**ÇANKAYA UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**



Computer Engineering

Literature Review Report

**Mine Detection Drone**

Ahmet Utku Gökmen

201911402

**Table of Contents**

| **Title** | **Page Number** |
| --- | --- |
| Abstract | 3 |
| Introduction | 3 |
| ESP Modules for Drones | 4 |
| Drone Software Stacks | 5 |
| An Overview of the Software Components | 6 |
| Programming Languages | 7 |
| Commercial Products | 8 |
| Conclusion | 14 |
| References | 15 |

**Abstract**

The purpose of this literature review report is to explain the software engineering aspects of developing a mine detection drone. We aim to investigate existing projects and tech relevant to this drone, mainly from the software side. The perspective from the software engineering side in the development of this project is vital in the modus operandi of this machine to work as intended. This literature review delves into the software-related challenges and solutions in the context.

**Introduction**

Mines have been used from the past to present for a considerable time range. Mines are placed on the designated land and used for various purposes, such as slowing down the speed of military operations, causing material damage to enemy forces, and causing human casualties. As of present, there are no such detectors that give highly accurate enough results for detecting mines. Mine scanning is very slow and dangerous because it is carried out on land. Land mine scanning operations need to be accelerated and made more reliable. When we consider this problem with the technological knowledge of the 21st century, it can be clearly seen that minesweeping operations can also be carried out from the air. In this period where the rotary-wing unmanned aerial vehicle projects are developing more and more and the importance of the miscellaneous projects being developed for the defense industry is increasing, it is envisaged that mine detectors can be integrated into unmanned aerial vehicles. In this study, we will design a modular mine detector suitable for rotary-wing unmanned aerial vehicles. Carrying out minesweeping operations from the air will facilitate operations and make them more reliable in places where the military cannot reach the areas or in areas where land transportation is difficult. The aspects that distinguish the aerial minesweeping method from mine detectors integrated into unmanned land vehicles are the many beneficial aspects such as not having the possibility of stepping on mines, the situation where the land vehicles become unusable when a mine is stepped on and the material damage that will occur due to this situation, and the fact that the secrecy of military operations is not protected.

In this study, mechanical engineer candidates, software engineer candidates, and computer engineer candidates will work together to design a rotary-wing aircraft that can detect aerial mines at low cost with high quality for use in the military.

As a starting point, we conducted a literature search that revealed the current state of the academic literature on these aircrafts’ software. In doing so, we gained insight into which aspects or technologies have already been studied or implemented. Additionally, it allowed us to identify gaps in the existing literature.

These gaps will serve as starting points for potential future contributions.

**ESP Modules for Drones**

The ESP32 module is generally a more powerful choice than the ESP8266, with more processing power and additional features. For a mine sensing drone project, both are capable of carrying and operating various sensors, or other relevant parameters that change according to the payloads.

The ESP32 is open-source, supported by an active community, and programmable using popular development environments like Arduino IDE and PlatformIO.

**ESPcopter** is a small-sized drone that is programmable and wirelessly networkable. The drone is based on the ESP8266 module, which is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability. The ESPcopter can be controlled using smart devices such as smartphones, tablets, remote controls, joysticks, or hand motion sensors. It can also be programmed to fly autonomously. The drone has been designed to be hackable, affordable, and easily programmable. Users can deploy development modules to add custom features or program ESPcopter as they wish. STEM students can use ESPcopter to learn programming from beginner to advanced level. The ESPcopter team has developed a complete blockly-programming website for students who are new to coding. However, students of a more advanced level can program ESPcopter with Arduino and design their own smartphone application for it. Other programming languages for drones include C and C++45. By getting such valuable training in drone software, students can then continue by developing software in related areas such as aircraft technologies, aerospace, and the defense industry.

**Detailed comparison together with other models:**

* **ESP32-S2:** This model has a Tensilica Xtensa 32-bit LX7 processor (up to 240MHz), 320KB SRAM, 128KB ROM, and supports Wi-Fi 4. It lacks Bluetooth support.
* **ESP32-S3:** This model has a Tensilica Xtensa 32-bit LX7 dual-core processor (up to 240MHz), 512KB SRAM, 384KB ROM, and supports Wi-Fi 4 and Bluetooth 5.0.
* **ESP32-C2:** This model is built around a RISC-V 32-bit, single-core processor. It has 272 KB of SRAM (16 KB dedicated to cache) and 576 KB of ROM. It supports Wi-Fi 4 and Bluetooth 5 (LE). It is designed to target simple, high-volume, and low-data-rate IoT applications, such as smart plugs and smart light bulbs.
* **ESP32-C3:** This model has a RISC-V 32-bit processor (up to 160MHz), 400KB SRAM, 384KB ROM, and supports Wi-Fi 4 and Bluetooth 5.0.
* **ESP32-C6:** This model has a RISC-V 32-bit processor (up to 160MHz), 400KB SRAM, 384KB ROM, and supports Wi-Fi 6 and Bluetooth 5.0.
* **ESP H2:** This model has a single-core, 32-bit RISC-V microcontroller that can be clocked up to 96 MHz. It has 320 KB of SRAM with 16 KB of Cache, 128 KB of ROM, and supports IEEE 802.15.4 connectivity on top of Bluetooth LE.
* **ESP32:** This model has a Tensilica Xtensa dual-core LX6 processor (up to 240MHz), 520KB SRAM, and supports Wi-Fi and Bluetooth.
* **ESP8266:** This model has a Tensilica L106 single-core processor (up to 160MHz), and supports Wi-Fi.

**Comparison Review:**

* **ESP32-S2/S3:** These models have a powerful processor and support Wi-Fi, which could be useful for transmitting data back to the controller. However, they lack Bluetooth support.
* **ESP32-C2:** This model supports Wi-Fi 4 and Bluetooth 5 (LE). It is designed for simple, high-volume, low-data-rate IoT applications.
* **ESP32-C3/C6:** These models support both Wi-Fi and Bluetooth, which could provide flexibility in communication options. The ESP32-C6 also supports Wi-Fi 6 for potentially faster data transmission.
* **ESP32:** This model also supports both Wi-Fi and Bluetooth. It has a powerful dual-core processor which could be beneficial for processing sensor data.
* **ESP8266:** This model supports Wi-Fi and has a less powerful processor compared to the ESP32 models. It might be suitable if the project doesn’t require heavy data processing.
* **ESP H2:** This model supports Bluetooth LE and IEEE 802.15.4 connectivity, which might be useful for low-power, short-range communications.

For drone projects, the ESP32 and ESP32-S2/S3 models are often preferred due to their powerful processors and Wi-Fi capabilities, which are useful for transmitting data back to the controller. The ESP32 model, in particular, has been used in various drone projects and has strong community support.

In our project, we will use the ESP8266 for being more suitable at a low cost.

**Drone Software Stacks**

ArduPilot and Betaflight are popular open-source flight control software platforms that support a wide range of flight controllers. ArduPilot in particular is known for its flexibility and extensive sensor support.

**ArduPilot** is an open-source autopilot software suite primarily designed for controlling unmanned aerial vehicles (UAVs), but it can also be used for ground and water-based vehicles. The software provides a wide range of capabilities for autonomous and semi-autonomous operation, making it a popular choice for hobbyists, researchers, and professionals in the field of robotics and drone technology.

**Key Aspects of the ArduPilot Software:**

1. **Autopilot Features:** ArduPilot is capable of providing autonomous flight control, navigation, and mission planning for various types of vehicles. It supports features like GPS-based waypoint navigation, altitude control, automatic takeoff and landing, and return-to-home functions. (Which are especially important in this project.)
2. **Vehicle Compatibility:** ArduPilot supports a variety of vehicle types, including multirotors (drones), fixed-wing aircraft, helicopters, and ground-based rovers or boats. This versatility makes it suitable for a wide range of applications.
3. **Open Source:** ArduPilot is open-source, which means that its source code is freely available for modification and distribution. This openness has led to a large and active community of developers and users who contribute to its development and provide support.
4. **Customization:** Users can customize ArduPilot to meet their specific needs and integrate additional hardware and sensors. It can be tailored for various applications, from aerial photography and agricultural crop monitoring to search and rescue missions.
5. **Mission Planning:** ArduPilot allows users to plan complex missions by defining waypoints, geofences, and other parameters through user-friendly interfaces or scripting. The software can execute these missions autonomously, following the predefined instructions.
6. **Telemetry and Communication:** ArduPilot supports communication with ground control stations, enabling real-time monitoring of vehicle status and the ability to adjust mission parameters on-the-fly.
7. **Safety Features:** The software includes several safety mechanisms, such as failsafes and geofencing, to prevent accidents and protect vehicles from flying or driving into restricted areas.
8. **Simulation:** Users can test and simulate their vehicle configurations and missions in a software environment before deploying them in the real world, helping to reduce the risk of costly errors.
9. **Ecosystem:** ArduPilot is part of a broader ecosystem that includes compatible hardware and accessories, further expanding the possibilities for customization and integration.
10. **Community and Documentation:** ArduPilot has an active and supportive community of users and developers. There are extensive documentation, forums, and online resources available for users to learn and troubleshoot issues.

**An Overview of the Software Components**

Controlling a drone involves several software components. Here’s a high-level overview:

1. **Drone Control Apps**: These are mobile or desktop applications that provide a user interface to control the drone. These apps often provide live video feed from the drone, controls for maneuvering, and options for setting flight paths.
2. **Flight Management Software**: This software is responsible for managing the drone’s flight. It takes input from the control app and translates it into commands for the drone’s motors.
3. **Sensor Data Processing**: Drones have various sensors (like GPS, altimeter, gyroscope) that provide data about its environment. The software must process this data in real-time to ensure stable flight.
4. **Image/Video Processing**: If the drone has a camera, there will be software to process the images/videos it captures. Some drones also have software for object detection, tracking, and other AI tasks.
5. **Communication Software**: This software handles communication between the drone and the control app. This could be via Wi-Fi, radio signals, or cellular network.

**Programming Languages**

When working with ESP modules, languages like C/C++ are used for low-level firmware development, which is often necessary for drone control systems. Additionally, Python can also be used for high-level scripting, data processing, and communication with the ground station. This may also depend on the specific drone hardware and software stack chosen.

Arduino programming primarily uses a simplified version of C and C++, often referred to as “Arduino Language” or the “Arduino IDE” (Integrated Development Environment). Here’s why C and C++ are preferred for Arduino-based drone projects:

1. **Simplicity:** Arduino programming languages are designed to be relatively simple and user-friendly. They are accessible to beginners with little programming experience. This is especially important in educational and hobbyist contexts, making it easier for enthusiasts to get started with drone projects.
2. **Community and Resources:** Arduino has a large and active community, along with a vast repository of libraries and examples written in C and C++. This extensive support network simplifies development and troubleshooting for drone projects. Many resources, tutorials, and code snippets are readily available.
3. **Performance:** C and C++ provide fine-grained control over hardware, making them suitable for drone applications where precise timing and control are essential. This is important for tasks like flight stabilization, sensor integration, and motor control. (Which are especially important in this project.)
4. **Portability:** Arduino code written in C/C++ is highly portable. It can be easily adapted to other microcontroller platforms or embedded systems, allowing for flexibility in hardware choices if needed.
5. **Extensibility:** Arduino’s simplified C/C++ language allows you to incorporate low-level code and directly interact with the hardware, making it suitable for advanced users who need more control over their drone systems.
6. **Access to Libraries:** Arduino’s extensive library ecosystem provides pre-written code for common functions and sensors, simplifying the development process. This can significantly accelerate project development for drone applications.

While C and C++ are preferred for Arduino-based drone projects due to the reasons mentioned above, it’s worth noting that other programming languages like Python or JavaScript can be used for certain aspects of a drone project, especially in higher-level tasks or ground control station applications. However, for low-level flight control, hardware interaction, and sensor integration, C/C++ remains the go-to choice for Arduino-based drone development.

In our project, we will use C for the reasons specified above.

**Commercial Products**

We have found three projects similar in various aspects to our project as a result of our research, two in hardware and one in software.

**Drone for detecting and removing land mines [7]:**

This current invention centers around a specialized drone engineered to address the critical issue of detecting and safely removing landmines hidden beneath the surface. To elaborate further, it introduces a comprehensive drone system that functions as a two-fold solution: first, a detection drone is deployed to meticulously identify the location of the landmines, and second, a removal drone is activated to safely dispose of any identified landmines. This cohesive system seamlessly integrates the detection and removal processes, allowing the drone to efficiently perform these tasks, thereby mitigating the direct risks faced by humans when handling landmines.

What sets this innovation apart is its ability to enhance safety and efficiency by delegating these hazardous tasks to drones. By deploying these drones in high-risk regions like military borders, where the potential for conflict exists, it enables the proactive detection and elimination of landmines. This not only reduces the risks associated with landmine removal but also contributes to the overall safety and security of such areas. In essence, this innovation represents a significant advancement in addressing the ongoing threat of landmines and offers a more secure and efficient approach to landmine detection and removal in dangerous environments.

A black and white drawing of a drone

Description automatically generated

**Figure 1:** is a configuration example diagram of a detection drone according to the present invention.

A drawing of a drone

Description automatically generated

**Figure 2:** is a configuration example diagram of a removal drone according to the present invention.

A drawing of a ball

Description automatically generated

**Figure 3:** is a perspective view of the explosives in the present invention.

A drawing of a machine

Description automatically generated

**Figure 4:** is a block diagram of the explosives of the present invention.

**Method for detecting and removing land mines using drones [8]:**

The current invention is centered on a method designed for the identification and eradication of landmines, whether they are on the surface or buried beneath the ground. More specifically, it focuses on a method that employs drones to detect and remove landmines. In this approach, a detection drone responsible for pinpointing landmines and a removal drone tasked with eliminating the identified landmines function as a single aircraft. This innovative approach transfers tasks that are inherently perilous for humans, such as landmine detection and removal, to the capabilities of drones.

What distinguishes this method is its capacity to enhance safety and efficiency by delegating these hazardous responsibilities to drones. By deploying these drones to high-risk zones like military borders where the potential for conflict exists, they are empowered to detect and eliminate landmines. This not only minimizes the risks associated with landmine handling but also contributes to the overall safety and security of such areas. In essence, this method signifies a significant advancement in addressing the ongoing threat of landmines and offers a more secure and efficient approach to landmine detection and removal in dangerous environments.

A drawing of a drone

Description automatically generated

**Figure 1:** is a schematic diagram of configuration for detection and removal drones according to the present invention.

A drawing of a drone

Description automatically generated

**Figure 2:** is a schematic diagram of configuration for detection and removal drones according to the present invention.

A drawing of a drone

Description automatically generated

**Figure 3:** is a schematic view showing the structure of the removing drones according to the present invention.

A drawing of a rectangular object with numbers

Description automatically generated

**Figure 4:** is a view showing a configuration example of a storage box and a shock absorber according to the present invention.

A drawing of a drone

Description automatically generated

**Figure 5:** is a schematic diagram of an embodiment of the present invention.

A drawing of a drone

Description automatically generated

A drawing of a drone

Description automatically generated

**Figures 6 & 7:** shows examples of the present invention.

**Custom drone with Arduino Mega 2560 [9]:**

**Description:** Build a customized drone for specific applications, such as mine detection, aerial mapping, or environmental monitoring. The Arduino Mega 2560 will serve as the flight controller and central processing unit for integrating various sensors and achieving autonomous flight capabilities. Customize the drone’s hardware and software to meet your specific project requirements.

**Features:**

* **Hardware Customization:** You can integrate specialized sensors like cameras, LiDAR, or gas detectors, depending on your application.
* **Autonomous Flight:** Implement mission planning and autonomous flight capabilities for data collection or specific tasks.
* **Data Processing:** Process and analyze data collected during drone missions to derive actionable insights.
* **Versatility:** The Arduino Mega 2560 offers a large number of I/O pins for sensor and peripheral integration, making it suitable for a wide range of applications.

Summary - Why use Arduino Mega 2560:

1. **Versatility:** The Arduino Mega 2560 provides a high number of digital and analogue I/O pins, allowing you to integrate a wide range of sensors and peripherals for specialized drone applications.
2. **Customization:** Having the flexibility to tailor the drone to specific needs, whether it’s for mine detection, environmental monitoring, aerial mapping, or any other task.
3. **Open Source:** Arduino is an open-source platform, which means a wealth of online resources, libraries, and communities for support and development can be accessed.
4. **Learning Opportunities:** Working with the Arduino Mega 2560 on custom drone projects offers excellent opportunities for learning about drone technology, sensor integration, and autonomous control.
5. **Cost-Effective:** Using the Arduino Mega 2560 can be a cost-effective option for building a custom drone, especially if the board is already available or if it is preferred to use an affordable and widely available microcontroller.
6. **Scalability:** It can be given a start as a relatively simple project and progressively more features and sensors can be added.

**Note:** While the Arduino Nano option is suitable for smaller drone projects, it may have limitations in terms of processing power and I/O pins compared to larger Arduino boards like the Mega 2560. Therefore, it’s essential to choose projects that align with the Nano’s capabilities and limitations.

Detailed comparisons:

**Arduino Nano:**

* **Pros:**
  1. **Compact Size:** The Arduino Nano is small and lightweight, making it ideal for smaller-sized drones with limited space for components.
  2. **Affordable:** It’s a cost-effective option, which is great for hobbyist or budget-friendly projects.
  3. **Lightweight:** Nano drones are typically lightweight, which can be advantageous for indoor flight or applications where weight matters.
  4. **Aerial Imaging:** Suited for applications like aerial photography and surveillance, as it can easily accommodate a small camera.
  5. **Learning and Experimentation:** Ideal for beginners or those looking to learn about drone technology and programming.
  6. **Indoor and Outdoor Use:** Depending on the project design, Nano drones can be flown both indoors and outdoors, providing flexibility for various applications.
* **Cons:**
  1. **Limited I/O Pins:** The Nano has a limited number of I/O pins compared to larger Arduino boards, which may restrict the number of sensors and peripherals it can integrate.
  2. **Processing Power:** It has less processing power compared to the Mega 2560, which can limit the complexity of the drone’s operations and control.
  3. **Customization:** Limited room for adding multiple sensors and components, making it less suitable for highly customized and complex drone projects.

**Arduino Mega 2560:**

* **Pros:**
  1. **Versatile I/O Pins:** The Arduino Mega 2560 offers a large number of digital and analog I/O pins, enabling extensive sensor and peripheral integration for a wide range of drone applications.
  2. **Processing Power:** It has more processing power and memory, allowing for complex flight control algorithms and data processing.
  3. **Customization:** Suitable for highly customized drone projects, as it can accommodate a wide variety of sensors and peripherals.
  4. **Open Source Support:** Arduino’s open-source community and resources offer extensive support for project development and troubleshooting.
  5. **Learning Opportunities:** Working with the Mega 2560 provides opportunities for in-depth learning about drone technology, electronics, and programming.
* **Cons:**
  1. **Larger Size and Weight:** The Arduino Mega 2560 is larger and heavier than the Nano, potentially impacting drone agility and maneuverability.
  2. **Higher Cost:** It is generally more expensive than the Nano, which can be a consideration for budget-conscious projects.
  3. **Limited Suitability for Nano Drones:** Due to its larger size, it may not be practical for nano or micro drone designs where compactness is essential.

We will be using the Arduino Mega 2560 in our project.

**Conclusion**

In conclusion, the study aims to design a modular mine detector suitable for rotary-wing unmanned aerial vehicles, with the goal of accelerating and improving the reliability of minesweeping operations. This innovative approach of conducting minesweeping operations from the air presents several advantages over traditional land-based methods, such as minimizing the risk of land vehicles triggering mines, enhancing the secrecy of military operations, and reducing the potential for material damage caused by mine explosions.

The review of existing commercial products, such as specialized drones for detecting and removing landmines, highlights the growing interest in leveraging drone technology for mine detection and removal. These solutions offer significant advancements in safety, efficiency, and overall security, particularly in high-risk regions and military border areas.

Regarding programming languages and software stacks, C/C++ and Python are identified as essential for low-level firmware development, high-level scripting, data processing, and communication with ground stations. Additionally, open-source flight control software platforms like ArduPilot provide flexibility and extensive sensor support for drone control systems.

In summary, the study’s exploration of existing technologies, commercial products, and ESP modules, along with an understanding of relevant programming languages and software stacks, serves as a foundation for the development of a cost-effective, high-quality rotary-wing aircraft for mine detection in military applications. The integration of these technologies and the collaboration of mechanical engineers, software engineers, and computer engineers offer promising potential for addressing the challenges of aerial minesweeping in the 21st century.

**References**

1. “ESPcopter: A Programmable, Networkable Micro Drone Based on ESP8266,” Espressif Systems.
   * This source provides detailed information about the ESPcopter, a drone based on the ESP8266 module. It discusses the drone’s features, capabilities, and potential applications, particularly in the field of education.
2. “Comparison of ESP32 Models,” Gist.
   * This source offers a comprehensive comparison of various ESP32 models. It covers their technical specifications, including processor type, memory, and connectivity options, providing valuable insights for those looking to choose the right ESP module for their projects.
3. “Espressif Products Overview,” Espressif Systems.
   * This source provides an overview of the products offered by Espressif Systems, including various ESP modules. It serves as a useful reference for understanding the range of options available for IoT applications.
4. “ESP-Drone documentation - Espressif Systems”.
   * This source provides comprehensive documentation on the ESP-Drone, an ESP32/ESP32-S2/ESP32-S3 based flying development board provided by Espressif.
5. “ESP32 Drone | Hackaday.io”.
   * This project on Hackaday.io details the creation of a low-cost drone using the ESP32, providing insights into the challenges and solutions encountered during the build.
6. “ESP32 LoRa Controlled Drone Engine - Instructables”.
   * This tutorial on Instructables guides you through the process of controlling a drone engine using ESP32 and LoRa, offering practical knowledge on drone hardware and software.
7. “Drone for detecting and removing land mines”.
   * KR102299872B1
8. “Method for detecting and removing land mines using drones”.
   * WO2019050298A1
9. “ELEGOO Mega 2560 The Most Complete Starter Kit Tutorial”
   * Online tutorial provided by ELEGOO which makes similar guides on STEM subjects.