PROFESSIONAL PROJECT 1

PROJECT TITLE: DEVELOPMENT AND DEISGN OF A DRONE PAYLOAD DEPLOYER ATTACHMENT

By Malek Itani
Mechatronics Engineer

ABSTRACT

This project involved designing and fabricating a payload attachment system for a client's drone, enabling it to carry and deploy a small payload of approximately 100g. The attachment was created using 3D printing and an Arduino-based control system to allow remote operation. Testing confirmed the attachment's functionality and safety for payload deployment. This report details the design process, electronics integration, and testing outcomes.

Keywords: Drone Payload, 3D Printing, Arduino, Remote Deployment

1. INTRODUCTION

The primary objective of this project was to create an attachment for a drone to carry and remotely deploy a payload of up to 100g. The design considerations included weight minimization, durability, and secure attachment to the drone structure. Key components included a 3D-printed casing, an Arduino microcontroller for controlling the deployment mechanism, and an external remote-control system to operate the device.

FIGURE 1: DRONE PAYLOAD SYSTEM

2. MATERIALS AND METHODS

All the relevant materials (components, parts, devices) that have been used in the work must be clearly described from the technical point of view. The principle of operation of your circuit should be describe in the methods section. Use a schematic of your circuit if possible. Use a flowchart to describe your code. Subtitles should be used when necessary.

- **Fusion 360 CAD Software**

Fusion 360 is a 3D CAD software used for designing and simulating engineering components. It enables precise modeling of parts and assemblies; its user-friendly interface and versatile features make it ideal for creating realistic mechanical designs and 3d printing.

FIGURE 2: AUTODESK FUSION 360

- **3D Printer and PLA Filament**

A 3D printer with PLA filament was used to fabricate the payload attachment, ensuring a lightweight, durable structure for drone mounting. PLA was selected for its strength-to-weight ratio, ease of use, and suitability for drone applications, adding minimal weight and preserving battery efficiency.

FIGURE 3: 3D PRINTING

- **Arduino Nano:**

is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Arduino NANO board has 22 digital input/output pins and 8 analog input pins including 6 PWM output pins. Its input voltage is 7 to 12V and it operates at 5V. [3]

FIGURE 4: ARDUINO NANO

- **5V Servo:**

A type of motor commonly used in robotics, automation, and hobby projects. These servos are designed to rotate to a specific angle or position, and can be controlled using a pulse-width modulation (PWM) signal. The rotation of the servo is controlled by adjusting the duration of the PWM signal, which determines the position of the servo's output shaft.

FIGURE 5: MG90 SERVO

- **XL6009 DC-DC Boost Converter**

An XL6009 boost converter was used to regulate the power supply, ensuring stable voltage for the Arduino and servo motor. This converter was chosen for its efficiency in stepping up voltage, allowing consistent performance of the payload deployment system without overloading the drone’s power source.

FIGURE 6: XL6009 MODULE

- **Lipo batteries**

A 300mAh LiPo battery was selected to power the attachment independently, ensuring reliable operation of the Arduino and servo without drawing from the drone’s main battery. This lightweight battery provided sufficient capacity for multiple deployments while maintaining a compact design ideal for aerial applications.

FIGURE 7: 300MAH LIPO BATTERY

- **HC-12 Radio Transceiver**

The HC-12 is a versatile and widely used wireless transceiver module that operates on the 433 MHz frequency band. Known for its long-range communication capabilities, it can transmit data up to 1,000 meters in open spaces., making it suitable for a variety of applications such as remote-control systems, sensor networks, and IoT projects.

FIGURE 8: HC-12 TRANSCEIVER

- **TP 4056 Lipo Charger**

The TP4056 is a popular lithium-ion battery charger module known for its compact size and efficient charging capabilities. Designed to charge single-cell Li-ion batteries, it features a constant current/constant voltage (CC/CV) charging method, ensuring safe and optimal charging. It has built-in protection against overcharging, overheating, and short circuits.

FIGURE 9: TP 4056

- **LM2596 Buck Converter**

An LM2596 buck converter was used to step down the voltage from the LiPo battery, providing a stable, lower voltage suitable for the Arduino and other electronics. This converter was chosen for its reliability and efficiency in maintaining steady power, crucial for consistent control of the payload deployment mechanism.

FIGURE 10: LM2596

2.1 3D Modeling and Design

A lightweight attachment was designed to securely fit onto the drone, providing a compartment for housing electronics and a mechanism for payload release. The attachment consists of four main parts: the base housing that attaches to the drone, a cover that secures to the base with screws, and two legs that raise the drone off the ground, compensating for the absence of built-in legs on this model. To hold the drone to the chassis, the 2 pieces of the chassis will be mounted with screws and nuts. The nuts have slots built into the design with holes so that the nuts won’t vibrate over time and loosen. The housing includes dedicated mounts for the servo locking mechanism, circuit board, batteries, and support legs. The complete design is shown below.

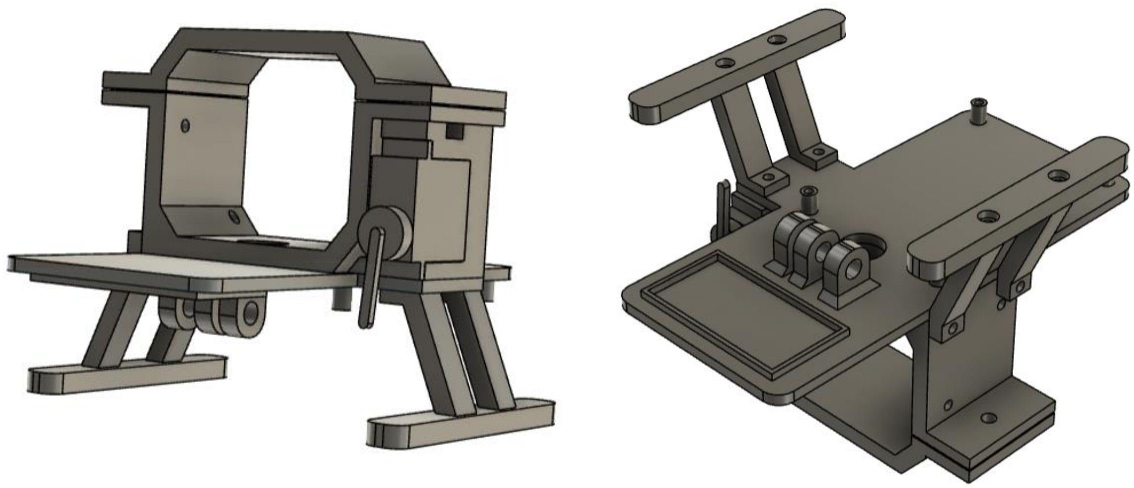

FIGURE 11: DRONE ATTACHMENT DEISGN

To release the payload, which will be held by a string, a slider crank mechanism will be used using a servo, a piece of steel wire, and a metal pin.

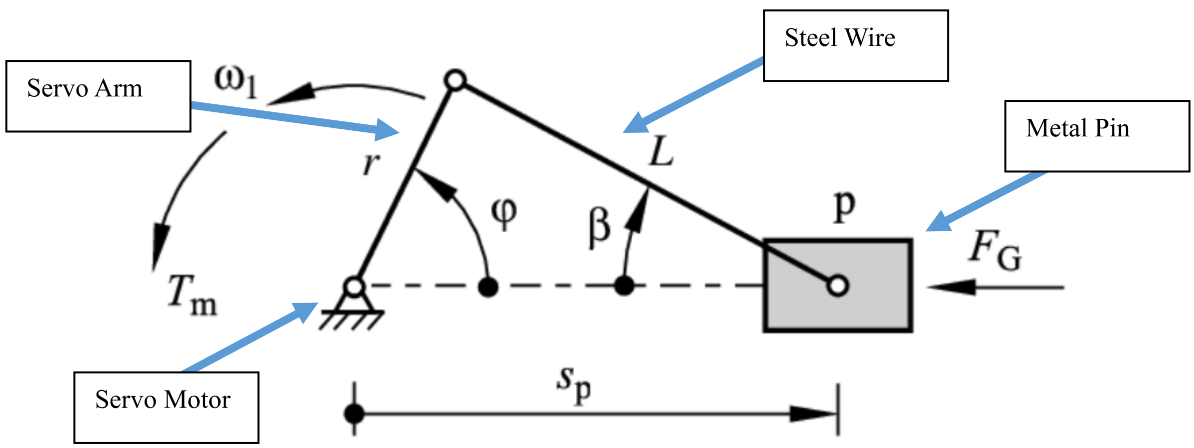

FIGURE 12: SLIDER CRANK MECHANISM

A custom remote control was also designed and 3D printed to manage payload deployment. The remote features a dedicated LED indicator window, providing visibility of the device's power status. A recessed opening was included for a push button, allowing easy control activation. The interior has mounting points to secure the circuit board with screws, ensuring stability during use, and a designated slot for the radio module's antenna to extend, optimizing signal transmission. The design details are illustrated below.

FIGURE 14: REMOTE CONTROL DESIGN

2.2 Electronic Circuit Design and Wiring

The main function of this system is to release a payload using a remote control with a range up to 1000m. the release mechanism is mechanically covered in the design section. To control the slider mechanism, a servo motor is used. The servo's position is fine-tuned to open and close the slider at the appropriate angles. The trigger that comes from the Arduino to control the servo to the open or close position comes from the HC-12 wireless receiver when a button is pressed from the remote control. To power the system and to keep everything light, small lipo batteries (300mah) batteries were used. 2 batteries were used in parallel to give a higher capacity giving a total capacity to 600mah. These batteries have a nominal voltage of 3.7 volts. To charge the batteries, a TP4056 charging module was wired in parallel to the batteries and fixed with Kapton tape. However, the Arduino and the servo operate normally at 5V (the Arduino can operate at 3.3v), so in order to use the batteries properly, a boost module (XL6009) is connected between the battery and the Arduino which will boost the voltage of the battery to 5 volts. An important note is that there seems to be an issue with the system functioning properly when using this battery setup. Normally, the output of the boost module is connected to the Vin of the Arduino. However, in this setup, the servo kept twitching and resetting. After troubleshooting intensely, this appeared to be a power problem. The way to solve this was to connect the output of the boost module into the 5v pin of the Arduino instead of the Vin. The Hc-12 needs a decoupling capacitor between the Vcc and ground any small value would work, a 47uF capacitor was used. As this module runs on serial communication, the Tx and Rx pins were connected to any 2 digital pins. Its wiring is shown in the figure below. Two of these transceivers will be used. The one on the drone will act as the receiver, and the other will act as the transmitter. More information is shown in reference [1]. The circuit Diagram is shown below:

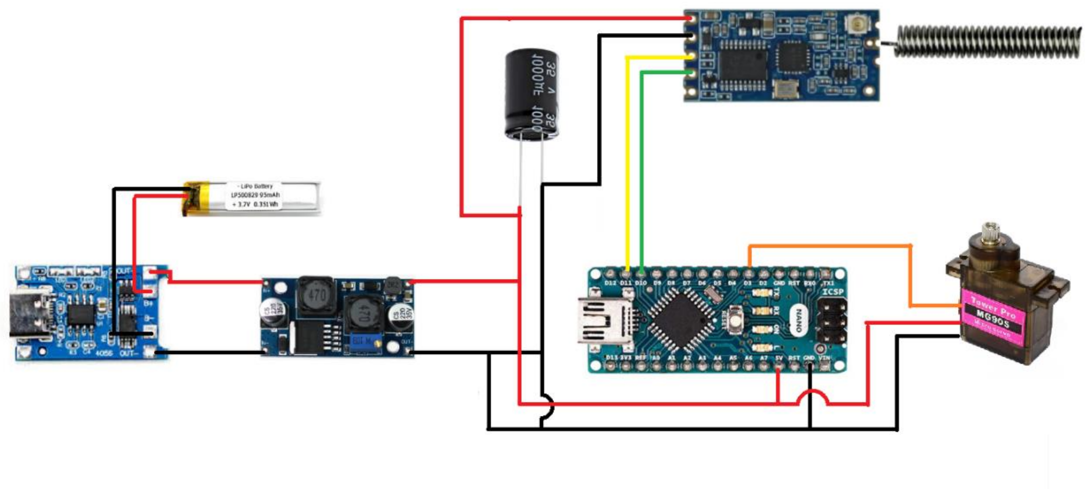

FIGURE 13: MAIN CIRCUIT WIRING

The remote control is also similar to the main circuit on the drone however it only uses the hc-12 module, a push button, and a Led. The pushbutton is connected between a digital pin on the Arduino, and a 10 kilo-ohm resistor to ground, normally a pullup or a pulldown resistor setup is preferred but this setup worked with no problem. The LED had a 220-ohm resistor to ground to extend its life and to limit the current flowing through it. Finally, the system is operated with a 9-volt battery so its replaceable, and compact. The battery voltage was converted to 5 volts with a LM2596 Buck Module. In this setup the output can be connected to the Vin pin of the Arduino with no problem.

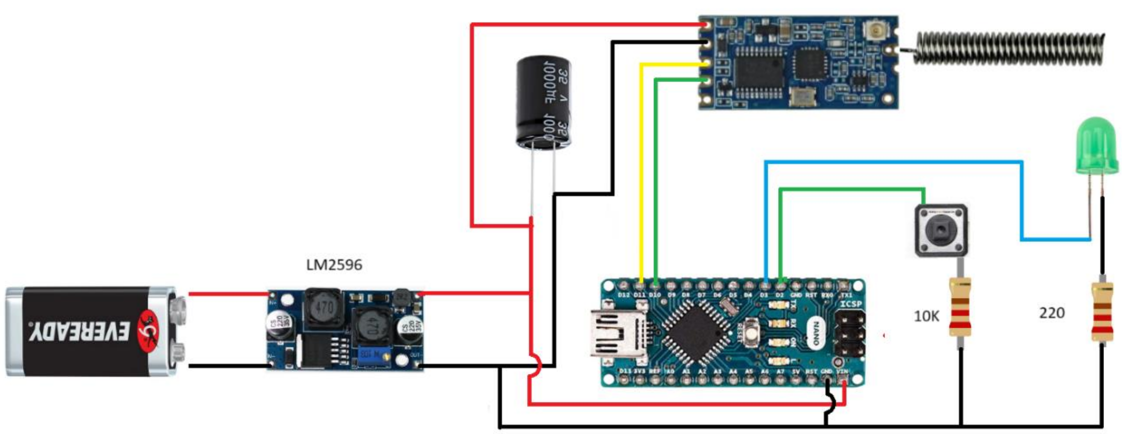

FIGURE 14: REMOTE CONTROL CIRCUIT WIRING

2.3 Coding

2.3.1 Receiver Code

This Arduino code controls a servo motor using an HC-12 wireless transceiver module for long-range communication. It employs the Servo and SoftwareSerial libraries to interface with the servo and communicate with the HC-12. The setup initializes the servo on pin 9, establishes serial communication with both the computer and the HC-12 at 9600 baud, and sets an initial state variable (state = 0). In the main loop, the code checks for incoming data from the HC-12 and alternates between two states based on this data to control the servo. In State 0, if the received data equals 1, the servo rotates to 45°, and the state switches to 1. In State 1, receiving a 1 signal again moves the servo to 105°, and the state resets to 0. This setup results in the servo toggling between 45° and 105° each time the HC-12 receives a signal of 1. Unused code sections hint at possible future extensions, such as resetting received data or sending data back to the HC-12 from the serial monitor.

2.3.2 Transmitter Code

This Arduino code enables wireless communication using an HC-12 module and includes a button press to trigger the transmission of a message. The program starts by including the SoftwareSerial library to facilitate communication with the HC-12 module on pins 10 and 11, with a message value of 1 stored in message. A button is connected to pin 3, configured as an input with an internal pull-up resistor, meaning the default button state is HIGH when not pressed. In the setup function, serial communication is initiated with both the computer and the HC-12 module at 9600 baud. In the main loop, the program reads the button state and prints it to the serial monitor for debugging. When the button is pressed (button state is LOW), the code sends message (which is 1) via the HC-12 module and sets pin 4 to HIGH, turning on an LED connected to that pin as an indicator. When the button is released (button state returns to HIGH), the LED is turned off. A delay (500) is included to add a brief pause, ensuring stable button detection and preventing multiple transmissions from a single press. This code allows simple wireless message transmission based on a button press, with an LED indicating the message status.

3. RESULTS AND DISCUSSION

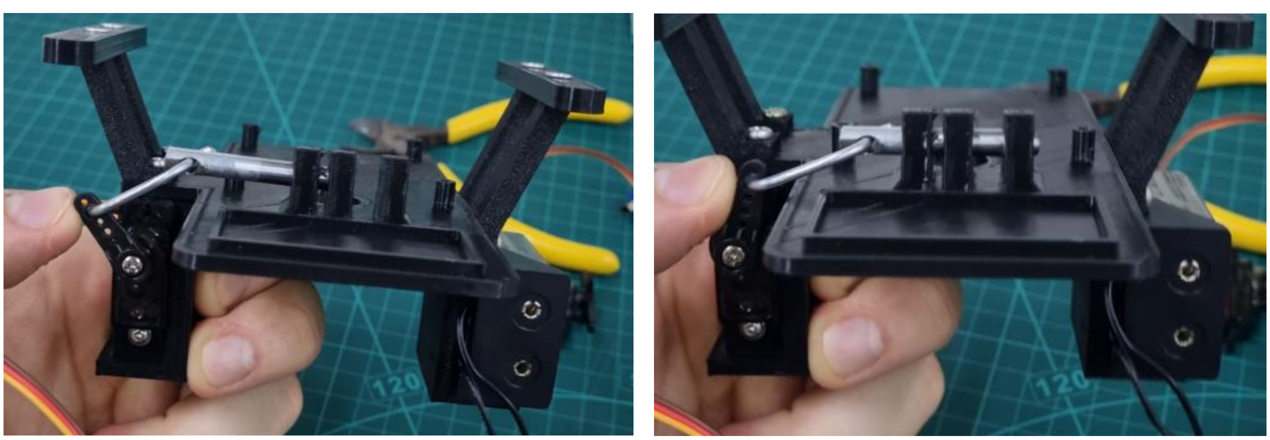

FIGURE 15: PAYLOAD LOCKING MECHANISM

FIGURE 16: FINAL PRODUCT.

Video of How it was Made: https://www.youtube.com/watch?v=iz78Hvmi7rU

4. CONCLUSION

In conclusion, the drone payload attachment project successfully demonstrates the integration of a versatile payload system designed for easy attachment and detachment, enhancing the functionality of drones for various applications. The system was carefully engineered to ensure stability and secure handling of the payload, while also maintaining the drone's flight performance. Through rigorous testing and adjustments, the project has proven the effectiveness of the attachment mechanism, allowing for quick adaptations to different payloads. This project highlights the potential of drones in fields such as delivery, surveillance, and environmental monitoring, offering a practical and scalable solution for payload integration. Further improvements can focus on optimizing the system for weight reduction, expanding payload capacities, and ensuring seamless compatibility with a wider

REFERENCES

https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/

Appendix:

Reciever Code :

```
#include <Servo.h>
#include <SoftwareSerial.h>


Servo myservo;
int state = 0;

SoftwareSerial HC12(10, 11);  // HC-12 TX Pin, HC-12 RX Pin
uint8_t text = 0;
void setup() {
  myservo.attach(9);
  delay(100);
  Serial.begin(9600);  // Serial port to computer
  HC12.begin(9600);    // Serial port to HC12
}

void loop() {
  if (HC12.available() && state == 0) {  // If HC-12 has data
    text = HC12.read();
    Serial.println(text);
    Serial.println("We are in state 0, HC12 avai");

    if (text == 1) {
      myservo.write(45);
      state = 1;
      Serial.println("State has switched to 1");
```

```
    }
  }

  if (HC12.available() && state == 1) {
    text = HC12.read();
    Serial.println(text);
    Serial.println("HC is availabe and state is 1");

    if (text == 1) {
      myservo.write(105);
      Serial.println("HC is availabe and state will switch to zero");
      state = 0;
    }
  }

  //else {
  //   text = 0;
  // }
  // while (Serial.available()) {  // If Serial monitor has data
  //   HC12.write(Serial.read());
  //   // Send that data to HC-12
  // }
}
```

Transmitter Code:

```
#include <SoftwareSerial.h>
uint8_t message = 1;
SoftwareSerial HC12(10, 11);  // HC-12 TX Pin, HC-12 RX Pin
int button = 3;
int buttonstate = 3;

void setup() {
  pinMode(button, INPUT_PULLUP);
  Serial.begin(9600);  // Serial port to computer
  HC12.begin(9600);    // Serial port to HC12
}

void loop() {
  buttonstate = digitalRead(button);
  Serial.println(buttonstate);
  if (buttonstate == LOW) {  // If HC-12 has data
    HC12.write(message);
    digitalWrite(4, HIGH);
  } else {
    digitalWrite(4, LOW);
  }
  delay(500);
```

```
}
```