к квалификации персонала

Для работы с программным обеспечением роботов необходимо разделение пользователей на:

- Оператор:

Должен обладать компетенциями работы с автономными мобильными роботами. Должен курировать операционные процессы сортировки.

- Инженер информационных систем:

Должен отвечать за взаимодействие системы RMS с внешними системами типа WMS, WCS, ERP, а также решать возможные проблемы в системах связи либо в системах интеграции.

- Администратор:

Обладает максимальным уровнем доступа к ПО, имеет возможность осуществлять корректировку информации и расширенные настройки системы (при взаимодействии с сервисной службой).

4 ПОДГОТОВКА К РАБОТЕ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ

4.1 Предварительные условия для установки

Установка системы RMS необходима для ее последующей настройки и организации процессов штатной работы. *Установка и настройка системы должны осуществляться специалистом компании разработчика.* Возможность самостоятельной установки конечным пользователем возможна только в случае постоянного курирования процессов со стороны компании разработчика.

Для корректной установки системы необходимо следующее оборудование:

- целевой сервер, на который происходит установка системы;
- рабочая станция разработчика, с которой осуществляется процесс непосредственной установки;
- флеш-накопитель для предустановки операционной системы.

4.2 Подготовка рабочей станции разработчика к установке ПО

Перед проведением процедуры установки системы RMS необходимо осуществить предварительную настройку рабочей станции разработчика. На ней должно быть предустановлено следующее ПО:

- операционная система Ubuntu 22.04;
- пакет python3;
- пакет python3-pip;
- пакет python3-venv;
- пакет openssh-client;
- пакет git.

4.2.1 Установка Ubuntu 22.04

Для установки операционной системы Ubuntu 22.04 необходимо проделать следующие действия:

На официальном сайте <https://releases.ubuntu.com/22.04/> необходимо скачать образ операционной системы для рабочего стола.

Затем необходимо записать скачанный ранее образ на флеш-накопитель и сделать его загрузочным. Для записи образа необходимо использовать специальное ПО, предназначенное для данных целей. К примеру, для Ubuntu linux для данных целей можно использовать утилиту Etcher.

После создания загрузочной флешки, необходимо загрузиться с нее и, при отображении загрузочного экрана, соответствующего рисунку 1, нажать на «Try or install ubuntu» и следовать дальнейшим указаниям инсталлятора.

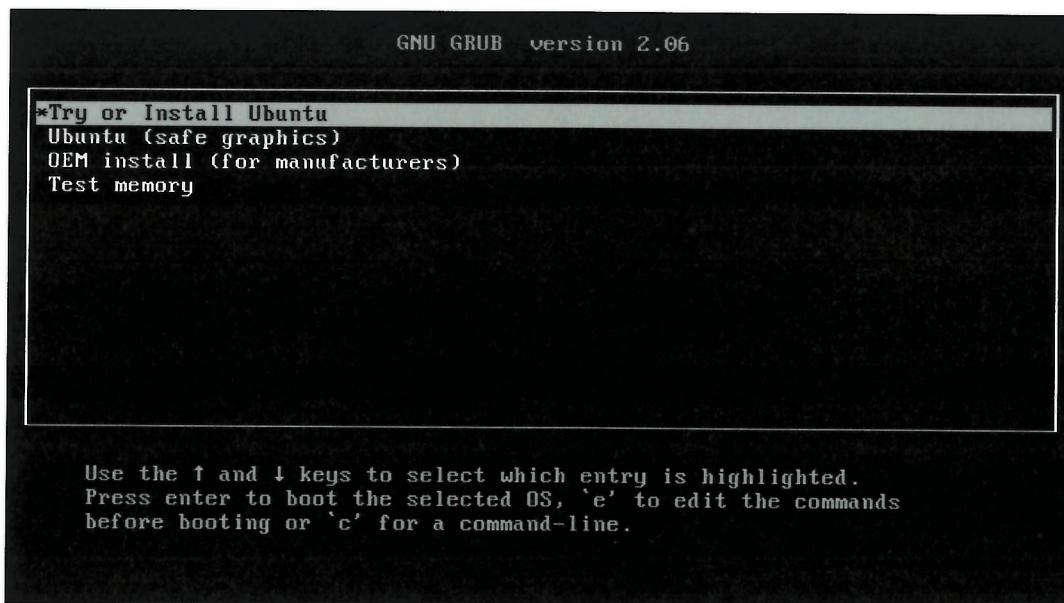


Рисунок 1 – Загрузочный экран установки ОС Ubuntu

После установки системы необходимо создать пользователя, подключить рабочую станцию к интернету и приступить к установке необходимых пакетов.

4.2.2 Установка python3

Перед установкой пакета python3 необходимо проверить его наличие в системе. Делается это командой «python3» в терминале. Если пакет установлен в системе, то результат выполнения команды выведет приветственное сообщение, схожее с тем, что представлено на рисунке 2.

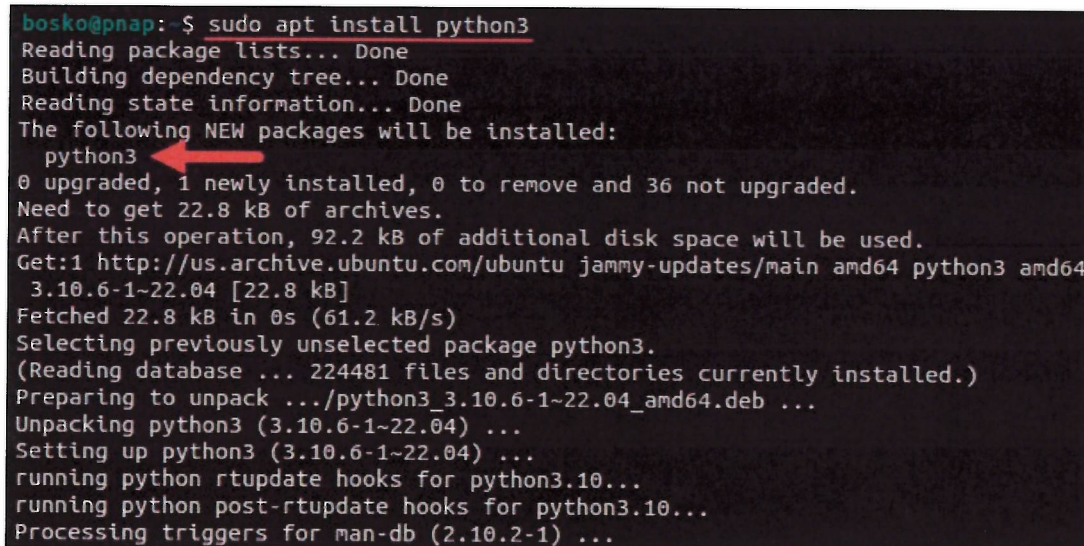
```
bosko@pnap:~$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
bosko@pnap:~$
```

Рисунок 2 – Сообщение наличия пакета python3 в системе

В случае отсутствия данного пакета, необходимо его установить, выполнив в терминале последовательно команды.

- sudo apt update;
- sudo apt install python3;

При успешной установке пакета терминал выведет результат, схожий с рисунком 3.



```
bosko@pnap: ~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
python3
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 22.8 kB of archives.
After this operation, 92.2 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3 amd64
3.10.6-1-22.04 [22.8 kB]
Fetched 22.8 kB in 0s (61.2 kB/s)
Selecting previously unselected package python3.
(Reading database ... 224481 files and directories currently installed.)
Preparing to unpack .../python3_3.10.6-1-22.04_amd64.deb ...
Unpacking python3 (3.10.6-1-22.04) ...
Setting up python3 (3.10.6-1-22.04) ...
running python rtupdate hooks for python3.10...
running python post-rtupdate hooks for python3.10...
Processing triggers for man-db (2.10.2-1) ...
```

Рисунок 3 – Установка пакета python3 в системе

4.2.3 Установка необходимых пакетов

Пакет python3-pip является инструментом, который помогает устанавливать необходимые для проекта библиотеки и модули и управлять ими.

Для установки пакета python3-pip необходимо убедиться, что пакет python3 (п. 4.2.2.) уже установлен в системе. После этого необходимо в терминале выполнить следующую команду:

- sudo apt install -y python3-pip

Пакет python3-venv является виртуальной средой, которая обеспечивает изолированное пространство для проектов Python на рабочей станции. Это позволяет обеспечить стабильность среды разработки и бесперебойность работы основной системы.

Для установки пакета python3-venv также необходимо убедиться, что пакет python3 уже установлен в системе. После этого необходимо в терминале выполнить следующую команду:

- sudo apt install -y python3-venv

Пакет `openssh-client` является пакетом, позволяющим осуществлять подключение по `ssh` для удаленного доступа к серверу и проведения различных работ на нем.

Для установки пакета `openssh-client` необходимо в терминале выполнить следующую команду:

- `sudo apt install openssh-client`

Пакет `git` является пакетом, позволяющим осуществлять контроль версий, отслеживать и управлять изменениями в коде или образе системы RMS. Также данный пакет позволяет работать с репозиторием, в котором содержатся файлы, необходимые для установки системы RMS

Для установки пакета `git` необходимо в терминале выполнить следующую команду:

- `sudo apt install git`

После установки всех данных пакетов можно приступить к непосредственной настройке рабочей станции.

4.2.4 Предварительная настройка рабочей станции

Для осуществления предварительной настройки рабочей станции необходимо выполнить следующие действия:

- Убедиться, что рабочая станция подключена к сети, имеющей выход в интернет.

- Клонировать репозиторий `rms-server` при помощи пакета `git`. Для этого необходимо создать соответствующую папку на рабочей станции, открыть в этой папке терминал и выполнить команду:

- `git clone https://URL/rms-server.git`

В данной команде `URL` является адресом сайта, на котором расположен репозиторий `rms-server`. Адрес сайта и доступ к нему есть у разработчиков системы.

- Перейти в терминале в корень скопированного репозитория и выполнить там следующие команды:

- python3 -m venv venv
- . venv/bin/activate
- python3 -m pip install -r requirements.txt

После выполнения данных команд настройка рабочей станции будет завершена.

4.3 Подготовка целевого сервера к установке ПО

Также перед проведением процедуры установки системы RMS необходимо осуществить предварительную настройку целевого сервера. На нем должно стоять следующее ПО:

- Ubuntu 22.04;
- openssh-server.

4.3.1 Установка Ubuntu 22.04

Для установки пакета Ubuntu 22.04 следует воспользоваться инструкцией из пункта 4.2.1. Принцип установки полностью совпадает за исключением того, что при создании пользователя необходимо создать пользователя с именем rms, для которого доступно повышение привилегий в системе (вызов sudo).

4.3.2 Установка необходимых пакетов

Пакет openssh-server является пакетом, позволяющим осуществлять допускать к себе подключение клиентов по ssh. В данном случае клиентом является рабочая станция, на которой установлен пакет openssh-client

Для установки пакета openssh- server необходимо в терминале выполнить следующую команду:

- sudo apt install openssh- server.

4.3.3 Предварительная настройка целевого сервера

Для осуществления предварительной настройки сервера необходимо выполнить следующие действия:

- Убедиться, сервер подключен к сети, имеющей выход в интернет.

– Убедиться, что сервер и рабочая станция подключены к одной и той же сети.

– Убедиться, что рабочая станция имеет возможность стабильно подключиться к серверу по ssh.

4.4 Настройка сервера RMS с рабочей станции разработчика

Перед началом работы следует получить ключевую пару ssh (ansible_id_ed25519, ansible_id_ed25519.pub) для дальнейшего доступа к серверу и сохранить её локально на рабочей станции разработчика в директории ~/.ssh/ с учётом требуемых прав доступа.

На рабочей станции разработчика следует открыть файл deploy/inventory.yml в данном репозитории и внести следующие изменения:

– в секцию hosts добавить новую секцию с названием сервера с суффиксом bare (название можно выбрать любое);

– в новой секции указать следующие пары ключей-значений:

- ansible_host: - IP-адрес сервера;
- ansible_connection: ssh (тип подключения, всегда указывать ssh)
- ansible_user: rms - имя пользователя на сервере;
- ansible_password: - пароль пользователя rms на сервере;
- ansible_become_pass - пароль пользователя rms на сервере.

– добавить ещё одну секцию без суффикса bare;

– в ней указать следующие пары ключей-значений:

- ansible_host: - IP-адрес сервера (тот же, что и выше);
- ansible_connection: ssh (тип подключения, всегда указывать ssh)
- ansible_user: ansible-deploy - имя пользователя на сервере (обязательно ansible-deploy).

Пример файла с корректно добавленными секциями представлен на рисунке 4.

```
all:
  hosts:
    deploy-server-bare:
      ansible_host: 10.77.0.94
      ansible_connection: ssh
      ansible_user: rms
      ansible_password: myVerySecretPassword01
      ansible_become_pass: myVerySecretPassword01
    deploy-server:
      ansible_host: 10.77.0.94
      ansible_connection: ssh
      ansible_user: ansible-deploy
```

Рисунок 4 – Корректный файл с добавленными секциями

Затем, в корне репозитория на рабочей станции разработчика выполнить команды:

- . venv/bin/activate
- export ANSIBLE_HOST_KEY_CHECKING=False
- ansible-playbook -i deploy/inventory.yml -l deploy-server-bare deploy/bootstrap.yml
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/docker.yml
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/nvidia-cuda.yml
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/nvidia-docker.yml
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/mosquitto.yml
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/xvfb-remote.yml

После успешного выполнения последней команды сервер должен перезагрузиться, следует дождаться его полной загрузки перед продолжением.

4.5 Деплой системы RMS на сервер

Для деплоя RMS требуется добавить в директорию deploy файл deploy.yml из RMS.

Предварительно разработчику требуется получить токен для доступа в хранилище контейнеров.

Затем на рабочей станции разработчика необходимо выполнить следующие команды в корне репозитория:

- . venv/bin/activate
- export ANSIBLE_HOST_KEY_CHECKING=False
- export CI_JOB_TOKEN="токен для доступа в хранилище образов (CI_REGISTRY)"
- export CI_REGISTRY="git-registry.comitas.ru:5000" # адрес хранилища образов
- export CI_REGISTRY_IMAGE="git-registry.comitas.ru:5000/robotics/rms" # базовый адрес образа
- export CI_COMMIT_REF_SLUG="devel" # здесь должна быть указана версия RMS
- ansible-playbook -i deploy/inventory.yml -l deploy-server deploy/deploy.yml

После выполнения последней команды запустится процедура установки системы RMS на сервер. После ее завершения и получении сообщения об успешном деплое необходимо перезагрузить сервер, дождаться его загрузки. Затем можно приступить к запуску и настройке системы RMS.

5 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Вопросы возникающие в ходе работы с программным обеспечением следует направлять в службу поддержки по адресу info@comitas.ru.

Все обращения рассматриваются в рабочее время (Европе GMT+3), ответы и оказание поддержки в штатном режиме предоставляются не позднее 1 часа с момента обращения.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (стр.) в документе	№ документа	Входящий № сопроводительного документа	Подпись	Дата
	измененных	Замененных	новых	Аннулированных					

СОСТАВИЛИ

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпи сь	Дата

СОГЛАСОВАНО

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпи сь	Дата