Arduino-Based Decibel Detection System for Study Environment

Jiang Yuhan Guo Hangkai

Abstract

This study tackles the issue of elevated noise levels in study rooms. A device was designed to detect noise levels in various areas of a study room and provide reminders when levels exceed a set threshold. Within a 16m x 16m study room, the 16 device was spaced 4 meters apart. Each device has an Arduino UNO R3, sound sensors, and LEDs, noise levels were monitored, and alerts were issued to noisy areas. The system successfully localized noise sources and issued alerts accordingly. Advantages include higher accuracy in noise source localization and a targeted application for study rooms.

Introduction

The study room has been shown to be prone to elevated occupied noise levels that reduce learning efficiency. This experiment discusses a current study that designed a device that can detect the noise level in each area of the study room, and provide reminders for areas with noise exceeding the standard to reduce the noise source in the study room.

There is no noise level standard for self-study rooms in our country. According to the Technical Specification for Classification of Sound Environment Functional Zones

(GB/T 15190-2014), the study room belongs to Class 1 environmental functional zones. According to the “Environmental Quality Standards for Noise” (GB 3096-2008), the daytime environmental noise limit for Class 1 functional areas is 55 decibels, and the nighttime environmental noise limit is 45 decibels. According to the “Code for Design of Library Buildings” (JGJ38-2015), the allowable noise limit for the reading room is 45 decibels. Based on the above specifications, this article selects 45 decibels as the limit for environmental noise in the study room.

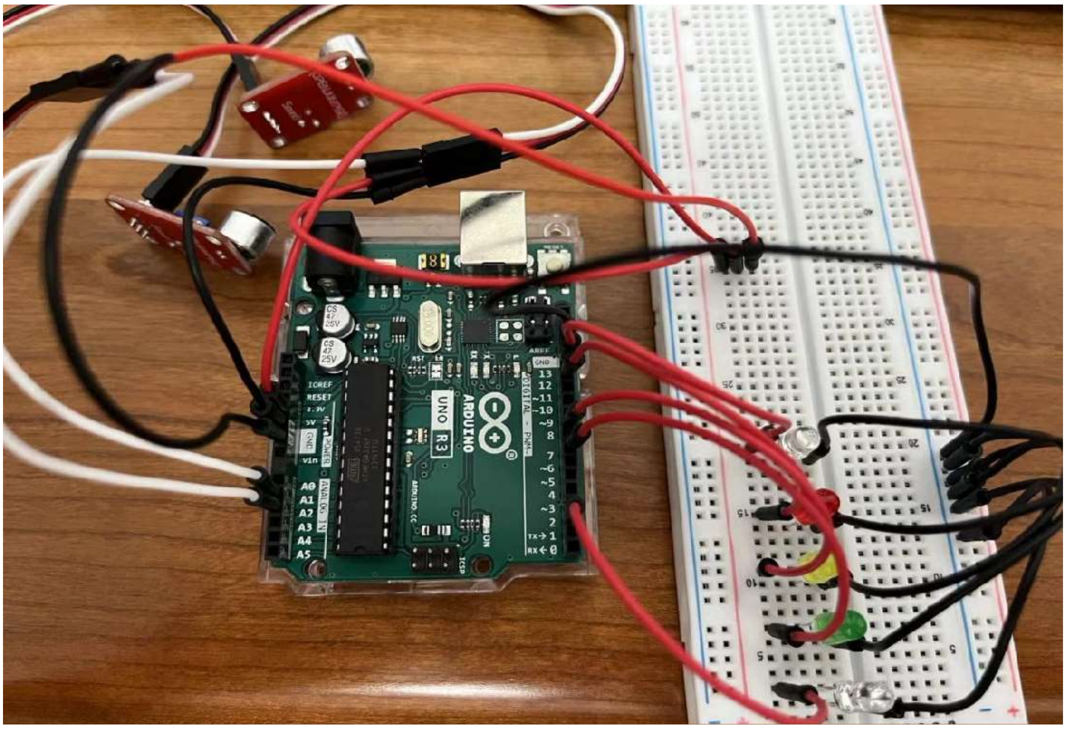
Due to the ability of sound to propagate in the air, a sound sensor receives data that exceeds a threshold, but the noise source may not necessarily come from the local area, it may also come from adjacent areas. Within a 16-meter by 16-meterstudy room, 16 sensors are placed in a grid formation to monitor noise levels. Each sensor is spaced 4 meters apart.

When adjoining sensors detect a higher noise level than this sensor, it is determined that the noise source comes from adjoining areas. This study used Arduino UNO，it is a board to get started with electronics and coding. Arduino UNO can be paired with many accessories. Through several sound sensors, it can read sound data from multiple areas, compare sound data exceeding the threshold with sound data from surrounding areas exceeding the threshold, comprehensively determine and locate the noise source, and issue reminders to the noise source.

Methods/Experimental design

Circuit design

This experiment uses the Arduino UNO R3 development board, sound sensors, and LEDs. Due to the interface limitations of the Arduino UNO R3 development board, the "ANALOG IN" interface is insufficient. Therefore, only 2 sound sensors are connected in this experiment, and the sound in the other 14 areas uses preset data methods; Use 5 LEDs, two of which correspond to areas with sound sensor inputs, and the other 3 LEDs correspond to areas with preset sound data exceeding 45dB.



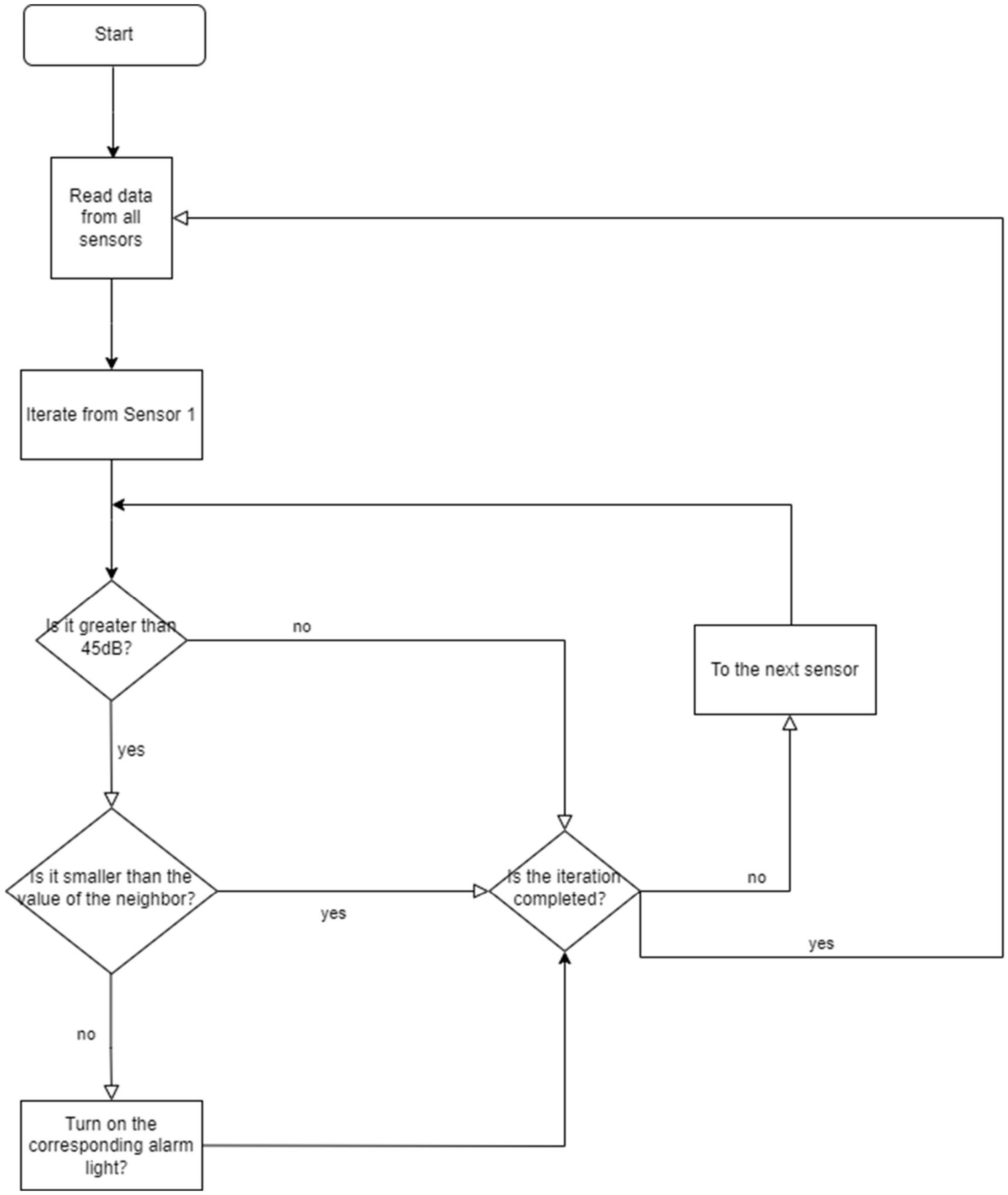
Program processing logic

Each of the sensor has a led near to it and the noise level detected by each of the sensor will be recorded. The LED is used to alert the students from the area which is the source of the noise. The LED numbers are from 1 to 16, and the 16 sound sensors are divided into 4 rows and 4 columns. The corresponding relationship between the sound sensor area and the LED is shown in the following figure:

	0	1	2	3
0	LED 1	LED 2	LED 3	LED 4
1	LED 5	LED 6	LED 7	LED 8
2	LED 9	LED 10	LED 11	LED 12
3	LED 13	LED 14	LED 15	LED 16

The program keeps looping, reading from sensor 1 to sensor 16. If a sensor records a noise level exceeding 45 decibels, and the adjoining sensor detects a noise level greater than the previous sensor, it is assumed that the noise source originates from the adjacent area and no alert is issued for that area. No alerts will be issued.

If a sensor records a noise level exceeding 45 decibels, and none of the adjoining sensor detects a noise level greater than the current sensor, it is assumed that the noise source originates from the current area and alert is issued for that area.


Start
Read data from all sensors
Iterate from Sensor 1
Is it greater than 45dB?
no
yes
To the next sensor
Is it smaller than the value of the neighbor?
yes
Is the iteration completed?
no
yes
no
Turn on the corresponding alarm light?

Program

```
void LEDOnOff(int num) {
  digitalWrite(num, HIGH);
  delay(100);
  digitalWrite(num, LOW);
}

void setup() {
  Serial.begin(9600);
  for (int i = 1; i <= 16; i++) {
    pinMode(i, OUTPUT);
  }
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
}


void loop() {
  int array[4][4];
  array[0][0] = 30;
  array[0][1] = analogRead(A1); //sensor A1 （2nd LED）
  Serial.print("A1=");
  Serial.print(array[0][1]);
  Serial.println("dB");
  array[0][2] = 40;
  array[0][3] = 35;
  array[1][0] = 29;
  array[1][1] = 40;
  array[1][2] = 60; // 7th LED
  array[1][3] = 80; // 8th LED
  array[2][0] = 30;
  array[2][1] = 30;
  array[2][2] = 28;
  array[2][3] = 51;
  array[3][0] = analogRead(A2); // sensor A2 （13th LED）
  Serial.print("A2=");
  Serial.print(array[3][0]);
  Serial.println("dB");
  array[3][1] = 40;
  array[3][2] = 40;
  array[3][3] = 43;
```

```
  int LED[4][4] = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12},
    {13, 14, 15, 16}
  };


  for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
      if (array[i][j] >= 45) {
        bool noiseFromNeighbor = false;
        if (i - 1 >= 0 && array[i - 1][j] > array[i][j] + 11) {
          noiseFromNeighbor = true;
        }
        if (i + 1 <= 3 && array[i + 1][j] > array[i][j] + 11) {
          noiseFromNeighbor = true;
        }
        if (j - 1 >= 0 && array[i][j - 1] > array[i][j] + 11) {
          noiseFromNeighbor = true;
        }
        if (j + 1 <= 3 && array[i][j + 1] > array[i][j] + 11) {
          noiseFromNeighbor = true;
        }
        if (!noiseFromNeighbor) {
          Serial.print("[");
          Serial.print(i);
          Serial.print("][");
          Serial.print(j);
          Serial.print("]:");
          Serial.print(array[i][j]);
          Serial.println("dB");
          LEDOnOff(LED[i][j]);
        }
      }
    }
  }
  delay(200);
}
```

Experimental results

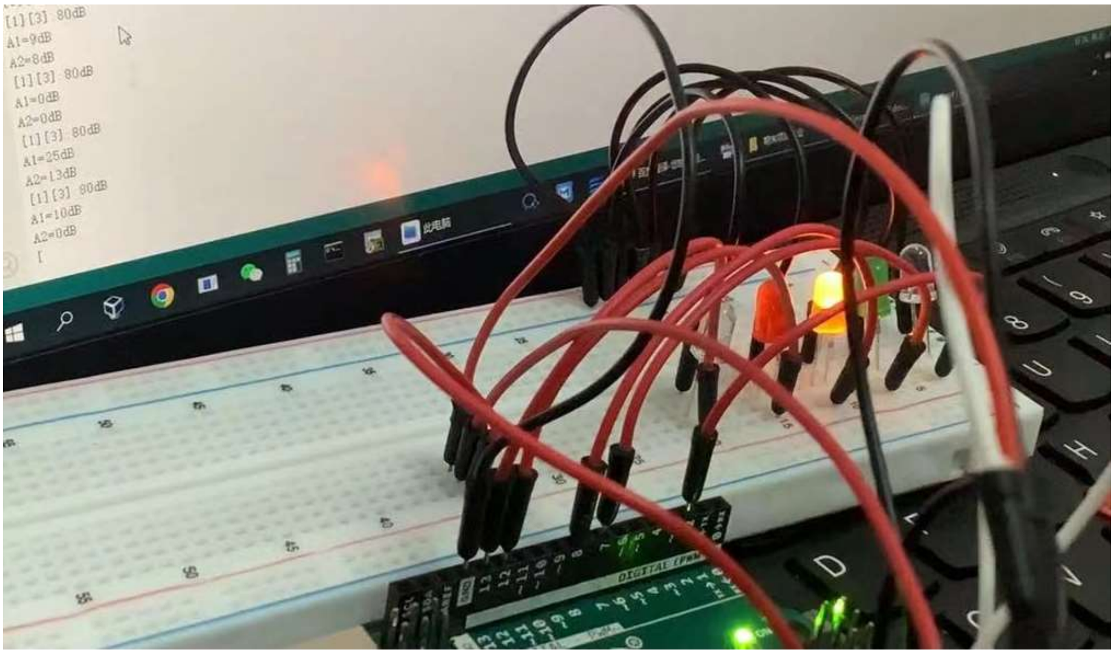
Simulated input data

	0	1	2	3
0	30	Sensor A1	40	35
1	29	40	60	80
2	30	30	28	51
3	Sensor A2	40	40	43

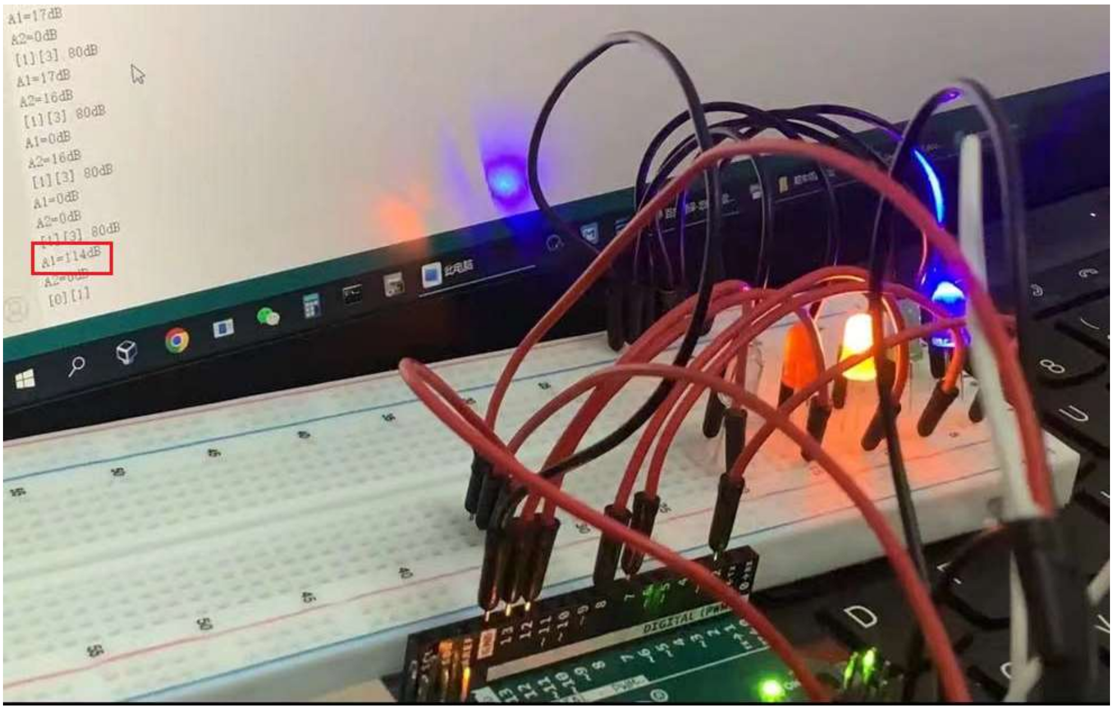
Results

The 8th LED lights is on, while the 2nd and 13th LED lights are judged based on sensor input data. If the data is greater than 45dB, the LED lights are on, otherwise the LED lights are not on.The 7th and 12th LED are close to the 8th LED, although they exceed 45 decibels, the LED does not light up.

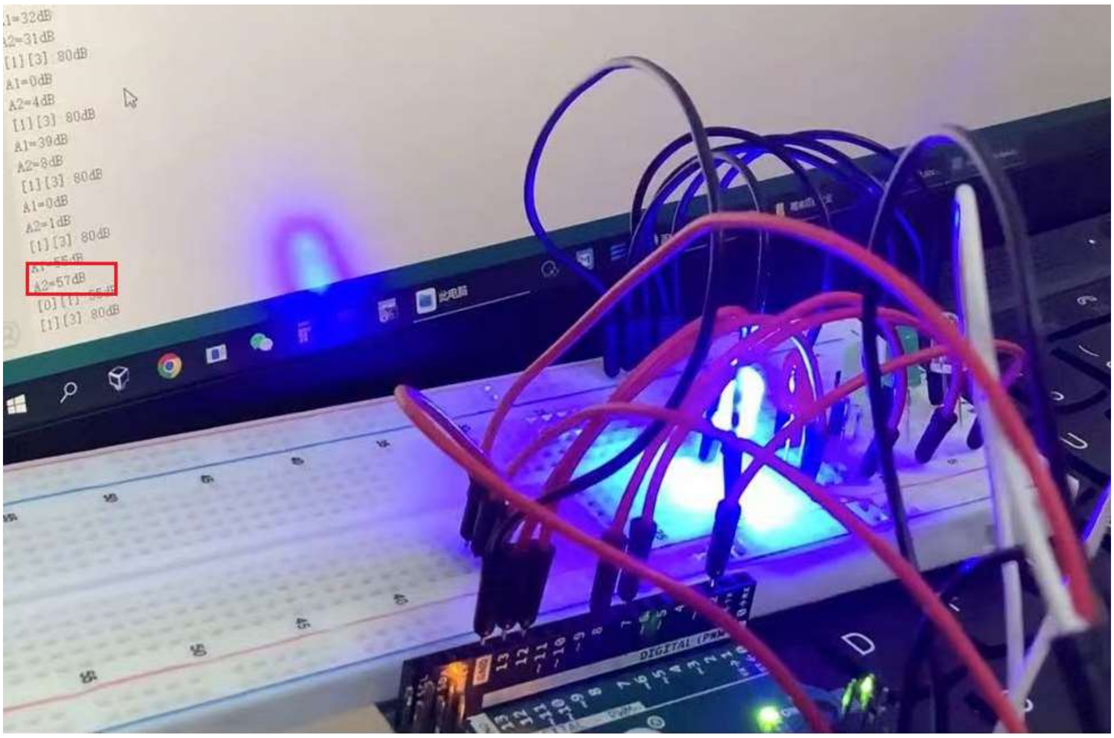
	0	1	2	3
0	30	Sensor A1	40	35
1	29	40	60	80
2	30	30	28	51
3	Sensor A2	40	40	43



All sound sensor input values are less than 45dB.Only the 8th LED light is on.



The input value of A1 sound sensor is greater than 45dB.The 2nd and 8th LED lights are on.



The input value of A2 sound sensor is greater than 45dB.The 13^{th} and 8^{th} LED lights are on.

Due to the flashing interval of the LED lights being 100 milliseconds, the output of the serial port monitor and the LED status are not completely synchronized.

Discussion

Advantages

1. **Higher Accuracy in Noise Source Localization**
    - **Number of Sensors**: Our project uses 16 sensors arranged in a grid pattern in a 16m × 16m study room. This layout allows for a more comprehensive monitoring of noise levels throughout the entire area, which is superior to their project that uses a single sensor to monitor the whole environment.
2. **Focus on Specific Application Scenarios**
    - **Study Room Design**: Unlike their project's broad application scope, our project is more targeted and effective in the study room setting. We have considered the noise standards and environmental characteristics of study rooms, aiming to avoid affecting students who are not making noise. (Edgar, Aakansha, et al.)

Disadvantages

1. **Sensor Cost and Maintenance**
    - **High Cost**: Using 16 sensors increases hardware costs compared to a project that uses only one sensor.
    - **Complex Maintenance**: More sensors mean more maintenance work, such as calibration, troubleshooting, and replacement.
2. **Limitations of Sensor Arrangement**

- **Fixed Arrangement**: The fixed arrangement of sensors may not be suitable for all environments, requiring repositioning for different scenarios.

3. **Power Consumption and Power Management**
   - **High Power Consumption**: Multiple sensors and real-time data processing can lead to high system power consumption, complicating power management.
   - **Battery Life Issues**: In battery-powered situations, using multiple sensors might affect the device’s battery life.
4. **Feedback Limitations**
   - **Limited Feedback Methods**: Displaying information about noise-exceeding areas through LEDs is relatively simple.

Conclusion

Our project has advantages in multi-sensor arrangement, noise source localization accuracy, and specific application scenario design. Especially in environments requiring precise noise monitoring and control, the multi-sensor grid layout and real-time alert features can provide higher monitoring accuracy. However, there are several issues with the practical operation of this project, such as sensor maintenance, layout limitations, and power management.

Future work

1. **Improving System Stability and Reliability**
    - **Environmental Adaptability**: Improve the design of sensors and devices to ensure stable operation under various environmental conditions.
    - **Fault Detection and Recovery**: Add fault detection and automatic recovery features to ensure long-term stable operation of the system.
2. **Optimizing Hardware and Software Design**
    - **Modular Design**: Adopt a modular design to make the system easier to maintain and upgrade.
    - **Low-Power Design**: Optimize hardware circuits and software algorithms to reduce system power consumption and extend device lifespan.
3. **Enhancing Alert and Control Functions**

- **Various Alert Methods**: Grade noise levels and use different LED light effects for different noise levels.
- **Remote Monitoring**: Implement remote monitoring and management through a cloud platform to improve system management efficiency.

4. **Providing Detailed Noise Data Recording and Analysis**
    - **Data Recording Function**: Add noise data recording and storage functions, allowing users to view historical noise data and understand noise trends.
    - **Data Analysis and Report Generation**: Provide noise data analysis functions and generate detailed noise reports to help users better understand noise conditions.

Reference

Edgar, A., Aakansha, et al. (n.d.). Noise Detector Alarm Device.

Technical Specification for Classification of Sound Environment Functional Zones. (2014). GB/T 15190-2014.

Environmental Quality Standards for Noise. (2008). GB 3096-2008.

Code for Design of Library Buildings. (2015). JGJ38-2015.