

# Библиотека RubiRobot

© 2017 Слинкин Д.А.

Библиотека предназначена для управления роботами на базе контроллера Lego EV3 (в дальнейшем - EV3).

ОС EV3: **Linux Debian** проекта **ev3dev**.

Язык программирования: **FreePascal**.

Лицензия: **LGPL v3.0**

Версия **0.1**

## Требования к ОС Linux Debian на EV3:

1. Пакет **fpс** для компиляции на EV3 программ с использованием RubiRobot.
2. Пакет **ttf-dejavu** для корректного запуска на EV3 программы с использованием RubiRobot и графическим выводом текста на дисплей контроллера.

## Модули библиотеки

Модули библиотеки можно условно разделить на 3 части:

**1. Базовый модуль uev3.** Любая программа, использующая библиотеку, должна подключить как минимум модуль uev3 и первым действием вызвать функцию инициализации библиотеки ev3init(). Без этого обращение к библиотеке будет завершаться ошибкой.

### 2. Модули поддержки 6 основных подсистем EV3.

Модуль	Описание
rubiroScreen	Обработка дисплея EV3
rubiroSound	Обработка аудио-системы EV3
rubiroLeds	Обработка цветовой индикации EV3
rubiroButtons	Обработка кнопок EV3
rubiroMotors	Обработка двигателей EV3
rubiroSensors	Обработка датчиков EV3

### 3. Вспомогательные модули

Модуль	Описание
uev3devices	Низкоуровневая поддержка подключаемых устройств.
uev3sysfs	Низкоуровневое чтение-запись данных для sysfs-файлов
uev3fb, uev3freetype, uev3ftfont, uev3image	Низкоуровневая поддержка дисплея
uev3const, uev3func, uev3utf8	Общепотребительные константы, переменные, функции

## Классы и объекты библиотеки

Библиотека предоставляет для программисту набор классов и несколько predefined объектов.

### Предопределенные объекты (создаются вызовом `ev3init()`)

Объект	Класс	Модуль	Краткое описание
ev3	TEv3	uev3	Может использоваться для ожидания возникновения событий на других объектах. Применяется редко. Необходимость в создании других экземпляров отсутствует.
ev3buttons	TEv3Buttons	rubiroButtons	Обеспечивает обработку кнопок EV3. Используется повсеместно. Необходимость в создании других экземпляров отсутствует.
ev3leds	TEv3Leds	rubiroLeds	Обеспечивает обработку цветоиндикации EV3. Используется повсеместно. Необходимость в создании других экземпляров отсутствует.
ev3screen	TEv3Screen	rubiroScreen	Обеспечивает обработку дисплея EV3 в графическом режиме. Используется повсеместно. Необходимость в создании других экземпляров отсутствует.
ev3sound	TEv3Sound	rubiroSound	Обеспечивает поддержку аудио EV3. Используется повсеместно. Необходимость в создании других экземпляров отсутствует.
ev3DeviceFactory	TEv3DeviceFactory	uev3devices	Фабрика устройств. Обеспечивает инициализацию устройств и хранит набор подключенных устройств. Автоматически обновляется: при попытке инициализировать новое устройство, при попытке обращения к отключенному устройству. В последнем случае осуществляет несколько попыток обновления со звуковой и цветовой индикацией. Активно используется для внутренних целей, необходимость доступа к объекту извне возникает крайне редко.

Объекты подключаемых устройств (моторов и датчиков) должны создаваться и инициализироваться программистом самостоятельно, для них отсутствуют predefined объекты.

### Классы подключаемых устройств

Класс	Модуль	Краткое описание
TEv3InfraSensor	rubiroSensors	Инфракрасный датчик (код 45509). Позволяет определять расстояние до препятствия (до примерно 70 см), направление и расстояние до маяка (код 45508), набор нажатых кнопок на маяке. Операции на маяке проводятся с учетом выбранного канала связи.
TEv3GyroSensor	rubiroSensors	Гироскопический датчик (код 45505). Позволяет определить скорость и направление при движении в горизонтальной плоскости, скорость и угол наклона.
TEv3UltraSensor	rubiroSensors	Ультразвуковой датчик (код 45504). Позволяет определять расстояние до препятствия в миллиметрах, сантиметрах, метрах; оценивать наличие шумов от других датчиков; получать данные сразу, либо ожидать изменившегося значения. Штатно поддерживает режим постоянного непрерывного определения расстояния. Экспериментально поддерживает режим одноразового определения расстояния (не рекомендуется к использованию)

Класс	Модуль	Краткое описание
TEv3ColorSensor	rubiroSensors	Датчик цвета/света (код 45506). Поддерживает режимы определения цвета, определения составляющих цвета (RGB), определения уровня отраженного света и определения освещенности. Позволяет получать данные сразу, либо ожидать изменившегося значения
TEv3TouchSensor	rubiroSensors	Датчик касания (код 45507). Позволяет определить текущее состояние кнопки, позволяет ожидать нажатия, отжатия и щелчка.
TEv3MediumMotor	rubiroMotors	Средний мотор (код 45503). Позволяет запускать мотор с остановкой по истечении времени, по достижению количества оборотов или градусов, а также без ограничений. Дает возможность реверса. Перед запуском и при запуске позволяет указать скорость и один из трех способов остановки, а также в любой момент времени получить эти данные. Позволяет определить текущую скорость и позицию, ждать остановки мотора.
TEv3LargeMotor	rubiroMotors	Большой мотор (код 45502). Возможности эквивалентны среднему мотору.
TEv3Rule	rubiroMotors	Рулевое управление. Позволяет использовать одновременно пару больших или средних моторов. Позволяет запускать моторы с ожиданием остановки по по достижению количества оборотов или градусов. Позволяет запускать моторы без ожидания с остановкой по истечении времени. Дает возможность реверса. Перед запуском и при запуске позволяет указать скорость и один из трех способов остановки, а также в любой момент времени получить эти данные. Позволяет определить текущую скорость и позицию, ждать остановки моторов.

## Особенности библиотеки RubiRobot

1) Применение типа Variant для большинства параметров позволяет по разному формировать и интерпретировать данные. Обычно это строки и числа разных типов. Дополнительно, модуль uev3 содержит 286 переменных типа Variant, с именами a,a0,a1...a9,b,b0,b1,...z,z0,z1...z9. Это позволяет быстро создавать программы, не отвлекаясь на объявление переменных. Исключения составляют объекты подключаемых устройств (датчики и моторы), которые в обязательном порядке следует объявлять, т. к. тип Variant несовместим с объектным типом.

2) Большинство методов классов в качестве результата возвращает ссылку на текущий объект. Это позволяет осуществлять последовательный вызов методов объекта без повторного указания объекта. Например:

```
var LM: TEv3LargeMotor;
begin
  ev3init();
  LM:=TEv3LargeMotor.Create;
  ...
  LM.setReverse(true).run(100).wait();
  ...
end.
```

3) Запись и чтение значений датчиков, моторов, кнопок и цветоиндикаторов производится (за редким исключением) с использованием виртуальных файлов в файловой системе sysfs, которые, в свою очередь, обслуживаются драйверами соответствующих устройств. Слишком частое чтение или запись в эти файлы может вызвать некорректную работу драйверов и, как следствие, замедление работы, отказ в доступе к файлам или даже зависание контроллера. Поэтому в библиотеке предусмотрены отдельные таймауты на чтение (uev3sysfs.getInterval) и запись данных (uev3sysfs.setInterval). По умолчанию они равны соответственно 10 и 20

миллисекунд, но могут быть изменены как глобально, так и индивидуально для каждого класса или объекта. Для подавляющего большинства ситуаций таймаут на чтение достаточен, однако таймаут на запись в некоторых случаях придется увеличить.

Первым признаком недостаточности используемых таймаутов может служить процесс `systemd_udevd`, который начинает использовать более 10% процессорного времени и не уменьшает свою загрузку по окончании работы программы (проверяется утилитой `top` в `ssh`-сессии).