# Model of the BOEING MCAS System

Michael Iheagwara, Len Johns Shaji,
Ethan Tran, Muskaan Trivedi, and Anh Nguyen

December 09, 2023

ECE-3155 Electronics Laboratory
Dr. John C Wolfe

# Abstract

The engineering project, centered around the Maneuvering Characteristics Augmentation System (MCAS), has undergone a transformative evolution. Initially conceived as a wind tunnel for comprehensive aircraft testing, the plan faced feasibility challenges. Additionally, efforts to construct a GY-521 chip PCB encountered functionality issues. A dynamic problem-solving approach was adopted, leading to an alternative strategy featuring a servo controller circuit with 555 timers. Two servo motors integrated on the sides contributed to dynamic balance, while Arduino technology enabled the simultaneous display of the plane's coordinates and the control of the rear tail wing servo motors. In the context of the MCAS system, these motors played a vital role in tail wing adjustments for stability during ascent and descent. This project not only addresses the initial challenges but also underscores the adaptability and resilience of the engineering design, particularly within the framework of the MCAS system.

# Introduction

The goal of this project is to design and prototype a replica of the Boeing 737's MCAS system. This aircraft stabilizing system could be used by a commercial airline and aircraft operators who plan on operating the Boeing 737 MAX aircraft. There are 62 operators that currently use at least one variant of the 737 MAX [2]. The MCAS unit must comply with the rules and regulations set by the Federal Aviation Administration or the European Union Aviation Safety Agency. It must be designed to work seamlessly and safely in the 737 MAX aircraft and not interfere with other systems of the aircraft. The system must also be able to accurately provide the corrective stabilizing measures when encountered with unsafe conditions. Additionally, the system must be user-friendly with clear displays and controls. In this model of the MCAS system, the clear displays and user-friendliness are incorporated through the LCD screens and LED lights which indicate the roll and pitch of the plane as well as the upward or downward direction.

# Theoretical Considerations and Design Approach

## Circuit Overview

The overall circuit consisted of three major modules: the arduino circuit, the GY-521 breakout board, and the servo motor controller. The GY-521 is the most essential part of the design. Part of our project involved designing a PCB based on the breakout board. This was a board that held a special chip on it, called the MPU-6050, and is commonly used as an Inertial Measurement Unit (IMU) [4]. The MPU6050 is a Micro-Electro-Mechanical Systems (MEMS) that essentially acts as both a 3 axis gyroscope and 3 axis accelerometer [4]. It is capable of measuring displacement, velocity, acceleration, and orientation [4]. In particular, we mainly used it as a gyroscope, where

it was able to provide us with the raw data that we needed to feed into the arduino. As a gyroscope the chip uses a phenomenon known as the Coriolis Effect to detect changes in the chip's angular velocity [4]. Upon detection, the chip is able to provide raw data that can be used to find the X, Y, and Z axis values.

The purpose of the servo motor controller is to control the angle of the GY521 board. In order to control the angle that a servo motor rotates at we need to feed the motor with a specific waveform signal, known as a Pulse Width Modulation, or PWM [5]. This signal acts similar to a quick switch, in which there is a high state and a low state. Based on how long the high state is relative to the low state, we are able to control the width of our pulse, as well as the angle that the servo motor was able to rotate to. In order to produce this wave, we used a specific configuration of the 555 timer circuit, known as the astable configuration [6]. This configuration produces a continuous pulse signal, as opposed to a single pulse like the monostable configuration, allowing the servo to update when the pulse width is changed.

Each time the servo motor rotated, the MPU6050 would calculate its relative position and then send that raw data to the Arduino microcontroller. The microcontroller would then send data to four components: an LCD screen that displayed the pitch and the roll of the plane, and LCD to display whether the MCAS was on or off, an array of LEDs, and servo motors that were used to control the back wing stabilizers The figure below provides a labeled image of the entire project design, where the individual components, and power supplies are labeled.

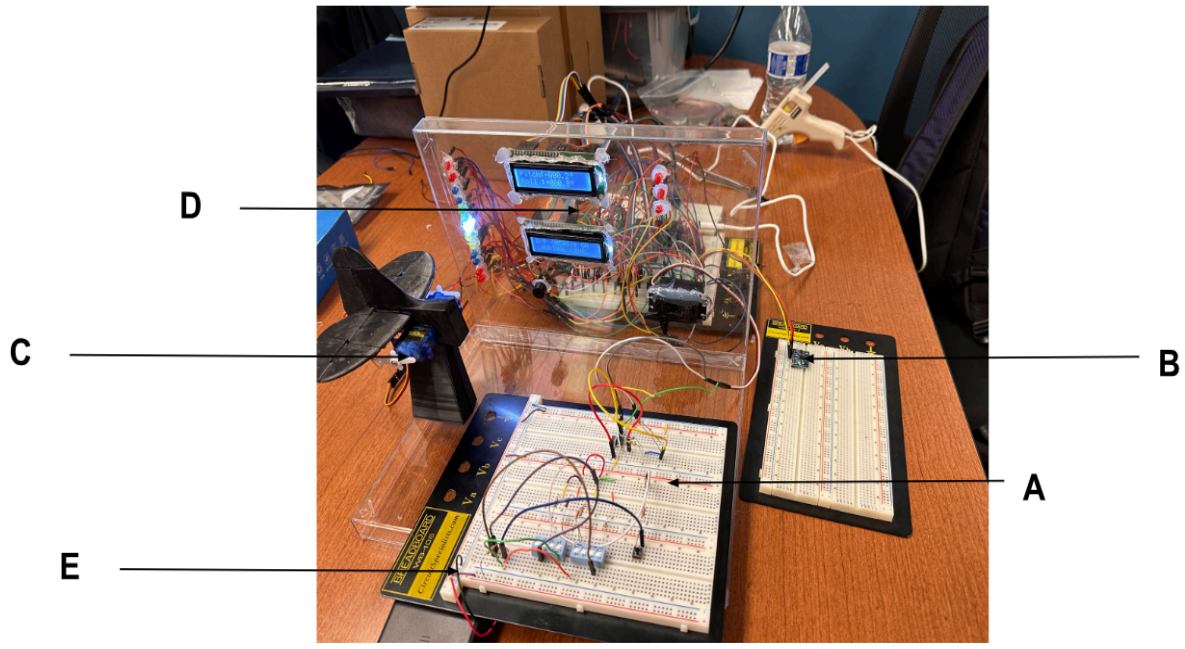


***Figure 1: Finished Project Design. A. Servo Controller; B. GY-521; C. Horizontal Stabilizer Servos; D. Arduino Circuit; E. Servo Controller Power Supply (6V)***

## Microcontroller Programming Overview

In the course of programming the microcontroller, a comprehensive system was devised involving the integration of two LCD displays, a passive buzzer, 13 LEDs, and 2 servo motors, alongside an accelerometer (MPU-6050). The initiation of the program involves the calibration of the accelerometer to establish the true level of the aircraft. Subsequent to calibration, the accelerometer transmits data to the Arduino, which processes the raw data into discernible values, facilitating the presentation of the aircraft's pitch and roll.

To enhance data accuracy, a specialized filter was implemented to mitigate the impact of engine vibration and wind interference on the accelerometer readings, specifically pertaining to pitch and roll. The pitch data serves as an input to regulate the servo motors, adjusting the horizontal stabilizers to achieve the desired degree and ensuring the seamless functioning of the MCAS (Maneuvering Characteristics Augmentation System). An indicator, represented by a white LED, signals the operational status of the MCAS system.

Moreover, the system incorporates 12 LEDs, with 6 indicating a pitch-up condition and another 6 denoting a pitch-down scenario, effectively conveying the Angle of Attack (AOA) of the aircraft. A predefined limit for the AOA, set at ±15°, serves as a critical threshold. If the AOA surpasses this limit, the MCAS system disengages, setting the horizontal stabilizer to 0° and activating a passive buzzer to alert the pilot to the system's deactivation.

The secondary LCD display conveys critical information, indicating the status of the MCAS system. In instances where the AOA falls below -15°, a "Pull UP" message is displayed; conversely, if the AOA exceeds 15°, a "Push DOWN" message is presented. Automated corrective measures are initiated if the AOA is adjusted beyond the preset limit, automatically reactivating the MCAS system and aligning the horizontal stabilizers to correct the aircraft's AOA accordingly.

## Task:

Our project consisted of four major steps: the circuit design and simulation; the programming of the microcontroller, the circuit build and testing; and the overall project build. We divided the work by having two people involved with the circuit design and simulation, one person involved with the microcontroller programming, and two people involved with the circuit and overall project build. We first started with the circuit design team, designing and simulating the servo controller circuit on Multisim and designing a PCB copy of the GY521 breakout board. Simultaneously, our programmer was programming the LCDs, LEDs, and servo motor to react based on the data collected from the GY521.  Finally our physical build group built the servo

controller circuit on a breadboard, and tested the circuit to compare its PWM output to the graphed voltage output of the circuit's simulation.

# Experimental Procedure

## Construction of the Circuit:

We utilized a 5 [V] voltage supply to power the circuit incorporating a 555 timer, which served to control a servo motor responsible for indicating the coordinates of the airplane.

The 555 timer IC comprises 8 pins with specific roles (Figure 1). Pin 1 (GND) is the ground reference, while Pin 2 (TRIG) triggers the timer. Pin 3 (OUT) produces the pulse waveform. Pin 4 (RESET) halts the timer. Pin 5 (CV) allows external voltage control. Pin 6 (THR) marks the end of the timing cycle. Pin 7 (DIS) discharges the capacitor, and Pin 8 (VCC) connects to the power supply (4.5 [V] to 15 [V]). Operating in astable and monostable modes, the 555 timer is versatile in the oscillator, pulse generation, and frequency division applications.

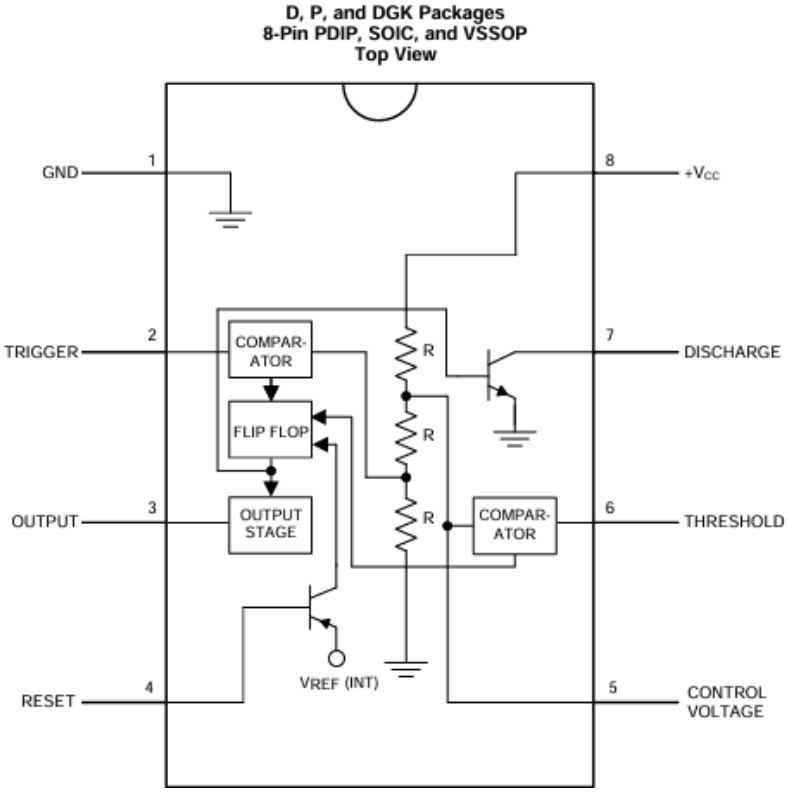


***Figure 2: Pin Configuration and Functions of 555 timer***

The figure below is the circuit schematic for the servo motor controller design utilizing the 555 timer. Each resistor value and the voltage supply for the circuit have been meticulously labeled. The output from the 555 timer will subsequently be linked to the servo motor to provide the clock signal. This configuration aims to precisely control the servo motor's movement, leveraging the timer's capability to generate an accurate and adjustable pulse. The labeled resistor values play a crucial role in determining the timing components of the circuit, contributing to the overall functionality and synchronization of the system.

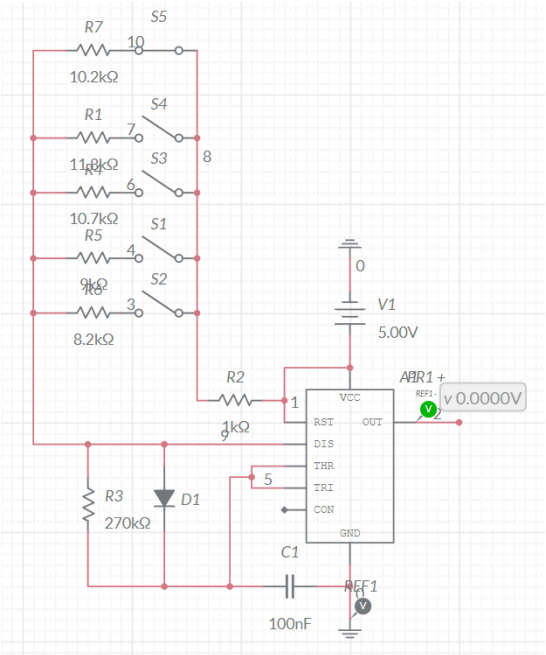


***Figure 3: Schematic for the circuit using 555 timer***

## Circuit/Module Testing:

Our comprehensive testing strategy covered both individual circuit modules and the overall system.

For the 555 timer circuit, getting the correct PWM wave was essential for making the servo controller work. Therefore we started by testing the Multisim circuit to get an initial idea of the resistor and capacitor values needed. The table below demonstrates us using an iterative design process to determine what resistor and capacitor values would give us a high state or ON state time frame of about 1.5 seconds and a total period of 20 ms (50 Hz) (Table 1). In the table the Ra represents the resistor that would be our reset resistor or the resistor that would set the servo to 0

degrees. Using proportions we were able to find the other values that were needed. This will be discussed later in the results.

| Ra (Reset resistor) | Rb (parallel to diode) | C | ON Time | Period (T) |
|---|---|---|---|---|
| 55 kΩ | 220 kΩ | 1 nF | 76.99 ms | 230.01 ms |
| 30 kΩ | 220 kΩ | 1 nF | 43.4 ms | 196.52 ms |
| 10 kΩ | 220 kΩ | 1 nF | 14.22 ms | 169.41 ms |
| 10 kΩ | 220 kΩ | 100 nF | 1.5883 ms | 16.94 ms |
| 10 kΩ | 290 kΩ | 100 nF | 1.5883 ms | 18.30 ms |
| 10 kΩ | 260 kΩ | 100 nF | 1.5883 ms | 19.739 ms |
| 12 kΩ | 270 kΩ | 100 nF | 1.5883 ms | 20.4 ms |

***Table 1: Iterative Process for Finding Components Needed to Produce a PWM Wave with a Period of 20 ms and ON Cycle of about 1.5 ms***

## PCB Testing:

When it came to constructing the PCB of the GY-521 breakout board, there were limited ways of testing the circuit. In order to get a PCB manufactured, using the JLCPCB it must pass a design test, where the program EasyEDA, checks to see if the PCB has all the correct connections. After the PCB was manufactured, we used a multimeter to check the overall resistance of the board and the voltage at each pin of the board. Finally, we connected the board to the Arduino and gave it a basic script to test and see if the PCB was providing raw data.

## Microcontroller Program Testing:

The Microcontroller system  upon testing performs the following functionalities:

1. **Initialization:**
    - Custom characters, including a degree symbol, are created for the LCD display.
    - Pins for components and servos are configured.
    - MPU-6050 registers are set up for calibration.

2. **Startup Display:**
    - The system displays team members' names on the LCD, scrolling horizontally.
    - LEDs and LCD indicators signal the initiation of the MCAS (Maneuvering Characteristics Augmentation System) and calibration process.
3. **Calibration:**
    - The system calibrates the MPU-6050 accelerometer and gyroscope to establish accurate reference values for pitch and roll.
4. **Real-time Monitoring:**
    - The loop continuously reads raw data from the MPU-6050, adjusts for calibration offsets, and computes pitch and roll angles.
    - A complementary filter is applied to dampen pitch and roll angles.
5. **LCD Display:**
    - Pitch and roll values are displayed on the LCD in real-time.
    - LEDs provide a visual representation of the pitch angle, with specific LEDs lighting up based on the angle range.
6. **Servo Motors Control:**
    - Servo motors are controlled based on the pitch angle to adjust the horizontal stabilizers, simulating the functionality of the MCAS System.
7. **Audio Feedback:**
    - A buzzer is activated to provide an audible alert when the MCAS system is turned off or when the pitch angle exceeds predefined limits.
8. **Limit Handling:**
    - If the pitch angle exceeds a specified limit, the MCAS system is deactivated, and the horizontal stabilizer is set to 0°.
    - The LCD displays corresponding messages, indicating either "Pull UP" or "Push DOWN" based on the angle of attack.
9. **Automatic MCAS Reactivation:**
    - If corrective action is taken within the defined angle of attack limits, the MCAS system is automatically reactivated.
10. **Servo Motor Control Based on Accelerometer Data:**
    - Servo motors are controlled based on accelerometer data to further adjust the plane's orientation.
11. **Error Handling:**
    - The system provides visual and audio cues in case of errors or extreme pitch angles.

The code integrates sensor data processing, control logic, and user feedback mechanisms to simulate the behavior of a Boeing MCAS system. The Program was coded in Arduino IDE using C# and an Arduino Mega was used as the microcontroller. There were no errors upon testing the program and the code worked perfect

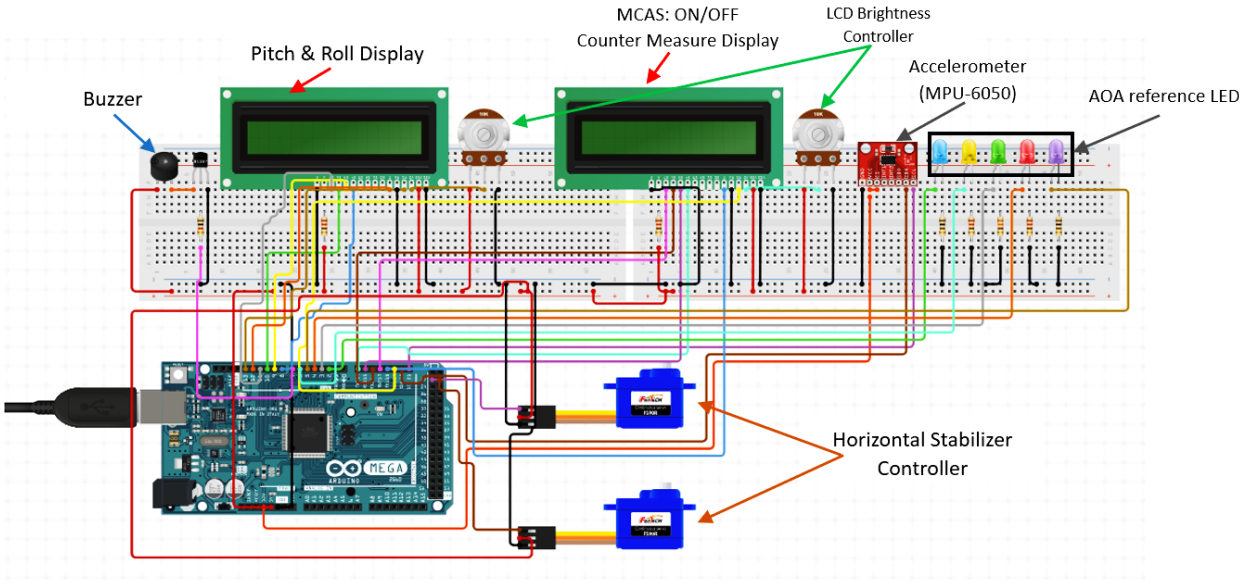


***Figure 4: Microcontroller Circuit Design***

# Results

## Data Collection

Through our calculations, we determined that the on-off cycle for the servo motor is 1.5ms and 18.5 ms, corresponding to a servo motor position of 0 degrees. Initially, we attempted to construct the circuit using a potentiometer. However, the servo motor exhibited limited functionality as the 10k potentiometer was beyond its operational range. As an alternative approach, we designed the circuit employing five distinct resistor values each in combination with a switch. To choose the proper resistor values we used the final values received from our iterative testing to select resistor values above and below the reset values. The following table presents the data collected, including the corresponding on cycle times for each resistor value.

| **ON Time (ms)** | **Resistances [kΩ]** |
|---|---|
| 2.2602 | 15 |
| 1.6583 | 10.7 |
| 1.5883 | 10.2 (Reset) |

| | |
|---|---|
| 1.4202 | 9 |
| 1.3077 | 8.2 |

***Table 2: Data collecting for servo motor controlling by 555 time***

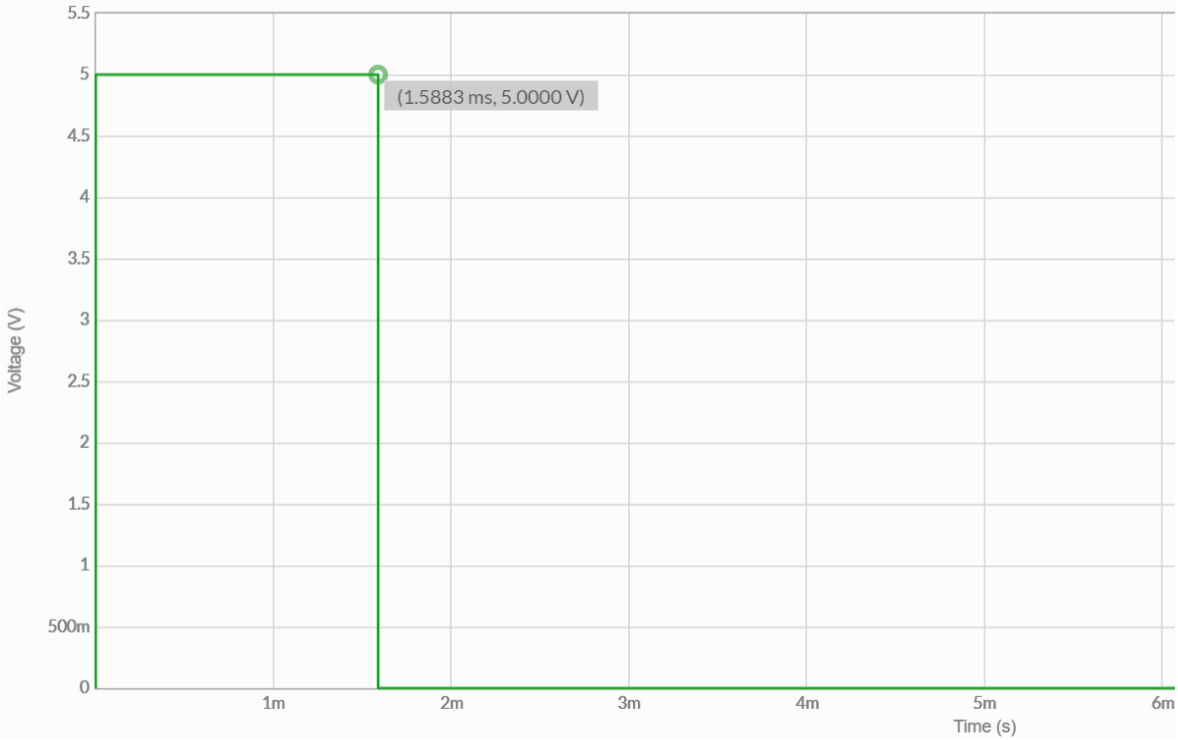


***Figure 5: Output signal for servo motor controlling by 555 timer***

## PCB Results:

Unfortunately, the PCB did not work, and we ended up using the real GY-521 board. While the board was producing the correct voltage values, it was not sending data to the arduino, and therefore can not be used to control the servo motor stabilizer.

# Discussion

The project initially set out with the ambitious aim of developing a MCAS, envisioning the fabrication of the entire 3D-printed airplane and the construction of a wind tunnel for comprehensive testing. However, a significant realization dawned on the team- the absence of

prior experience in mechanical engineering. Faced with this challenge and acknowledging the need for a more pragmatic approach, the team tactfully adjusted its course.

Complications arose during attempts to construct a printed circuit board PCB for the GY-521 chip, a crucial component of the MCAS system. One team member encountered difficulties in achieving voltage testing integrity, particularly with SDL and SCA connections. Despite earnest efforts to rectify the issues, a definitive solution remained elusive, prompting the team to reevaluate their strategy.

In a decisive shift, the team opted for an alternative strategy, steering away from the complex wind tunnel concept and embracing a servo controller circuit driven by 555 timers. This revised approach involved a more manageable 3D printing task - creating only the back of the tail wing - and integrating two servo motors on the sides for dynamic balance. The circuit process ensued, commencing with the use of a potentiometer to control the input into the 555 timer. However, this introduced a fresh challenge as the potentiometer provided resistor values outside the operational range, resulting in erroneous outputs from the 555 timers.

Undeterred by this setback, the team pivoted once more, exploring a different method. Five distinct resistor values and a switch were employed, and meticulous calculations were conducted to determine optimal on-and-off cycles for precise control of the servo motor. This strategic adjustment proved successful, culminating in the accurate timing of the 555 timers in the servo controller circuit.

The GY-521 chip, integral to the MCAS system, played a pivotal role by transmitting raw data to the Arduino, which was meticulously programmed to interpret and display the airplane's coordinates. Noteworthy ingenuity was displayed by a team member who incorporated an LED light, serving as an intuitive indicator of the airplane's directional movement - upward or downward. Despite the initial setbacks and shifts in strategy, the project successfully evolved into a functional MCAS system, demonstrating the team's ability to navigate challenges and deliver a robust engineering solution.

# Conclusion

In conclusion, our pursuit of developing a model of the Boeing MCAS system was marked by a series of formidable challenges requiring multiple adjustments and redesigns. In the face of setbacks, our team demonstrated commendable resilience and steadfast commitment to the project's objectives. Each difficulty encountered served as a catalyst for refinement, enabling the enhancement of our model through the assimilation of valuable insights into electrical engineering intricacies. Ultimately, through collaborative expertise and unwavering dedication, our team achieved the creation of a functional prototype.

# References


[1] Brady, Chris. “The Boeing 737 MAX MCAS System.” *The Boeing 737 Technical Site*, Boeing , www.b737.org.uk/mcas.htm. Accessed 9 Dec. 2023.

[2] Boon, Tom, et al. “1160 Active Aircraft: Which Airlines Fly the Most Boeing 737 MAX?” *Simple Flying*, 8 Dec. 2023, simpleflying.com/boeing-737-max-airlines/ Accessed 9 Dec. 2023.

[3] “Software Update.” *Boeing*, www.boeing.com/commercial/737max/737-max-software-updates.page. Accessed 09 Dec. 2023.

[4] “How Does the MPU6050 Accelerometer & Gyroscope Sensor Work and Interfacing It with Arduino.” *Arduino MPU6050 Tutorial - How MPU6050 Module Works and Interfacing It with Arduino*, circuitdigest.com/microcontroller-projects/interfacing-mpu6050-module-with-arduino. Accessed 9 Dec. 2023.

[5] “What Is a Servo Motor? - Understanding the Basics of Servo Motor Working.” *Servo Motor Basics, Working Principle & Theory*, circuitdigest.com/article/servo-motor-working-and-basics. Accessed 09 Dec. 2023.

[6] *LM555 Timer Datasheet (Rev. D) - Texas Instruments India*, www.ti.com/lit/ds/symlink/lm555.pdf. Accessed 09 Dec. 2023.

[7] Brokking, Joop. “MPU-6050 6DOF IMU Tutorial for Auto-Leveling Quadcopters with Arduino Source Code.” *YouTube*, YouTube, 21 May 2016, www.youtube.com/watch?v=4BoIE8YQwM8&ab_channel=JoopBrokking.

[8] *Datasheet - Stm32f103x8 Stm32f103xb - Stmicroelectronics*, www.st.com/resource/en/datasheet/stm32f103c8.pdf. Accessed 09 Dec. 2023.

[9] *Dynamic Wi-Fi Reconfigurable FPGA Based Platform for Intelligent ...*, www.researchgate.net/publication/221914353_Dynamic_Wi-Fi_Reconfigurable_FPGA_Based_Platform_for_Intelligent_Traffic_Systems. Accessed 09 Dec. 2023.

[10] Shaji, Len Johns. “Boeing MCAS System Code.” *Google Colab*, Google, 9 Dec. 2023, colab.research.google.com/drive/1n_nTLrFRWuAN0PyE-mJSvCwFsJPNbwVc?usp=sharing.

# Acknowledgement


- 3D Lab Print
- Joop Brokking
- Bernard Li
- IEEE - University of Houston
- Dr. John C Wolfe

Michael Iheagwara - Circuit Designer
Len Johns Shaji - Programmer and 3D Designer
Ethan Tran - Assembler
Muskaan Trivedi - Circuit Designer
Anh Nguyen - Circuit Tester


***Figure 6: from left to right - Muskaan Trivedi, Michael Iheagwara, Len Johns Shaji, Anh Nguyen, and Ethan Tran***