

§4 Заключительный этап: командная часть

В командной части заключительного этапа участники разрабатывали автоматизированную систему управления летательным аппаратом мультироторного типа и демонстрировали полученный результат на конкретном летательном аппарате в условиях замкнутого объема. Продолжительность командной части — 3,5 дня.

4.1. Набор заданий

Команды выполняли набор заданий. Проверка задания выполняется на рабочем месте команды или специальном стенде по заданным объективным критериям. За каждое выполненное задание команда получает баллы. Команда, которая первая правильно выполнит задание получала максимальный балл.

Задание	Описание задачи
1. Сборка летательного аппарата	Сборка квадрокоптера из комплектующих по инструкции
2. Подключение переключателя	Подключение специального устройства для обеспечения ручного и автоматического полета
3. Настройка аппарата	Настроить аппарат по соответствующим рекомендациям организаторов и изучить возможности программного обеспечения
4. Калибровка датчика расстояния	Изучить принцип работы оптического датчика, провести его калибровку, используя эталонное средство измерения и показать полученный результат
5. Запуск двигателей	Написание программы запуска двигателей и проверка её работоспособности
6. Взлет на определенную высоту	Разработать алгоритм и программное обеспечение, позволяющее летательному аппарату в автоматизированном режиме взлетать на определенную высоту и зависнуть на ней на определенное время
7. Полет по заданной траектории с препятствиями	Разработать алгоритм и программное обеспечение, позволяющее летательному аппарату в автоматизированном режиме обеспечить полет по определенной траектории, при этом аппарат должен распознавать препятствия (стенки) и не врезаться в них. Затем также продемонстрировать полученный результат в реальном полете.

4.2. Описание конструктора - учебно-тренировочного комплекса на базе летательного аппарата мультироторного типа «Колибри»

Каждая команда получает следующий набор для конструирования и инструменты:

- Пульт дистанционного управления.

- Приемник радиуправления.
- Рама пластиковая из двух деталей.
- Защита винтов.
- Электромотор и регулятор – 4 шт..
- Пропеллеры – 4шт.
- Датчик расстояния – 3 шт.
- Полетный контроллер.
- Индикатор.
- Плата питания.
- Контроллер ARDUINO и плата переходник.
- Набор проводов.
- Зарядное устройство.
- Аккумулятор 1300 мАч – 2шт.
- Набор винтов М3 и гаек.
- Отвертка.
- Длинногубцы.
- Набор пластиковых стоек.
- USB кабель для подключения полетного контролера.
- USB кабель для подключения ARDUINO.
- Контейнер пластиковый (цвет синий) 340x230x160 мм.
- Ноутбук с установленным программным обеспечением.

Собранный из этого конструктора летательный аппарат способен совершать полет как в замкнутом пространстве, так и на открытой территории

Упакованный конструктор и летательный аппарат в собранном виде из данного конструктора представлены ниже на соответствующих рисунках.



Рис. Упакованный конструктор

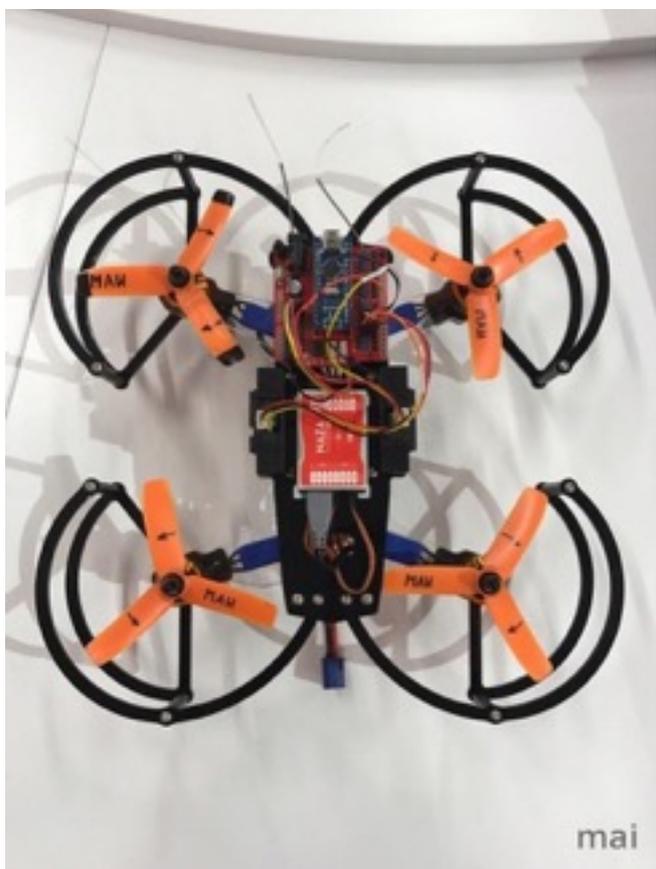


Рис. Летательный аппарат в собранном виде

4.3. Описание стенда для испытаний

Стенд для испытаний представляет собой пластиковый куб, со следующими размерами 300см*300см*200см (длина, ширина, высота). Данный стенд предназначен для демонстрации полета летательного аппарата в автоматизированном режиме и отработки соответствующих задач каждой командой. Также стенд обеспечивает безопасность полетов. Внешний вид стенда представлен ниже на рисунке.

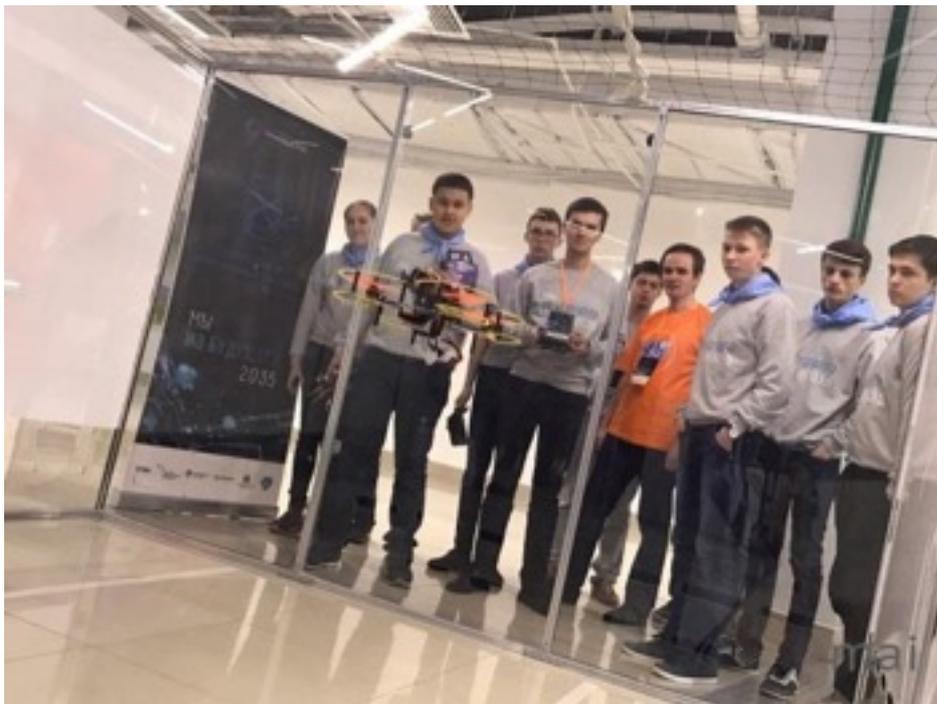


Рис. Внешний вид стенда и полет летательного аппарата

4.4. Программирование полета аппарата

Программирование полета аппарата осуществляется в среде ARDUINO.

Среда ARDUINO: <http://arduino.ru/>

Ниже на рисунке представлен внешний вид среды разработки программного обеспечения ARDUINO.



Рис. Среда разработки ARDUINO.

4.5. Условие финальной задачи

Наиболее сложные задания 6 и 7 должны быть выполнены на описанном стенде, причем качество результата полета контролируется опытными специалистами. Каждая команда имеет право выполнять задачу несколько раз, до тех пор, пока она не захочет перейти к выполнению следующей задачи. Каждая команда понимает сколько баллов она может заработать, оценивая текущую ситуацию у других команд.

4.6. Что требует решение задачи

Для решения поставленной задачи (начиная с самого простого задания №1 и до самого сложного задания №7) учащиеся должны обладать следующими знаниями и навыками:

- Умение работать с инструкциями по сборке.
- Умение разрабатывать алгоритмы и программное обеспечение для микроконтроллера в среде ARDUINO.
- Понимать принципы управления летательным аппаратом мультироторного типа.
- Уметь работать с датчиками и понимать принципы измерений.

4.7. Критерии оценки

Оценка работы происходит следующим образом: команда, выполнившая первая правильно определенное задание набирает максимальное количество баллов. Из трех участвующих команд, та команда, которая правильно и быстрее всех выполняет задание набирает 3 балла. Вторая команда, выполнившая правильно задание набирает 2 балла. Команда, выполнившая задание позже остальных набирает 1 балл.

Решение:

Пример программы в среде ARDUINO, обеспечивающей решение финального задания:

```
1 #define pin1 A5
2 #define pin2 A6
3 #define pin3 A7
4 const double division_ADC = 0.0048828125;
5 double dist1;
6 double dist2;
7 double dist3;
8 double distA;
9 double sum1 = 0;
10 double sum2 = 0;
11 double sum3 = 0;
12 int count1 = 0;
13 int count2 = 0;
14 int count3 = 0;
15 #include <Servo.h>
16 Servo serv1;
17 Servo serv2;
18 int flag = 1, p, k = 0, l = 1, v=0;
19
20
21 void setup() {
22
23 Serial.begin(9600);
24 serv1.attach(3);
25 serv2.attach(4);
```

```

26 serv2.writeMicroseconds(1500);
27
28}
29
30 void loop() {
31   for(int i = 0; i < 150; i++){
32     dist1 = analogRead(pin1)*division_ADC;
33     dist1 = 27.86*pow(dist1, -1.15);
34     if(dist1 >= 75){count1+=1;}
35     sum1+=dist1;
36     if (dist1 >= 75 && dist1 <= 85){k = dist1;}
37     dist2 = analogRead(pin2)*division_ADC;
38     dist2 = 27.86*pow(dist2, -1.15);
39     if(dist2 >= 80){count2+=1;}
40     sum2+=dist2;
41     dist3 = analogRead(pin3)*division_ADC;
42     dist3 = 27.86*pow(dist3, -1.15);
43     if(dist3 >= 80 || dist3<0){count3+=1;}
44     sum3+=dist3;
45
46   }
47   int k1 = sum1/150;
48   int k2 = sum2/150;
49   int k3 = sum3/150;
50   if (count1 >= 70) {k1 = 80;}
51   if (count2 >= 70) {k2 = 80;}
52   if (count3 >= 70) {k3 = 80;}
53   Serial.print("Left: ");
54   Serial.print(k3);
55   Serial.print(", down: ");
56   Serial.print(k1);
57   Serial.print(", right: ");
58   Serial.println(k2);

```

```
59 sum1 = 0;
60 sum2 = 0;
61 sum3 = 0;
62 count1 = 0;
63 count2 = 0;
64 count3 = 0;
65
66 if (k2 <= 50 && k == 1){
67     serv1.writeMicroseconds(1200);
68     l = 1;
69     k = 0;
70 }else if(k2 <=30 && k == 1){
71     l = 1;
72     k = 0;
73     serv1.writeMicroseconds(1000);
74 }
75 if (k3 <= 50 && l == 1){
76     serv1.writeMicroseconds(1700);
77     l =0;
78     k = 1;
79 }else if(k3 <=30 && l == 1){
80     l = 0;
81     k = 1;
82     serv1.writeMicroseconds(2000);
83 }
84 if (l == 1 && k3 >= 55){
85     serv1.writeMicroseconds(1200);
86 }
87 if (k == 1 && k2 >= 55){
88     serv1.writeMicroseconds(1700);
89 }
90}
```