Motivating Computing and Engineering Undergraduate Research via Educational and Scientific Drones

Case Study: Hexacopter Drone with Raspberry Pi Modules

Eric S. McDaniel[1],
David A. Juckem, Warren S. Vaz
Department of Computer Science[2]
Department of Engineering Technology
University of Wisconsin – Fox Valley
Menasha, Wisconsin, 54914
{mcdae6861[1], juckd6040}@uwc.edu,
warren.vaz@uwc.edu[2]

Saleh M. Alnaeli
Mathmatics, Statics and Computer
Science Department
University of Wisconsin – Stout
Menomonie, Wisconsin, 54751
alnaelis@uwstout.edu

[2]corresponding author

Abstract

Presented is a complete design guide of an educational scientific hexacopter, an unmanned aerial vehicle commonly known as a drone. It is used as a framework to increase students' motivation in computer science and engineering and aimed at promoting interdisciplinary research in colleges. The construction of a custom-built drone is described with all the technical challenges and solutions. The drone is designed to carry a cluster of single-board computers (Raspberry Pi) and programmable modules. Modules use programming languages that are typically taught in introductory computer science courses (e.g. C/C++, Python, and Java). Modules (e.g. sensors) are mounted on to the human-controlled hexacopter drone and designed to automate the collection of data, requiring a pilot only to fly the drone. The presented project has the potential to be adapted to offer students flexible, enjoyable, and creative methods to collect real data to be used for many research projects.

1 Introduction

As the demand for technically-centered jobs has dramatically increased (e.g. companies require millions of professionals by 2020 according IBM and other relevant studies), both communities from academia and research believe that it is very important that the number of students proficient in STEM fields needs to be increased to meet current and future job market demands. Promoting and motivating students to excel in fields such as engineering and computer science as early as possible plays a big role in meeting the community's expectation and future job market needs. In order to increase students' motivation in science and engineering, they need to be exposed and introduced to STEM research as early as possible in enjoyable and engaging ways. Additionally, students need to learn the importance of the material and curriculum that are taught in K-12 and college. These goals can be met by providing and supporting instructors to prepare them to teach STEM concepts in enjoyable and effective ways.

Among the affordable technologies and devices that have introduced to support educators to educate young students in engineering and computing, and to create easier access to STEM education is Raspberry Pi. The Raspberry Pi is a small, single-board computer supported by the Raspberry Pi Foundation that was developed in 2014 for young students to engage in the study of computer science. The Raspberry Pi, or Pi, is an all-in-one computer, capable of running an operating system and offers methods to interface with the device using built-in input and output ports [1]. Programs can be written or downloaded in popular programming languages taught in schools such as C/C++, Python, and Java. Students can use this to develop an understanding of the foundation of computer science, such as logic, syntax, problem solving. Modules can be purchased by third-party manufacturers to supplement and extend the capabilities of the Pi, such as sensors, microswitches, continuous servomotors, and others.

From a broader perspective, multicopter drones, a type of unmanned aerial vehicle, have grown in popularity and are found in educational, recreational, and commercial environments. Hobbyists can purchase prebuilt drones or custom drones for recreation. Private industry and government regulatory bodies have used them as well, such as contracting firms who use drones to capture images of areas in construction sites that are difficult to view [2]. Some cities use them to image the condition of bridges and evaluate their decay without the effort and labor of having a person safely evaluate areas that are difficult to reach. Drones, in addition, can be used in educational environments as well, where concepts such as aviation, aerodynamics, etc. can be applied and studied. Some students have used drones to supplement their knowledge of environmental science by using them to evaluate the effect droughts have on their local community [3].

In this work, the combination of a custom-built drone and a Raspberry Pi with modules is detailed, and consideration is given to the capabilities of students, reasonable budgets, and learning outcomes. The device is shown in Figure 1. With the drone and Pi combination, there are limitless options in which investigators can test their product and collect real-world results.

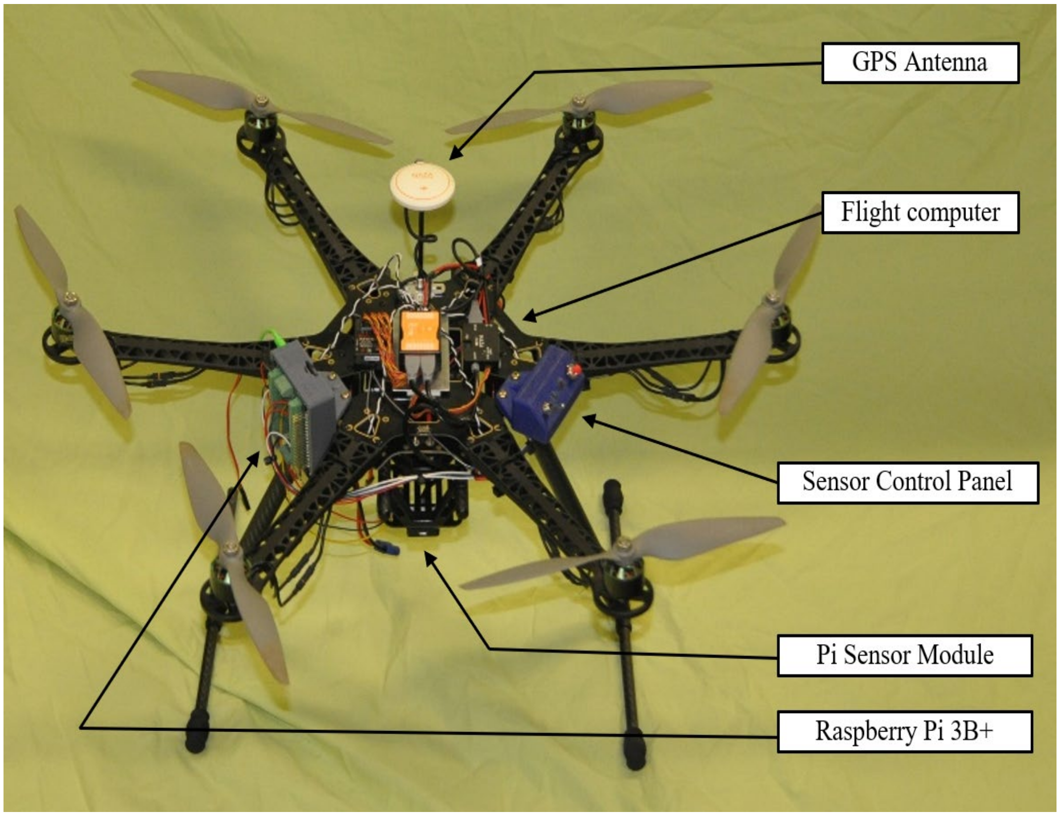

Figure 1: Hexacopter drone with Raspberry Pi module.

In this work, a drone module was designed (drone, mounted sensors, and Raspberry Pi's) and can be used to host some servers that can be accessible by the students in real time via web interfaces. This work can motivate college and well as high school students to study STEM in general, and computer science and engineering in particular. Additionally, the project can be easily be delivered to local schools in the form of educational workshops that help students appreciate the material they study in their programs that are relevant to the computing and engineering domains. The project was designed and implemented by students and faculty members from two different disciplines (computer science and mechanical engineering), which speaks volumes about the opportunities it provides to undergraduate students to collaborate and conduct interdisciplinary research at even the freshman and sophomore levels.

This particular drone's purpose is to collect data related to atmospheric pollution and emissions. The accompanying components, however, are modular and can be replaced with relevant modules for future studies. Federal and state environmental agencies delegate industries the responsibility to maintain a permissible level of pollution in the environment [4]. Drones offer scientists and engineers a flexible and creative solution to collect and analyze atmospheric pollution data because they can ascend to areas that are difficult or expensive to see and collect data from. Utilizing such drones can offer cities, manufacturing plants, and regulatory bodies a solution to monitor environmental compliance and affordably measure the concentration of selected gases within a predefined geospatial area.

The use of drones can advance understanding of emissions and carbon footprint by providing tangible evidence and showing where a concentration of pollution exists, such as at dense traffic intersections. Results can also be plotted using GIS software to visualize the data in a map.

Building the drone applies skills taught in mechanical engineering, electrical engineering, and teaches circuit design. Programming the data collection provides an avenue for computer science and programming concepts to be applied. Students can learn about aviation, atmospheric sciences, and geographical information systems (GIS). Students can also encounter and handle general problems solving, critical thinking, and analysis. The construction of the scientific drone was split into a series of independent, sequential stages. The first stage was to design, build, and test the drone and its stable flight. The second stage was the physical mount of the Raspberry Pi and accompanying sensors to the drone, accounting for the weight distribution and balance to maintain a stable flight. The final stage was the software implementation of the modules, allowing the Pi to collect relevant and purposeful data for analysis.

2 Drone Design, Assembly, and Performance

This section explains the design and building aspects of the scientific drone, shown in Figure 1, and some of the considerations that need to be taken into account when building it. The section gives a rough idea about some of the key components that need to be used and their estimated cost, all detailed in Table 1.

2.1 Initial Design, Considerations, and Component Selection

Accommodating the student's demographics, age, experience, and the nature of the project's interdisciplinary scope, designing a drone from ground zero was originally perceived as impractical. Focusing on the use of off-the-shelf components, especially with respect to the current maturity level of drones and their designs, was decided to be the most practical route. A hexacopter desgin was chosen over a quadcopter design as it provides an extra margin of safety for both the drone and any bystanders on the ground. Compromises include lower flight times and higher upfront cost, but this is believed to be a justifiable trade-off. Table 1 shows the frame, motors, and propellers along with the electronics chosen for the initial build.

Q	Item Description	Part Number	Supplier	Weight gm ea	Total Weight	Price ea	Total
3	Brushless Motor CW /	9536000006-0	Hobbyking	116	348	$40.50	$121.50

	CCW 1each						
3	CCW CW 9x4.5 prop set	9329000129	Hobbyking	18	54	$3.31	$9.93
6	ESC (electronic speed control)	586000005-0	Littlebee	11	66	$15.31	$91.86
1	Skyline 32 w/ baseflight and cleanflight	045000041-0	Hobbyking	5	5	$25.69	$25.69
1	Misc wiring and connectors		Hobbyshop	25	25	$30.00	$30.00
1	Misc Hardware and fittings		Hobbyshop	40	40	$25.00	$25.00
1	Main Frame	426000015-0	Hobbyking	445	445	$44.10	$44.10
1	Receiver	DSMX	Spectrum	3	3	$24.99	$24.99
1	Battery Charger	DYNC2015	Horizon Hobby	0	0	$69.90	$69.90
1	ESC Battery 11.1v	VNR15026	Horizon Hobby	435	435	$75.99	$75.99
1	PDB	090000026-0	Hobbyking	11	11	$5.90	$5.90
1	Transmitter	DX6	Horizon Hobby	*	*	$199.00	$199.00
6	3.5mm connectors	258000084	Hobbyking	20	120	$5.45	$32.70
1	Lipo Balancing Charger		Hobbytown			$149.99	$149.99
1	GPS Flight controller upgrade	DJI-Acc-Naza-M-V2	Helipal	40		$159.00	$159.00
2	4S Lipo Batteries		Hobbytown	528		$129.99	$259.98
Totals				1,697	1,552		$1,325.53

Table 1: Hexcopter atmospheric testing drone components and parts.

2.2 Construction and Providing Disciplinary Learning Outcomes

While straightforward, the assembly of the components offer sufficient challenges to capture the attention of students. In addition, students are provided exposure to the mechanical engineering discipline due to the opportunity to design custom components. Exposure to electrical engineering is limited with respect to the motor and flight control, however, there exists opportunity to work with electrical components in the design and application of the drone's sensors, modules, and its control. Control via the use of onboard computers can range anywhere from microcontrollers such as the BASIC Stamp or Arduino, to a complete Raspberry Pi single-board computer. Selecting and implementing any modules used for computation are limited by only budget and imagination.

The airframe is a screw-together erector set, however careful planning of how components will mount to the frame is needed in advance to prevent, or at least reduce, disassembly to gain access to areas for modularity. The flight control board and radio receiver should be mounted in the center of the airframe followed by the electronic speed controllers (ESCs) and motors out on the arms. Lastly, wiring all these components together and establishing communication between them completes the initial build of the drone.

3 Computational Hardware and Modular Peripherals

Automating the collection and processing of atmospheric data required the installation of a small, lightweight computer on the drone itself. Many options exist that can achieve this task, notably the Raspberry Pi and the Arduino Uno. The Raspberry Pi 3B+ single-board computer was chosen over other options because of its weight, flexibility, cost, and the accessibility of components. The Raspberry Pi 3B+ weighs 42 grams, making it an ideal choice considering the restrictions of the drone's payload [5]. With an upfront cost of $35 and a plethora of accessories available from third-party sensor kits, the Raspberry Pi, or simply Pi, offers investigators flexibility with respect to what their project goals include. In addition, the Raspberry Pi is a product of the Raspberry Pi Foundation, whose collective mission is to promote computer science education around the world, which stays consistent with mutualistic goals in mind [1].

Decisions were made to determine how much processing power was necessary to collect atmospheric data. However, this was found to limit the drone's payload weight and negatively impact performance. A compromise was made, which limited possible ideas such as utilizing a cluster computer or mounting a wireless receiver. Using a single Pi and the DHT11 Humidity and Temperature sensor from SunFounder Electronics [6], the Pi was fitted into a 3D-printed enclosure that was custom-designed on SolidWorks to reduce as much weight as possible. The Pi was then mounted in front of the drone's body, between the intersection of the two arms. A shield was fastened above the Pi to accommodate modularity if future decisions are made to replace sensors or add more variety. A 3D-printed control panel was designed to help interface with the Pi. A smaller, NiCad battery was fastened under the chassis of the drone's body to power the Pi. Power from the drone could have accommodated the Pi, however keeping these two systems independent of each other was found to be best for public safety and maximizing the flight performance of the drone.

4 Software Design and Implementation

Many operating systems have been designed to modified lightweight, ARM-architecture computers, such as Microsoft Windows IoT and modified distributions of Linux. The Raspberry Pi Foundation supports and maintains a distribution of Linux optimized for the Raspberry Pi called Raspbian, which is based on Debian Linux [7]. Raspbian was found to be the best fit because it simplified customization of the operating system such as booting with a graphical (GUI) desktop or a command-line shell. Raspbian also includes many libraries of code that can be imported for future Raspberry Pi projects, simplifying the process for students who intend to mount different sensors. In addition, Raspbian is UNIX-based and s a distribution of Linux, so many tools native to Linux were available for use, such as SSH, SCP, Git, and others.

Using the DHT11 sensor, SunFounder Electronics provides sample code for use with their modules [6]. This code, written in Python 3.4, was modified and adapted to the demands of this experiment, where the logic that facilitates the Pi's communication with the module was contained into one function utilizing BCM pin 17 of the Raspberry Pi's GPIO expansion board. As the main method iterates, a function call collects the data from the module once every second. Because the module was part of an inexpensive kit, the module would often provide a corrupt reading if a checksum invalidates a reading. The code was later revised to reiterate data collections over failed checksums as fast as possible to keep a stream of data collection as consistently spaced as possible. Separate function calls were added to facilitate the storage and presentation of the data. The script would create a basic text file with a title formatted based on the system's current date/time and append the reading at every iteration. Additional function calls were made to safely power down the Pi once a predetermined number of readings were collected.

The methods used to write and maintain the source code used on the Pi evolved as the project elapsed. Originally, the Python script was revised directly on to the SD card of the Pi that would be mounted on to the drone. Once the Pi had been properly fitted on to the drone, debugging code revisions became more involved and time-consuming due to the inaccessibility of the Pi's input/output ports. Solutions to work around this problem without frequent dismounts include code revisions on the same SD card applied to a separate Pi, or mounting the SD card as a physical storage volume on to a separate computer running Linux to amend files and extract data.

5 Flight Performance and Design Modification

This section discusses some of the expected challenges students and instructors could experience while building and testing the drone and the associated systems and some suggestions on how to resolve these technical challenges.

5.1 Resolving Drone Instability and Substituting Components

Issues developed following the installation of the original proposed drone components. The original flight control board had proven to be problematic going through the setup routine. These flight control boards are designed primarily for the small racing quadcopters and they often evolve quickly. The software needed to setup the boards, the firmware available for them, and the lack of reliable documentation inhibited a stable, reliable platform for use. The DJI NAZA V2 was chosen as an upgrade to a more mature and stable controller. It includes built-in GPS, return to home functionality, and scripted flight pattern capabilities. Although this controller is more expensive, it significantly improves the platform into a usable vehicle for research.

5.2 Initial Drone Flight Performance and Reconsiderations

The drone's initial flight performance, excluding the Raspberry Pi and its accompanying modules, proved sufficient to continue with minimal adjustment. However, adding the Pi, its components, and a separate power supply negatively impacted the drone's capabilities. Performance had degraded enough to warrant looking into a more powerful motor and propeller combination. No such action was taken, however considerations for alternative equipment will be addressed for future drone builds.

5.3 Failed Data Collection and Limited User Feedback

On November 21st, 2018, the drone and Pi combination saw its first outdoor flight in which real atmospheric data would be collected. The drone flight was safe, successful, and reached an altitude of approximately 200 feet for five minutes. The Pi's SD card was extracted for review and it was found that no measurements were taken of the flight. While difficult to assert exactly why the Pi failed to record data, the best prediction of what had occurred is likely a result of a permission error when the Python script is unable to create the directory in which to store the data. A directory of the same name cannot be created if one already exists, throwing an exception and halting the script. The code was later improved to elegantly handle this scenario by checking if such a directory exists before attempting to create one.

Following the failed first flight data collection, the event exposed a larger problem to be handled. Because the Pi is running headless, no feedback was provided from the Pi to indicate its status and what it is doing. Investigators would have to wait until the drone lands to find out if the Pi had ever booted, launched the script, and recorded and properly stored the data. A solution to provide visual feedback led to the addition of two LEDs and a toggle switch to the Pi. The toggle switch wired to BCM pin 23 would prevent the script from collecting any data until the switch was flipped to complete the circuit. This allows investigators to exercise control over when to start recording the collection of data. A red LED wired to BCM pin 27 would illuminate only after the Pi boots, the script launches, the data file is opened to write data, and is ready to begin recording the data after the toggle switch is flipped on. An accompanying green LED wired to BCM pin 22 would slowly oscillate between on and off for every new measurement collected, spanning one full second.

5.4 Improving Control via Remote Shell Access

As the drone and Pi combination grew with complexity, the ability to modify the source code became more difficult and time-consuming. The I/O pins were inaccessible to connect the Pi to a monitor. The LEDs, toggle switch, shield, and interchangeable modules made utilizing a second Raspberry Pi expensive and unnecessarily redundant.

Mounting the Pi’s SD card as a physical storage volume on a separate computer running Linux prevents debugging as the modules were fixed to the Pi. The solution to address this inefficiency was to utilize the Secure Shell (SSH) network protocol.

The use of the SSH protocol give investigators more control over the Pi without physically handling the Pi, other than the required steps of connecting the battery and flipping the toggle switch. The script on the drone could be modified through terminal-based text editors via SSH on a terminal session from another computer running Linux. Investigators can run the script from their terminal session, and the code was implemented such that while the data is collected, formatted data would print to the window as well. Investigators can run and terminate the program as much as necessary, and shutdown the Pi if desired, without physical interaction of the Pi. This improvement led to a fork in the source code, where one implementation would be used for autonomous collection, while another version would be more interactive and allow for more control. To facilitate a connection, the host must be connected on the same local area network (LAN) as the client machine. The Pi was configured to automatically connect to a specific network at boot, and its IP address was obtained to make the connection.

5.5 Improving Collaboration with Git/GitHub Version Control

Maintaining the source code on one local device prohibited collaboration among multiple developers. Modifying copies of the code is an inefficient practice and introduces risk of error and loss. Limited version control and lack of physical access prohibited collective improvements on the software. This was addressed by utilizing git for version control and hosting the repository on GitHub for remote collaboration [8].

With the repository centralized onto a remote location, the code no longer needed to be modified directly via a text editor over an SSH connection. Code could be modified on any computer regardless of its operating system. These changes would be staged, committed, and push to the remote repository. Multiple branches would be utilized to ensure that the master branch is functional and free from developmental bugs. When appropriate, the Pi would then pull those changes from GitHub via SSH. This methodology would significantly improve the collaboration and accountability of code maintenance. This allows investigators to shift their focus from mundane coding complications to the purpose of the drone’s data collection.

5.6 Improving the Extraction of Atmospheric Data

With the Pi accessible via SSH connections and its accompanying software hosted remotely for modification and centralization, extracting atmospheric data is the only reason that would demand the dismount of the SD card and physically handling the Pi. In a follow-up test flight, this was addressed temporarily by using GitHub to store the results. A separate local branch titled "collections" was created which contained the data. This branch was then push onto the remote repository, then downloaded onto a separate computer. The "collections" local and remote branch was deleted after extraction.

Creating the separate branch eventually led to chaos where the branches faced merging conflicts due to having different histories once its parent branch was modified. While this method sufficed at the time, it created unnecessary labor. Consideration was given to elegantly extract the data from the Pi, and this resulted in the use of the Secure Copy Protocol (SCP). Based on the SSH protocol, the local host can obtain the collected atmospheric data from the remote host by transferring the file via SCP. Once authenticated, the wildcard character (*) made the transfer of all readings done with one command, eliminating chaos, unnecessary git branches, and allowed for investigators to make progress with the evaluation of the data. Subsequent drone flights proved this approach to work.

6 Conclusion

The drone and Raspberry Pi combination successfully completed stable flights, collecting and extracting data using efficient methods. In this case the temperature was measured using a temperature sensor and found to be 24 ˚C. The data, regardless of its end use, can be used for analysis for future scientific studies, and can also be graphically represented to visualize the result. By constructing a similar drone with a scientific purpose in mind, students get exposure to a large range of sub-disciplines. This exposure allows them to apply these concepts in a method that can be exciting, engaging, and enjoyable, promoting their desire to continue their education in STEM fields.

7 Future Work

The design of the drone and Raspberry Pi combination includes the modularity of interchangeable parts. With the drone successfully built and the ability to change Pi modules when desired, future work involves expanding the collection of the atmospheric data at a deeper, more investigative level.

These studies can include studying atmospheric carbon dioxide, carbon monoxide, or nitrogen oxide pollution. Measurements will be collected from areas susceptible to pollution concentration, providing scientists and investigators evidence to help design better traffic solutions.

8 Acknowledgment

The authors are grateful to the University of Wisconsin-Stout and the Professional Development Committee of the University of Wisconsin-Fox Valley (a campus of the University of Wisconsin-Oshkosh) for their support for this work. The authors are also grateful to the vast communities of developers who continue to develop the open-source tools that were used.

References

[1] The Raspberry Pi Foundation. “A small and affordable computer that you can use to learn programming” (2012). https://www.raspberrypi.org/about/.

[2] Bogue, Robert. “What Are the Prospects for Robots in the Construction Industry?” *The Industrial Robot* 45.1 (2018): 1-6.

[3] Gillani, Bijan, and Gillani, Roya. “From Droughts to Drones: An After-School Club Uses Drones to Learn about Environmental Science.” *Science and Children* 53.2 (2015): 50-54.

[4] United States Environmental Protection Agency. Regulations (2019). https://www.epa.gov/laws-regulations/.

[5] Pololu Robotics and Electronics. “Raspberry Pi 3 Model B” (2019). https://www.pololu.com/product/2759/specs.

[6] SunFounder. “Company focused on STEM education” (2019). https://www.sunfounder.com/b-sensor-kit.html.

[7] Raspberry Pi Foundation. Raspbian (2019). https://www.raspbian.org/RaspbianAbout.

[8] Atmospheric-sensing repository. GitHub Repository (2019). https://github.com/McDanielES/atmosphere-sensing.