

Порядок создания новых роботов для ДинРобот-3

Общие положения

Программный комплекс «ДинРобот-3» позволяет создавать системы управления для различных роботов, в т.ч. созданных силами других производителей.

Программный комплекс «ДинРобот-3» рассматривает робота, как совокупность различных аппаратных устройств. Для каждого такого устройства в программном комплексе должен быть предусмотрен драйвер. Для большинства таких устройств программный комплекс имеет встроенный драйвер. Для тех устройств, для которых нет встроенного драйвера, можно написать DLL-библиотеку, в которой реализовать управление этим неизвестным устройством.

Аппаратные устройства в программном комплексе «ДинРобот-3» объединяются в логические устройства, такие как:

- моторы;
- сенсорные датчики;
- устройства управления подсветкой;
- лидары;
- камеры ил камеры глубины;
- дальномеры.

Эти логические устройства ссылаются на физический драйвер и его канал управления, реализуя для них преобразование сырых данных в удобные для использования значения. Например, пересчет сырых показаний датчиков положения моторов в градусы или сантиметры.

Кроме того, объединение драйверов в логические устройства позволяет создать сквозную нумерацию для каждого класса устройств. Например, пусть драйвер устройства 0 позволяет управлять тремя моторами с номерами каналов 0, 1 и 2. А драйвер устройства 1 позволяет управлять двумя моторами с номерами каналов 0 и 1. За счет объединения драйверов в логические моторы можно создать сквозную нумерацию моторов от 0 по 4, при этом моторы 0, 1 и 2 будут связаны с каналами 0, 1 и 2 устройства 0, а моторы 3 и 4 будут связаны с каналами 0 и 1 устройства 1.

Логические моторы могут объединяться в устройства более высокого уровня, такие как:

- шасси;
- голова;
- лицо робота;
- манипуляторы.

Такое объединение позволяет абстрагироваться от вопросов управления конкретными моторами, а управлять совокупностью моторов, как единым устройством.

Файл hwconfig.txt

Файл «hwconfig.txt» в программном комплексе «ДинРобот-3» задает аппаратную конфигурацию робота. Запущенный программный комплекс «ДинРобот-3» отслеживает измерения в этом файле и пересчитывает его «налету» без перезагрузки программы в течение 4-5 секунд.

Для создания нового робота необходимо собрать файл аппаратной конфигурации «hwconfig.txt» из кусков блоков, расположенных в папке «RobotConfigs/CONFIG_BLOCKS/HW_CONFIG_TXT». Для этого следует использовать любой текстовый редактор.

Рекомендуется начать разработку «hwconfig.txt» со вставки содержимого файла «robot.txt»:

```
[РОБОТ]
; Название робота
model = Маша

; иконка робота 256x256 из папки WEBControl/robots
icon = masha.jpg

; (bool) производить калибровку при старте ДинРобот-3
CALIB_AT_START = false
```

Здесь следует указать название модели Вашего робота (любой текст), создать файл с иконкой, положить ее в папку «WEBControl/robots», а параметре «icon» указать имя файла этой иконки. Параметр «CALIB_AT_START» рекомендуется временно сделать «false». Это предотвратит запуск моторов робота, прежде чем разработчик правильно установит все их параметры. На время разработки рекомендуется калибровать каждый мотор отдельно, задавая для него оптимальные параметры конфигурации, и лишь затем разрешать калибровку всего робота при старте.

Камеры (если есть)

У робота может быть одна или несколько камер. Это могут быть USB-камеры, IP-камеры или RTSP-камеры (RTSP-камеры поддерживаются только под Windows).

Так называемая DEVICE-камера подразумевает, что драйвер этой камеры реализован в драйвере устройства (см.ниже). Для такой камеры указывается номер устройства из списка hwconfig.txt и номер канала.

Также камера может быть камерой симулятора робота. В этом случае картинка с камеры будет создаваться путем 3D-рендера по средствам OpenGL.

Для каждой камеры в создаваемый файл «hwconfig.txt» следует добавить текстовый блок соответствующей камеры из файла подпапки «Cameras»:

– usb-camera.txt

- ip-camera.txt
- rtsp-camera.txt
- device-camera.txt
- emulator-camera.txt

```

...
;-----
; USB-Камера
;-----
[CAMERA0]

; Название камеры в интерфейсе управления робота
NAME = Лицевая камера

; Тип
TYPE = USB

; (Только windows) часть названия камеры или пусто,
; если камера обнаруживается по HW INDEX
; Поставьте знак "?". В этом случае камера будет обнаружена по HW_INDEX,
; а в HW_NAME будет вписано ее название
HW_NAME = C920

; (int) (только windows) аппаратный номер камеры или -1 (не используется,
если камера обнаруживается по HW_NAME, но может подстраховать)
HW_INDEX = 0

; (Только linux) название видеоустройства
DEVICE = /dev/video0

; (int) ширина кадра в видеорежиме
WIDTH = 640

; (int) высота кадра в видеорежиме
HEIGHT = 480

; (int) сколько пустых кадров делать перед тем, как сделать фото (в некоторых
камер первые фото после включения плохого качества)
FRAMES_BEFORE_PHOTO = 0

; (float) угол обзора по горизонтали в градусах (внимание, нужно измерять,
т.к. в документации приводится обычно угол обзора по диагонали, и то
неправильный)
FOV = 45.0

; (float) масштаб по глубине для камер глубины, или 0,
; если это не камера глубины
DEPTH_SCALE = 0.0

; (bool) Использовать ли эту камеру глубины для обнаружения препятствий
USE_DEPTH_CAMERA_FOR_OBSTACLES = false

; (bool) перевернуть ли картинку на 180 градусов
ROT180 = 0

; Перечень номеров логических камер, с которыми данная камера
; не совместима (через запятую или пробел)
UNCOMPATIBLE_WITH =

```

```

;----- кинематика -----

;(bool) Признак того, что камера на подвижной голове
ON_THE_HEAD = true

;(bool) Признак того, что камера за подъемником
ON_THE_LIFT = true

; матрица положения камеры на роботе, относительно последней оси
; вращения головы (если ON_THE_HEAD) или начала координат робота.
; Задается в виде:
;   1 число - положение по Z (см)
;   3 числа - X,Y,Z (см)
;   4 числа - X,Y,Z, OX, OY ,OZ (см, градусы)
;   12 чисел - матрица 4x3
;   16 чисел - матрица 4x4
; Матрица должна приводить к ориентации камеры, смотрящей вдоль оси Y,
; ось X - вправо по изображению, ось Z вверх по изображению
MATRIX = 0 7.0 100.0

```

Блоки камер следует пронумеровать. Т.е. вместо «CAMERAх» вписать «CAMERA0», «CAMERA1» и т.д. это будут логические номера камер в системе. Логические номера не обязательно должны совпадать с физическими номерами камер в системе.

Для USB-камер под Windows в поле HW_NAME следует указать часть полного названия камеры. Например, если камера называется «HD WEBCAM C920 PRO», достаточно указать «C920». При запуске ДинРобот-3 пишет в консоли названия камер, которых он обнаруживает в системе.

Если по HW_NAME камера не обнаруживается, то следует указать параметр HW_INDEX – порядковый номер камеры (при запуске ДинРобот-3 в консоли нумерует камеры). При этом если в параметре «HW_NAME» указать значение «?», то ДинРобот-3 сам при запуске камеры впишет в «hwconfig.txt» название камеры.

Если на роботе одноименные камеры, то обнаруживать их придется только по HW_INDEX. Параметр HW_NAME при этом нужно сделать пустой строкой. К сожалению, в последнее время производители стали выпускать камеры, которые работают под управлением Universal Camera Driver. Такие камеры имеют одно и то же название, даже для камер разных производителей.

Под Linux параметры HW_INDEX и HW_NAME игнорируются. Вместо них следует указать параметр DEVICE, в котором следует указывать название видеоприбора, например «/dev/video0».

Параметрами WIDTH и HEIGHT следует задать разрешение, в котором камера будет работать в видеорежиме. Для Windows разрешение кадра в фоторежиме определяется автоматически. Для Linux размер кадра видеоизображения следует задавать параметрами PHOTO_WIDTH, PHOTO_HEIGHT.

Отдельно следует сказать про параметр `UNCOMPATIBLE_WITH`. В этом параметре следует через пробел или запятую перечислить логические номера камер, с которыми данная камера несовместима. Дело в том, что далеко не все USB-камеры могут одновременно работать. Скажем больше, обычно две камеры одного и того же производителя обычно друг с другом одновременно не работают. По мнению некоторых экспертов, проблема в том, что камеры не могут поделить между собой пропускную возможность USB, по мнению автора статьи, проблема в криво написанных драйверах, т.к. почему-то две камеры двух разных производителей обычно могут поделить между собой USB, а две одинаковые камеры – не могут.

Так или иначе, если две камеры не могут работать друг с другом одновременно, их следует указать в параметрах `UNCOMPATIBLE_WITH`. Т.е. если камера 0 не работает одновременно с камерой 1, то в параметре `UNCOMPATIBLE_WITH` камеры 0 следует указать 1, а в параметре `UNCOMPATIBLE_WITH` камеры 1 следует указать 0.

Параметр `ON_THE_HEAD` (камера на голове) обязательно требуется выставлять, в противном случае алгоритмы управления головой, включая ручное управление, будут неправильно ей управлять.

Параметр `ON_THE_LIFT` (камера на подъемнике) определяет, стоит ли камера после подъемного устройства робота (у большинства роботов такого подъемника нет, поэтому значение этого параметра не играет никакой роли). В противном случае при расчете кинематики подъем подъемника будет добавляться к координате Z камеры.

Для камер глубины важно указать параметр `USE_DEPTH_CAMERA_FOR_OBSTACLES`, определяющий, следует ли данную камеру глубины использовать для обнаружения препятствий на маршруте. Для обычных камер значение этого параметра игнорируется.

Для тех камер, с которых возможно будет рассчитываться кинематика (пересчет 3D-координат), например, для камер глубины, важно указать параметры подсекции «кинематика».

Аудиосистема

Параметры аудиосистемы в файл «`hwconfig.txt`» следует вставить из файла «`audio.txt`». Эти параметры универсальны для 99.9% систем.

Драйвера устройств

В создаваемый файл «`hwconfig.txt`» следует добавить блоки драйверов устройств, из которых состоит аппаратная часть робота.

Перечень встроенных в «ДинРобо-3» устройств находится в подпапке: «`ROBOT_CONFIGS/HW_CONFIG/Devices`».

Если же разработчику робота необходимо создать драйвер под своё собственное устройство, необходимо для этого разработать драйвер в формате DLL-файла. Исходный код и примером создания такой DLL лежит в папке «`DevicePlugin`». Разработав такой драйвер, его также можно

подключить к «ДинРобот-3», указав в качестве параметра «type» в секции нужного устройства значение «mydriver.dll» – название DLL-файла разработанного драйвера.

Построение робота предлагается рассмотреть на примере.

Допустим, что некий гипотетический робот состоит из одной платы «MASHA2024», которая управляет лифтом (канал 2), поворотом головы (канал 3) и наклоном головы (канал 4), а вместо колес используются мотор-колеса от гироскутеров проекта VESC Tool. Колеса соединены по CAN-шине и подключены по одному USB-порту. Основное колесо левое (CAN_ID=0), правое колесо ведомое (CAN_ID=1).

Для рассматриваемого гипотетического робота в «hwconfig.txt» необходимо вставить блоки из файлов подпапки «Devices»:

- Masha2024.txt;
- VESC.txt.

Если драйвер устройства написан с помощью DLL-файла, то для регистрации этого драйвера используйте текстовый блок из файла «dll.txt».

```
...
[DEVICE0]
;-----
; Плата Masha2024 производство ИП Евстигнеев Дмитрий Валерьевич
; Совместима с Masha1
;-----
; Моторы:
; Канал 0 - левое колесо   speed,pwm,sensor 16 bit
; Канал 1 - правое колесо speed,pwm,sensor 16 bit
; Канал 2 - подъемник (если есть)      calib,pos,pwm,sensor 16 bit
; Канал 3 - поворот головы calib,pos,pwm,sensor 16 bit
; Канал 4 - наклон головы (если есть)  calib,pos,pwm,sensor 16 bit
; Дальномеры:
; Каналы 0-3
; Датчики:
; Канал 0 - зарядка (показание АЦП 16)
; Канал 1 - подключение зарядки (0 или 1)
; Подсветка (если есть):
; канал 0 - RGB
; Мимика, если есть HW_LIPS
;-----
; тип
TYPE = MASHA2024

; COM-порт
COMM = COM3

; (bool) есть ли подъемник
HAS_LIFT = true

; (bool) есть ли наклон головы
HAS_PITCH = true

; (bool) Есть ли аппаратные губы
HW_LIPS = false

; (int) Положение нуля поворота головы (метки)
```

```

HEAD_STOPPER_POS = 1200

; (int) ШИМ калибровки поворота головы (0-127)
HEAD_CALIB_PWM = 30

; (int) Положение нуля наклона головы (метки)
PITCH_STOPPER_POS = -900

; (int) ШИМ калибровки наклона головы (0-127)
PITCH_CALIB_PWM = -25

; (int) ШИМ калибровки подъемника (0-127)
LIFT_CALIB_PWM = -70

[DEVICE1]
;-----
; Колеса VESC. Опционально колеса могут быть соединены по CAN-шине
;-----
; Моторы:
; Канал 0 - основное колесо  pwm,speed,sensor 32 bit
; Канал 1..255 - колеса, соединенные по CAN-шине, номер
;                      канала=CAN_ID  pwm,speed,sensor 32 bit
;-----
; тип
TYPE = VESC

; COM-порт (скорость 115200)
COMM = COM4

; (bool) Использовать аппаратную поддержку скорости
; (обычно не работает на малых оборотах)
USE_HW_RPM_CONTROL = false

; (float) Пропорциональный коэффициент программного ПИ-регулятора скорости
PI_P = 10.0

; (float) Интегральный коэффициент программного ПИ-регулятора скорости
PI_I = 20.0

; (int) максимальное значение DutyCycle (умноженное на 100000)
MAX_DUTY = 10000

; (int) максимальное значение RPM (градусы в секунду при
; программной поддержки, обороты в минуту при оригинальной
; аппаратной поддержке)
MAX_RPM = 900

```

Вставляя блоки из файлов, необходимо следить за нумерацией устройств, заменяя «DEVICE x » на «DEVICE0» и «DEVICE1» соответственно. Также следует следить и при необходимости исправлять остальные параметры этих устройств (в частности исправлять нумерацию COM-портов).

Логические моторы (если есть)

Далее следует определить логические моторы, для каждого из них рекомендуется использовать текстовый блок из файла «motor.txt»:

```

...
;-----
; Логический мотор 0
;-----
[MOTOR0]

; (устройство:канал), управляющий мотором
DEVICE = 1:0

; имя в интерфейсе
NAME = Левое колесо

; (bool) Признак мотора колеса
WHEEL_MOTOR = true

; (bool) имеется ли управление по скорости
HAS_SPEED = true

; (bool) имеется ли управление по ШИМ
HAS_PWM = true

; (bool) имеется ли калибровка
HAS_CALIB = false

; (int) сколько битный сенсор
SENSOR_BIT_COUNT = 16

;-----
; Логический мотор 1
;-----
[MOTOR1]

; (устройство:канал), управляющий мотором
DEVICE = 1:1

; имя в интерфейсе
NAME = Правое колесо

; (bool) Признак мотора колеса
WHEEL_MOTOR = true

; (bool) имеется ли управление по скорости
HAS_SPEED = true

; (bool) имеется ли управление по ШИМ
HAS_PWM = true

; (bool) имеется ли калибровка
HAS_CALIB = false

; (int) сколько битный сенсор
SENSOR_BIT_COUNT = 16

;-----
; Логический мотор 2
;-----
[MOTOR2]

; (устройство:канал), управляющий мотором
DEVICE = 0:2

; имя в интерфейсе
NAME = Лифт

```



```

; (bool) Признак мотора колеса
WHEEL_MOTOR = false

; (int) показание датчика мотора на нуле
SENSOR0 = 0

; (float) значение на нуле в градусах или сантиметрах
VALUE0 = 0.0

; (int) показание датчика в отличном от нуля положении,
; заданный относительно SENSOR0, для удобства коррекции нуля
SENSOR1+ = 8337

; (float) значение в градусах или сантиметрах, соответствующее SENSOR1
VALUE1 = 20.0

; (int) ограничение на перемещение. мин.значение датчика
MIN_SENSOR = 0

; (int) ограничение на перемещение. макс.значение датчика
MAX_SENSOR = 8337

; (float) люфт в градусах или сантиметрах
LUFT = 0.0

; (bool) люфт зависит от гравитации, при этом значение 0,
; является центром гравитации
HAS_GRAVILUFT = false

; (int) период (мс) отключения после начала движения путем ШИМ=0.
; Применяется, только если HAS_PWM
; При значении 0 не применяется.
; Игнорируется для WHEEL_MOTOR=true
OFF_PERIOD = 2000

; (bool) имеется ли управление по положению
HAS_POSITION = true

; (bool) имеется ли управление по скорости
HAS_SPEED = false

; (bool) имеется ли управление по ШИМ
HAS_PWM = true

; (bool) имеется ли калибровка
HAS_CALIB = true

; (int) сколько битный сенсор
SENSOR_BIT_COUNT = 16

;-----
; Логический мотор 3
;-----
[MOTOR3]

; (устройство:канал), управляющий мотором
DEVICE = 0:3

; имя в интерфейсе
NAME = Поворот головы

; (bool) признак мотора шасси
WHEEL_MOTOR = false

```

```

; (int) показание сенсора на нуле
SENSOR0 = 0

; (float) значение на нуле в градусах или сантиметрах
VALUE0 = 0.0

; (int) показание сенсора в отличном от нуля положении
SENSOR1+ = 1400

; (float) значение в градусах или сантиметрах, соответствующее SENSOR1
VALUE1 = 45.0

; (int) ограничение на перемещение. мин.значение сенсора
MIN_SENSOR = -1200

; (int) ограничение на перемещение. макс.значение сенсора
MAX_SENSOR = 1200

; (float) люфт в градусах или сантиметрах
LUFT = 0.0

; (bool) люфт зависит от гравитации, при этом значение 0, является центром
гравитации
HAS_GRAVILUFT = 0

; (bool) имеется ли управление по положению
HAS_POSITION = 1

; (bool) имеется ли управление по скорости
HAS_SPEED = 0

; (bool) имеется ли управление по ШИМ
HAS_PWM = 1

; (bool) имеется ли калибровка
HAS_CALIB = 1

; (int) сколько битный сенсор
SENSOR_BIT_COUNT = 16

OFF_PERIOD = 8000

;-----
; Логический мотор 4
;-----
[MOTOR4]

; (устройство:канал), управляющий мотором
DEVICE = 0:4

; имя в интерфейсе
NAME = Наклон головы

; (bool) признак мотора шасси
WHEEL_MOTOR = false

; (int) показание сенсора на нуле
SENSOR0 = 0

; (float) значение на нуле в градусах или сантиметрах
VALUE0 = 0.0

```

```

; (int) показание сенсора в отличном от нуля положении
SENSOR1+ = 1400

; (float) значение в градусах или сантиметрах, соответствующее SENSOR1
VALUE1 = 45.0

; (int) ограничение на перемещение. мин.значение сенсора
MIN_SENSOR = -800

; (int) ограничение на перемещение. макс.значение сенсора
MAX_SENSOR = 800

; (float) люфт в градусах или сантиметрах
LUFT = 0.0

; (bool) люфт зависит от гравитации, при этом значение 0,
; является центром гравитации
HAS_GRAVILUFT = 5

; (bool) имеется ли управление по положению
HAS_POSITION = 1

; (bool) имеется ли управление по скорости
HAS_SPEED = 0

; (bool) имеется ли управление по ШИМ
HAS_PWM = 1

; (bool) имеется ли калибровка
HAS_CALIB = 1

; (int) сколько битный сенсор
SENSOR_BIT_COUNT = 16

OFF_PERIOD = 4000

```

Следует обратить внимание на последовательность нумерации логических моторов «[MOTOR0]», «[MOTOR1]» и т.д., а также на параметр «DEVICE» в каждой записи о логическом моторе, который должен ссылаться на конкретное устройство («DEVICEх», вставленное в «hwconfig.txt» ранее) и конкретный канал этого устройства.

Дальномеры (если есть)

Для каждого дальномера в файл «hwconfig.txt» следует вставить текстовый блок из файла «rangefinder.txt».

```

...
;-----
; Дальномер
;-----
[RANGEFINDER0]

; (device:channel) Устройство и номер канала дальномера
device = 0:0

; (int) Длина последовательности фильтра скользящего среднего
filter = 2

```

```

; (float) Минимальное показание (см), ниже которого дальномер
; считается дефектным и показания игнорируются
minValue = 3.0

; (float) Максимальная дальность (см), свыше которой показания похожи на шум
maxValue = 100.0

; (float) Координата X (см) размещения дальномера на роботе
x = -22.0

; (float) Координата Y (см) размещения дальномера на роботе
y = 0.0

; (float) Угол (градусы) ориентации дальномера относительно
; робота (0 - вперед, слева-направо)
angle = -90.0

; (float) Угол обзора (сканирования) дальномера (градусы)
fov = 30.0

; (bool) Используется для навигации
useForNavigation = true

; (bool) Используется для обнаружения препятствий
useForObstacles = true

; (bool) Используется как датчик обнаружения лестниц (смотрит в пол)
useForStairDetector = false

; (float) Мин.граница нормального показания дальномера-детектора
; лестниц (см)
stairDetectorMinValue = 5.0

; (float) Макс.граница нормального показания дальномера-детектора
; лестниц (см)
stairDetectorMinValue = 15.0

```

При вставке блока следует заменять «RANGEFINDERx» на «RANGEFINDER0», «RANGEFINDER1» и т.д., это будут логические номера дальномеров.

Для каждого дальномера следует установить ссылку «device», в которой следует указать номер устройства и номер канала конкретного дальномера.

Важно указать параметры «x», «y» и «angle», определяющие положение дальномера на роботе.

Параметр «useForNavigation» определяет, используется ли дальномер для навигации. Для навигации используется строго боковые дальномеры и строго передние дальномеры. Периферийные дальномеры лучше не использовать для навигации – данные от них неоднозначны, что сбивает навигацию. Кроме того, чем больше дальномеров, тем больше от них ложных срабатываний.

Остальные параметры дальномера следует установить согласно комментариям к этим параметрам.

Сенсоры (если есть)

К сенсорам не относятся датчики обратной связи на моторах, лидары и дальномеры. Остальные датчики, измеряющие состояние робота, относятся к понятиям «сенсор». Например, к сенсорам относятся датчики заряда батареи и датчик подключения зарядки.

Для каждого такого сенсора следует в файле «hwconfig.txt» вставить текстовый блок из файла «sensor.txt».

Во вставленном блоке вместо «[SENSORx]», следует задать нумерацию «[SENSOR0]», «[SENSOR1]». Эти номера будут являться логическим номерами сенсоров.

В каждой записи сенсора следует указать параметр «DEVICE» в виде ссылки на устройство и его канал этого устройства, отвечающего за конкретный физический сенсор.

Остальные параметры сенсоров следует установить согласно комментариям к этим параметрам.

Лидары (если есть)

Если на роботе есть лидары, то для каждого из них следует вставить в «hwconfig.txt» текстовый блок из файла lidar.txt. При этом нужно заменить «[LIDARx]» на «[LIDAR0]», «[LIDAR1]» и т.д.

В каждом блоке лидара следует указать параметр «DEVICE», в котором следует указать ссылку на устройство и канал драйвера устройства, управляющего лидаром.

Остальные параметры лидаров следует установить согласно комментариям к этим параметрам.

Шасси (если есть)

Далее имеет смысл определить для робота шасси, если у робота есть шасси. Для этого следует использовать текстовый блок из файла «wheels.txt».

```
...
;-----
;
; Шасси
;
;-----
[WHEELS]

; Тип шасси
;TYPE=NONE
TYPE=2WHEELS
;TYPE=3ROT
;TYPE=3OMNI
;TYPE=4OMNI
;TYPE=DRIVER

; (bool) Поддерживается ли режим стрейфа
CAN_STRAFE = false

; (bool) Поддерживается ли отдельный режим поворотов на месте
CAN_TURN = true

; (bool) Поддерживается ли режим автоматического движения
```

```

CAN_AUTOMAT = true

; (устройство) Номер устройства управления шасси, только если
; TYPE=DRIVER, или -1
DEVICE = -1

; (float) Полурастояние между боковыми колесами (см)
WHEEL_X = 17.2

; (float) положение колес по Y относительно центра робота (см):
; для трехколесных шасси колес 0 и 2, для четырехколесных шасси колес 0 и 1,
; для других - игнорируется
WHEEL_Y0 = 0

; (float) положение колес по Y относительно центра робота (см):
; для трехколесных шасси колеса 1, для четырехколесных шасси колес 2 и 3,
; для других - игнорируется
WHEEL_Y1 = -20.0

; (float) сколько сантиметров хода на один тик датчика обратной связи
CM_PER_TICK = 0.02252

...

; коэф.коррекции показания одонометрии по Y
Y_CORRECTION_K = 1.0

; коэф.коррекции показания одонометрии по X
X_CORRECTION_K = 1.0

; коэф.коррекции показания одонометрии по углу поворота
A_CORRECTION_K = 1.0

; (int) логический номер мотора колеса 0 (не используется при TYPE=DRIVER)
MOTOR0 = 0

; (float) коэф.передачи мотора 0 для движения вперед
; (коррекция неоднородности, может быть отрицательный)
MOTOR0_K = 1.0

; (float) коррекция неоднородности хода назад для мотора 0,
; относительно хода вперед (величина положительная)
MOTOR0_BACK_K = 1.0

; (int) логический номер мотора колеса 1 (не используется при TYPE=DRIVER)
MOTOR1 = 1

; (float) коэф.передачи мотора 1 для движения
; вперед (коррекция неоднородности, может быть отрицательный)
MOTOR1_K = 1.0

; (float) коррекция неоднородности хода назад для мотора 1,
; относительно хода вперед (величина положительная)
MOTOR1_BACK_K = 1.0

```

Здесь, в принципе, каждый параметр подписан. Следует пояснить лишь некоторые параметры.

Параметр TYPE:

<i>TYPE</i>	<i>Описание</i>
NONE	Значение по умолчанию, означает, что шасси отсутствует.
2WHEELS	Шасси, состоящее из двух ведущих колёс (правое и левое), а также одного или двух опорных колёс. Либо двух гусениц (левой и правой).
3ROT	Шасси, состоящее из трёх колёс, каждое колесо и ведущее, и поворотное.
3OMNI	Шасси, состоящее из трёх OMNI-колёс.
4OMNI	Шасси, состоящее из четырёх OMNI-колёс.
DRIVER	Шасси управляется непосредственно через драйвер одного из устройств без использования промежуточной сущности «мотор». Такое шасси у роботов R.Bot100 и KIKI.

Задавая тип DRIVER, разработчик робота должен указать в параметре DEVICE номер устройства, управляющего шасси. Моторы при этом указывать не следует.

В противном случае следует указать номера логических моторов, входящих в состав шасси, с помощью параметров «MOTOR0», «MOTOR1» и т.д.

Параметрами «MOTOR0_K», «MOTOR1_K» и т.д. следует задать коэффициент передачи от мотора до колеса. Обычно 1, но может быть и (-1), если колесо крутится в обратном направлении относительно движения шасси вперед. Кроме того, этими параметрами можно компенсировать неоднородность скоростей вращения колёс.

В некоторых редких случаях (обычно из-за дефектов датчиков колёс) может получиться так, что колесо вперед и назад вращается с разной скоростью. Это можно компенсировать параметрами «MOTOR0_BACK_K», «MOTOR1_BACK_K» и т.д. Нормальное значение этого коэффициента равно 1.0. Но можно задать большее или меньшее положительное значение для компенсации неоднородности скоростей вперед и назад.

Важным коэффициентом является CM_PER_TICK. Здесь следует задать, сколько сантиметров проходит шасси за одну метку датчика шасси.

Задавая геометрию расположения колёс параметрами WHEEL_X, WHEEL_Y0, WHEEL_Y1, разработчик определяет базовую модель расчета кинематики шасси. Однако может получиться так, что показания одонометрии не совпадает с реальным перемещением робота по карте. Скорректировать показания одонометрии можно коэффициентами Y_CORRECTION_K, X_CORRECTION_K, A_CORRECTION_K.

Голова (если есть)

Далее имеет смысл определить для робота параметры головы, если у робота есть поворотная голова. Для этого следует использовать текстовый блок из файла «head.txt». Параметры в этом текстовом блоке следует установить согласно комментариям к этим параметрам.

Система питания (если есть)

Если робот способен определяет свой уровень заряда батареи, то для него следует в файле «hwconfig.txt» прописать блок системы питания. Текст этого блока следует скопировать из файла «power.txt».

```

;-----
;
; Система питания
;
;-----
[POWER_SYSTEM]

; (int) номер лог.сенсора показания заряда батареи
BATTERY_SENSOR = 0

; (int) номер лог.сенсора датчика подключения зарядки
CHARGE_SENSOR = 1

; (float) минимальный критический уровень заряда батареи,
; при котором нужно срочно искать зарядку
MIN_POWER_LEVEL = 20.0

```

Здесь в параметре BATTERY_SENSOR следует указать логический номер сенсора, отвечающего за определение уровня заряда батареи.

Если у робота есть сенсор подключения зарядного устройства, то логический номер этого сенсора следует указать в параметре «CHARGE_SENSOR».

Манипуляторы (если есть)

Если у робота есть манипуляторы, то для каждого из них следует использовать текстовый блок из файла «arm.txt».

```

...
;-----
;
; Рука робота
;
;-----
[ARM0]

; человеческое название
NAME = Манипулятор

; перечень логических моторов, относящихся к данной руке, через пробел или
любой другой разделитель
MOTORS = 1 2 3 4 5 6

;-----
; матрица крепления нулевого звена руки. Разделители между числами любые
; Если задано:
;   1 число - положение по Z (см),
;   3 числа - x,y,z (см).
;   6 чисел - x,y,z (см), ox,oy,oz (градусы)
;   12 чисел - матрица 4x3
;   16 чисел - матрица 4x4
MATRIX0 = 0 0 0

```



```

;----- звенья -----
; (int) номер логического мотора звена
JOINT0_MOTOR = 1

; ось вращения или перемещения
JOINT0_AXIS = OZ

; (float) коэффициент передачи (может быть отрицательный)
JOINT0_K = 1.0

; матрица звена после мотора. Формат тот же, что и у параметра MATRIX0
JOINT0_MATRIX = 100.0

; (int) номер логического мотора звена
JOINT1_MOTOR = 2

; ось вращения или перемещения
JOINT1_AXIS = OX

; коэффициент передачи (может быть отрицательный)
JOINT1_K = 1.0

; матрица звена после мотора. Формат тот же, что и у параметра MATRIX0
JOINT1_MATRIX = 100.0 0 0
;
;
; ...
;
;-----

; (bool) может ли осуществлять линейную интерполяцию (только поддержка
драйвером)
CAN_LINEAR_INTERPOLATION = true

; (bool) может ли определять свое местоположение (программным или аппаратным
способом)
CAN_KINEMATIC = true

; (bool) может ли производить расчет обратной кинематики
CAN_IK = true

; (bool) драйвер может сам производить расчет обратной кинематики
CAN_DRIVER_IK = true

; (bool) драйвер сам поддерживает поименованные жесты
CAN_DRIVER_GESTURES = false

; (устройство:канал), если драйвер сам поддерживает управления
; кинематикой всей руки
KINEMATIC_DEVICE = 0:0

; имя файла хранения жестов данной руки
GESTURE_FILE = arm0_gestures.txt

```

Руки следует нумеровать ARM0, ARM1 и т.д. У робота может быть до 16 рук.

Для каждой руки следует определить перечень логических моторов, участвующих в кинематике данной руки. Этот перечень следует указать в параметре «MOTORS», разделяя номера моторов пробелами или запятыми. Мотор следует включать в перечень, если его можно отнести к понятию «Жест рукой». Например, моторы пальцев робота-аниматроника имеет смысл включать в перечень, т.к. они являются непосредственным элементом жеста. А вот мотор захватного устройства манипуляционного робота включать в данный перечень неудобно, т.к. для него жест – это скорее точка, в которой он может прибывать как в состоянии закрытого, так и открытого захватного устройства.

Если манипулятор робота имеет встроенную систему управления (например, как устройство «xArm»), то для него следует указать параметр «KINEMATIC_DEVICE=0:0». В этом параметре следует номер драйвера устройства и номер канала, через который производится управление манипулятором. Драйвер устройства может самостоятельно производить расчет прямой и обратной задачи кинематики, а также реализовывать движение в обобщенной и линейной системе координат.

Если же у манипулятора нет встроенной системы управления, то строчку «KINEMATIC_DEVICE» следует удалить или закомментировать. В этом случае управлением звеньями будет заниматься сам «ДинРобот-3».

Если управление манипуляторами не подразумевает решения прямой и обратной задач кинематики, то нет смысла указывать параметры MATRIX0 и JOINTx_xxx.

Если же планируется решение прямой и обратной задач кинематики (например, за счет использования функций движения к точкам, указанным в мировой (линейной) системе координат), то для манипулятора следует указать матрицы преобразования звеньев.

В параметре MATRIX0 следует указать матрицу преобразования точки крепления манипулятора к роботу. Эту матрицу можно задать с помощью:

- 1 числа (подразумевается положение по Z, см);
- с помощью 3 чисел через пробел или запятую (подразумевается X, Y, Z, см);
- с помощью 6 чисел – подразумевается X,Y,Z (см), OX,OY,OZ (градусы);
- с помощью 12 чисел – подразумевается матрица 4x3;
- с помощью 16 чисел – подразумевается матрица 4x4.

Параметром JOINT n _MOTOR (где n – номер звена: 0,1,2,...) следует задать номер логического мотора, который управляет звеном n .

Параметром JOINT n _AXIS (где n – номер звена: 0,1,2,...) следует задать ось вращения или перемещения, по которой происходит движение с помощью мотора. Следует указывать:

- X – линейное перемещение вдоль оси X в системе координат текущего звена.
- Y – линейное перемещение вдоль оси Y в системе координат текущего звена.

- Z – линейное перемещение вдоль оси Z в системе координат текущего звена.
- OX – вращение вокруг оси X в системе координат текущего звена.
- OY – вращение вокруг оси Y в системе координат текущего звена.
- OZ – вращение вокруг оси Z в системе координат текущего звена.

Параметром `JOINT n _MATRIX` (где n – номер звена: 0,1,2,...) следует указывать матрицу преобразования звена (до точки крепления следующего звена или конца рабочего органа манипулятора). Формат тот же, что и для параметра `MATRIX0`.

Параметров `JOINT n _xxx` должно быть столько, сколько звеньев у робота.

Также для руки следует указать ее возможности.

Параметр `CAN_LINEAR_INTERPOLATION` (bool) задает возможность реализации линейного перемещения рабочего органа манипулятора в мировой системе координат. Линейное перемещение не поддерживается самим «ДинРобот-3». Реализация такого движения возможна, только если за движение отвечает сам драйвер устройства.

`CAN_KINEMATIC` – (bool) задает возможность решения прямой задачи кинематики аппаратным способом (с помощью драйвера) или программным способом (по средствам «ДинРобот-3»). В последнем случае указание параметров `JOINT n _MATRIX` и `MATRIX0` является обязательным.

`CAN_IK` – (bool) задает возможность решения обратной задачи кинематики аппаратным способом (с помощью драйвера) или программным способом (по средствам «ДинРобот-3»). В последнем случае указание параметров `JOINT n _MATRIX` и `MATRIX0` является обязательным.

`CAN_DRIVER_IK` – (bool) флаг указывает, что аппаратный драйвер поддерживает решение обратной задачи кинематики. В противном случае обратная задача кинематики решается по средствам «ДинРобот-3».

`CAN_DRIVER_GESTURES` – (bool) указывает, что аппаратный драйвер поддерживает поименованные жесты на аппаратном уровне. Т.е. сам драйвер запоминает точки (жесты) под определенным названием. И впоследствии драйверу можно дать команду двигаться в точку (жест), указанный по его названию.

`GESTURE_FILE` – (string) здесь следует указать файл, в котором хранятся жесты и их координаты. «ДинРобот-3» отслеживает изменения в этом файле и при необходимости перечитывает этот файл «налету». Рекомендуются для каждого робота, а иногда даже каждого мероприятия указывать отдельный файл (для удобства создания резервных копий).

Подсветка

В программном комплексе «ДинРобот-3» имеется возможность управления подсветкой робота. Этой подсветкой можно управлять из скрипта, а можно привязать к:

- эмоциям робота;
- конкретному мотору робота;
- шасси робота.

В случае с привязкой управление подсветкой производится автоматически.

Для каждого устройства подсветки в файле «hwconfig.txt» следует задать текстовый блок из файла:

- «LEDS/led.txt» – если это простая подсветка, управляемая скриптами робота;
- «LEDS/led-emotion.txt» – если подсветка связана с эмоциями робота;
- «LEDS/led-motor.txt» – если подсветка связана с конкретным логическим мотором;
- «LEDS/led-wheels.txt» – если подсветка связана с направлением движения шасси.

Каждый текстовый блок следует пронумеровать, указав вместо «[LEDx]» строки «[LED0]», «[LED1]» и т.д. Эти номера будут логическими номерами устройств подсветки.

В каждом блоке следует указать параметр «DEVICE», указав номер устройства и номер канала управления подсветкой в нем.

Значения подсветки (например, параметр «NORMAL_EMOTION_VALUE») следует задавать в том формате, который требует сам драйвер подсветки. Так, например, часть устройств управления подсветки требуют в качестве параметра значение 0 или 1 (выключено или включено). Другие устройства (например, диспенсор стаканчиков) имеют код режима подсветки. А некоторые устройства поддерживают подсветку в виде RGB значения (последний может задаваться в формате «#rrggbb»).

Кроме того, система управления подсветкой имеет встроенную возможность автоматического моргания подсветки с заданной частотой. Для этого следует задать параметр «xxxxx_BLANK».

Лицо робота

Если робот имеет лицо (а виде аватарки или механическое), то для него в файл «hwconfig.txt» следует вставить текстовый блок из файла «robotface.txt».

```
...
;-----
;
; Лицо робота (как аппаратное, так и аватарка)
;
;-----
[ROBOTFACE]

; (int) Номер мотора 0 лица или -1
MOTOR0 = -1

; (float) Значение мотора 0 для нейтрального рта
MOTOR0_0_VALUE = 0
```

```

; (float) Значение мотора 0 для буквы "A"
MOTOR0_A_VALUE = 10.0

; (float) Значение мотора 0 для буквы "O"
MOTOR0_O_VALUE = 7.0

; (float) Значение мотора 0 для буквы "U"
MOTOR0_U_VALUE = 7.0

; (float) Значение мотора 0 для буквы "E"
MOTOR0_E_VALUE = 5.0

; (float) Значение мотора 0 для моргания
MOTOR0_BLINK_VALUE = 0.0

; (float) Значение мотора 0 для поднятия бровей
MOTOR0_BROW_VALUE = 0.0

; (float) Значение мотора 0 для легкого движения глаз влево
MOTOR0_LEFT_VALUE = 0.0

; (float) Значение мотора 0 для легкого движения глаз вправо
MOTOR0_RIGHT_VALUE = 0.0

; (float) Значение мотора 0 для подмигивания влево
MOTOR0_BLINK_LEFT_VALUE = 0.0

; (float) Значение мотора 0 для подмигивания вправо
MOTOR0_BLINK_RIGHT_VALUE = 0.0

; (float) Значение мотора 0 для сердечек 1
MOTOR0_HEART1_VALUE = 0.0

; (float) Значение мотора 0 для сердечек 2
MOTOR0_HEART2_VALUE = 0.0


; (int) миллисекунд на удержание эмоции
EMOTION_PERIOD = 8000

; (int) миллисекунд на мимику разговора
MIMIC_PERIOD = 150

; (int) миллисекунд на поднятие бровей
BROW_PERIOD = 400

; (int) миллисекунд на легкий взгляд влево-вправо
EYS_PERIOD = 200

; (int) миллисекунд на мигание
BLINK_PERIOD = 200

; (int) миллисекунд на подмигивание влево или вправо
BLINK_SIDE_PERIOD = 2200

; (int) миллисекунд на переход между сердечками (глаза сердечками)
HEART_TRANSACTION_PERIOD = 100

; (int) миллисекунд на такт сердцебиения (глаза сердечками)
HEARTBEAT_PERIOD = 500

; (int) максимальный период между морганиями и другими мимиками лица (мс)
BLINK_INTERVAL = 5000

```

Если робот имеет механическое лицо, то в этом текстовом блоке [ROBOTFACE] следует указать перечень логических моторов, управляющих лицом: MOTOR0, MOTOR1, MOTOR2 и т.д. Нумерация не должна иметь пропусков номеров. Если вместо значения установить (-1), то это считается за отсутствие мотора.

Для каждого такого мотора следует установить параметры:

Параметр	Назначение
MOTOR n _0_VALUE	Положение сервы, соответствующее закрытому рту.
MOTOR n _A_VALUE	Положение сервы, соответствующие фонеме «А»
MOTOR n _O_VALUE	Положение сервы, соответствующие фонеме «О»
MOTOR n _U_VALUE	Положение сервы, соответствующие фонеме «У»
MOTOR n _E_VALUE	Положение сервы, соответствующие фонеме «Е»
MOTOR n _BLINK_VALUE	Положение сервы, соответствующие морганию глазами
MOTOR n _BROW_VALUE	Положение сервы, соответствующие приподнятым бровям
MOTOR n _LEFT_VALUE	Положение сервы, соответствующее повороту глаз влево.
MOTOR n _RIGHT_VALUE	Положение сервы, соответствующее повороту глаз вправо.
MOTOR n _HEART1_VALUE	Глаза-сердечки, положение 1
MOTOR n _HEART2_VALUE	Глаза-сердечки, положение 2

В таблице n – номер мотора в блоке [ROBOTFACE].

Также при необходимости можно поменять параметры, отвечающие за периоды времени тех или иных движений лица. Эти параметры влияют как на поведение механического лица, так и на аватарку робота.

Окно аватарки (если есть)

Если лицо робота выполнено в виде аватарки, то в «hwconfig.txt» следует установить параметры окна этой аватарки. Для этого в «hwconfig.txt» следует вставить текстовый блок из файла «avatar.txt».

В этом текстовом блоке следует задать положение окна так, чтобы оно оказалось на дисплее, связанным с лицом робота.

Обычно используется расширение экрана Windows на два дисплея. Первый дисплей – основной. Второй – лицо робота. Второй экран ставится справа от первого и выравнивается с ним по верхней границе.

Поэтому окно аватарки должно быть в координатах, сразу за правой границей основного экрана. Например, если основной экран имеет разрешение 1920x1080, то окно аватарки должно быть расположено в координатах (1920, 0).

config.txt

Файл «config.txt» в программном комплексе «ДинРобот-3» задает программную конфигурацию робота. Запущенный программный комплекс «ДинРобот-3» отслеживает измерения в этом файле и пересчитывает его «налету» без перезагрузки программы в течение 4-5 секунд.

При создании нового робота необходимо собрать файл конфигурации «config.txt» из кусков блоков, расположенных в папке «RobotConfigs/CONFIG_BLOCKS/CONFIG_TXT». Для этого следует использовать любой текстовый редактор.

Общая часть

Для вставки общей части файла программной конфигурации следует использовать текстовый блок из файла «common.txt».

```
[CONSOLE]

; (int) кодовая страница консоли (только для Windows)
CODE_PAGE = 1251

; (int) сколько дней хранить ошибки или 0, чтобы отключить хранение ошибок
ERROR_LOG_DAYS = 1

;-----
;
; Параметры запуска браузера Chrome
;
;-----
[CHROME]

; (bool) использовать ли браузер Chrome
USE = 1

; команда запуска (для Windows только команда, для Linux
; вместе с аргументами)
COMMAND = chrome

; аргументы запуска (только для windows)
ARGUMENTS = --kiosk --disable-pinch --overscroll-history-navigation=0 --
autoplay-policy=no-user-gesture-required --ignore-certificate-errors --allow-
running-insecure-content --disable-infobars --always-authorize-plugins
http://127.0.0.1:81/index2.html

;COMMAND = export DISPLAY=:0.0;google-chrome --kiosk --password-store=basic -
-autoplay-policy=no-user-gesture-required --ignore-certificate-errors --
allow-running-insecure-content --disable-infobars --always-authorize-plugins
--disable-pinch --overscroll-history-navigation=0
http://127.0.0.1:81/index2.html&

; только для Linux: команда закрытия chrome
KILL_COMMAND = killall google-chrome
```

```

;-----
;
; Скрипты
;
;-----
[SCRIPTS]

; название папки скриптов со знаком "/" в конце
PATH = iscript/

; название главного запускаемого файла скриптов
FILE = main.i

; название файла чат-бота
CHATBOT = masha2.chatbot

; (bool) запускать ли скрипты по умолчанию (или ждать команды
; запуска из интерфейса)
AUTORUN = false

;-----
;
; WEB-сервер экранного контента
;
;-----
[WEB_CONTENT_SERVER]
; (int) порт HTTP-сервера или 0
HTTP_PORT = 81

; (int) порт HTTPS-сервера или 0
HTTPS_PORT = 0

; рабочая папка (со знаком "/" на конце)
DIRECTORY = WEBContent/Masha/

; файл ключа для HTTPS
PRIVATE_KEY_FILE = cert/private.key

; файл публичного сертификата для HTTPS
PUBLIC_CERT_FILE = cert/server.pem

;-----
;
; WEB-сервер управления роботом
;
;-----
[WEB_CONTROL_SERVER]

; (int) порт HTTP-сервера или 0
HTTP_PORT = 80

; (int) порт HTTPS-сервера или 0
HTTPS_PORT = 0

; рабочая папка (со знаком "/" на конце)
DIRECTORY = WebControl/

; логин доступа к серверу из локальной сети или пустая строка
SERVER_LOGIN =

```



```

; пароль доступа к серверу из локальной сети или пустая строка
SERVER_PASSWORD =

; файл ключа для HTTPS
PRIVATE_KEY_FILE = cert/private.key

; файл публичного сертификата для HTTPS
PRIVATE_KEY_FILE = cert/server.pem
;-----
; Проверка подключения интернета
;-----
[INTERNET_CHECKER]

; какой хост пинговать, чтобы определить наличие интернета
URL = https://ya.ru

;-----
;
; Туннель
;
;-----
[TUNNEL]

; хост сервера туннеля или пустая строка
HOST = 89.111.132.208

; (int) порт сервера туннеля
PORT = 8080

; логин робота для сервера туннеля
LOGIN = masha2321

; пароль робота для сервера туннеля
PASSWORD = 111

;-----
;
; Внутренняя база данных
;
;-----
[INNER_DB]

; название файла внутренней базы данных
FILE = innerDB.txt

;-----
;
; Вывод звука
;
;-----
[AUDIO_OUT]

; (int) уровень мимики при воспроизведении wave/mp3. 0-32367
MIMIC_LEVEL = 8000

```

В этом блоке в секции «[CHROME]» следует установить параметр USE в зависимости от необходимости запуска браузера «Google Chrome». Через этот браузер осуществляется вывод на экран робота экранного контента, а также ведется распознавание речи. При необходимости можно также

поправить параметры запуска этого браузера, но в большинстве случаев эти параметры остаются неизменными.

В секции «[SCRIPTS]» устанавливаются параметры запуска скриптов iScripts, на которых реализована система управления поведением робота. Параметр FILE этой секции определяет название главного файла скрипта робота.

В параметре CHATBOT следует установить название файла с чат-ботом от робота. Наличие этого параметра определяет наличие в интерфейсе управления роботом вкладки «Чат-бот». Если параметр пуст или отсутствует, то вкладки «Чат-бот» в интерфейсе не будет.

Параметр AUTORUN определяет необходимость запуска скриптов робота сразу после запуска «ДинРобот-3». После окончательной подготовки всего робота этот параметр следует установить в значение «true». На момент разработки робота параметр рекомендуется установить в значение «false».

В секции «[WEB_CONTENT_SERVER]» следует указать параметр DIRECTORY. Здесь должен быть указан путь к папке, в которой лежит текущий экранный контент робота. Следует не забыть про символ «/» в конце.

Обратим внимание, что экранный контент робота открывается с порта 81. Можно указать и другой порт, но следует учитывать, что с порта 80 (порт по умолчанию для http) открывается контент управления роботом, поэтому его нельзя использовать.

При необходимости работы по HTTPS можно указать порт для HTTPS-сервера. Правда, следует учитывать, что сертификаты для HTTPS все равно будут самоподписанные (на localhost никто не может выдать настоящий сертификат).

В секции «[WEB_CONTROL_SERVER]» следует указать в параметре DIRECTORY путь в папке с файлами интерфейса управления роботом. Этот путь обычно «WEBControl/». Следует не забывать про символ «/» в конце.

При необходимости можно установить логин и пароль для доступа к интерфейсу управления параметрами «LOGIN» и «PASSWORD». Этот пароль будет запрашиваться при входе на страницу стандартными средствами http.

В секции «[TUNNEL]» следует указать путь к серверу туннеля параметром HOST. Строку следует закомментировать, если сервер туннеля не используется.

Следует отметить, что в отличие от «ДинРобот-1», в «ДинРобот-3» все роботы работают на одном порту туннеля. Разделение роботов происходит по логину и паролю доступа к этому серверу. На самом сервере теперь ничего прописывать не надо (в отличие от «ДинРобот-1»). **Однако каждый робот теперь должен иметь уникальный логин и пароль от сервера туннеля.** Логин и пароль указывается в секции «[TUNNEL]» параметрами «LOGIN» и «PASSWORD».

Аватар

Если робот имеет в качестве лица экран с аватаркой, то в «config.txt» следует вставить секцию «[AVATAR]», текст которой следует вставить из файла «avatar.txt».

```
...
;-----
;
; Аватарка
;
;-----
[AVATAR]
; (int) период для сервиса "Робоаватар" (сек)
ROBOAVATAR_PERIOD = 20

; Файл модели аватарки
;FILE = AVATAR/Snowgirl/model.xml
FILE = AVATAR/DubayGirl/model.xml
```

Здесь в параметре FILE следует указать путь к текущей аватарки робота.

Следует заметить, что само окно аватарки формируется не данной секцией, а задается через секцию «[AVATAR]» файла «hwconfig.txt». Поэтому, если в «hwconfig.txt» не будет задано окно аватарки, аватарка не появится.

Синтез речи

Если робот использует систему синтеза речи, то в «config.txt» следует вставить текст из файла «speech_synth.txt».

```
;-----
;
; Синтезатор речи
;
;-----
[SPEECH]

; имеется ли система синтеза речи
ENABLE = true

; язык по умолчанию, используется как приставка в следующих параметрах.
; из iscript можно также использовать SetSpeechDriver("RU") для
; переключения языка
LANGUAGE = RU

LANGUAGES = RU, EN

; (int) Пол робота (0 - Муж. 1 - жен)
ROBOT_GENDER = 1

;-----
; параметры синтезатора для RU
;-----
; драйвер
;RU_DRIVER = NONE
;RU_DRIVER = SAPI
```

```

RU_DRIVER = SAPIx86
;RU_DRIVER = RHVOICE

; (voicelist) голос
;RU_VOICE = Elena
RU_VOICE = Milena

; тональность по умолчанию (int). Обычно 0
RU_DEF_PITCH = 0

; коэф.управления тональностью (float), соответствующий
; одному ">" или "<" во фразе. Обычно 1.0
RU_PITCH_K = 1.0

; относительная скорость (int). Обычно 0
RU_DEF_RATE = 0

; частота дискретизации голоса
RU_WAVE_RATE = 22000

;-----
; параметры синтезатора для EN
;-----
; драйвер
;EN_DRIVER = NONE
EN_DRIVER = SAPI
;EN_DRIVER = RHVOICE

; (voicelist) голос
EN_VOICE = Zira

; тональность по умолчанию (int). Обычно 0
EN_DEF_PITCH = 0

; коэф.управления тональностью (float), соответствующий
; одному ">" или "<" во фразе. Обычно 1.0
EN_PITCH_K = 1.0

; относительная скорость (int). Обычно 0
EN_DEF_RATE = 0

; частота дискретизации голоса
EN_WAVE_RATE = 16000

```

Следует пояснить, что в отличие от предыдущих версий, в «ДинРобот-3», каждая фраза ставится в очередь на произношение совместно с параметрами синтезатора речи, с которыми она должна быть произнесена. Сделано это для того, чтобы можно было бы сказать одну фразу, скажем, на русском, другую – на английском. Физически синтезатор речи переключается лишь в тот момент, когда подошла очередь на произношение очередной фразы.

Поэтому в «config.txt» определяется синтезатор речи для каждого языка, и определяется параметром LANGUAGE, какой язык будет по умолчанию. Перечень же всех поддерживаемых языков указан в параметре LANGUAGES.

В качестве синтезатора речи в параметре «xx_DRIVER» можно выбрать:

- SAPI – встроенный в Windows 64-битный синтезатор речи.
- SAPIx86 – встроенный в Windows 32-битный синтезатор речи.
- WaveTable – синтезатор речи на основе таблицы Wave-файлов, предварительно записанных диктором.
- RHVoice – только для Linux.

Следует понимать, что язык для системы, это не какой-то набор звуков, это полноценный программный синтезатор речи со своими DLL, скомпилированными для 64-битных или 32-битных систем. Причем 64-битное приложение, коим является «ДинРобот-3», может подключать к себе лишь 64-битные DLL-файлы. Если язык не был скомпилирован под 64-битную версию Windows, то его невозможно воспроизвести встроенным синтезатором речи «SAPI».

Однако в ДинРобот-3 предусмотрена возможность подключения 32-битных языков через запуск отдельного 32-битного мини-приложения. Для этого в качестве драйвера языка следует указать значение «SAPIx86». Но запуск такого приложения и обмен данными с ним является для системы относительно накладным. Поэтому не стоит злоупотреблять этой возможностью (не стоит использовать 32-битный синтезатор для языка, если есть его 64-битный аналог).

Отдельно следует сказать про параметр «xx_WAVE_RATE». Он задает дискретизацию Wave-файлов, в которые синтезируется речь. Некоторые языки (например «Elena»), имеют дискретизацию 16000 Гц (это явно следует из полного названия голоса), а некоторые языки поддерживают дискретизацию 22050 Гц (что тоже следует из названия голоса). Под каждый язык лучше ставить соответствующую дискретизацию. Голос «Elena», например, при установке дискретизации свыше 16000 Гц в пустые дискреты, образованные за счет того, что при большей дискретизации самих дискрет становится больше, вставляет какой-то шум.

Распознавание речи

Если робот использует систему распознавания речи, то в «config.txt» следует вставить файлы текст из файла «speech_recognizer.txt».

Система распознавания лиц

Если роботу требуется система распознавания лиц, то в «config.txt» нужно вставить текст из файла «faces.txt».

```
...
;-----
;
; Кручение головой для трекинга и поиска лиц
;
;-----
[FACE_TRACKING]

; использовать ли кручение головой для трекинга лиц
USE = true

; (int) период (мс) блокировки трекинга головы после команд
; управления головой вручную или из скрипта
```

```

LOCK_FACE_TRACKING_PERIOD = 8000

; (int) период (мс) управления поворотом головы при трекинге лица
FACE_TRACKING_PERIOD = 2000

; (int) период (мс) управления креном головы при трекинге или поиске
; лиц (кратно FACE_TRACKING_PERIOD)
FACE_ROLL_PERIOD = 8000

; (float) коэф.сглаживания трекинга лиц [0-1] (обычно 0.2),
; чем меньше, тем приближение поворота головы к лицу осуществляется более
; мелкими шажками
FACE_TRACKING_SMOOTHER = 0.7

; (float) ширина мертвой зоны трекинга лиц (градусы)
FACE_TRACKING_DEADZONE_WIDTH = 6.0

; (float) высота мертвой зоны трекинга лиц (градусы)
FACE_TRACKING_DEADZONE_HEIGHT = 4.0

; (int) период (мс) засыпания трекинга головой, если лиц нет
FACE_SEARCH_SLEEP_PERIOD = 900000

; (int) период (мс) следования моментов поворота головы в поиске лиц
(случайность в пределах 70%-100% от этой величины)
FACE_SEARCH_PERIOD = 16000

; (int) период (мс) следования моментов покачиваний головы (квантуется
моментами FACE_SEARCH_PERIOD)
FACE_ROLL_PERIOD = 24000

; (float) амплитуда поворота головы в поиске лиц (градусы)
FACE_SEARCH_ANGLE_AMP = 12.0

; (float) амплитуда наклона головы в поиске лиц (градусы)
FACE_SEARCH_PITCH_AMP = 5.0

; (float) амплитуда качаний головой в поиске лиц (градусы)
FACE_SEARCH_ROLL_AMP = 5.0

;-----
;
; Система распознавания лиц
;
;-----
[FACE_SYSTEM]
; (bool) используется ли база данных лиц
USE_DB = true

; путь к базе данных лиц, обязательно со знаком "/" в конце
DB_PATH = FaceDB/

; (int) логический номер камеры
CAMERA = 0

; (int) минимальная граница от края кадра для запоминания лица (% от ширины)
REMEMBER_LEFT_RIGHT_MARGIN = 10

; (int) минимальная граница от верхнего края кадра за запоминания лица

```

```

; (% от высоты)
REMEMBER_TOP_MARGIN = 4

; (int) минимальная граница от нижнего края кадра за запоминания лица
; (% от высоты)
REMEMBER_BOTTOM_MARGIN = 4

; (int) минимальная ширина лица для запоминания (% от ширины)
REMEMBER_MIN_FACE_WIDTH = 12

; (int) минимальное качество детектирования лица для запоминания
REMEMBER_MIN_CONFIDENCE = 0.6

; (int) интервал (мс) с момента включения режима запоминания лица (за это
время выбирается лучший ракурс)
REMEMBER_PERIOD = 4000

; (int) минимальная ширина лица (% от ширины кадра)
MIN_FACE_WIDTH = 8

; (int) ширина лица чтобы здороваться (% от ширины кадра)
HELLO_FACE_WIDTH = 9

; (int) максимальная ширина лица (% от ширины кадра)
MAX_FACE_WIDTH = 50

; (int) минимальная ширина лица (% от ширины кадра) при сканировании новых
фото в базе данных лиц
DB_MIN_FACE_WIDTH = 10

; (int) максимальная ширина лица (% от ширины кадра) при сканировании новых
фото в базе данных лиц
DB_MAX_FACE_WIDTH = 80

; (float) минимальное качество детектирования лица, чтобы его распознавать.
От 0 до 1. Обычно 0.4
ДЕТЕКТ_MIN_CONFIDENCE = 0.4

; (float) порог похожести для распознавания. От 0 до 1. Обычно 0.95
MATCH_MIN_CONFIDENCE = 0.95

; (float) порог похожести при слежении. От 0 до 1, Обычно 0.7
TRACKING_MIN_CONFIDENCE = 0.7

; (int) время (мс), через которое будет предложено познакомиться
ACQUAINTANCE_PERIOD = 32000

; (int) период (мс) на которое лицо может скрыться из кадра
FACE_OUT_PERIOD = 10000

; (int) период (мс), прежде чем система понимает, что перед ней сменилась
неизвестная персона
NEW_PERSON_PERIOD = 4000

; (int) период (мс) с момента обнаружения лица до момента события * hello
HELLO_PERIOD = 1500

; (int) период (сек) хранения временных персон
TEMP_PERSON_PERIOD = 86400

```

Здесь параметры системы трекинга лиц и параметры системы распознавания лиц. Все параметры подписаны, при необходимости их можно отредактировать.

Фотосервисы (если есть)

Если экранный контент робота содержит фотосервисы, такие как «сделать фото», «робо-аватар», «фото в образе», в «config.txt» следует включить секцию «[PHOTO_SERVICE]» из файла «photoservice.txt».

Печать фото на локальном принтере

Если планируется печать фото на локальном (для робота) фотопринтере то в файл конфигурации «config.txt» следует вставить текст из файла «printer.txt».

```

;-----
;
; Принтер
;
;-----
[PRINTER]

; (bool) используется ли локальный принтер, подключенный непосредственно
; к роботу
LOCAL_PRINTER = true

```

В этой секции необходимо установить параметр LOCAL_PRINTER равным true.

По умолчанию параметр LOCAL_PRINTER имеет значение false, при этом робот реализует очередь на печать, которую следует забирать на принтер по сети с помощью программы «PrintClient». Скачать программу PrintClient можно из интерфейса управления роботом.

Обратная задача кинематики

Если робот оснащен манипуляторами, а решение обратной задачи кинематики предусмотрено средствами «ДинРобот-3», то в «config.txt» нужно вставить секцию из файла «ik.txt». Здесь содержатся параметры решения обратной задачи кинематики, которые при необходимости можно скорректировать.

Запись видео (видеорекордер)

Если робот записывает видео (например, записывает видеоотзывы), то в «config.txt» необходимо вставить секцию из файла «videorecorder.txt».

Здесь параметры видеорекордера, которые при необходимости можно поправить.

Отправка почты

Если робот отправляет фото на почту, то в «config.txt» следует прописать параметры smtp-сервера, которые можно взять из файла «smtp.txt».

Навигация и движение по карте

В зависимости от используемых аппаратных средств движение по карте может осуществляться на основе дальномеров и камеры, или на основе лидара.

Если используется лидар, то в «config.txt» следует вставить секцию с локальной картой местности из файла «localmap.txt», а также систему навигации из файла «NAVIGATORS/depth_navigator.txt». При необходимости параметры из вставляемых секций можно скорректировать.

Если используются камера и дальномеры, то в «config.txt» следует вставить секцию навигатора из файла «NAVIGATORS/visual_navigator.txt». При необходимости вставляемые параметры можно скорректировать.

В обоих случаях в файл «config.txt» следует вставить текст из файла «imoving.txt» с параметрами системы управления движением, которые также можно скорректировать под конкретного робота.

Парковка на зарядку

Если робот обладает камерой наведения на зарядное устройство по QR-коду, то в «config.txt» следует вставить секцию из файла «parking.txt». Здесь параметры наведения на парковку, которые при необходимости можно скорректировать.

Организация WiFi-сети для робота и его оператора при использовании программного комплекса «ДинРобот-3»

При организации связи между роботом и оператором программный комплекс «ДинРобот-3» выступает в роли http-сервера, а оператор – клиентом. Для связи с роботом оператор использует Web-браузер (рекомендуется Google Chrome).

При невозможности организовать прямой доступ оператора к локальной сети робота (робот и оператор находятся в разных локальных сетях), имеется возможность подключения к роботу через Интернет-туннель – специальный Web-сервер с прямым IP-адресом, позволяющий связать робота, подключающегося к этому серверу как клиент, и оператора, использующего Web-браузер.

Рекомендованные варианты организации WiFi-сети управления роботом:

Вариант 1. Встроенный в робота WiFi-роутер и USB WiFi-адаптер

Робот имеет два WiFi-устройства: 1) встроенный в робота WiFi-роутер для связи с оператором, подключенный кабелем к Ethernet-входу бортового компьютера. Вторым концом Ethernet кабель подключается ко входу LAN роутера (внутренняя сеть); 2) USB WiFi-адаптер для связи бортового компьютера с внешней сетью Интернет.

В бортовом компьютере в настройках TCP/IPv4 по кабельному соединению требуется установить большое значение метрики, например: 9999. Также, чтобы не искать всякий раз после включения IP-адрес робота в

локальной сети, рекомендуется бортовому компьютеру по кабельному соединению установить статический IP-адрес (обычно используется 11.0.0.2), при этом в настройках роутера нужно прежде установить адрес внутренней (локальной) сети 11.0.0.1.

В настройках TCP/IPv4 для USB WiFi-адаптера требуется установить маленькое значение метрики, например 1. Через USB-WiFi адаптер робота следует подключить к внешней WiFi-сети с Интернет.

Оператор робота будет подключать свой ноутбук или смартфон к WiFi-сети встроенного роутера. Эта сеть будет без Интернета, но она будет предоставлять связь с бортовым компьютером робота. После подключения ноутбука ко встроенному роутеру робота оператору необходимо будет в браузере ввести IP-адрес бортового компьютера робота (в предложенном примере, это «http://11.0.0.2»).

При подключении смартфона к роботу на смартфоне нужно будет отключать мобильные данные, иначе смартфон будет отправлять данные не в ту сеть.

При подключении ноутбука к роботу по WiFi необходимо отключаться от других сетевых подключений (например, от кабельного подключения ноутбука) по той же причине.

Никогда не следует подключить USB WiFi-адаптер ко встроенному в робота роутеру, иначе операционная система запомнит от него пароль, и при каждом удобном случае будет подключаться к этому роутеру (ведь у него выше уровень сигнала).

Преимущества такой схемы подключения в том, что связь с роботом можно будет установить даже в чистом поле, где нет Интернета. Все же интернет не так критичен для дистанционного управления роботом, а связь с роботом у оператора будет всегда, причем в этой сети только оператор и робот, что повышает стабильность такого соединения.

Вариант 2. Встроенный в робота WiFi-роутер с подключенным в него 4G-модемом

Робот имеет встроенный в робота WiFi-роутер для связи с оператором, подключенный кабелем к Ethernet-входу бортового компьютера. Второй конец Ethernet кабель подключается ко входу LAN роутера (внутренняя сеть). К роутеру по USB подключается 4G-модем, с которого всей внутренней сети робота раздается мобильный Интернет. Роутер должен иметь такую возможность (не у каждого роутера имеется возможность подключения 4G-модема).

Чтобы не искать IP-адрес робота в локальной сети после включения робота, рекомендуется бортовому компьютеру по кабельному соединению установить статический IP-адрес (обычно используется 11.0.0.2), при этом в настройках роутера нужно прежде установить адрес внутренней (локальной) сети 11.0.0.1.

Оператор робота будет подключать свой ноутбук или смартфон к WiFi-сети встроенного роутера. После подключения к роутеру робота оператору

необходимо будет в браузере ввести IP-адрес бортового компьютера робота (в предложенном примере, это «http://11.0.0.2»).

При подключении смартфона к роботу на смартфоне нужно будет отключать мобильные данные, иначе смартфон будет отправлять данные не в ту сеть.

При подключении ноутбука к роботу по WiFi необходимо отключаться от других сетевых подключений (например, от кабельного подключения ноутбука) по той же причине.

Преимущества такой схемы подключения в том, что и на работе и у оператора появляется Интернет. Недостаток в том, что Интернет – мобильный, и качество его приема в некоторых местах может быть плохим. Кроме того, 4G-модем должен иметь SIM-карту, зарегистрированную на чье-то имя.

Вариант 3. Встроенный в робота WiFi-роутер и подключенный к бортовому компьютеру 4G-модем

Робот имеет встроенный в робота WiFi-роутер для связи с оператором, подключенный кабелем к Ethernet-входу бортового компьютера. Второй конец Ethernet кабель подключается ко входу LAN роутера (внутренняя сеть).

К бортовому компьютеру робота по USB подключается 4G-модем, раздающий роботу мобильный Интернет. На бортовой компьютер робота необходимо будет установить драйвер и программное обеспечение используемого 4G-модема, и настроить автоматическое подключение к Интернету при запуске операционной системы.

Чтобы не искать IP-адрес робота в локальной сети после включения робота, рекомендуется бортовому компьютеру по кабельному соединению установить статический IP-адрес (обычно используется 11.0.0.2), при этом в настройках роутера нужно прежде установить адрес внутренней (локальной) сети 11.0.0.1.

Оператор робота будет подключать свой ноутбук или смартфон к WiFi-сети встроенного роутера. Эта будет сеть без Интернета, но через нее можно подключиться к роботу. Оператору робота необходимо будет в браузере ввести IP-адрес бортового компьютера робота (в предложенном примере, это «http://11.0.0.2»).

При подключении смартфона к роботу на смартфоне нужно будет отключать мобильные данные, иначе смартфон будет отправлять данные не в ту сеть.

При подключении ноутбука к роботу по WiFi необходимо отключаться от других сетевых подключений (например, от кабельного подключения ноутбука) по той же причине.

Преимущества такой схемы подключения в том, что и на работе и у оператора появляется Интернет. Недостаток в том, что Интернет – мобильный, и качество его приема в некоторых местах может быть плохим. Кроме того, 4G-модем должен иметь SIM-карту, зарегистрированную на чье-то имя.

Вариант 4. WiFi-адаптер на работе, подключаемый к внешнему WiFi-роутеру

Робот имеет USB WiFi-адаптер или иной встроенный в бортовой компьютер WiFi-адаптер. Через этот адаптер робот подключается к внешнему WiFi-роутеру, который тем или иным способом подключен к сети Интернет.

Рекомендуется в настройках роутера установить для робота фиксированный IP-адрес. В противном случае IP-адрес робота будет плавающим.

Оператор робота будет подключать свой ноутбук или смартфон к WiFi-сети этого внешнего роутера. После подключения к роутеру робота оператору необходимо будет в браузере ввести IP-адрес бортового компьютера робота.

При подключении смартфона к роботу на смартфоне нужно будет отключать мобильные данные, иначе смартфон начинает отправлять данные не в ту сеть.

При подключении ноутбука к роботу по WiFi необходимо отключаться от других сетевых подключений (например, от кабельного подключения ноутбука).

Преимущества такой схемы подключения в том, что и на работе и у оператора будет Интернет. Недостаток в том, что без именно этого внешнего роутера у оператора не будет никакой возможности подключиться к роботу. Придется всегда иметь с собой именно этот роутер (что не очень удобно на выездных мероприятиях), для этого роутера всегда должна быть обеспечена розетка, и его всегда нужно подключать кабелем по WLAN или 4G-модемом к какому-либо внешнему Интернету.

Вариант 5. Подключение через http-туннель

В настройках программного комплекса «ДинРобот-3» имеется возможность подключения программы «ДинРобот-3» в качестве клиентского соединения к Интернет-серверу туннеля, имеющего прямой IP-адрес в сети Интернет. Подключившись к этому Интернет-серверу в качестве TCP-клиента, между роботом и сервером образовывается постоянное двухстороннее TCP-соединение. Робот сообщает Интернет-серверу свой уникальный логин и пароль.

Оператор также подключается к этому Интернет-серверу туннеля с помощью браузера. При первом подключении Интернет-сервер туннеля запрашивает у оператора логин и пароль от робота, после чего ищет робота по логину и паролю в перечне подключенных к этому серверу роботов, и перенаправляет http-запросы оператора на робота через установленное с ним TCP-соединение.

Робот при этом должен быть подключен к сети Интернет любым и ранее перечисленных способов.

Оператор робота также должен быть подключен к сети Интернет любым способом.

Для своих пользователей разработчик программного комплекса организовал такой сервер Интернет-туннеля на своём сервере «89.111.132.208» порт 8080. Однако разработчик не гарантирует, что этот сервер будет работать вечно.

Достоинством такого подключения является то, что робот и оператор могут находиться в разных сетях. Недостаток в том, что скорость обмена данными через Интернет-туннель ниже, чем при прямом соединении. Кроме того ресурсы сервер-туннеля ограничены, поэтому не стоит злоупотреблять подключением через туннель.

Запуск робота

Чтобы запустить робота после создания config.txt и hwconfig.txt необходимо запустить на роботе программу «ДинРобот-3» (dynrobot3.exe). При этом должно появиться консольное окно Windows, в котором будет отображаться консоль программы ДинРобот-3.

Чтобы зайти на робота необходимо использовать Web-браузер (Google Chrome, Mozilla FireFox или Yandex Browser). Браузер можно запустить на внешнем компьютере оператора, подключенного к сети робота, или на самом роботе.

В командной строке браузера следует указать адрес: «http://11.0.0.2» (IP-адрес робота, адрес может не совпадать с указанным в примере). Если браузер запущен прямо на самом роботе, то адрес будет «http://127.0.0.1».

Если по данному адресу нет ответа, или открывается что-то отличное от интерфейса управления «ДинРобот-3», то следует убедиться, что на бортовом компьютере робота нет других WEB-серверов на порту 80. Решением проблемы может стать отключение Firewall (Брандмауэра Windows или иного антивирусного ПО на бортовом компьютере робота), или внесение программы «ДинРобот-3» в список исключений для этого антивирусного ПО.

Если связь не работает при подключении с удаленного компьютера, имеет смысл проверить саму сеть. Сделать ПИНГ IP-адреса бортового компьютера с компьютера управления. Убедиться, что IP-адрес бортового компьютера и компьютера оператора находятся в одной локальной подсети (обычно, если на бортовом компьютере IP-адрес: 11.0.0.2, то и у компьютера управления должен быть IP-адрес: 11.0.0.xxx). На компьютере оператора также следует отключить прочие сетевые подключения или расставить в них метрики.

Если подключение состоялось, то следует проверить работу камер, управления шасси, работу отдельных моторов робота.