

EXPLICACIÓN DE LA LÓGICA PROGRAMÁTICA DEL SENSOR MAX4466



Código

```
// Lectura del sensor de sonido MAX4466
unsigned long startMillis = millis(); // Inicio de la ventana de muestra
float peakToPeak = 0; // Amplitud pico-pico

unsigned int signalMax = 0; // Valor máximo de la señal
unsigned int signalMin = 4095; // Valor mínimo de la señal para ESP32 (12-bit
ADC)

// Recolectar datos durante la ventana de muestra
while (millis() - startMillis < sampleWindow) {
    sample = analogRead(sensorRuido); // Obtener lectura del micrófono

    if (sample > signalMax) {
```

```
    signalMax = sample; // Guardar el valor máximo
}
if (sample < signalMin) {
    signalMin = sample; // Guardar el valor mínimo
}
}

peakToPeak = signalMax - signalMin; // Amplitud pico-pico
int db = map(peakToPeak, 0, 4095, 40, 100); // Ajustar según el sensor
```

Explicación:

1. **unsigned long startMillis = millis();**

Esta línea almacena el tiempo actual en milisegundos usando la función `millis()`, que devuelve el tiempo transcurrido desde que el ESP32 comenzó a ejecutarse. Aquí, se usa para marcar el inicio de la ventana de muestra durante la cual se realizarán las lecturas del sensor de sonido.

2. **float peakToPeak = 0;**

Se declara una variable `peakToPeak` para almacenar la amplitud pico a pico del sonido, es decir, la diferencia entre el valor máximo y mínimo de la señal medida durante la ventana de muestra.

3. **unsigned int signalMax = 0;**

Aquí se declara la variable signalMax e inicialmente se establece en 0. Esta variable almacenará el valor máximo de la señal registrada por el sensor durante el periodo de muestreo.

4. **unsigned int signalMin = 4095;**

Similar a signalMax, esta variable signalMin se inicializa en 4095, el valor más alto posible que puede leer el ADC (Convertidor Analógico a Digital) de 12 bits de la ESP32. Este valor representa la señal mínima medida. Como se espera que el sensor de sonido tenga lecturas más bajas, este valor será actualizado con la señal mínima real registrada.

5. **while (millis() - startMillis < sampleWindow)**

Este bucle while recolecta datos de sonido durante un intervalo de tiempo especificado por sampleWindow. El bucle continúa ejecutándose mientras el tiempo transcurrido desde el inicio de la ventana (calculado como millis() - startMillis) sea menor que sampleWindow.

6. **sample = analogRead(sensorRuido);**

Dentro del bucle while, el valor de la señal del sensor de sonido es leído a través del pin analógico conectado al sensor. El método analogRead() devuelve un valor entre 0 y 4095 (para la ESP32, que tiene un ADC de 12 bits).

7. **if (sample > signalMax)**

Si el valor leído (sample) es mayor que el valor actualmente almacenado en signalMax, el nuevo valor se asigna a signalMax. Esto asegura que

signalMax siempre contenga el valor máximo medido durante el periodo de muestra.

8. **if (sample < signalMin)**

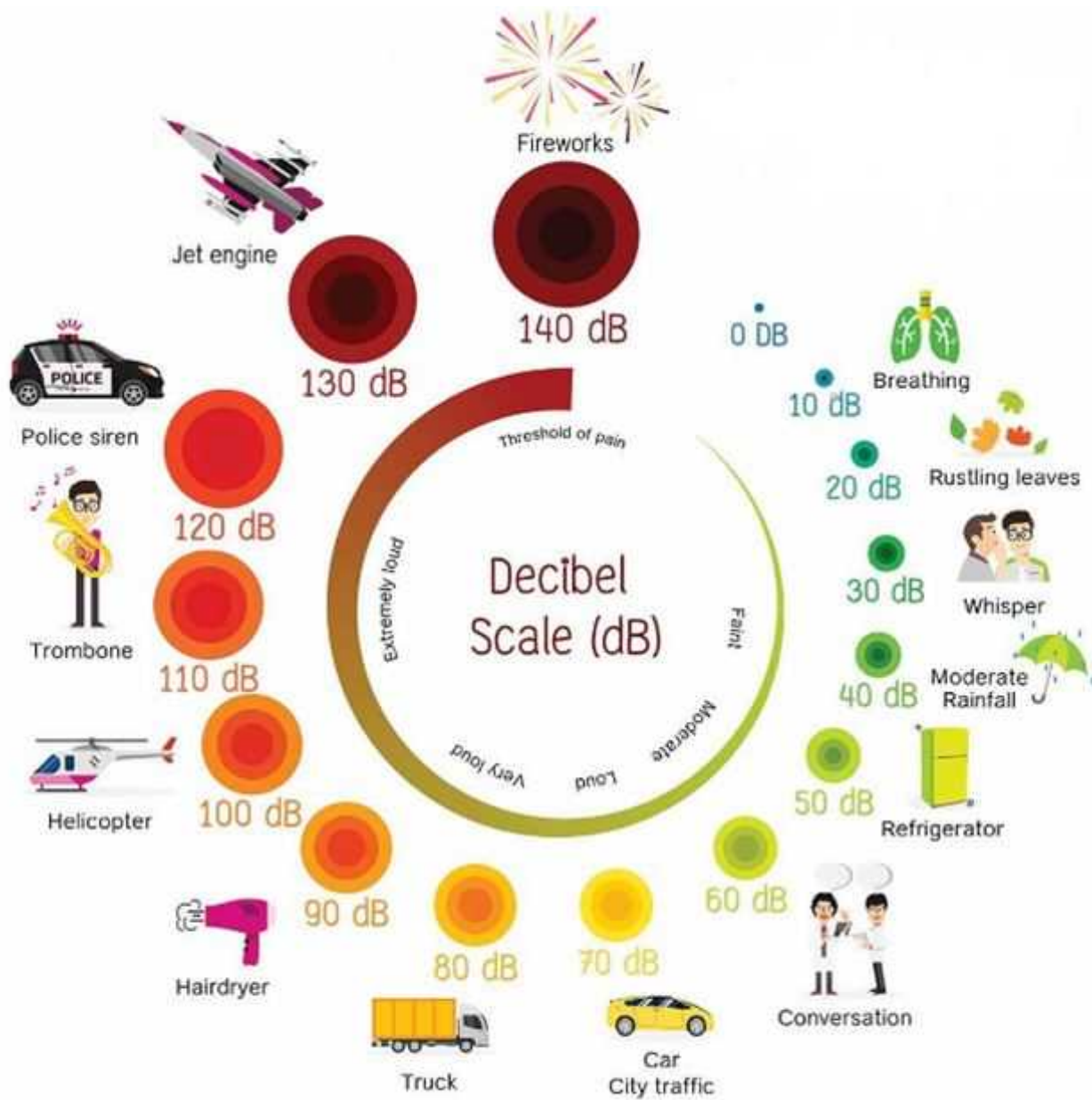
Si el valor leído (sample) es menor que signalMin, se actualiza signalMin con el valor más pequeño registrado hasta el momento. De esta forma, signalMin contendrá el valor mínimo medido durante la ventana de muestra.

9. **peakToPeak = signalMax - signalMin;**

Una vez que el bucle while ha terminado (es decir, cuando se ha alcanzado el final de la ventana de muestra), se calcula la amplitud pico a pico de la señal de sonido. Esto se hace restando signalMin de signalMax. El resultado, almacenado en peakToPeak, representa la diferencia entre los puntos más alto y más bajo de la señal, lo cual es un indicador de la intensidad del sonido.

10. **int db = map(peakToPeak, 0, 4095, 40, 100);**

Esta línea utiliza la función map() para convertir el valor de peakToPeak en una estimación aproximada de los decibelios. Aquí, se está asignando el rango de valores que el ADC puede leer (0 a 4095) al rango de decibelios esperado (40 a 100). Este rango se estima en base a un valor mínimo de 40dB (ruido nocturno en horas de la madrugada) y a un valor máximo de 100dB (ruido diurno en horas pico durante el día). El valor resultante en db representa el nivel de sonido en decibelios aproximado basado en la señal pico a pico medida.



Resumen del Proceso:

1. **Inicialización:** Se definen las variables para almacenar los valores máximos y mínimos de la señal, así como la amplitud pico a pico.
2. **Muestreo:** Durante un periodo definido por `sampleWindow`, el sensor de sonido es muestreado repetidamente, y se actualizan las variables para almacenar los valores máximos y mínimos medidos.
3. **Cálculo del pico a pico:** Después de la ventana de muestreo, se calcula la diferencia entre el valor máximo y el valor mínimo (amplitud pico a pico).
4. **Conversión a decibelios:** El valor pico a pico es convertido en una estimación de nivel de decibelios utilizando la función `map()`.

Este código mide la intensidad del sonido a través de las variaciones en el voltaje de salida del sensor de sonido MAX4466, y luego lo traduce a un nivel de decibelios aproximado que se puede mostrar o utilizar para otras funciones, como activar el LED si los niveles son demasiado altos.