

```

const int interruptPin = 2; // Definerer hvilken pin der skal bruges
volatile int count = 0; // Definerer en tæller
const unsigned long timeInterval = 60000;
unsigned long lastReset = 0;

uint8_t Protocol[7] = {255, 4, 113, 0, 0, 0, 254};

void setup() {
    // Initialiserer serielle kommunikation
    Serial.begin(19200);
    Serial1.begin(19200);
    // Sætter interruptPin som input
    pinMode(interruptPin, INPUT);
    // Aktiverer interrupts for pin 2 (INT0 på de fleste Arduino-modeller)
    attachInterrupt(digitalPinToInterrupt(interruptPin), incrementCount, RISING);
    //Serial.print(myFloat);
    //Serial.println(binaryInt, BIN); // Print in binary format
}

void initializeArray(float hz) {
    // Split the float into integer and fractional parts

    uint8_t Protocol[7] = {255, 4, 113, 0, 0, 0, 254};

    uint8_t one = (uint8_t)hz;
    uint8_t two = (uint8_t)((hz - one) * 100);

    Protocol[3] = one;
    Protocol[4] = two;

    int sum = 0;
    for (int i = 0; i < 6; ++i) {
        sum += Protocol[i];
    }
    Protocol[5] = sum;

    Serial.print("Array elements: ");
    for (int i = 0; i < 7; ++i) {
        Serial.print(Protocol[i]); // Output each element
        Serial.print(" "); // Separate elements by a space
    }
    Serial.println(); // End the line
    Serial1.write(Protocol[7]);
}

void loop() {

```

```

// Få aktuelt tidspunkt
unsigned long currentMillis = millis();

// Tjek om det er tid til at udføre nulstillingen
if (currentMillis - lastReset >= timeInterval) {
    // Gem tidspunktet for nulstillingen
    lastReset = currentMillis;
    // Beregn frekvensen
    float hz = count / (timeInterval / 1000.0); // Konverterer timeInterval til
sekunder
    // Nulstil tælleren
    count = 0;
    // Udskriv frekvensen
    Serial.print("Frekvens (Hz): ");
    Serial.println(hz);
    initializeArray(hz);
    // Udskriv nulstilling besked
    Serial.println("Nulstilling udført!");
}

// Andre handlinger kan udføres her
}

// Funktion kaldt af interrupt
void incrementCount() {
    // Tæller op
    //delay(1000);
    count++;
    //delay(1000);
    Serial.println(count);
}

```