// Task 1 - Output Digital Signal

int outputPin = 9; // Set output pin

int state = 0; // Set initial state

// Task 2 - Measure Frequency of Square Wave Signal

int inputPin2 = 2; // Set input pin for Task 2

int frequency\_task\_2 = 0; // Set initial frequency for Task 2

// Task 3 - Measure Frequency of Second Square Wave Signal

int inputPin3 = 3; // Set input pin for Task 3

int frequency\_task\_3 = 0; // Set initial frequency for Task 3

// Task 4 - Read Analog Input and Compute Filtered Analog

#define ANALOG\_IN A0

#define LED\_PIN 13

float lastFourReadings[4];

int currentReadingIndex = 0;

float avgAnalogIn = 0.0;

float maxRange = 3.3;

void setup() {

pinMode(outputPin, OUTPUT); // Set output pin as output

pinMode(inputPin2, INPUT); // Set input pin as input for Task 2

pinMode(inputPin3, INPUT); // Set input pin as input for Task 3

pinMode(LED\_PIN, OUTPUT);

Serial.begin(9600); // Initialize serial communication

}

void loop() {

switch (state) {

case 0:

digitalWrite(outputPin, HIGH);

delayMicroseconds(200);

state = 1;

break;

case 1:

digitalWrite(outputPin, LOW);

delayMicroseconds(50);

state = 2;

break;

case 2:

digitalWrite(outputPin, HIGH);

delayMicroseconds(50);

state = 3;

break;

case 3:

digitalWrite(outputPin, LOW);

delayMicroseconds(50);

state = 4;

break;

case 4:

digitalWrite(outputPin, LOW);

delayMicroseconds(50);

state = 5;

break;

case 5:

digitalWrite(outputPin, HIGH);

delayMicroseconds(50);

state = 6;

break;

case 6:

digitalWrite(outputPin, HIGH);

delayMicroseconds(30);

state = 7;

break;

case 7:

digitalWrite(outputPin, LOW);

delayMicroseconds(50);

state = 8;

break;

case 8:

digitalWrite(outputPin, HIGH);

delayMicroseconds(50);

state = 0;

break;

default:

break;

}

delayMicroseconds(3700); // Wait for the remaining time in the 4ms period

}

// Task 2 - Measure Frequency of Square Wave Signal

int inputPin = 2; // Set input pin for Task 2

int frequency\_task\_2 = 0; // Set initial frequency for Task 2

void loop() {

int period = pulseIn(inputPin, HIGH); // Measure period of input signal

frequency\_task\_2 = (int)(1000000.0 / period); // Convert period to frequency in Hz

frequency\_task\_2 = constrain(frequency\_task\_2, 0, 99); // Scale frequency between 0 and 99

Serial.print(frequency\_task\_2);

Serial.print(",");

Serial.println(0); // Output frequency value and 0 (since it's Task 2)

delay(20); // Wait for 20ms before next measurement

}

// Task 3 - Measure Frequency of Second Square Wave Signal

int inputPin2 = 3; // Set input pin for Task 3

int frequency\_task\_3 = 0; // Set initial frequency for Task 3

void loop() {

int period = pulseIn(inputPin2, HIGH); // Measure period of input signal

frequency\_task\_3 = (int)(1000000.0 / period); // Convert period to frequency in Hz

frequency\_task\_3 = constrain(frequency\_task\_3, 0, 99); // Scale frequency between 0 and 99

Serial.print(0);

Serial.print(",");

Serial.println(frequency\_task\_3); // Output 0 and frequency value (since it's Task 3)

delay(8); // Wait for 8ms before next measurement

}

// Task 4 - Read Analog Input and Compute Filtered Analog Value

#define ANALOG\_IN\_PIN A0 // Set analog input pin

#define LED\_PIN 13 // Set LED pin for output

float lastFourReadings[4]; // Array to store last 4 readings

int currentReadingIndex = 0; // Index of the current reading

float avgAnalogIn = 0.0; // Initialize average analog input value

void setup() {

pinMode(LED\_PIN, OUTPUT); // Set LED pin as output

Serial.begin(9600); // Initialize serial communication

}

void loop() {

// Read the analog input

int analogIn = analogRead(ANALOG\_IN\_PIN);

float analogInVolts = (analogIn / 1023.0) \* 5.0; // Convert analog reading to volts

// Store the analog input value in the array

lastFourReadings[currentReadingIndex] = analogInVolts;

currentReadingIndex = (currentReadingIndex + 1) % 4;

// Calculate the average of the last four readings

avgAnalogIn = (lastFourReadings[0] + lastFourReadings[1] + lastFourReadings[2] + lastFourReadings[3]) / 4.0;

// Check if the average analog input value is above a certain threshold

if (avgAnalogIn > 2.0) {

digitalWrite(LED\_PIN, HIGH); // Turn on the LED

} else {

digitalWrite(LED\_PIN, LOW); // Turn off the LED

}

// Output the average analog input value and the Task 2 and Task 3 frequencies to the serial port

Serial.print(avgAnalogIn, 2); // Output the average analog input value with 2 decimal places

Serial.print(",");

Serial.print(frequency\_task\_2); // Output the Task 2 frequency

Serial.print(",");

Serial.print(frequency\_task\_3); // Output the Task 3 frequency

Serial.println();

// Delay for a short period of time before the next loop iteration

delay(100);

}