

ПРОГРАММИРОВАНИЕ В TRIK (ЧАСТЬ 1)

Вебинар 3

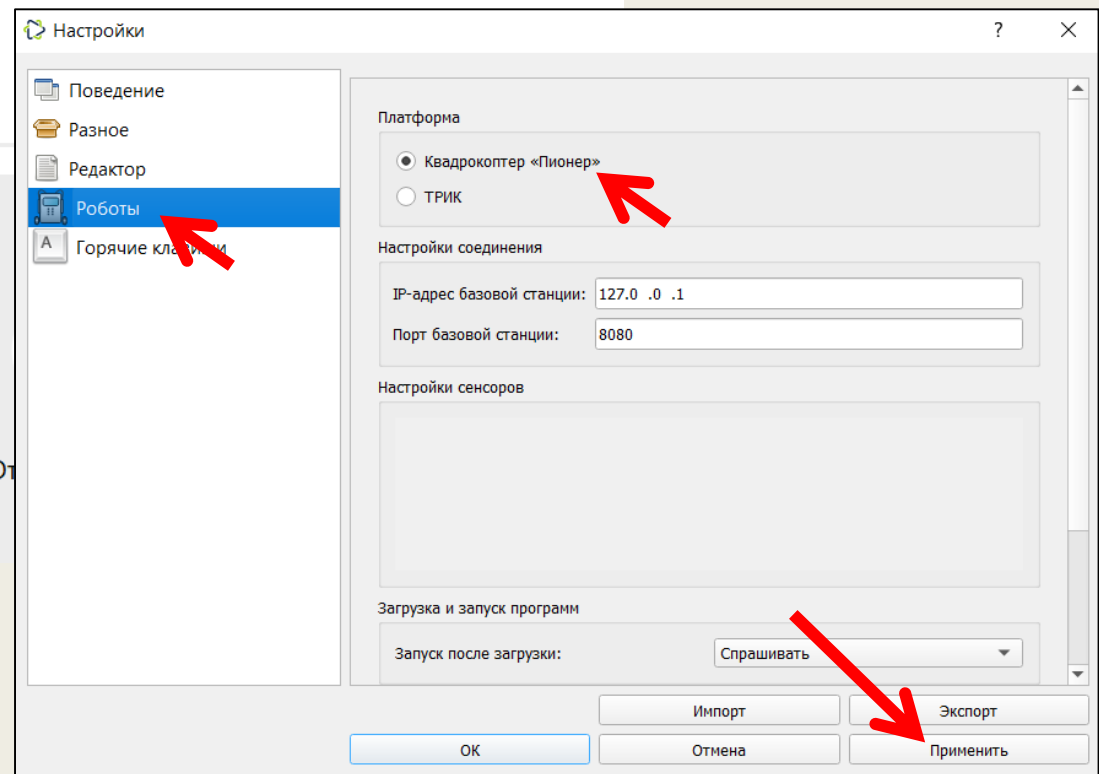
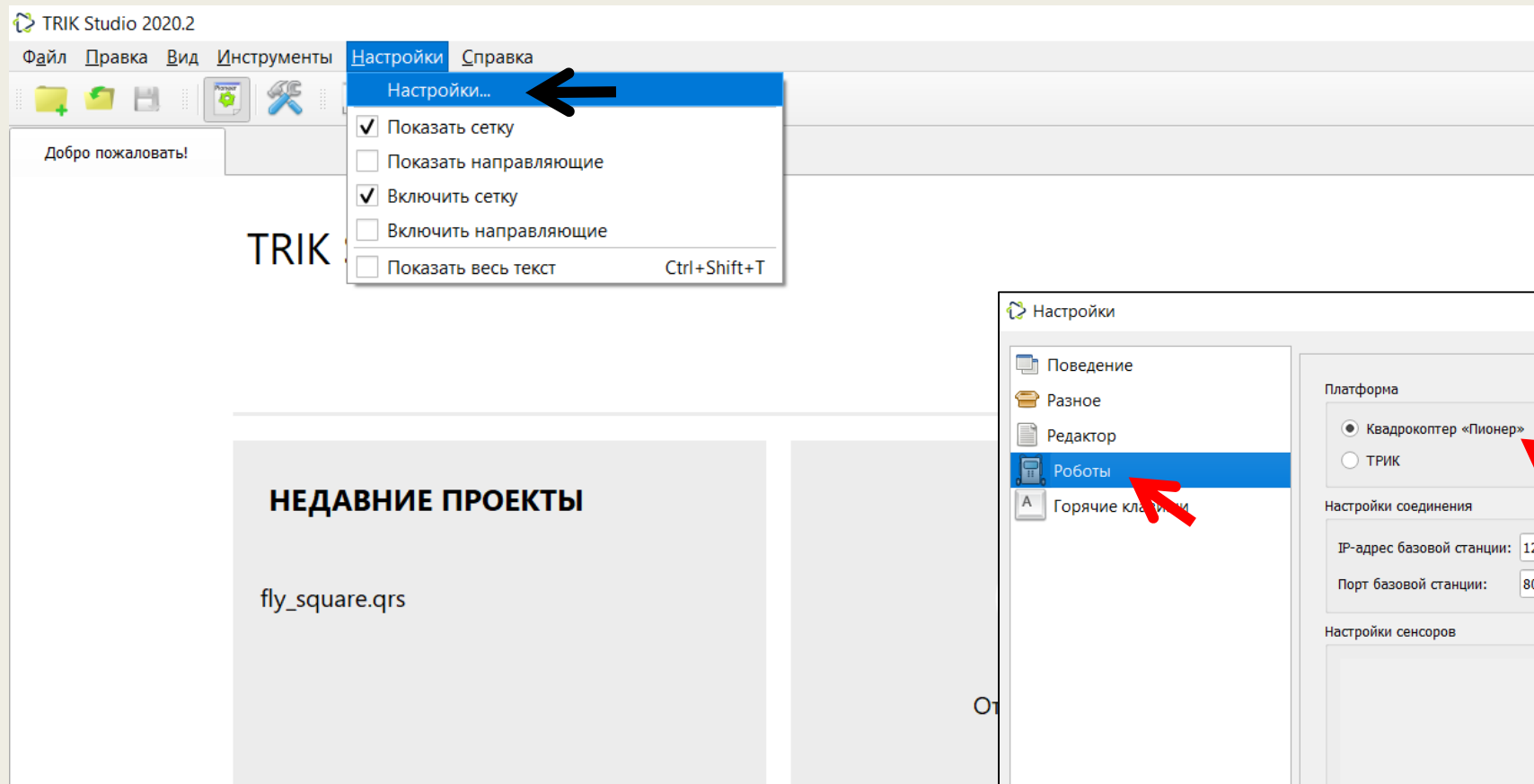
GEOSCAN



ИнтЭРА



TRIK STUDIO



Интерфейс TRIK STUDIO

1. Рабочая область

2. Палитра

3. Блок

4. Генерация кода для дрона

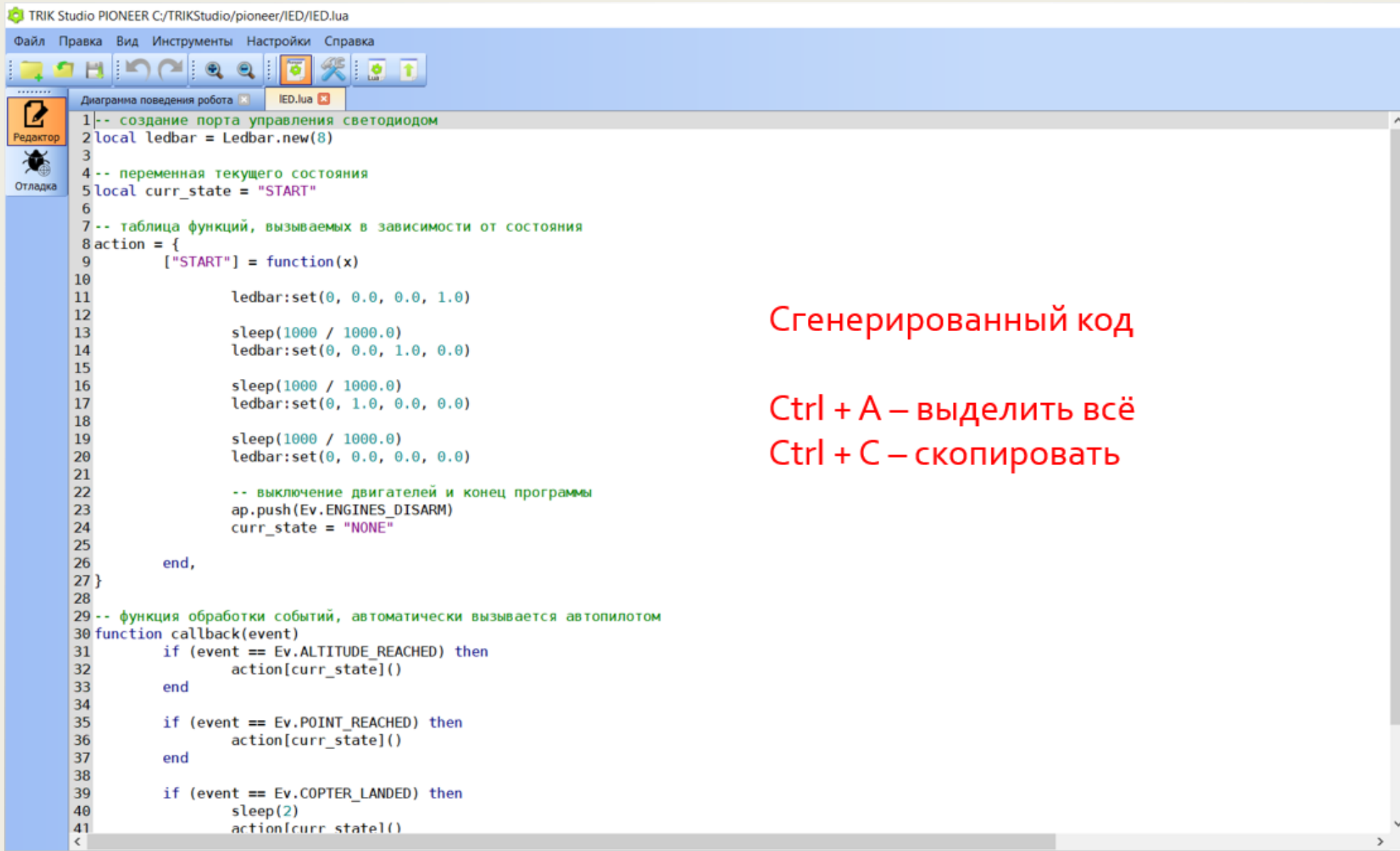
5. Отладчик

Ошибки

- [13:20:54]: **ОШИБКА:** От этого элемента должна отходить в точности одна связь
- [13:20:54]: **ОШИБКА:** Генерация закончилась с ошибкой. Возможные причины: внутренняя ошибка генератора или слишком сложная структура программы. Обратитесь к разработчикам.

Режим редактирования - нажмите Ctrl+2 или кликните здесь для переключения в режим отладки

Интерфейс TRIK STUDIO



```
TRIK Studio PIONEER C:/TRIKStudio/pioneer/IED/IED.lua
Файл Правка Вид Инструменты Настройки Справка
Диаграмма поведения робота IED.lua
1 -- создание порта управления светодиодом
2 local ledbar = Ledbar.new(8)
3
4 -- переменная текущего состояния
5 local curr_state = "START"
6
7 -- таблица функций, вызываемых в зависимости от состояния
8 action = {
9     ["START"] = function(x)
10
11         ledbar:set(0, 0.0, 0.0, 1.0)
12
13         sleep(1000 / 1000.0)
14         ledbar:set(0, 0.0, 1.0, 0.0)
15
16         sleep(1000 / 1000.0)
17         ledbar:set(0, 1.0, 0.0, 0.0)
18
19         sleep(1000 / 1000.0)
20         ledbar:set(0, 0.0, 0.0, 0.0)
21
22         -- выключение двигателей и конец программы
23         ap.push(Ev.ENGINES_DISARM)
24         curr_state = "NONE"
25
26     end,
27 }
28
29 -- функция обработки событий, автоматически вызывается автопилотом
30 function callback(event)
31     if (event == Ev.ALTITUDE_REACHED) then
32         action[curr_state]()
33     end
34
35     if (event == Ev.POINT_REACHED) then
36         action[curr_state]()
37     end
38
39     if (event == Ev.COPTER_LANDED) then
40         sleep(2)
41         action[curr_state]()
42     end
43 end
```

Сгенерированный код

Ctrl + A – выделить всё

Ctrl + C – скопировать

Интерфейс Pioneer Station

The screenshot shows the GEOSCAN Pioneer Station 1.9.1 software interface. The window title is "GEOSCAN Pioneer Station 1.9.1". The interface includes a toolbar with buttons for "Старт прогр." (Start program), "Ост. прогр." (Pause program), and "Ост. моторы" (Stop motors). Below the toolbar is a menu bar with "Редактор кода" (Code editor) and "Параметры автопилота" (Autopilot parameters). The main area is a code editor with a line number "1" on the left. A "Загрузить" (Load) button is visible in the toolbar. In the bottom right corner, there is a "Подключение" (Connection) dropdown menu. Red circles highlight the "Старт прогр." and "Загрузить" buttons. Red arrows point from the instructions to these buttons and the connection menu.

1. Написать или загрузить код

2. Подключить дрон по USB

3. Загрузить код на дрон

4. Нажать старт, если не нужно летать

Интерфейс Pioneer Station

GEOSCAN Pioneer Station 1.9.1


Старт прог. Ост. прог. Ост. моторы № 2 0,03 В

Редактор кода Параметры автопилота

```
1 -- создание порта управления светодиодом
2 local ledbar = Ledbar.new(8)
3
4 -- переменная текущего состояния
5 local curr_state = "START"
6
7 -- таблица функций, вызываемых в зависимости от состояния
8 action = {
9   ["START"] = function(x)
10     ledbar:set(0, 0.0, 0.0, 1.0)
11
12     sleep(1000 / 1000.0)
13     ledbar:set(0, 0.0, 1.0, 0.0)
14
15     sleep(1000 / 1000.0)
16     ledbar:set(0, 1.0, 0.0, 0.0)
17
18     sleep(1000 / 1000.0)
19     ledbar:set(0, 0.0, 0.0, 0.0)
20
21
22     -- выключение двигателей и конец программы
23     ap.push(Ev.ENGINES_DISARM)
24     curr_state = "NONE"
25
26   end,
27 }
28
29 -- функция обработки событий, автоматически вызывается автопилотом
30 function callback(event)
31   if (event == Ev.ALTITUDE_REACHED) then
32     action[curr_state]()
33   end
34
35   if (event == Ev.POINT_REACHED) then
36     action[curr_state]()
37   end
38
39   if (event == Ev.COPTER_LANDED) then
40     sleep(2)
41     action[curr_state]()
42   end
43 end
```

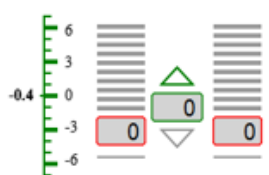
Ctrl+V – загрузим наш код и запустим

№ 2 0,03 В



0.0 В

№: 2
GPS:0
Канал:1



Высота Ват,м Газ,% V верт GPS,м/с V гор GPS,м/с

Отказы: LINK_ERROR

Датчик	Значение
Тангаж, °	-0,18
Крен, °	0,72
Рыскание, °	0,0
Занято памяти на LUA, КБ (из ...	25,7

Устройство	Версия
Autopilot	1.4.4922

Тип подключения: serial://COM5 Подключение

Блоки TRIK STUDIO



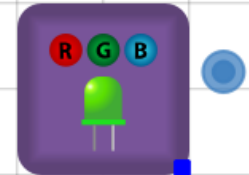
Начало программы



Конец программы

Светодиод

Номер: 0



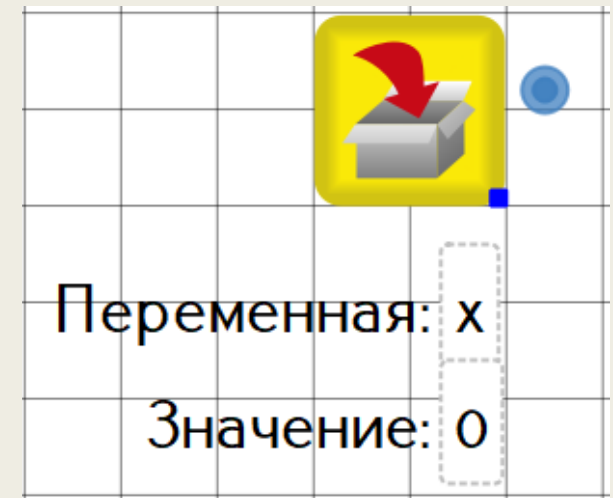
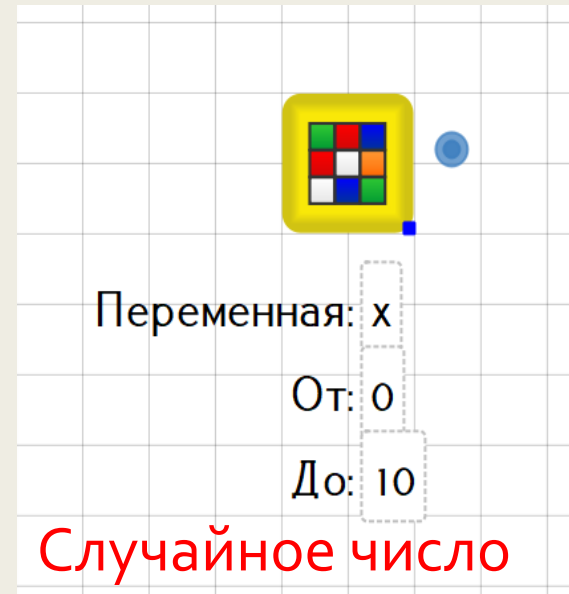
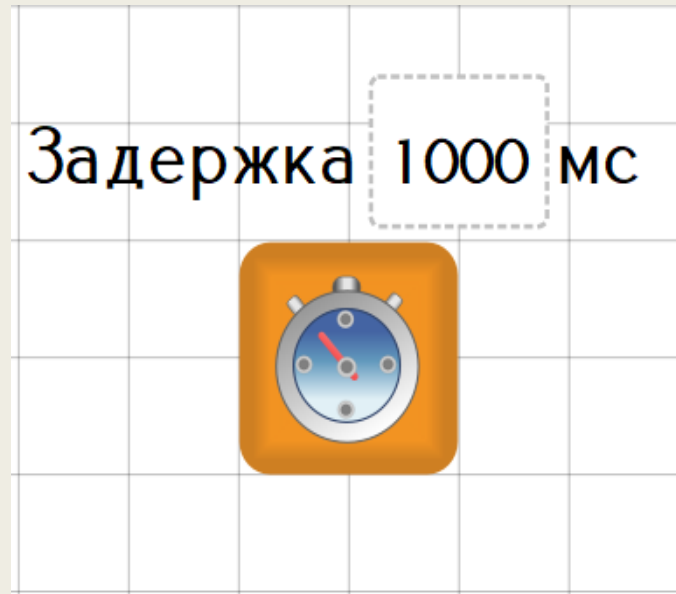
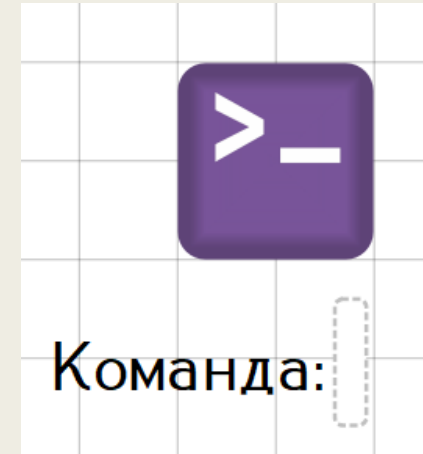
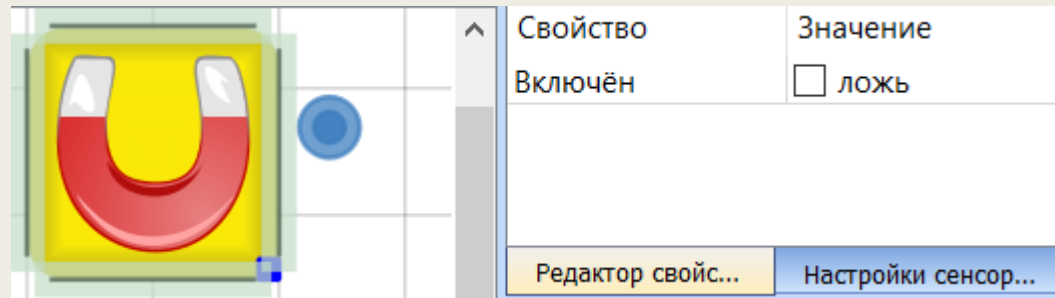
Красный: 0.0

Зелёный: 0.0

Синий: 0.0

Блоки TRIK STUDIO

Магнитный захват



Блоки управлением полётом TRIK STUDIO

Взлёт

Посадка

Лететь в точку

X: 0
Y: 0
Z: 0
Time:

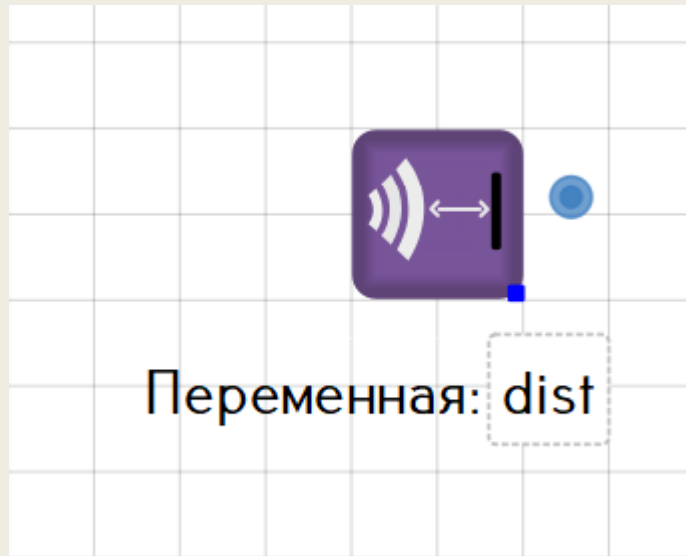
The image shows three yellow square blocks on a grid. The first block has a green upward arrow and is labeled 'Взлёт' (Takeoff). The second block has a green downward arrow and is labeled 'Посадка' (Landing). The third block has a green arrow pointing up and to the right, a red square target, and a blue dot; it is labeled 'Лететь в точку' (Fly to point). Below the third block are input fields for X: 0, Y: 0, Z: 0, and Time:.

Угол: 0.0

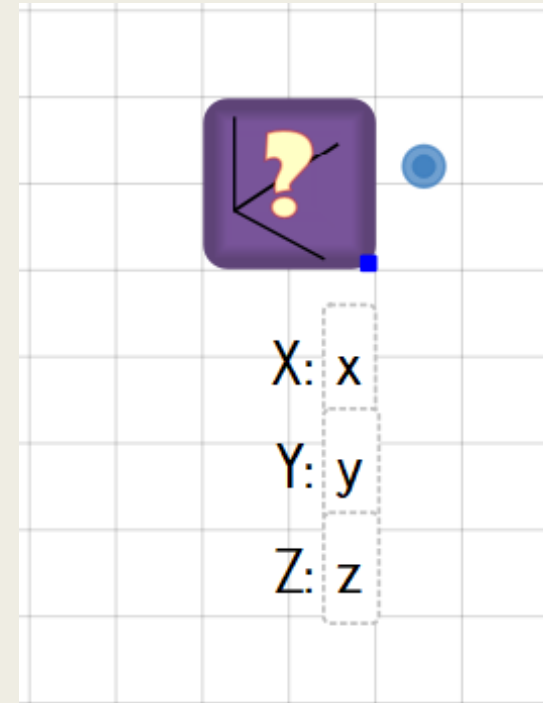
Рысканье

The image shows a yellow square block on a grid with a red triangle and two curved black arrows indicating rotation. Above the block is a dashed box containing the text 'Угол: 0.0'. Below the block is the label 'Рысканье' (Roll).

Специальные блоки



Прочитать датчик



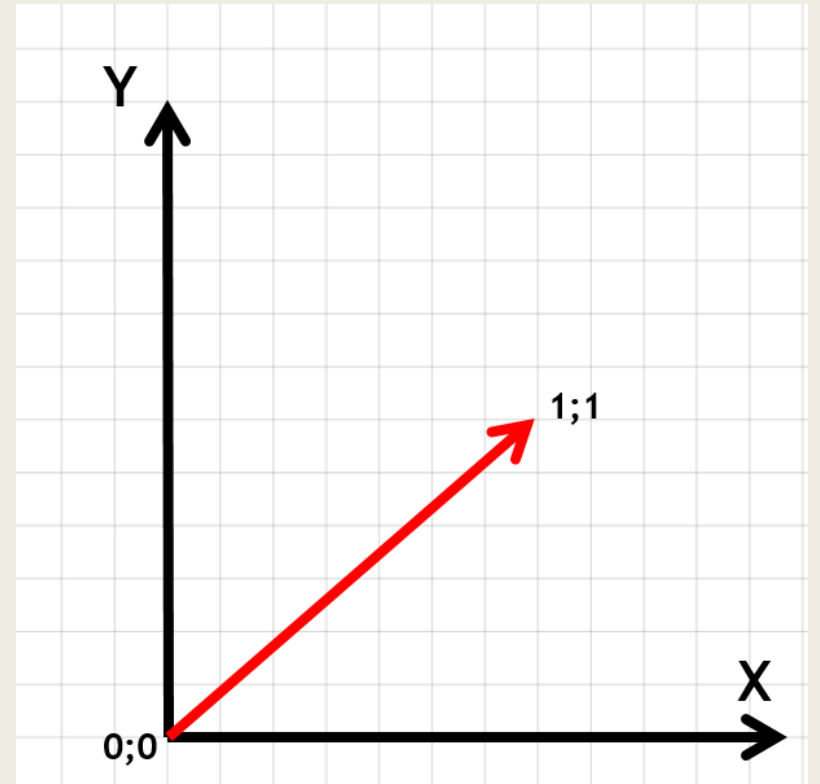
Получить точку (ЛК)

Задача

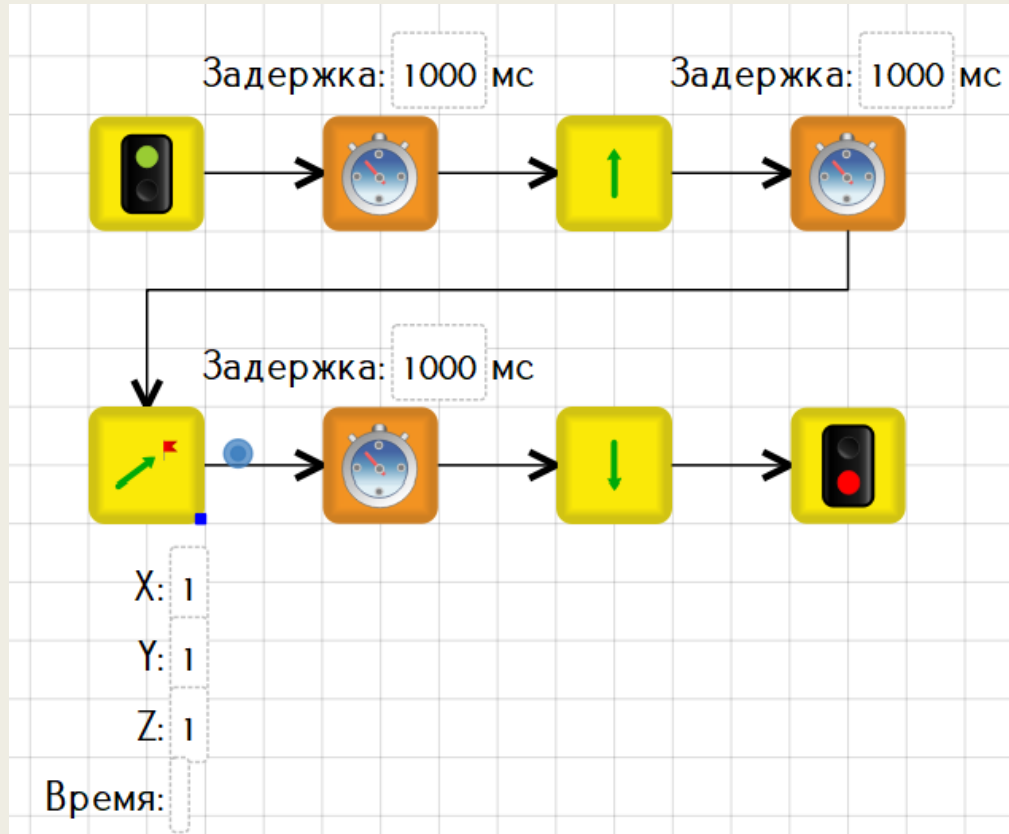
Взлететь.

Пролететь в точку на высоте 1 метр в точку $x=1, y=1$.

Сесть.



Решение



```
1 -- переменная текущего состояния
2 local curr_state = "_TIMER_1"
3 -- таблица функций, вызываемых в зависимости от состояния
4 action = {
5     ["_TIMER_1"] = function (x)
6         sleep(1000 / 1000.0)
7         ap.push(Ev.MCE_PREFLIGHT)
8         sleep(2)
9         ap.push(Ev.MCE_TAKEOFF)
10        -- переход в следующее состояние
11        curr_state = "_TIMER_2"
12    end,
13    ["_TIMER_2"] = function (x)
14        sleep(1000 / 1000.0)
15        ap.goToLocalPoint(1, 1, 1)
16        -- переход в следующее состояние
17        curr_state = "_TIMER_3"
18    end,
19    ["_TIMER_3"] = function (x)
20        sleep(1000 / 1000.0)
21        ap.push(Ev.MCE_LANDING)
22        -- переход в следующее состояние
23        curr_state = "_FINAL_NODE_1"
24    end,
25    ["_FINAL_NODE_1"] = function (x)
26        -- выключение двигателей и конец программы
27        ap.push(Ev.ENGINES_DISARM)
28        curr_state = "NONE"
29    end,
30 }
31 -- функция обработки событий, автоматически вызывается автопилотом
32 function callback(event)
33     if (event == Ev.TAKEOFF_COMPLETE) then
34         action[curr_state]()
35     end
36     if (event == Ev.POINT_REACHED) then
37         action[curr_state]()
38     end
39     if (event == Ev.COPTER_LANDED) then
40         sleep(2)
41         action[curr_state]()
42     end
43 end
44 -- вызов функции из таблицы состояний, соответствующей первому состоянию
45 action[curr_state]()
```

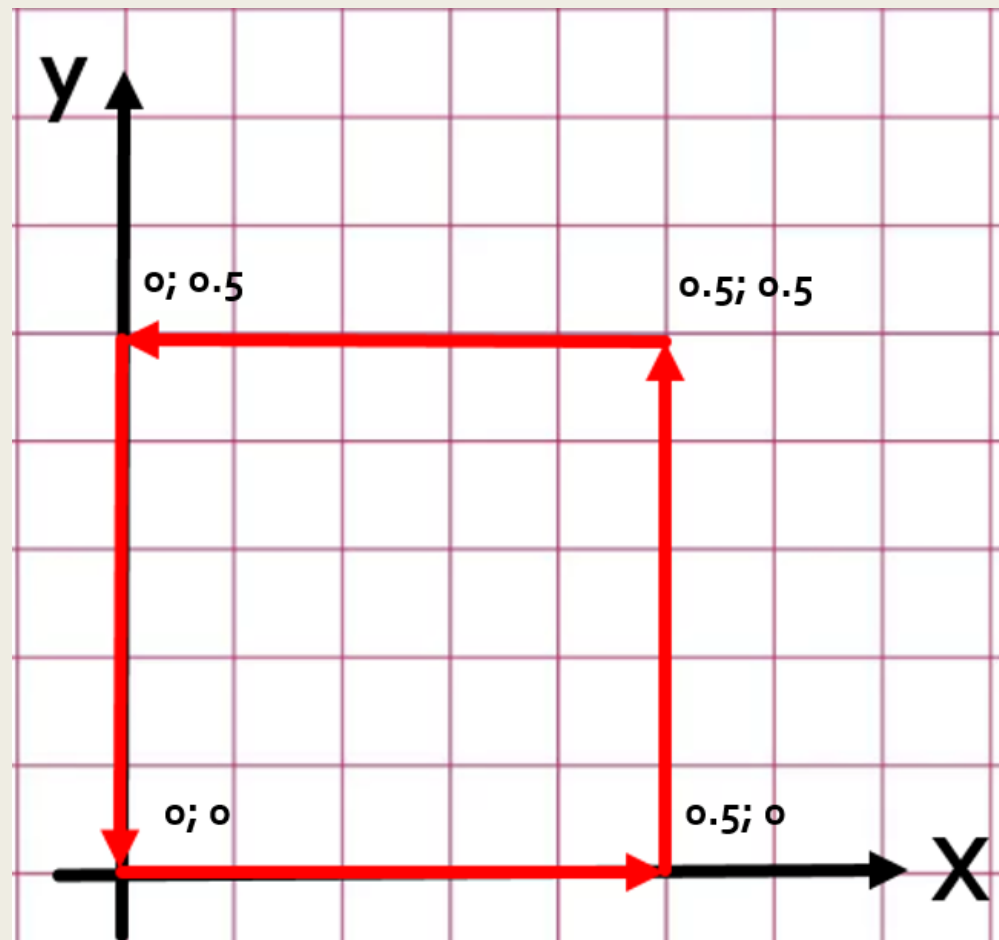
Задача

- Взлететь.

Пролететь по квадрату со стороной 0.5 м на высоте 0.4 м.

Сесть.

- При взлёте, посадке и на каждом повороте менять цвет лампочек на плате квадрокоптера.



Контакты

- Грибова Лада Александровна (программист)

- Почта: l.gribova@geoscan.aero

- WhatsApp: +7 965 704 53 03

Москва, ул. Большая Грузинская 12к2, Бизнес-центр Николаевский