

РЕАЛИЗАЦИЯ АЛГОРИТМА ВЫСОКОТОЧНОЙ ПОСАДКИ НА МЕТКУ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА «ПИОНЕР» С ИСПОЛЬЗОВАНИЕМ ПРОГРАММИРУЕМОЙ КАМЕРЫ OPENMV

Е.И. Червинко, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, proleaveprod@gmail.com;

И.А. Иванов, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича.

УДК 621.396.1

Аннотация. В работе представлена реализация алгоритма высокоточной посадки беспилотного летательного аппарата «Пионер» на метку с использованием программируемой камеры *OpenMV*. Разработка и реализация алгоритма происходила во время прохождения факультативного курса «Основы устройства, программирование и пилотирование квадрокоптеров» на базе научно-образовательного центра «ТИОС» СПбГУТ. Основной задачей работы являлось получение практических навыков по алгоритмизации автономных полетов дронов с применением дополнительных совместимых модулей.

Ключевые слова: разработка электроники; робототехника; алгоритмизация; программирование на *Lua*; программирование на *Python*; БПЛА, мультикоптер.

IMPLEMENTATION OF THE ALGORITHM FOR HIGH-PRECISION LANDING ON THE LABEL OF THE PIONEER UNMANNED AERIAL VEHICLE USING THE OPENMV PROGRAMMABLE CAMERA

E.I. Chervinko, St. Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruevich, proleaveprod@gmail.com;

I.A. Ivanov, St. Petersburg State University of Telecommunications named after Prof. M. A. Bonch-Bruevich.

Annotation. The paper presents the implementation of the algorithm for high-precision landing on the label of the unmanned aerial vehicle «Pioneer» using the programmable camera *Open MV*. The development and implementation of the algorithm took place during the optional course «Fundamentals of the device, programming and piloting of quadrocopters» on the basis of the scientific and educational center «*TIOS*» SPbGUT. The main objective of the work was to gain practical skills in algorithmization of autonomous drone flights using additional compatible modules.

Keywords: electronics development; robotics; algorithmization; *Lua*; *Python*; UAV; multicopter.

Введение

Осуществление высокоточной посадки позволяет не только расширить сферу применения БПЛА, но также дает возможность применять автономные устройства для обслуживания летательных аппаратов. Примером такого устройства является станция автоматической смены аккумулятора БПЛА коптерного типа. В ходе реализации работы была поставлена задача разработать алгоритм высокоточной посадки БПЛА на станцию перезарядки, после которой следовала автономная механическая замена аккумулятора. В процессе такой работы одной из самых существенных задач является высокоточная посадка дрона на специальную платформу. Эти факторы повлияли на создание алгоритма посадки БПЛА на специальную *april*-метку (*AprilTag*) [1].

Технико-технологическое обеспечение работы

В качестве БПЛА использовался квадрокоптер «Пионер» от компании «Geoscan» (рис. 1). БПЛА «Пионер» оснащен базовой платой, выполняющей основные функции при использовании квадрокоптера, а именно: включение и выключение двигателей, обработка операций, получение команд с пульта управления, оценка параметров барометра и гироскопа и т.д. [2].



Рисунок 1

В качестве основного подключаемого модуля была использована программируемая камера *OpenMV* (рис. 2). Данная камера может самостоятельно обрабатывать видеопоток и посылать команды управления на квадрокоптер «Пионер» [3]. Камера взаимодействует с основной платой через интерфейс *UART* [4].



Рисунок 2

Для обозначения места посадки была использована технология *AprilTag*. *AprilTag* – это библиотека тестов машинного зрения, широко используемая в областях дополненной реальности, роботов и калибровки камер. С помощью специальных знаков (похожих на *QR*-коды, но с меньшей сложностью для соответствия требованиям реального времени) знаки могут быть быстро обнаружены и рассчитаны относительные положения БПЛА в пространстве [1]. *AprilTag* позволяет камере получить точные данные о положении дрона, обработать их и передать необходимую информацию на основную плату. На рис. 3 показан пример *AprilTag*.



Рисунок 3

Реализация алгоритма высокоточной посадки на метку беспилотного летательного аппарата «Пионер» с использованием программируемой камеры *OpenMV*

Написание программной модели для квадрокоптера «Пионер» осуществлялось на языке программирования *Lua* [5]. Программирование камеры *OpenMV* осуществлялось на языке программирования *MicroPython*. Основное взаимодействие между камерой и квадрокоптером происходит через *UART* интерфейс, по которому передаются как основные командные запросы, так и полезная информация [2].

При выполнении работы были разработаны две блок-схемы. Первая блок-схема (рис. 4) отражает принцип работы алгоритма на стороне квадрокоптера «Пионер», вторая блок-схема (рис. 5) отражает принцип работы алгоритма на стороне камеры *OpenMV*. На рис. 4 показана схема работы алгоритма на стороне квадрокоптера «Пионер».

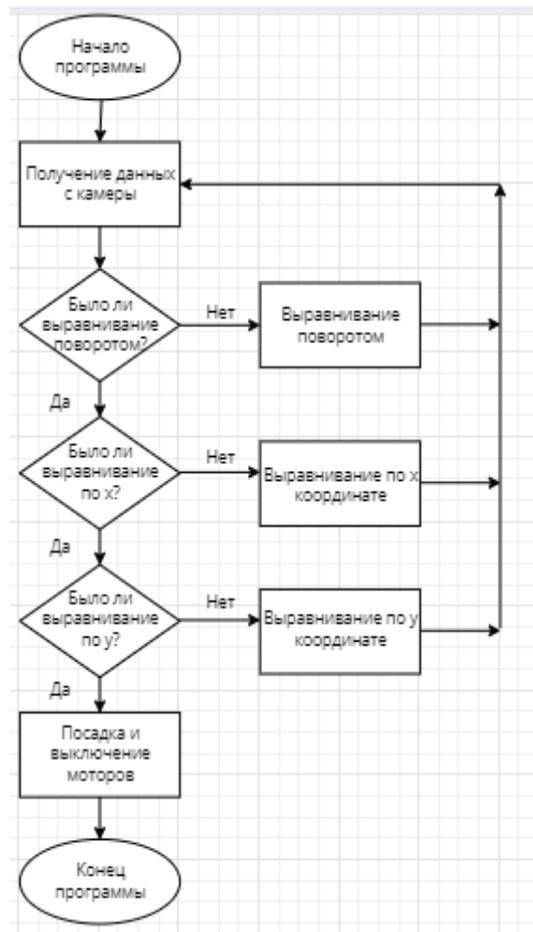


Рисунок 4

Программа выполняется при нажатии кнопки «Старт» на плате полетного контроллера. Основными функциональными блоками выполнения программы являются:

1) *Получение данных с камеры.* За получение данных с камеры ответственна функция, которая периодически читает приходящие через интерфейс *UART* сообщения:

```
function getData() -- функция приема байта данных
  buf = uart:read(uart:bytesToRead() or '0')
  if (#buf == 0) then buf = '\0' end
  if (strUnpack ~= nil) then
    local b = strUnpack("B", buf)
    return b
  else
    return 20
  end end
```

Полученное число передается другой функции, где происходит его обработка. При реализации алгоритма были выделены специальные числа, при получении которых происходит переключение флагов, отражающих текущее состояние дрона. Например, при получении числа 10 дрон закончит выравнивание поворотом и перейдет к выравниванию по оси *X*.

2) *Проверка текущего состояния дрона.* При помощи оператора ветвления определяется текущее состояние квадрокоптера и предпринимаются определенные действия. Если дрон выровнялся по координатам *x* и *y*, а также имеет допустимое рыскание, то произойдет его посадка.

```
if(yawFlag==false)then
  if(negNumFlag==false)then changeColor(0,0,0.5) ap.updateYaw(x)
  else changeColor(0.2,0.2,0) ap.updateYaw(x*(-1)) end
elseif(xFlag==false)then
  if(negNumFlag==false)then changeColor(0.2,0,0.2) ap.goToLocalPoint(lx-
0.05,ly,lz,1)
  else changeColor(0,0.2,0.2) ap.goToLocalPoint(lx+0.05,ly,lz,1) end
elseif(yFlag==false)then
  if(negNumFlag==false)then changeColor(0,0,0.5) ap.goToLocalPoint(lx,ly-
0.05,lz,1)
  else changeColor(0.2,0.2,0) ap.goToLocalPoint(lx,ly+0.05,lz,1)end
else
  if(x==2)then changeColor(0,0.5,0)else changeColor(0.5,0,0)end end
```

Стоит отметить, что в данном случае посадка произойдет только, если *ID* тега будет равно 2. Соответственно, существует возможность выполнения действий, привязанных к определенному *ID* тега.

Теперь рассмотрим работу алгоритма программы на стороне камеры *OpenMV*.

На рис. 5 показана схема работы алгоритма на стороне камеры *OpenMV*.

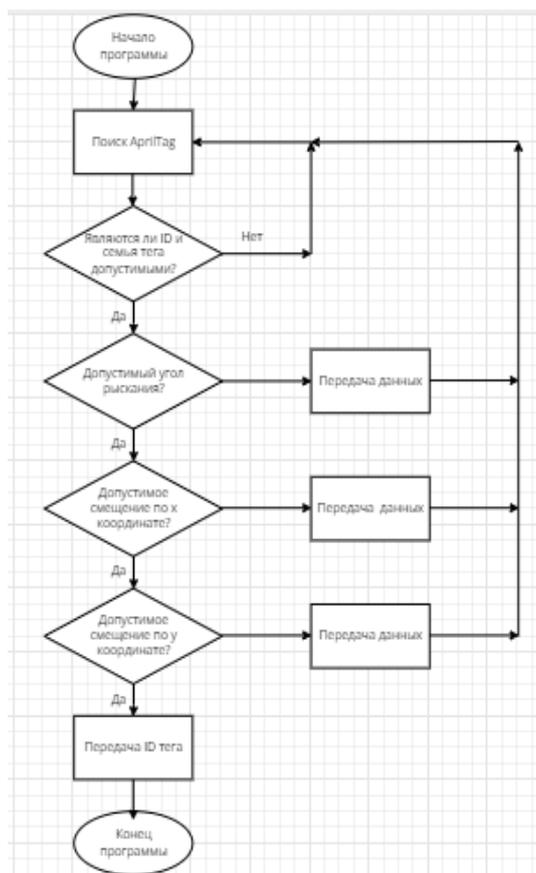


Рисунок 5

1) Начало программы: программа начинает исполняться после запуска камеры *OpenMV*.

2) Поиск *AprilTag*: камера делает снимок через определенный временной интервал и пытается обнаружить *AprilTag*.

3) В случае обнаружения *AprilTag* происходит проверка на соответствие *ID* тега допустимым значениям. Если *ID* тега подходит, то камера начинает считывать данные с тега, получая угол рыскания, смещение по осям *x* и *y*.

4) При помощи оператора ветвления определяется текущее состояние дрона, и камера передает необходимую информацию на основную плату. Для выравнивания поворотом передается количество градусов на которое необходимо повернуться, для выравнивания по координатам передаются специальные символы, сообщающие в диапазоне отрицательных или положительных чисел где, находится квадрокоптер на осях *x* и *y* относительно *AprilTag*.

5) Если положение дрона находится в допустимых значениях, то на контроллер передается *ID* тега для выполнения посадки или других запрограммированных действий.

В качестве практической реализации алгоритма была выбрана посадка на метку, обозначенную *AprilTag*. Разработка и написание алгоритма, а также тестирование системы проводилось на базе лаборатории научно-образовательного центра «ТИОС».

Во время выполнения посадки дрон находился на высоте 1 м от пола. На этой высоте камера смогла обнаружить подходящий *AprilTag* и передавала необходимые данные на полетный контроллер до момента окончания посадки. При этом дрон указывал светодиодной индикацией смену каждой фазы выравнивания.

На рис. 6 изображен вид станции замены АКБ для БПЛА на основе алгоритма посадки по *AprilTag*.



Рисунок 6

Заключение

Решение проблемы высокоточной посадки квадрокоптера решает огромный спектр задач в стремительно развивающейся сфере БПЛА, однако самым актуальным в этой работе является ее внедрение в более масштабный проект по разработке интеллектуальной системы замены аккумулятора для БПЛА. На данный момент продолжается улучшение алгоритма посадки квадрокоптера на *april*-метку. Проект позволит решить одну из основных проблем использования БПЛА коптерного типа – невозможность длительного эксплуатирования БПЛА без участия оператора. В дальнейшем планируется расширение возможных действий, выполняемых в зависимости от *ID* тега, а также снижение погрешности при выравнивании и посадке.

Литература

1. AprilTag – Официальный сайт проекта. – URL: <https://april.eecs.umich.edu/software/apriltag> (дата обращения 2022-07-26).
2. Геоскан Пионер Документация. – URL: <https://docs.geoscan.aero/ru/master/index.html> (дата обращения 2022-07-26).
3. Программируемая камера OpenMV. Документация. – URL: <https://docs.geoscan.aero/ru/master/module/openMV.html> (дата обращения 2022-07-25).
4. Магда Ю.С. Программирование последовательных интерфейсов. – БХВ-Петербург, 2009.
5. Иерусалимски Р. Программирование на языке Lua. – Litres, 2022.

Работа выполнена в рамках прикладных научных исследований СПбГУТ, регистрационный номер 122020100465-3 от 01.02.2022 в ЕГИСУ НИОКТР.