

Класс TEv3Buttons, модуль rubiroButtons

```
TEv3Buttons=class(TEv3)
  constructor Create();
  destructor destroy;override;
{
  ждет нажатия или отпускание кнопки.
  кнопка кодируется событием вида '14:0', 'назад:нет'
  (отпускание клавиши Back),
  '108:1', 'Down:NO' (нажатие клавиши Down) и т.п.
  Регистр символов неважен.
  timeout - максимальный таймаут
  возвращает false при выходе по таймауту
}
function waitEvent(event:string; timeout:dword=high(dword)):boolean;
  override;
{
  заблокированно ждет нажатия или отпускания клавиши
  в Button возвращает клавишу - одну из констант bBACK, bCENTER, bUP,
  bLEFT, bRIGHT или bDOWN
  в Press возвращает true, если было нажатие
}
function wait(out Button:variant; out Press:variant):TEv3Buttons;
{
  ждать нажатия-отпускания набора кнопок
  завершение - при возникновении ВСЕХ событий из переданного массива
  каждая кнопка в массиве кодируется строкой (см.WaitEvent)
  процессе ожидания сопровождается миганием цветоиндикации
}
function waitAll(Buttons:array of const):TEv3Buttons;
{
  ждать нажатия-отпускания набора кнопок
  завершение - при возникновении ЛЮБОГО события из переданного массива
  каждая кнопка в массиве кодируется строкой (см.WaitEvent)
  процессе ожидания сопровождается миганием цветоиндикации
}
function waitAny(Buttons:array of const):TEv3Buttons;
// возвращает состояние кнопки BACK, true - при нажатии
property Back:variant read getBack;
// возвращает состояние кнопки CENTER, true - при нажатии
property Center:variant read getCenter;
// возвращает состояние кнопки UP, true - при нажатии
property Up:variant read getUp;
// возвращает состояние кнопки LEFT, true - при нажатии
property Left:variant read getLeft;
// возвращает состояние кнопки RIGHT, true - при нажатии
property Right:variant read getRight;
// возвращает состояние кнопки DOWN, true - при нажатии
property Down:variant read getDown;
end;
```

Доступ к кнопкам EV3 реализован через объект ev3buttons:TEv3Buttons, который автоматически создается при инициализации библиотеки функцией ev3init.

У EV3 существует 6 кнопок: UP, DOWN, LEFT, RIGHT, CENTER, BACK.

Нажатие на кнопку CENTER дополнительно генерирует перевод строки. Таким образом, завершение вызова процедуры readln можно обеспечить не клавишей Enter, а кнопкой CENTER.

Объект ev3buttons позволяет:

- 1) Получать текущее состояние кнопок (нажата или отжата)
- 2) Ожидать возникновения любого события на кнопках
- 3) Ожидать возникновения заданных событий на кнопках

Существует несколько способов как идентификации кнопок в методах ожидания, так и идентификации состояния кнопок, при этом регистр символов значения не имеет:

Идентификация кнопок:

Кнопка UP, варианты идентификации: 'up','вверх','103','bup'

Кнопка DOWN, варианты идентификации: 'down','вниз','108','bdown'

Кнопка LEFT, варианты идентификации: 'left','влево','105','bleft'

Кнопка RIGHT, варианты идентификации: 'right','вправо','106','bright'

Кнопка CENTER, варианты идентификации: 'center','центр','28','bcenter'

Кнопка BACK, варианты идентификации: любые строки, но рекомендуются следующие - 'back','cancel','назад','отмена','bback'

Идентификация состояния кнопок

Кнопка отжата: 'нет','no','0','"

Кнопка нажата: все остальные варианты

Примеры:

```
{
    Неблокирующее ожидание и вывод на экран нажимаемых кнопок EV3 (btn1.pp)
}
{$mode objfpc}
uses uev3, rubiroButtons;
begin
    ev3init();
    while not ev3buttons.back do begin
        if ev3buttons.up then writeln('UP');
        if ev3buttons.down then writeln('DOWN');
        if ev3buttons.left then writeln('LEFT');
        if ev3buttons.right then writeln('RIGHT');
        if ev3buttons.center then writeln('CENTER');
    end;
end.

{
    Блокирующее ожидание и вывод на экран нажимаемых/отпускаемых кнопок EV3
    (btn2.pp)
}
{$mode objfpc}
uses uev3, rubiroButtons;
var button, press: variant;
begin
    ev3init();
    while true do begin
        ev3buttons.wait(button, press);
        case button of
            bback: writeln('BACK: ', press);
            bcenter: writeln('CENTER: ', press);
            bleft: writeln('LEFT: ', press);
            bright: writeln('RIGHT: ', press);
            bup: writeln('UP: ', press);
            bdown: writeln('DOWN: ', press);
```

```
    else writeln('ERROR!');
  end;
  if (button=bback)and(not press) then break;
end;
end.
```

```
{
  Блокирующее ожидание различных комбинаций нажатия/отпускания кнопок LEFT и
  RIGHT (btn3.pp):
}
{$mode objfpc}
uses uev3, rubiroButtons;
begin
  ev3init();
  Writeln('Wait LEFT and RIGHT press');
  ev3buttons.waitAll(['left:да', 'right:1']);
  Writeln('Wait LEFT and RIGHT release');
  ev3buttons.waitAll(['влево:0', 'right:no']);
  Writeln('Wait LEFT or RIGHT press');
  ev3buttons.waitAny(['left:да', 'вправо:YES']);
  Writeln('end');
end.
```