This article presents a survey on publicly available open-source projects (OSPs) on quadrotor unmanned aerial vehicles (UAVs).

Recently, there has been increasing interest in quadrotor UAVs. Exciting videos have been published on the Internet by many research groups and have attracted much attention from the public [1]–[7]. Relatively simple structures of quadrotors has promoted interest from academia, UAV industries, and radio-control (RC) hobbyists alike. Unlike conventional helicopters, swashplates, which are prone to failure without constant maintenance, are not required. Furthermore, the diameter of individual rotors can be reduced as a result of the presence of four actuators [8].

Many research groups or institutions have constructed their own quadrotors to suit specific purposes. Successes have been reported from academia, such as the X4-flyer [9], OS4 [10], STARMAC [11], and Pixhawk [12] to mention a few. To the commercial market, the Draganflyer X4, Asctec Hummingbird, Gaui Quad flyer, Parrot ARDrone, and DJI Wookong have been introduced. At the same time, a number of OSPs for quadrotors have emerged as shown in Figure 1, with contributions from RC hobbyists, universities [12], [13], and corporations.
Quadrotor OSPs use community-hosting sites (e.g., Google code and Github) to create code, blueprints, or schematics, which are freely available under open-source licenses such as the general public license (GPL) [14]. These tools help talented independent developers to join OSPs freely from all over the world. This setting allows very fast development processes because new features can be tested not only by the developer but also by other people in the community. Feedback is given in real time from various conditions and configurations, which make open-source software more robust in a relatively short period of time.

OSPs have been successful in many disciplines and remain competitive with commercial alternatives with Linux being the most famous operating system. In the robotics area, more than 2,000 projects have been established based on the robot operating system [15] with its well-organized framework that encourages open-source development.

In the case of quadrotor OSPs, one of the main reasons to use them is flexibility in both hardware and software, which makes modification easier to meet the specific requirements of a user. In addition, OSPs allow researchers to replicate and extend the results of others and provide a baseline for comparison among various approaches.

In this article, we introduce eight quadrotor OSPs and compare them in terms of hardware and software to provide a compact overview for use in a variety of areas as well as in academic research.

### Open-Source Projects for Quadrotor UAVs

In this section, we introduce the quadrotor OSPs that are listed in Table 1. These projects have been selected based on the user volume, activity, and project maintenance status. All of these projects are still in development. Therefore, readers should note that the information described in this article is as of May 2012. We use the term OSP to refer to code, electronics, or auxiliary software such as the ground-control software (GCS), depending on the context.

#### Arducopter

Arducopter is a quadrotor autopilot project based on the Arduino framework developed by individual engineers worldwide, which is described earlier [Figure 1(a)]. A graphical-user-interface (GUI)-based software GCS is provided to tune control gains and display flight information [Figure 2(a)]. This project shares the same avionics platform with Ardupilot, which is a fixed-wing aircraft autopilot OSP. A helicopter autopilot is also supported. There are more than 30 contributors on the project Web site and it uses the GNU Lesser GPL (LGPL) [16].

#### Openpilot

Openpilot is an OSP led by RC hobbyists [Figure 1(b)] using GPL [14]. This project features a real-time operating system modified from FreeRTOS, which is an open-source operating system. Openpilot supports fixed-wing aircraft and helicopters with the same autopilot avionics. A GUI-based GCS is provided to tune

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Web site URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arducopter</td>
<td><a href="http://code.google.com/p/arducopter">http://code.google.com/p/arducopter</a></td>
</tr>
<tr>
<td>Openpilot</td>
<td><a href="http://www.openpilot.org">http://www.openpilot.org</a></td>
</tr>
<tr>
<td>Paparazzi</td>
<td><a href="http://paparazzi.enac.fr">http://paparazzi.enac.fr</a></td>
</tr>
<tr>
<td>Pixhawk</td>
<td><a href="http://pixhawk.ethz.ch">http://pixhawk.ethz.ch</a></td>
</tr>
<tr>
<td>Mikrokopter</td>
<td><a href="http://www.mikrokopter.de">http://www.mikrokopter.de</a></td>
</tr>
<tr>
<td>KKmulticopter</td>
<td><a href="http://www.kkmulticopter.com">http://www.kkmulticopter.com</a></td>
</tr>
<tr>
<td>Multiwii</td>
<td><a href="http://www.multiwii.com">http://www.multiwii.com</a></td>
</tr>
<tr>
<td>Aeroquad</td>
<td><a href="http://www.aeroquad.com">http://www.aeroquad.com</a></td>
</tr>
</tbody>
</table>

*Figure 1.* Open-source quadrotor autopilots. (a) Arducopter, (b) Openpilot, (c) Paparazzi, (d) Pixhawk, (e) Mikrokopter (photo courtesy of Inkyu-Sa), (f) KKmulticopter, (g) Multiwii (photo courtesy of Alexandre Dubus), and (h) Aeroquad (photo courtesy of Ian Johnston).
gains and receive flight data [Figure 2(b)]. On the Web site of this project, various videos are available describing gain tuning, assembly processes, and flight principles to help users.

**Paparazzi**

Paparazzi is an autopilot system oriented toward inexpensive autonomous aircraft of all types [Figure 1(c)]. It has been in development since 2003 [13]. Originally a fixed-wing autopilot, it now supports quadrotor configurations by modifying the control mixing rule. Nine different autopilot hardware systems are developed under the lead of the Paparazzi team at ENAC University. Paparazzi provides GUI-based GCS with flight-path scripting that makes mission planning in outdoors convenient [Figure 2(c)]. This project uses GPL for hardware and software.

**Pixhawk**

Pixhawk [12] uses onboard computer-vision algorithms developed by ETHZ Computer Vision Group [Figure 1(e)]. Among the projects introduced here, only the Pixhawk project has computer vision equipment which has been used in several papers [17]–[19]. It also provides GUI-based GCS [Figure 2(d)] called Qgroundcontrol, which is a separate OSP in collaboration with the MAVlink protocol project. The Pixhawk project is available under the GPL.

**Mikrokopter**

Mikrokopter is a quadrotor autopilot system developed by a subsidiary of HiSystems GmbH in 2006 [Figure 1(e)]. GUI-based software for gain tuning and health monitoring is provided as shown in Figure 2(e). Mikrokopter is operated in well-organized Internet shops for their autopilot boards. In 2010, the University of Tasmania and the Australian Antarctic Division made use of Mikrokopter to monitor moss beds in Antarctica.
The source code is available for noncommercial purposes only.

**KKmulticopter**

KKmulticopter is contributed by 20 people around the world [Figure 1(f)]. This project has targeted hobbyists who want to capture aerial photographs using quadrotors. The autopilot hardware of this project is the most basic among the projects described in this article. It is equipped only with a triaxis gyroscope for inertial measurement and an 8-b microcontroller for control. No GCS is provided and gains are tuned by variable resistors on board.

**Multiwii**

Multiwii is a quadrotor autopilot system developed by RC hobbyists [Figure 1(g)]. This project uses an Arduino board as a main processor while the sensor system can vary. This project aims to make the fabrication of electronics easy. It uses gyroscopes and accelerometers of the commercial off-the-shelf Wii motion controller from Nintendo, which needs less soldering. GUI-based GCS is provided as shown in Figure 2(f). GPL is used for this project.

**Aeroquad**

Aeroquad is a quadrotor autopilot based on Arduino [Figure 1(h)]. Similar to Multiwii, it uses a standard Arduino board instead of making its own fully fledged single board. Aeroquad also provides GUI-based GCS software as shown in Figure 2(g). Arducopter was separated from this project in May 2010. GPL is used by Aeroquad.

**Arduino Platform**

Although the Arduino platform is not a quadrotor autopilot, we introduce it here because the Arducopter, Multiwii, and Aeroquad projects all use it. Arduino is the name of both the open-source single-board microcontroller circuit and the integrated development environment (IDE). Arduino has a well-organized device driver library for different sensors and actuators. It is frequently used for rapid prototyping because of the following advantages:

- **Simple setup**: Arduino’s IDE is easy to install, and firmware can be easily downloaded via USB or RS-232 without an expensive JTAG interface.
- **Rich device drivers**: There are more than 100 libraries related to hardware peripherals and signal analysis on the Arduino platform.
- **Operating systems supported**: The Arduino development IDE is ported to Mac OS X, Windows, and Linux.

### Components of Open-Source Projects for Quadrotor UAVs

Figure 3 shows overall configuration of a typical quadrotor UAV, which consists of flight avionics, sensor systems, radio transmitters, receivers, and communication systems.
Flight Avionics

Flight avionics for the various projects mentioned in the section “Open-Source Projects for Quadrotor UAVs” are shown in Figure 1. Most of the introduced projects provide electronic schematics for self-production. Typically, flight avionics consists of a processor, input/output (I/O) pins, and sensors. The I/O pins connect an off-the-shelf electronic speed controller (ESC) and RC receiver to the flight controller. The sensor suite consists of a gyroscope, accelerometer, barometer, magnetometer, and global positioning system (GPS).

Table 2 describes flight avionics composition. Most flight avionics are full-fledged with six degrees of freedom (6DoF) inertial measurement unit (IMU), magnetometer,
and barometer. However, KKmulticopter has only three gyroscopes, because it is devoted to manual flight. KKmulticopter implements a stability augmented system (SAS), which will be discussed in the section “Open-Source Projects Internals.” Most flight controllers implement proportional-integral-derivative (PID) control for stabilization of the quadrotor, although the structure of the PID controllers between the projects varies slightly. This will also be discussed in detail in the section “Open-Source Projects Internals.”

**Sensors**

Detailed specification of the sensors used in the OSPs is given in Tables 3–6. For the accelerometer, five different chips are used in the OSPs, which are shown in Table 3. For the gyroscope, there are seven different chips used as listed in Table 4. Magnetometers are used to correct attitude information and estimate drift of gyroscopes. There are three different magnetometers used in the OSPs as shown in Table 5. Three types of barometers used to measure altitude are shown in Table 6.

**Radio Transmitters and Receivers**

Recently, some groups have modified off-the-shelf RC transmitters to fit their requirement such as complex control mixing or curve shaping of a stick. As a result, custom firmwares for a few RC transmitters have been released as open source. In addition, open-source RC transmitters and receivers have been emerging [Figure 4(a) and (b)]. The OpenLRS project was initiated for open-source RC radio transmitter and receiver development. The OSRC project has developed not only a radio part but also controller hardware as shown in Figure 4(c). These projects are useful when a flight-avionics package needs to be more compact without additional hardware such as an RC receiver.

**Communication Systems**

XBee is a popular communication system because of its simple setup, low cost, and reasonable communication range when compared with its size. All the projects addressed here use XBee.

Arducopter and Pixhawk implement the MAVLink protocol for ground control. One advantage of the MAVLink protocol is that one can use Qgroundcontrol without a need to develop separate GCS.

**Open-Source Projects Internals**

**Attitude Estimation**

Because a sensor suite is typically composed of a three-axis gyroscope and a three-axis accelerometer, which provide linear accelerations and angular rates only, a proper attitude estimation algorithm should be employed.

**Extended Kalman Filter**

The Openpilot and Pixhawk projects have designed an attitude estimation algorithm based on the extended Kalman filter (EKF). Here, we provide only an overview of the EKF-based attitude estimation of the Openpilot project, and the complete EKF algorithm can be found in [20].

Let \( p \) and \( v \) be three-dimensional (3-D) position and velocity in earth-fixed frame, \( q \) the quaternion, and \( b \) the gyro bias. Let \( R(q) \) and \( \Omega(q) \) be rotation matrix that converts body-fixed frame to earth-fixed frame and quaternion rates matrix, respectively, as a function of the unit quaternion. Let \( a \) denotes linear acceleration in body-fixed frame and \( \omega \) the angular velocity in body-fixed frame. Then, the state equation in discrete time can be written as

\[
\begin{bmatrix}
    p_k \\ v_k \\ q_k \\ b_k
\end{bmatrix} = \begin{bmatrix}
    v_{k-1} \\ \frac{1}{2} \Omega(q_{k-1}) \cdot a_{k-1} \\ w_{b,k-1}
\end{bmatrix},
\]

In (1), the gyro bias \( b \) is modeled with noise \( w_b \). The system input \( u \) consists of measurements of angular velocity \( \omega_m \) and linear acceleration \( a_m \):

\[
\begin{bmatrix}
    \omega_{m,k} \\ a_{m,k}
\end{bmatrix} = \begin{bmatrix}
    \omega_k - w_{\omega,k} + b_k \\ a_k - w_{a,k} - R(q_k)^T [0 \ 0 \ g]^T
\end{bmatrix},
\]

where \( w_{\omega} \) and \( w_{a} \) represent noise and \( g \) is gravitational acceleration. Substitution of (2) into (1) yields the following nonlinear model:

\[
\begin{bmatrix}
    p_k \\ v_{k-1} \\ R(q_{k-1})(a_{m,k-1} + w_{a,k-1}) + [0 \ 0 \ g]^T \\ w_{b,k-1}
\end{bmatrix} = \begin{bmatrix}
    v_{k-1} \\ \frac{1}{2} \Omega(q_{k-1})(\omega_{m,k-1} + w_{\omega,k-1} - b_{k-1}) \\ w_{b,k-1}
\end{bmatrix},
\]

Figure 4. Open-source RC transmitters and receivers. (a) OpenLRS transmitter, (b) openLRS receiver, and (c) OSRC transmitter.
where \( w_k = [w_{ax,k}, w_{ay,k}, w_{az,k}]^T \) is process noise. The nonlinear measurement model is (we omit time index \( k \) for notational simplicity)

\[
z_k = h(x_k) + v_k = \begin{bmatrix} \frac{p}{v} \\ \frac{m_b}{h_b} \end{bmatrix} = \begin{bmatrix} \frac{p}{v} \\ R_T^T(q)m_x \end{bmatrix},
\]

(4)

where \( m_b \) is the measurement of the magnetic field of the earth \( m_x \) in body frame, \( h_b \) is the height measured by the barometric sensor reading \( P_z \), and \( v_k \) is the measurement noise.

The states are estimated by the standard EKF algorithm and measurements from accelerometers, gyroscopes, magnetometers, GPS, and barometer are fused to estimate the states.

**Linear Complementary Filter**

The Mikrokopter project implements the linear complementary filter (LCF) and is shown in Figure 5 on each axis of the accelerometer and gyroscope. It is designed to fuse multiple independent noisy measurements of the same signal that have complementary spectral characteristics. The details of complementary filters can be found in [21] and [22].

Let \( y_u \) be the rate measurement of the angle \( \theta \) and \( y_x \) the angle measured by accelerometer. The complementary filter to estimate the angle \( \theta \) is given by

\[
\dot{\theta} = y_u + k_p(y_x - \hat{\theta}),
\]

(5)

where \( \hat{\theta} \) denotes the estimate of \( \theta \) and \( k_p \) is a gain that determines crossover frequency. The complementary filter described in (5) assumes that there is no steady-state estimation error. However, in practice, the gyro bias varies over time. To compensate for this, an integrator [Figure 5(b)] is added to obtain the following:

\[
\dot{\hat{\theta}} = y_u - \hat{\theta} + k_p(y_x - \hat{\theta}),
\]

(6)

\[
\dot{\hat{\theta}} = -k_I(y_x - \hat{\theta}).
\]

(7)

**Nonlinear Complementary Filters on the SO(3) Group**

An LCF is extended to the nonlinear SO(3) group [22] (Figure 6). The final form of the filter with bias estimate is given by

\[
\dot{\hat{R}} = \hat{R}(\Omega_y + \hat{\theta} + \lambda) \times
\]

(8)

\[
\dot{\hat{\theta}} = -k_I \hat{\theta}, \quad \hat{\theta}(0) = \hat{\theta}_0
\]

(9)

\[
\lambda = \text{vex}(\pi_\theta(\hat{\theta})), \quad \hat{R} = \hat{R}^T R_y
\]

(10)

where \( \pi_\theta(\hat{R}) = 1/2(\hat{R} - \hat{R}^T) \) and \( R, \hat{R} \in \text{SO}(3) \) are attitude estimate and estimate error, respectively, and the vex operator is the inverse operation of a skew-symmetric matrix.

**Controllers**

It is well known that the open-loop rotational dynamics of a quadrotor are unstable as studied in [23]. The identified model reveals that poles are located in the right-half plane

---

**Figure 6. NCF on SO(3) group [22].**
of the real-imaginary axis and damping ratio is negative. Therefore, it needs to be stabilized by a feedback control algorithm, for example, SASs [24].

SAS, which makes the aircraft stable via the rate measurement in the feedback loop, is popular in aircraft control [25]. SAS is shown in Figure 8 with a dotted-line box. It consists of rate feedback with gain. If SAS is applied to a quadrotor, damping is increased. As a result, the quadrotor becomes controllable by a user.

The KKmulticopter project implements its SAS exactly as shown in Figure 9(f). Consequently, it has only three gyroscopes. Because the SAS only provides rate regulation, an autopilot is required to maintain the attitude of a quadrotor. We describe different autopilot structures. We begin with a proportional-derivative controller to illustrate how different architectures produce different characteristics.

The Pixhawk project implements a single-feedback loop as shown in Figure 10(a). In this case, the controller is

\[ G_c(s) = K_P + K_Ds. \]  

Then, the unity-feedback closed-loop transfer function is

\[ Y = \frac{(K_P + K_Ds)G_P}{1 + (K_P + K_Ds)G_P}. \]

Desirable closed-loop poles can be achieved by adjusting \( K_P \) and \( K_D \). In addition to the noise problem due to differentiation, there is now a
closed-loop zero at $s = -K_P/K_D$, and this zero can cause a large overshoot in case of a step disturbance unless the plant poles are heavily damped.

To resolve the above problem, consider the inner-loop configuration of Figure 10(b),

$$G_c = K_P, H_i = K_D s. \quad (13)$$

Then, the close-loop transfer function is

$$\frac{Y}{R} = \frac{K_P G_P}{1 + (K_P + K_D s) G_P}. \quad (14)$$

Therefore, with rate feedback, we can achieve the same closed-loop poles, but without an undesirable zero. Gyroscopes directly provide the rate information of the aircraft attitude, so this particular compensator structure is useful. All the other OSPs employ inner-loop configurations, while Pixhawk uses the single-loop configuration.

The Arducopter project implements a controller based on an inner-/outer-loop structure as shown in Figure 9(a). To mitigate steady-state error, an integrator is added in the forward path. The controller of the Openpilot project [Figure 9(b)] closely resembles Figure 8, which is proposed in [24]. The Paparazzi project also has the same structure, but the derivative term in the forward path is removed as shown in Figure 9(c). Therefore, Paparazzi controller is exactly the same as Figure 8. The Multiwii project has the same control configuration as Paparazzi. The Pixhawk project implemented a standard proportional-integral (PI) controller in the forward path shown in Figure 9(d).

As mentioned earlier, the KKmulticopter project only implements SAS that controls the body rate as shown in Figure 9(e). This controller is particularly useful for system identification experiments. This controller only depends on the gyroscope output: $u = -K_y$. In this case, we can perform closed-loop system identification to obtain an open-loop system model. Consider the following linear system:

$$\dot{x} = Ax + Bu, \quad y = Cx, \quad u = -K_y. \quad (15)$$

Then, the closed-loop system becomes $\dot{x} = A_{cl}x$, where $A_{cl} = A - BKC$. Therefore, once we perform closed-loop identification to obtain $A_{cl}$, the open-loop A matrix can also be obtained. This is only possible when there is no

Table 7. Summarized control structure and the number of gains to be tuned.

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Number of Gains on Each Axis</th>
<th>Controller Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arducopter</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Openpilot</td>
<td>4 + 2 (antiwindup)</td>
<td>PI + PI</td>
</tr>
<tr>
<td>Paparazzi</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Pixhawk</td>
<td>3 + 1 (antiwindup)</td>
<td>PID</td>
</tr>
<tr>
<td>Mikrokopter</td>
<td>2 + 1 (antiwindup)</td>
<td>PI</td>
</tr>
<tr>
<td>KKmulticopter</td>
<td>1</td>
<td>P</td>
</tr>
<tr>
<td>Multiwii</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Aeroquad</td>
<td>3 + 1 (antiwindup)</td>
<td>PID</td>
</tr>
</tbody>
</table>

Figure 10. Feedback control with two configurations. $G_p$ denotes the plant and $H_i$ is the inner-loop controller. (a) Feedback control with a single-loop configuration. (b) Feedback control with an inner-loop configuration.

Figure 11. Attitude tracking result of the quadrotor. Arducopter autopilot is tested with ground truth. The delay between reference and roll angle is due to communication delay.
integrand involved and K is fully known, which is not always the case for typical quadrotors in the market.

Controller Parameters
Control structure and the number of gains to be tuned in each project are shown in Table 7. KKmulticopter is the simplest one, which has only one gain for tuning. Among many controller configurations, PI+P is dominant. P is for the inner loop (rate feedback), and PI is for the forward attitude error compensation.

Controller Evaluation
We have constructed quadrotors using five different autopilots among mentioned OSPs: the Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Multiwii. Among these projects, Arducopter, Paparazzi, and Multiwii share the same controller composition as shown in Table 7. For qualitative evaluation, we mount markers on a quadrotor to acquire ground-truth data from the Vicon system. The desired angle is transmitted to the Arducopter-based quadrotor while quadrotor attitude from the Vicon and the transmitted commands are recorded simultaneously. The satisfactory attitude tracking result is shown in Figure 11. The delay is due to RC signal processing.

Selection Guidelines
We have analyzed eight OSPs with attitude estimation algorithm, control configuration, electronic components, and features. A comparison of features between OSPs is given in Table 8.

Availability of Flight Avionics
All the projects we described provide electronic schematic and bill of materials to reproduce their flight avionics. However, it takes high initial cost to manufacture electronics individually. Only five projects among them are available for purchase now: Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Aeroquad. It is recommended to start with these projects if a reader prefers to avoid electronics fabrication.
Attitude Estimation Algorithm Development
For attitude estimation tests, Arducopter and Paparazzi will be a good choice. The other projects are equipped with two or more gyro chips, which are hard to be calibrated for alignment. Only Arducopter and Paparazzi are equipped with 6-DoF IMU in a single chip: MPU 6000. The dynamic range is the best among the accelerometers and gyroscopes as described in Tables 3–6.

Minimalistic Configuration
As studied in [23], an open-loop model can be easily identified when control input is fully known and no integrators exist in a controller as mentioned in the section “Attitude Estimation Algorithm Development.” Because SAS is implemented in [23] to identify the open-loop dynamics, KKmulticopter is a good choice to this end. The system is simple to understand and modify because a source code for attitude control is less than 500 lines in C.

GPS-Based Navigation
For GPS-based outdoor missions (e.g., waypoint navigation and hovering): Arducopter, Openpilot, Paparazzi, or Mikrokopter will be a good choice. Only these projects support GPS-based navigation. Although Multiwii has GPS, it only supports a homing capability to move a quadrotor back to the initial position.

Vision-Based Navigation
Only the Pixhawk project supports vision-based navigation capability. It can synchronize an IMU and a camera in hardware level, which allows tight integration of IMU measurements into the computer vision pipeline.

Open-Source Projects in Research

Vision-Based Navigation
The Pixhawk UAV is designed to be a research platform for computer-vision-based autonomous flight [28]. The Pixhawk team has constructed a localization test setup using augmented reality ToolKit+ (ARToolKit+). They successfully performed waypoint navigation using a camera on the localization test bed as shown in Figure 12.

In [26], adaptive image-based visual serving (IBVS) was integrated with adaptive sliding mode control based on Arducopter. Figure 13 shows the experiment in process where the inset picture is the image obtained from the onboard camera. The fiducial marker and its tracking result are shown.

Real-time vision-based localization was performed on a quadrotor system based on Arducopter [Figure 14(a)]. This quadrotor is equipped with a frontal-view grayscale USB2.0 camera with 640 × 480 pixel resolution. Image data from the camera are transferred to a single-board computer and processed in a real time [Figure 14(b)] to obtain the vehicle location based on a map created in advance.

Indoor Flight
A Mikrokopter-based quadrotor flew autonomously using a laser range finder (LRF) [29]. Equipped with LRF, Gumstix, and external IMU, it successfully performed autonomous indoor navigation without external localization sensors. Indoor position control based on an onboard LRF was performed on the Mikrokopter-based quadrotor platform shown in Figure 15 [32]. An autoregressive moving average with exogenous terms model of the stabilized Mikrokopter was identified in [30]. Recently, the quadrotor platform with shared autonomy was investigated for infrastructure inspection [33].

Multiagent-related research can be easily performed on the indoor quadrotor flight system. Especially, as the communication topology between agents can be user-defined within the GCS, various settings and algorithms can be exploited. Figure 16 shows three quadrotors in flight where an auction algorithm is being tested for online task
assignment. As described in Figure 17, each quadrotor is equipped with an onboard controller to track input commands sent by the GCS that collects position and/or attitude data of the quadrotors from the Vicon motion capture system. Data from the onboard vision sensors are sent to the GCS using a dedicated communication link.

**Conclusions**

This article has presented eight quadrotor OSPs with descriptions of their avionics, sensor composition, analysis of attitude estimation and control algorithms, and comparison of additional features. Several research projects that use OSPs as a main flight controller are described. Among the eight OSPs summarized in this article, we have implemented five by utilizing the benefit of OSPs that allow to build own systems at a low cost with less effort. To bring out continued improvements based on communities’ work, objective evaluations of OSPs remain an important open problem.

The meaning of OSP had been more about software, but it is expanding to hardware and even products. There is already a project that has open hardware blueprints and a 3-D model of the quadrotor airframe that can be ordered from 3-D printing services. Sharing the same platform will become easier with such services. We expect that more OSPs for UAV will be initiated in the future.

![Figure 16. Three quadrotors in an experiment for an online task assignment algorithm.](image)

![Figure 15. Mikrokopter-based quadrotor equipped with Gumstix and LRF [32].](image)

![Figure 17. Overall hardware architecture of the indoor quadrotor flight system for Figure 16. Multiple layers indicate one for each quadrotor.](image)
Acknowledgments
This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (nos. 20120000921 and 2012014219), and by the New and Renewable Energy Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (no. 20104010100490).

References

Hyon Lim, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hyonlim@snu.ac.kr.

Jaemann Park, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: pjim7981@snu.ac.kr.

Daewon Lee, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: dwsh001@snu.ac.kr.

H.J. Kim, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hjinkim@snu.ac.kr.