

```

int button = 3;
boolean start = true;
int lightsensor = 0;
bool measuring = false;
long lastswitch = 0;
long lastcalc = 0;
int s = 1000;
int index = 0;
int array[3000];
int sonarTimeout = 10000;
int trigPin = 35;
int echoPin = 33;

void setup() {
    // put your setup code here, to run once:
    pinMode(button, INPUT);
    pinMode(echoPin, INPUT);
    pinMode(trigPin, OUTPUT);
    Serial.begin(9600);

    // init array with 0s
    for (int i = 0; i < 3000; i++){
        array[i] = 0;
    }
}

void loop() {
    if((digitalRead(button)==HIGH || (millis()-lastswitch >= 30*s)) && measuring==true){
        measuring = false;
        docalc();
    }
    else if(digitalRead(button)==HIGH){
        measuring = true;
        delay(500); // to make the button not trigger again right away
        lastswitch = millis();
    }

    // initialization of measuring process
    if (measuring == true){
        index = 0;
        Serial.println("Measuring started!");
    }

    // measuring process
    while (measuring == true){
        array[index] = triggerAndFetchSonar();
        Serial.println(array[index]);

        // analyze sensor data in detail with these lines:
        //Serial.print(array[index]);
        //Serial.print(" at index ");
        //Serial.println(index);

        waitAndCheck(10);
    }
}

```

```

        index++;
        if (index >= 3000){
            measuring = false;
            docalc();
        }
    }
}

void waitAndCheck(int x){
    lastcalc = millis();
    while(millis()-lastcalc < (x)){
        if(digitalRead(button)==HIGH){
            measuring=false;
            docalc();
        }
        else if(digitalRead(button)==HIGH)
            measuring=false;
    }
}

/**
 * Sendet einen Ultraschall-Puls aus und wartet auf das Echo. * Rueckgabewert: Verzoegerung
 */
unsigned long triggerAndFetchSonar() {
    digitalWrite(trigPin , LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin , HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin , LOW);
    return pulseIn(echoPin, HIGH, sonarTimeout);
}

void insertion_sort(int a[], int size) {
    for (int i = 1; i < size; i++) {
        int j = i;
        while (j > 0 && a[j - 1] > a[j]) {
            // Swap a[j] and a[j - 1]
            int tmp = a[j];
            a[j] = a[j - 1];
            a[j - 1] = tmp;
            j--;
        }
    }
}

// calculate mini max, average and reset array + index afterwards
void docalc(){
    Serial.println("Measuring ended!");
    delay(500); // to make the button not trigger again right away
    int mini = 1023;
    int maxi = 0;

```

```

float average = 0;

// index ends up being always 1 higher than the last entrance that was filled
for (int i = 0; i < index; i++){
    if (array[i] < mini)
        mini = array[i];
    if (array[i] > maxi)
        maxi = array[i];
    average += array[i];
}
average = average / index;

Serial.print("DEBUG: the min is: ");
Serial.println(mini);

Serial.print("DEBUG: the max is: ");
Serial.println(maxi);

Serial.print("DEBUG: the avg is: ");
Serial.println(average);

insertion_sort(array, index);

Serial.print("The Median is: ");
Serial.println(array[index/2]);

// clean array up
for (int i = 0; i < 3000; i++){
    array[i] = 0;
}
}

```