

Класс TEv3Sound, модуль rubiroSound

```
TEv3Sound=class
  constructor create;
  destructor destroy;override;
  // Устанавливает уровень громкости (от 0 до 100%)
  function volume(vol:variant):TEv3Sound;
  {
    Подает звуковой сигнал заданной частоты, заданной длительности,
    с заданной паузой по окончании.
    Длительность по умолчанию = 200мс, пауза - 0мс
  }
  function beep(tone:variant):TEv3Sound;
  function beep(tone:variant;duration:variant):TEv3Sound;
  function beep(tone:variant;duration:variant;delay:variant):TEv3Sound;
  {
    Подает набор звуковых сигналов заданной частоты, заданной длительности,
    с заданной паузой по окончании. Информация о сигналах передается в
    массив тройками частота-длительность-пауза.
  }
  function beep(tdws:array of const):TEv3Sound;
  // Озвучивает файл в WAV-формате, имя которого передается в функцию
  function play(wavfile:variant):TEv3Sound;
  {
    Озвучивает переданный текст синтезатором речи,
    скорость по умолчанию = 160 слов в минуту
  }
  function speak(txt:variant):TEv3Sound;
  function speak(txt:variant; speed:variant):TEv3Sound;
  function speak(txts:array of const):TEv3Sound;
  function speak(txts:array of const; speed:variant):TEv3Sound;
  // устанавливает кол-во слов в минуту для синтезатора речи
  function setSpeakSpeed(speed:variant):TEv3Sound;
  // Ожидает завершения любых запущенных звуковых процессов
  function wait():TEv3Sound;
  {
    Ожидает завершения запущенных звуковых процессов
    Параметр what определяет тип процессов:
    'all' - все звуковые процессы
    'beep' - звуковой сигнал
    'play' - озвучивание WAV-файла
    'speak' - синтезатор речи
  }
  function wait(what:variant):TEv3Sound;
  // Прерывает все запущенные звуковые процессы
  function stop():TEv3Sound;
  // Прерывает запущенные звуковые процессы, параметр what - см. wait()
  function stop(what:variant):TEv3Sound;
  {
    устанавливает режим ожидания завершения звукового процесса перед
    стартом нового звукового процесса такого-же типа. Используется
    по умолчанию
  }
  function setWaitBeforeStart():TEv3Sound;
  {
    устанавливает режим прерывания звукового процесса перед
    стартом нового звукового процесса такого-же типа.
  }
  function setStopBeforeStart():TEv3Sound;
end;
```

Управление звуковой подсистемой реализовано через объект `ev3sound:TEv3Sound` модуля `rubiroSound`. `ev3sound` осуществляет внешний вызов следующих консольных linux-утилит: `amixer` (установка громкости), `beep` (подача звуковых сигналов), `play` (озвучивание WAV-файла), `espeak` (синтезатор речи). Таким образом, существует 3 типа звуковых процессов, определяемых идентификаторами `'beep'`, `'play'` и `'speak'`.

Библиотека настраивает синтезатор речи на использование русского языка. Это означает, среди прочего, что числительные будут произноситься по русски, даже если находятся внутри английского текста. Используя дополнительный словарь с сайта разработчика `espeak`, можно значительно улучшить произношение русского текста. Для этого следует загрузить архив http://espeak.sourceforge.net/data/ru_dict-48.zip, разархивировать находящийся в нем файл `ru_dict-48` и скопировать его на контроллер, заменяя файл `/usr/lib/arm-linux-gnueabi/espeak-data/ru_dict`.

Запуск каждого звукового процесса реализуется параллельно основной программе и параллельно другим типам звуковых процессов. Такое поведение можно изменить с помощью метода `wait`, который блокирует программу в ожидании завершения заданного типа (типов) звукового процесса.

Звуковые процессы разных типов не взаимодействуют между собой, что, с учетом параллельности их работы, позволяет возникать ситуациям, когда разные типы процессов будут озвучены не в том порядке, котором запускались. Например, последовательный запуск ресурсоемкого `espeak` и высокоскоростного `beep` без ожидания завершения процессов, скорее всего приведет сначала к подаче звукового сигнала, а затем - к озвучиванию текста синтезатором речи.

Запуск звукового процесса до завершения другого звукового процесса такого-же типа приведет либо к ожиданию завершения предыдущего процесса (метод `setWaitBeforeStart()`), либо к прерыванию предыдущего процесса (метод `setStopBeforeStart()`), после чего будет произведен старт нового звукового процесса.

Если программа завершит свою работу до окончания работы звуковых процессов, то они все будут принудительно остановлены.

Примеры:

```
{
  Демонстрация beep с обязательным ожиданием перед завершением программ
  (snd1.pp)
}
{$mode objfpc}
uses uev3, rubiroSound;
begin
  ev3init();
  ev3sound.volume(100).beep(450);
  ev3sound.volume(10).beep(450).wait();
end.

{
  Построчный синтезатор речи. Предназначен для запуска в ssh-сессии.
  (snd2.pp)
}
{$mode objfpc}
uses uev3, rubiroSound;
var s:string;
begin
  ev3init();
  ev3sound.volume(100);
```

```
while true do begin
  readln(s);
  if s='' then break;
  ev3sound.speak(s,100);
end;
end.
```

```
{
  Демонстрация espeak и beep (snd3.pp)
}
{$mode objfpc}
uses uev3,rubiroSound;
begin
  ev3init();
  ev3sound.speak('Имперский марш, слушать всем!',120).wait;
  ev3sound.beep([
    500,500,250,500,500,250,400,500,200,600,200,50,
    500,500,250,400,500,200,600,200,50,500,500,500,
    750,500,250,750,500,250,750,500,250,810,500,200,
    600,200,50,470,500,250,400,500,200,600,200,50,500,500,500
  ]).wait;
end.
```