

**Интеллектуальная Олимпиада
Приволжского федерального округа среди обучающихся школ и СПО 2025-2026 уч. г.
Задания по направлению «Программирование БПЛА»
Муниципальный этап**

Основные изучаемые понятия: поисковые траектории, повороты, циклы, стартовая высота, реакция на изменение параметра, обратная связь.

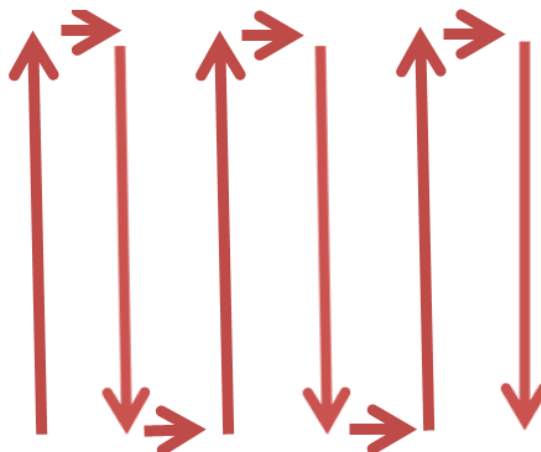
Постановка задачи, подготовка к работе:

Один из луноходов был потерян и не подает сигналов. Сектор поиска имеет плоскую поверхность, но достаточно обширен. Задача – произвести поиск лунохода по оптимальной траектории, обнаружить луноход с помощью высоотомера и сигнализировать об успехе миссии с помощью «кувырка».

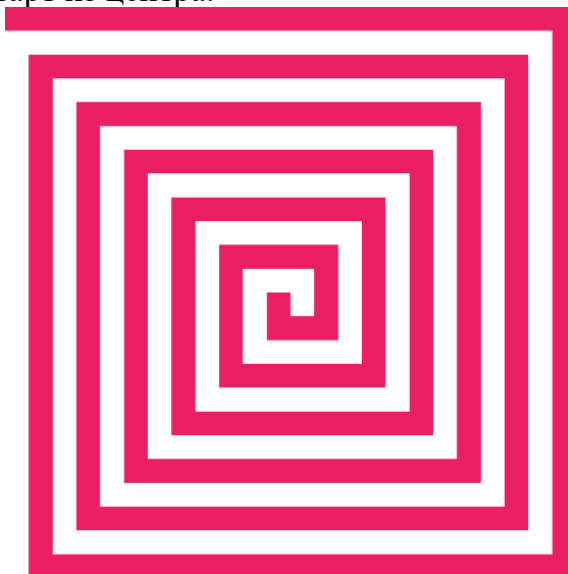
Сектор (квадрат) поиска обозначен на полу с помощью цветного скотча. Оптимально квадрат должен иметь сторону 1.5-2 метра. Возможно также использовать «лунное» поле из робототехнического комплекта LEGO или большую полетную карту Tello. В произвольное для каждой команды место сектора будет установлен луноход LEGO, причем это происходит после запуска программы командой. Оценивается алгоритм и время поиска объекта.

Обсуждение вариантов алгоритма:

1) Линейный проход от края до края сектора (его размеры известны) и обратно со сдвигом вбок. При этом квадрокоптер можно даже не поворачивать. Старт (в данном случае) из левого нижнего угла;



2) Проход по спирали. Старт из центра.



Возможен и любой другой вариант траектории, который кажется обучающимся эффективным.

**Интеллектуальная Олимпиада
Приволжского федерального округа среди обучающихся школ и СПО 2025-2026 уч. г.
Задания по направлению «Программирование БПЛА»
Муниципальный этап**

Принцип работы и применение высотомера

Инфракрасный высотомер, расположенный снизу квадрокоптера [Tello Edu](#), измеряет расстояние между квадрокоптером и поверхностью под ним.

Он отправляет инфракрасный сигнал и определяет время, потребовавшееся для его отражения от поверхности и возврата к датчику. Поскольку скорость света известна, можно использовать полученное время для расчета расстояния между квадрокоптером и объектом.

Функция `get_tof()`, входящая в библиотеку `tello_binom`, выдает данные о расстоянии от квадрокоптера до поверхности земли (или объекта под квадрокоптером) в миллиметрах.

Попробуем получить значение, выдаваемое функцией `get_tof()`, т.е. получить высоту квадрокоптера.

После взлета и стабилизации функция `get_tof()` вернет высоту квадрокоптера около 800 мм (с погрешностью в пределах 10%).

Мы сохраним это значение в переменной с именем `initial_height`, а затем используем его для сравнения с текущей высотой. При пролете над объектом (луноходом) она изменится.

Необходимо, чтобы квадрокоптер измерял расстояние до поверхности (или объекта) в непрерывном режиме, а не один раз. Для этого будем использовать цикл.

Например, с такой программой

```
...
for i in range(10): # Повторить 10 раз
    forward(20) # Вперед на 20 см
    current_height = get_tof()
    if current_height <= initial_tof - 120: # 120 мм на погрешность
        flip_forward()
        break
...
```

квадрокоптер пролетит 2 метра, через каждые 20 см проверяя, не уменьшилась ли высота по сравнению с исходной. Если это произошло, необходимо дать обратную связь и прекратить поиск.

00:00

00:46

Сканирование и обнаружение объекта на поверхности

Вариант программы линейного поиска объекта

```
from tello_binom import *
```

```
start() # Начало отправки команд квадрокоптеру
takeoff() # Взлет
obj_found = False # Объект пока не найден
initial_height = get_tof()
for i in range(10):
    for j in range(10):
        if i % 2 == 0: # Если шаг четный, летим вперед
            forward(20)
        else:
            backward(20)
```

**Интеллектуальная Олимпиада
Приволжского федерального округа среди обучающихся школ и СПО 2025-2026 уч. г.
Задания по направлению «Программирование БПЛА»
Муниципальный этап**

```
current_height = get_tof() # Текущая высота
if current_height <= initial_height - 120: # 120 мм погрешность
    flip_forward()# Кувырок
obj_found = True # Объект найден
break # Выход из внутреннего цикла
if obj_found:
    break # Выход из внешнего цикла
right(20) # Сдвиг вправо
land() # Посадка
```