

```

// PARCIAL 7 - DISEÑO CON uP y uC. 2024-1.
// NOMBRE: Diego Andrés García Díaz.
// CÓDIGO: 2195533.
// -----

// FRECUENCIA MÁXIMA ALCANZADA: 250 [kHz] - 310 [kHz].

// -----

// Definición de pines de los sensores táctiles y LED
const int sensorMinPin = 13; // Sensor táctil 1 (Ciclo de trabajo del 20% y frecuencia de 500
Hz)
const int sensorMaxPin = 14; // Sensor táctil 2 (Ciclo de trabajo del 75% y frecuencia de
5000 Hz)
const int sensorDecPin = 33; // Sensor táctil 3 (Ciclo de trabajo del 100% y frecuencia de
10000 Hz)
const int sensorIncPin = 32; // Sensor táctil 4 (Ciclo de trabajo del 50% y frecuencia de
1000 Hz)
const int ledPin = 16; // Pin del LED

// Variables para la frecuencia y el ciclo de trabajo actual
int dutyCycle = 128; // Ciclo de trabajo inicial del 50% para una resolución de 8 bits
int fActual = 1000; // Frecuencia inicial en Hz

// Prototipos de las funciones de interrupción
void touchCallbackMin();
void touchCallbackMax();
void touchCallbackDec();
void touchCallbackInc();

void setup() {
    // Inicializar los pines del LED y la comunicación serie
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);

    // Configuración inicial del canal PWM
    analogWriteResolution(ledPin, 8); // Establecer la resolución de PWM a 8 bits
    analogWriteFrequency(ledPin, fActual); // Establecer la frecuencia inicial
    analogWrite(ledPin, dutyCycle); // Establecer el ciclo de trabajo inicial

    // Inicializar la configuración de los pines táctiles
    touchAttachInterrupt(sensorMinPin, touchCallbackMin, 40); // Configurar interrupción táctil
    touchAttachInterrupt(sensorMaxPin, touchCallbackMax, 40); // Configurar interrupción táctil
    touchAttachInterrupt(sensorDecPin, touchCallbackDec, 40); // Configurar interrupción táctil
    touchAttachInterrupt(sensorIncPin, touchCallbackInc, 40); // Configurar interrupción táctil
}

void loop() {
    // Mostrar la frecuencia y ciclo de trabajo actual en el monitor serie
    Serial.print("Frecuencia actual: ");

```

```

Serial.print(fActual);
Serial.print(" Hz, Ciclo de trabajo: ");
Serial.print(map(dutyCycle, 0, 255, 0, 100));
Serial.println("%");

// Mostrar la señal de onda cuadrada en el Plotter Serie
unsigned long period = 1000000 / fActual; // Periodo en microsegundos
unsigned long highTime = period * dutyCycle / 255; // Tiempo en alto en microsegundos
unsigned long lowTime = period - highTime; // Tiempo en bajo en microsegundos

Serial.println((millis() % (highTime) < (lowTime)) ? 1 : 0);

// for (unsigned long i = 0; i < period; i += 10) {
//   if (i < highTime) {
//     Serial.println(1);
//   } else {
//     Serial.println(0);
//   }
//   delayMicroseconds(10); // Esperar 10 microsegundos antes de la siguiente iteración
// }
delay(30); // Esperar 100 ms antes de la siguiente iteración
}

// Funciones de interrupción para los sensores táctiles
void touchCallbackMin() {
  dutyCycle = map(30, 0, 100, 0, 255); // Cambiar el ciclo de trabajo al 20%
  fActual = 20000; // Cambiar la frecuencia a 500 Hz
  analogWriteFrequency(ledPin, fActual); // Aplicar cambios de frecuencia
  analogWrite(ledPin, dutyCycle); // Aplicar cambios de ciclo de trabajo
}

void touchCallbackMax() {
  dutyCycle = map(30, 0, 100, 0, 255); // Cambiar el ciclo de trabajo al 75%
  fActual = 30000; // Cambiar la frecuencia a 5000 Hz
  analogWriteFrequency(ledPin, fActual); // Aplicar cambios de frecuencia
  analogWrite(ledPin, dutyCycle); // Aplicar cambios de ciclo de trabajo
}

void touchCallbackDec() {
  dutyCycle = map(70, 0, 100, 0, 255); // Cambiar el ciclo de trabajo al 100%
  fActual = 40000; // Cambiar la frecuencia a 10000 Hz
  analogWriteFrequency(ledPin, fActual); // Aplicar cambios de frecuencia
  analogWrite(ledPin, dutyCycle); // Aplicar cambios de ciclo de trabajo
}

void touchCallbackInc() {
  dutyCycle = map(70, 0, 100, 0, 255); // Cambiar el ciclo de trabajo al 50%
  fActual = 25000; // Cambiar la frecuencia a 1000 Hz
  analogWriteFrequency(ledPin, fActual); // Aplicar cambios de frecuencia
  analogWrite(ledPin, dutyCycle); // Aplicar cambios de ciclo de trabajo
}

```

}