Координаты аэропортов

```
from m5stack import *
from m5ui import *
from uiflow import *
from m5mqtt import M5mqtt



airports = [(0, 0), (1, 3), (2, 5), (1, 8)]
airport_index = 2
```

Расположение надписей

```
label0 = M5TextBox(10, 29, "Text", lcd.FONT_Default, 0xFFFFFF, rotate=0)
label1 = M5TextBox(10, 63, "Text", lcd.FONT_Default, 0xFFFFFF, rotate=0)
label2 = M5TextBox(10, 100, "Text", lcd.FONT_Default, 0xFFFFFF, rotate=0)
label3 = M5TextBox(10, 147, "Text", lcd.FONT_Default, 0xFFFFFF, rotate=0)
label4 = M5TextBox(10, 190, "Text", lcd.FONT_Default, 0xFFFFFF, rotate=0)
setScreenColor(0x222222)
```

Расстояние между аэропортом и самолетом

```
def calculate_distance(point1, point2):
    return (                                    )
```

Расстояние между аэропортом и самолетом

```
def calculate_distance(point1, point2):
    return ((point1[0] - point2[0])**2 +
                (point1[1] - point2[1])**2)**0.5)
```

Функция публикации сообщения самолету

```
def notify_plane(plane_number):
    message = "{} {} {} {}".format(plane_number, airport_index,
                                   airports[airport_index][0],
                                   airports[airport_index][1])
    m5mqtt.publish('ligarobotov/lvl14/09/plane', message)
```

Функция обработки сообщения от самолета

```
def plane_message_recieve(topic_data):

    label0.setText(topic_data)

    plane_message = topic_data.split(' ')
    if len(plane_message) != 3:
        return 0
```

Функция публикации сообщения самолету

```
if len(plane_message) != 3:
    return 0


plane_number = int(         )
plane_x = float(            )
plane_y = float(              )
plane_coordinates = (plane_x, plane_y)
```

Функция публикации сообщения самолету

```
if len(plane_message) != 3:
    return 0


plane_number = int(plane_message[0])
plane_x = float(plane_message[1])
plane_y = float(plane_message[2])
plane_coordinates = (plane_x, plane_y)
```

Формирование списка с номером аэропорта и дистанцией до самолета

```
plane_coordinates = (plane_x, plane_y)


distances = []
    for i in range(len(airports)):
        distance = (i, calculate_distance(                    )
        distances.append(distance)
```

Формирование списка с номером аэропорта и дистанцией до самолета

```
plane_coordinates = (plane_x, plane_y)


distances = []
    for i in range(len(airports)):
        distance = (i, calculate_distance(plane_coordinates,
                                          airports[i]))
        distances.append(distance)
```

Сортировка списка координат

```
def sort_by_distance(x):
    return x[1]


distances.sort(key=sort_by_distance)
label3.setText(str(distances))


if                  == airport_index:
    notify_plane(plane_number)
```

Сортировка списка координат

```
def sort_by_distance(x):
    return x[1]


distances.sort(key=sort_by_distance)
label3.setText(str(distances))

if distances[0][0] == airport_index:
    notify_plane(plane_number)
```

```
m5mqtt = M5mqtt('', 'broker.hivemq.com', 1883, '', '', 300)
m5mqtt.subscribe('ligarobotov/lvl14/09/airport',
airport_message_recieve)
m5mqtt.subscribe('ligarobotov/lvl14/09/plane’, plane_message_recieve)
m5mqtt.start()
```