For noise disturbance detection in the environment

This code monitors the sound intensity using an LM393 sensor connected to an Arduino UNO board. The used sensor has only a digital output. Therefore, the number of times the sensor detects a sound is summed up over a sampling time called “SAMPLE_TIME”. Then the sum called “sampleBufferValue” is printed on a Serial Monitor and visualized with the Serial Plotter. Additionally, the code allows communication with a LED to provide a visual alarm if the “sampleBufferValue” surpasses a preset Threshold. Regarding the digital outputs, 0 means silence and 1 means noise.

Detailed explanation is given in the video tutorial

Sound_Detection.ino

```
 const int OUT_PIN = 12;    // The OUTPUT of the sound sensor is
connected to the digital pin D12 of the Arduino
 const int SAMPLE_TIME = 10;// The sampling time in milliseconds,  can
be set differently if required
 const int Threshold = 90;  // Threshold on decibel value for LED
switching ON, the value has been optimized with respect to
//the used sampling time (900 cumulative digital counts ≈ 90 dB from
"Schall")

 unsigned long millisCurrent;     // current time
 unsigned long millisLast = 0;    //previous time
 unsigned long millisElapsed = 0; // difference between current time
and previous time (time interval)

 int sampleBufferValue = 0;       // initiate the sum of digital
outputs over the sampling time
 int led = 8;                    // LED on pin 4 of Arduino
 int dB = 0;                     //initiate sound intensity dB value

void setup() {

  Serial.begin(9600);           //Arduino starts serial communication
with baud rate 9600
  pinMode(led,OUTPUT);          // the LED is connected as output for
alarm purpose

}

void loop() {

  millisCurrent = millis();                   //the current time is
assigned to the dedicated variable
  millisElapsed = millisCurrent - millisLast; //the elapsed time is
updated

  if(digitalRead(OUT_PIN) == HIGH){          //HIGH means noise
  sampleBufferValue++;                       //increments the sum
variable by 1
  }
   if (millisElapsed > SAMPLE_TIME) {        //if the elapsed time
surpasses the sampling time,
   //print the sampleBufferValue and test threshold for alarm

   dB = 0.0666 *(sampleBufferValue) + 30.223; //linear regression to
calculate the decibel value based of
   //the rough calibration of the sensor response
  Serial.println(dB);                        // print decibel values on
the Serial Monitor
  Serial.print("dB");                        // print sound unit
decibel


   if (sampleBufferValue > Threshold) {     // test if the threshold is
surpassed

    digitalWrite(led, HIGH);                //blink LED 2 ms ON and 1
ms OFF
    delay(2);
    digitalWrite(led, LOW);
    delay(1);
  }

  digitalWrite(led, LOW);                  // the LED is turned off to
be ready for the next sample
  sampleBufferValue = 0;                   // re-initialization of the
sampleBufferValue variable for the new sampling time
  millisLast = millisCurrent;              // update the previous time
to be the start for the next sample
  }
}
```

Back to report