MEC 104 Report

Lab 3 Open Project
(Library anti-occupancy system)

Group Number: A32

Group Member:

Student Name	Student ID
Zhongnan Li	1931254
Chenruiyuan Yu	1928118
Ding Wang	1927804

Abstract

The project we chose to achieve is called "Library anti-occupancy system". The goal of this project is to make a detector that can detect whether there are people or objects on the seat. After the person temporarily leaves the seat, it will count down. After the timer is over, the status of no one will be displayed. The starting point of this tool is to help libraries implement the rule of no more than 30 minutes to occupy a seat and provide evidence for managers and users. Taking into account the limitation of similar public resources, this can also be widely used in similar scenes where pressure can be used to judge the existence of objects. The detection part uses two pressure sensors connected in series to detect the pressure values on the table and the chair. An Arduino uno board is used in the calculation and storage part. An LCD screen is used in the display status part. Also, a RGB LED is used to show the states of the seat, where red means "using"; blue means "temporarily leave"; green means "empty" and white means "too heavy". This system can display three states: no one, temporarily away, and using. When there is no one, "no one" will be displayed by the LCD screen. A countdown will be started when the temporary departure starts. There will be no display when someone is using the seat to prevent affection to the user.

Content

1. Introduction

Currently, when the exam approaches, students go to the library self-study classroom to prepare for the exam, in which case the library and the classrooms reserved for self-study will always show that the demand exceeds the supply. However, according to the complaints of classmates, many seats are occupied over time by students who are not in use and not restricted by the administrator. Even if the administrator will use the method of issuing warning notes to remind users, the manual method is still inefficient and relies on human operation. This system hopes to reduce human operations, use the processor's ability to provide reference for users and provide a reference for the reasonable distribution of resources in public spaces. Considering the degree of visual recognition and the development of artificial intelligence, even the algorithms deployed on the Arduino are sufficient for this level of recognition. However, considering the sensitivity of facial information and the possible concerns about personal privacy when deploying cameras. We chose to use pressure sensors without obvious personal information for practice.

Figure 1: Occupation situation (XJTLU Library)

The first pressure sensors detect the pressure on the seat and returns three types of states: 0 means there is no pressure, 1 means there is pressure but not enough to be judged as human pressure, and 2 means that the pressure is large enough to judge as a person's weight. The second group of pressure sensors detects the pressure on the desktop and returns to two states: 0 means there is no pressure, 1 means there is something on the desk. In the two different state combinations and changes, the RGB light and the LCD screen will display different states. It mainly depends on the changes of the collected data and simple logic to control.

2. Main content

2.1 Theory

In the initial stage, when the pressure breaks through a certain pressure threshold, there is a breakthrough in the on-resistance. Before this threshold, the FSR is equivalent to a switch. When this threshold is exceeded. There is a continuous relationship between the resistance of the FSR and the pressure.

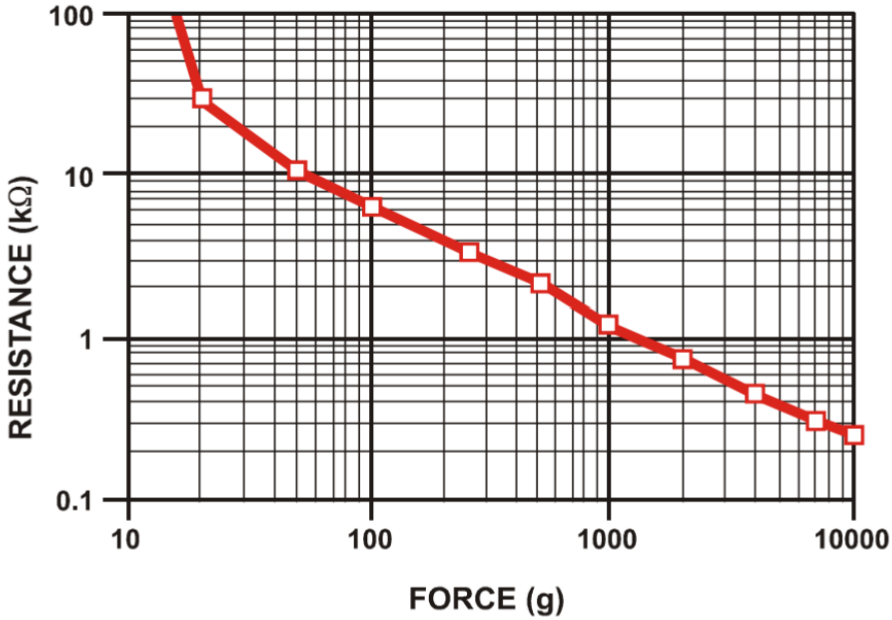


Figure 2: Force sensor data

This pressure sensor will output the voltage of the analog signal to realize the reading of the pressure state. As the report has mentioned before, the separation of the states is given below.

Table 1: Situation table for several states

A (sensor in seat)	B (sensor in desk)	X (LCD state)
0	0	0
0	1	1
1	0	1
1	1	1
2	0	2
2	1	2

The measurement of these different values requires experimental measurement in conjunction with equipment and scene. The values set in this simulation are tentative. In the code, it provides a unified and simple way to modify parameters. After assembling the equipment on the table and chair, use real people and real objects to read the parameters and manually reset the corresponding parameters.

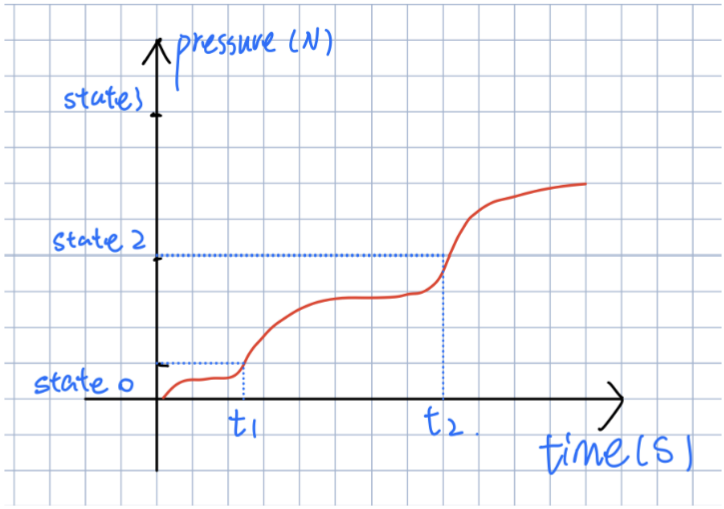


Figure 3: Schematic diagram of situation 1

Before t1, the state is 0—no one is using.
At t1, the timer starts counting down.
If t2-t1<30min the timer reset at t2 and state turns into 2—someone is using.
If t2-t1 in the timer reset at t1+30min and states firstly turns into 0—no one is using. At t2 the state turns into 2. Between t1+30min and t2 the seat allows anyone to use.

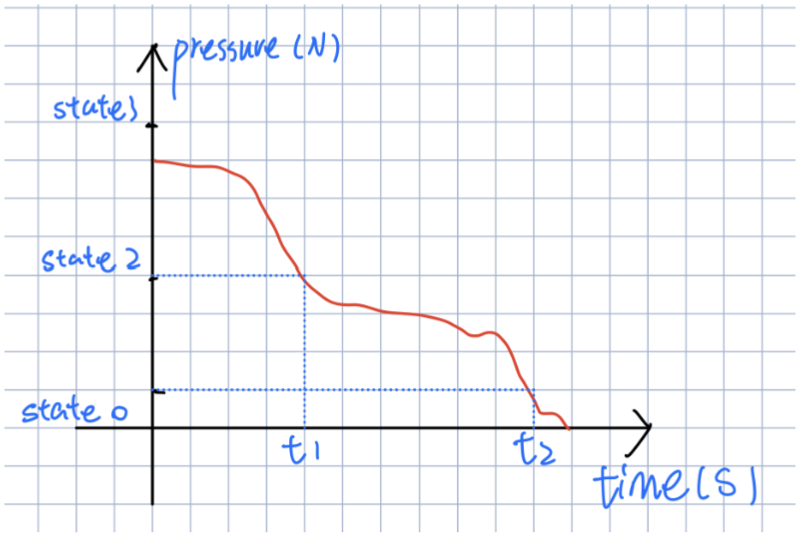


Figure 4: Schematic diagram of situation 2

Before t1, the state is 2— someone is using.
At t1, the timer starts counting down.
If t2-t1<30min the timer reset at t2 and state turns into 0—no one is using.
If t2-t1>30min the timer reset at t1+30min and states firstly turns into 0—no one is using. At t2 the state turns into 0. Between t1+30min and t2 the seat allows anyone to use.

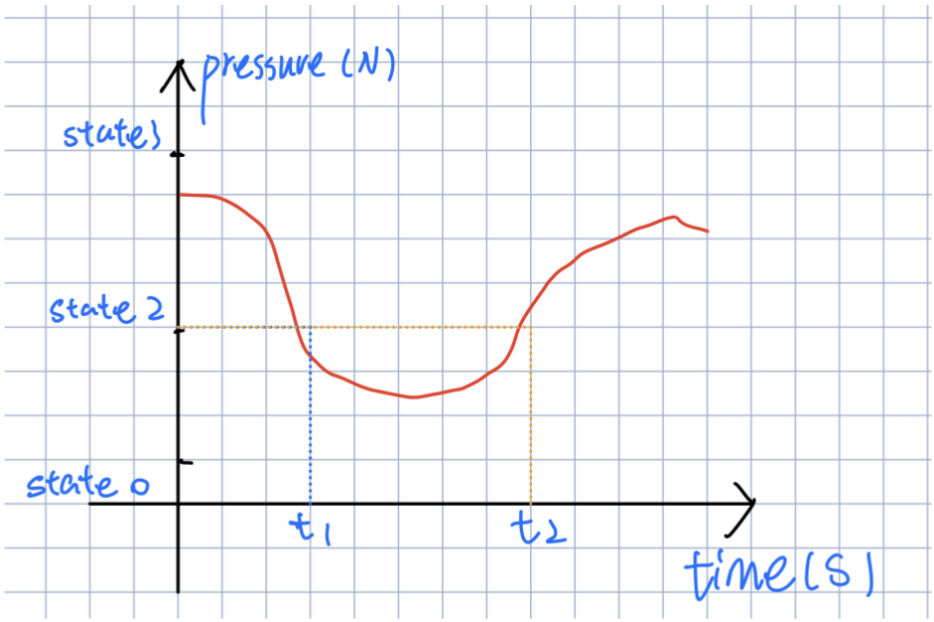


Figure 5: Schematic diagram of situation 3

Before t1, the state is 2—no one is using.
At t1, the timer starts counting down.
If t2-t1<30min the timer reset at t2 and state turns into 2—someone is using.
If t2-t1>30min the timer reset at t1+30min and states firstly turns into 0—no one is using. At t2 the state turns into 2. Between t1+30min and t2 the seat allows anyone to use.
In addition, if at any time the state turns to 0, the LCD will present “no one” at once.

2.2 Circuit Design

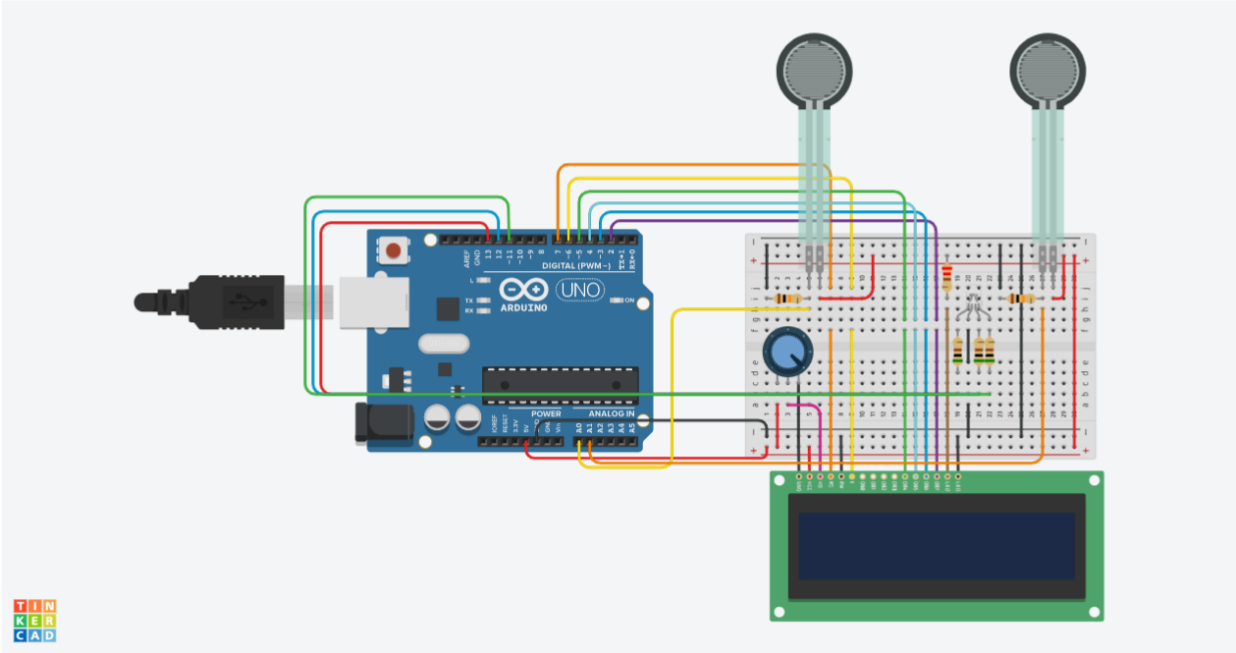


Figure 6: Experimental circuit

The circuit above is what we built to achieve “Library anti-occupancy system”. The specific circuit connections will be listed below. First, as the output terminals of Arduino, PWM pins 2, 3, 4, 5, 6, 7 are connected to a LCD to achieve the function of display of reminding words and time counter. The output pins 11, 12, 13 are connected to a RGB light to achieve the function of indicator light. The analog port A0 and

A1 are respectively connected to two force sensors to detect the force on the chair and desk. The specific theorem of how to connect these components are mentioned in the above theorem part.

2.3 Experimental Method

Code structure

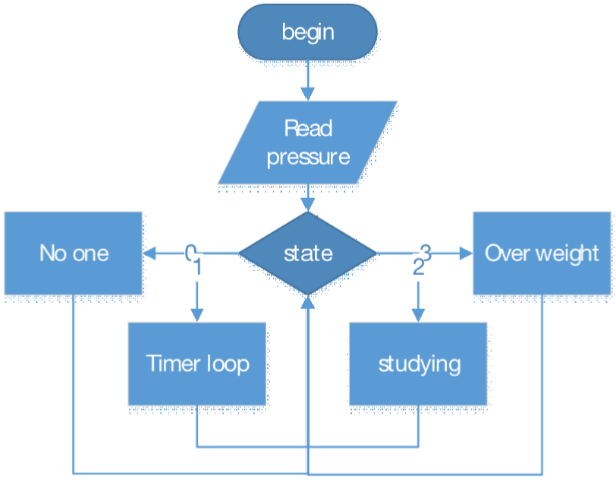


Figure 7: Main loop

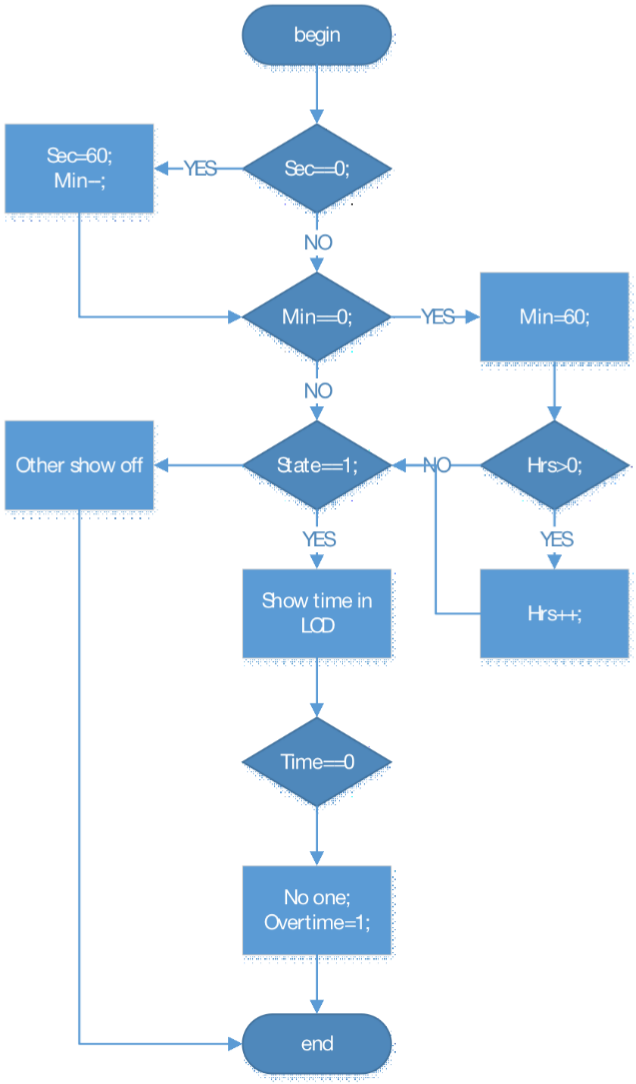


Figure 8: Loop for counter

Problems during debugging

For the logic part, we first neglected the situation which the state should change to 0 after the count down process even there is still a relevant force on the sensors. To solve this problem, we added a new parameter to control the situations the parameter will be reset after the state changed to actual no one and studying.

Table 2: The overtime situation

overtime	state	Present state
0	1	1
1	1	0

For the timer part, we first set the second and minute will change from 0 to 59. However, we separated count down process and digits change process. The code change from 0 to 60 if necessary.

2.4 Results

In this part, the experimental results will be explained in detail. When we click to start the simulation, the force sensors starts to detect the forces on the two sensors and displays the result in the LCD and RGB light change colors according to the states.

(1) The "Empty" state

When we start the simulation, the initial state is the "Empty" state, since no force on the two sensors. The RGB light is on green, and "Empty" displays on LCD.

The serial monitor keeps showing "Pressure on chair = 0 - No pressure
Pressure on desk = 0 - No pressure".

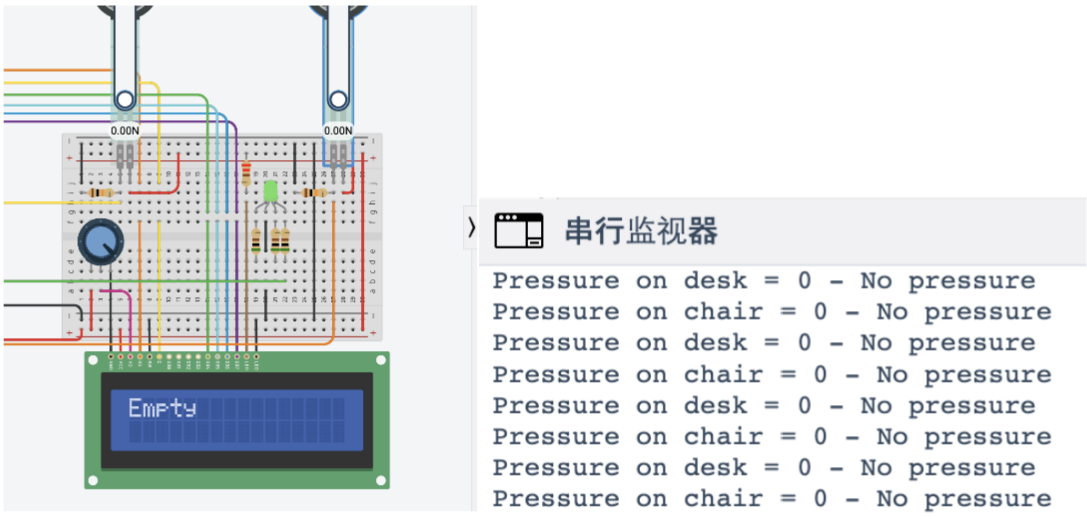
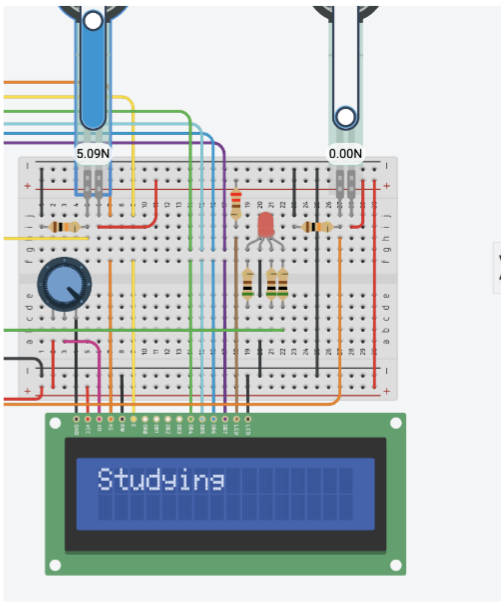


Figure 9: Result of state1 = 0 and state2 = 0

(2) The "Studying" state

And then, we change the force on the chair to a larger number, the sensor determines human in sitting on the chair, therefore the state changes to the "Studying" state. The RGB light is on red, and "Studying" displays on LCD. At this state, no matter we change the force on the desk, the state will not change.

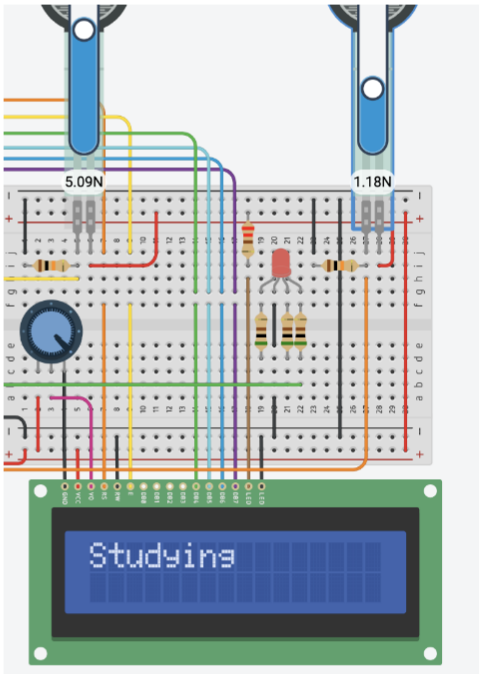
The serial monitor keeps showing "Pressure on chair = 858 - Student
Pressure on desk = 0 - No pressure" or "Pressure on chair = 858 - Student
Pressure on desk = 893 - Object".



串行监视器

```
Pressure on desk = 858 - Student
Pressure on chair = 0 - No pressure
Pressure on desk = 858 - Student
Pressure on chair = 0 - No pressure
Pressure on desk = 858 - Student
Pressure on chair = 0 - No pressure
Pressure on desk = 858 - Student
Pressure on chair = 0 - No pressure
```

Figure 10: Result of state1 = 2 and state2 = 0



串行监视器

```
Pressure on desk = 858 - Student
Pressure on chair = 893 - Object
Pressure on desk = 858 - Student
Pressure on chair = 893 - Object
Pressure on desk = 858 - Student
Pressure on chair = 893 - Object
Pressure on desk = 858 - Student
Pressure on chair = 893 - Object
```

Figure 11: Result of state1 = 2 and state2 = 1

(3) The “Overweight” state

And then, we change the force on the chair to the largest number, the sensor determines something overweight is on the chair, therefore the state changes to the “Overweight” state. The RGB light is on white, and “Too heavy!” displays on LCD. At this state, no matter we change the force on the desk, the state will not change.

The serial monitor keeps showing “Pressure on chair = 914 - Warning! Too heavy!
Pressure on desk = 0 - No pressure” or “Pressure on chair = 914 - Warning! Too heavy!
Pressure on desk = 847 - Object”.

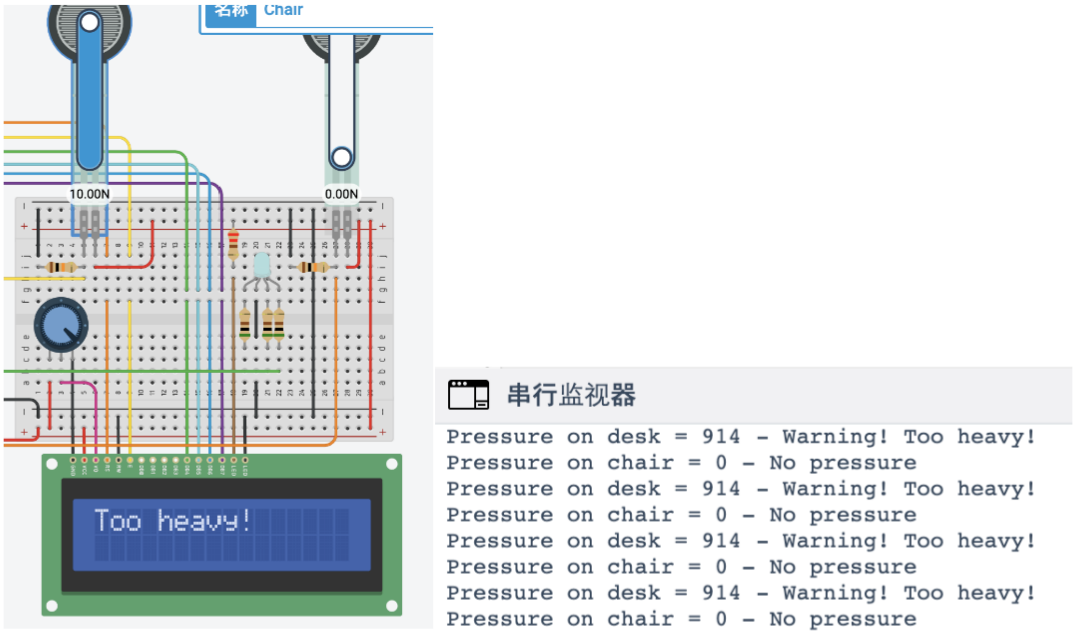


Figure 12: Result of state1 = 3 and state2 = 0

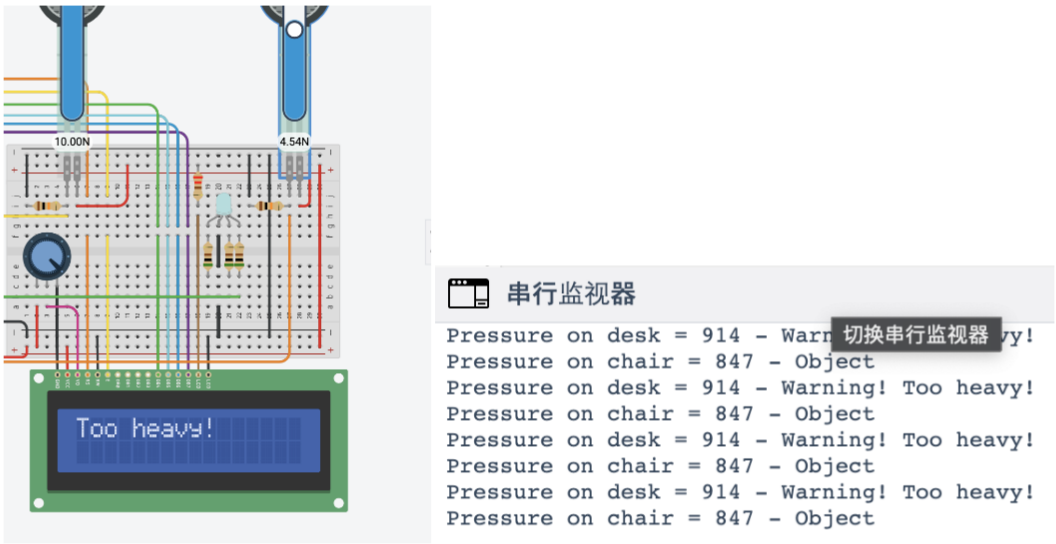
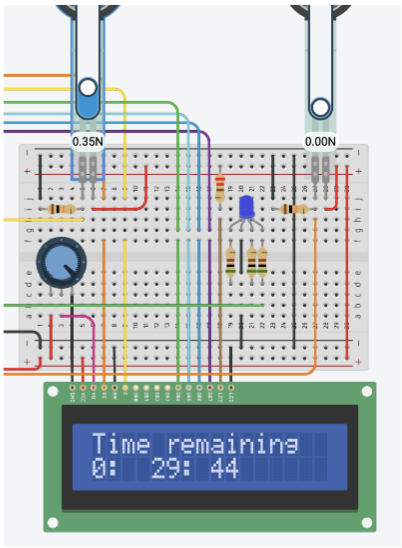


Figure 13: Result of state1 = 3 and state2 = 1

(4) The "Temporarily unoccupied" state

And then, we change the force on the chair to a smaller number, the sensor determines objects on the chair, therefore the state changes to the "Temporarily unoccupied" state. The RGB light is on blue, and the counter begins to start with the real timing displays on LCD. There are three situations will lead to this state, which are objects on the chair, and nothing on the desk; objects on the chair and desk; nothing on the chair, objects on the desk.

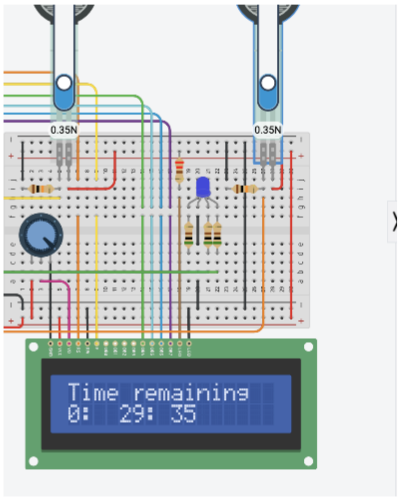
The serial monitor keeps showing the states of two sensors like the other states.



串行监视器

```
Pressure on desk = 413 - Object
Pressure on chair = 0 - No pressure
Pressure on desk = 413 - Object
Pressure on chair = 0 - No pressure
Pressure on desk = 413 - Object
Pressure on chair = 0 - No pressure
Pressure on desk = 413 - Object
Pressure on chair = 0 - No pressure
```

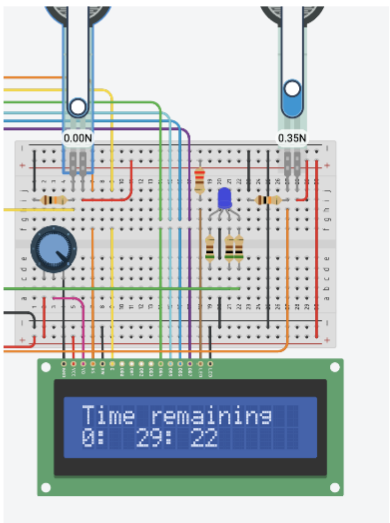
Figure 14: Result of state1 = 1 and state2 = 0



串行监视器

```
Pressure on desk = 413 - Object
Pressure on chair = 413 - Object
Pressure on desk = 413 - Object
Pressure on chair = 413 - Object
Pressure on desk = 413 - Object
Pressure on chair = 413 - Object
Pressure on desk = 413 - Object
Pressure on chair = 413 - Object
```

Figure 15: Result of state1 = 1 and state2 = 1



串行监视器

```
Pressure on desk = 0 - No pressure
Pressure on chair = 413 - Object
Pressure on desk = 0 - No pressure
Pressure on chair = 413 - Object
Pressure on desk = 0 - No pressure
Pressure on chair = 413 - Object
Pressure on desk = 0 - No pressure
Pressure on chair = 413 - Object
```

Figure 16: Result of state1 = 0 and state2 = 1

During the “Temporarily unoccupied” state, if we change the forces, the state will follows the change and become “Empty”, “Studying” or “Overweight” immediately.

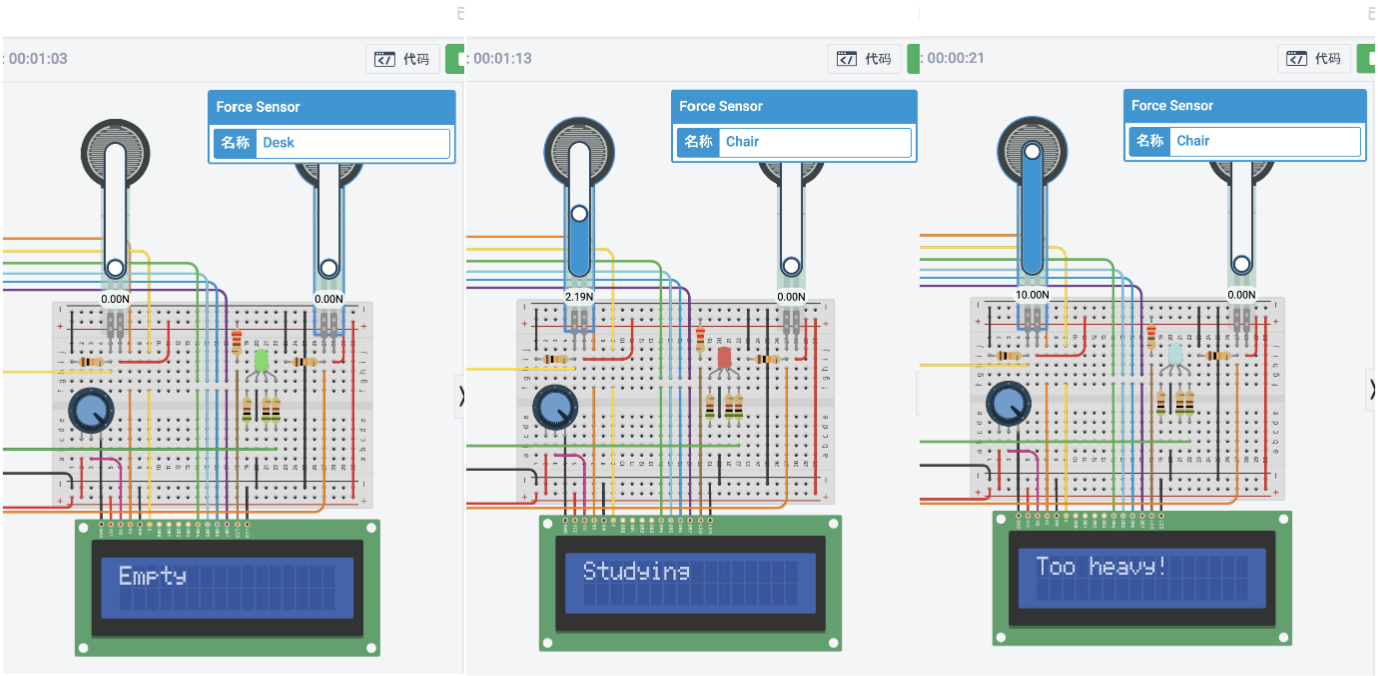


Figure 17: During "Temporarily unoccupied" state, situation for changes of state

(5) The "Overtime" state

During the "Temporarily unoccupied" state, when the time is run out, the state will automatically change to "Empty" state, which means the one who occupied the seat has no right to use the table.

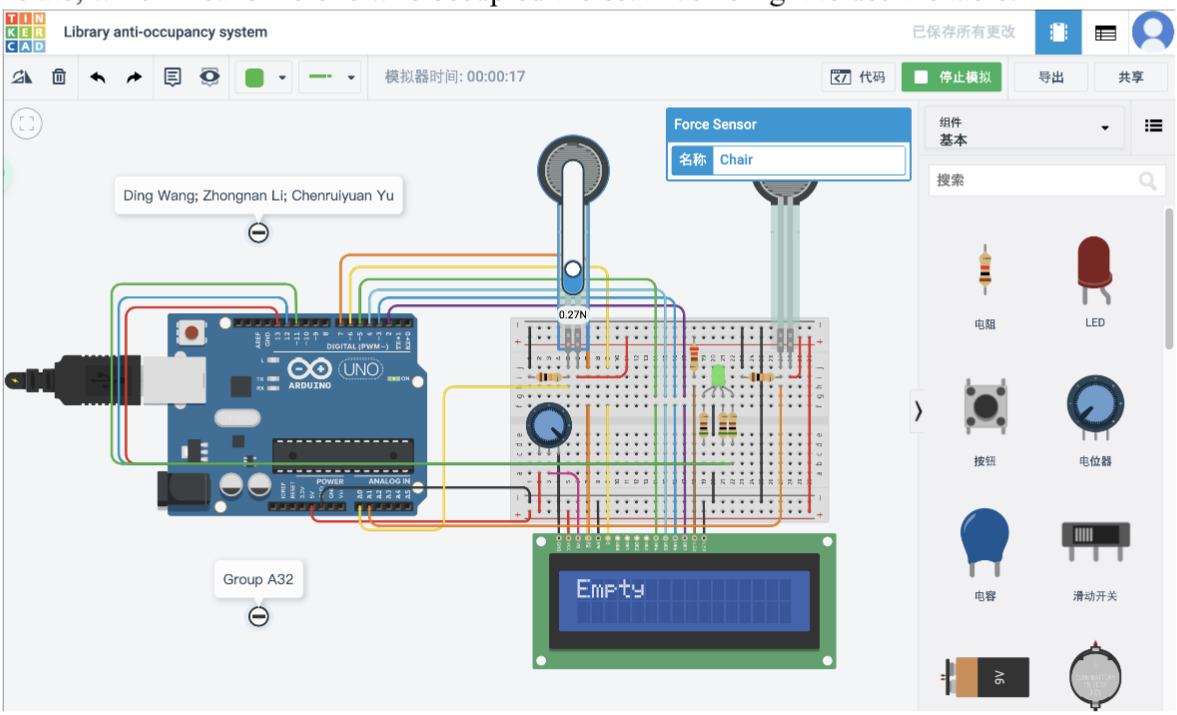


Figure 18: Overtime situation

2.5 Discussion

With the joint efforts of the team members, we have realized the operation of the project. In the testing process, timer was tested from 2 hours to ensure all digits are reliable and the shifting process are all works well. For the sensor states, it is changed from each other to test whether all of them works properly as expected.

2.6 Further study

In view of that this project starts from the needs of students, it can be effectively implemented as a part of the library and added to the daily management. As a prototype, we use Arduino which is relatively expensive and has more computing power than required. Multiple units can be processed by one processor. In addition to computing on the LCD, it can even connect to the network to update the seat status in real time so that more students who want to go to the public space for learning activities can provide reference.

3. Conclusion

Our group started from the needs in life, combined practice to the skills we have and finally designed the library anti-occupancy system. This system will provide a solution for a more reasonable allocation of existing resources. Better management and use of these resources are effective ways to increase the utilization of idle-free space. While writing code, we also provide approaches for subsequent modification of parameters and addition of content to increase its applicable scenarios. From the simulation results, this circuit and the corresponding code are reliable and effective, which shows that the system has made a success.

4. Author list and contributions

NO.	Name	Contribution description	Percentage (%)	signature
1	Zhongnan Li 1931254	Code construction and Part of debugging Modification of report and poster	35	Zhongnan.Li
2	Chenruiyuan Yu 1928118	Debugging for the code Report construction Modification of poster	35	Chenruiyuan.Yu
3	Ding Wang 1927804	Poster construction Video record Modification of report	30	Ding.Wang

5. References

[1] BASEMU, “Make an Arduino controlled timer”
Available:https://www.basemu.com/build-an-arduino-controlled-timer.html [Online]
[2] INTERLINK ELECTRONICS, “FSR Force Sensing Resistor Integration Guide and Evaluation Parts Catalog” Available: http://www.tinyos.net.cn/datasheet/fsrguide.pdf [Online]

6. Appendix

```
#include <LiquidCrystal.h>
#define sethrs 2
#define setmins 1
#define setsecs 1
```

```
// set pins for LCD
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int fsrPin = 0;          // A0
int secPin = 1;          // A1
int redPin = 13;         // red RGB
int bluePin = 12;        // blue RGB
int greenPin = 11;       // green RGB

int fsrReading;
int secReading;

int state1;
int state2;

int hrs = sethrs;       // for setting hours
int mins = setmins; // for setting minutes
int secs = setsecs; // for setting seconds

bool overtime=0;     // overtime situation
bool startTimer = false;
bool setTimer = true;
bool get_time = false;

unsigned long interval=1000; // waiting time
unsigned long previousMillis=0; // millis() returns the length of an unsigned number

// Function for color control
void color (int red, int green, int blue)
{
   analogWrite(redPin, red);
   analogWrite(bluePin, blue);
   analogWrite(greenPin, green);
}
void setup()
{
   pinMode(redPin, OUTPUT);
   pinMode(bluePin, OUTPUT);
   pinMode(greenPin, OUTPUT);
   Serial.begin(9600);
   lcd.begin(16, 2);
   lcd.setCursor(0,0);
   lcd.print("Library-");
   lcd.setCursor(0,1);
   lcd.print(" Seats");
   delay(1000);
```

```
    }

// The main loop
void loop(){
    fsrReading = analogRead(fsrPin);
    secReading = analogRead(secPin);
    Serial.print("Pressure on desk = ");
    Serial.print(fsrReading);

    // Determine the weight on the seat
    if (fsrReading < 10) {
        Serial.println(" - No pressure");
        state1 = 0;    // the situation of no force on the seat
    } else if (fsrReading >= 10 && fsrReading < 500) {
        Serial.println(" - Object");
        state1 = 1;    // the situation of objects on the seat
    } else if (fsrReading >= 500 && fsrReading < 900) {
        Serial.println(" - Student");
        state1 = 2;    // the situation of students on the seat
    } else {
        Serial.println(" - Warning! Too heavy!");
        state1 = 3;    // the situation of overweight thing on the seat
    }
    Serial.print("Pressure on chair = ");
    Serial.print(secReading);

    // Determine the weight on the desk
    if (secReading < 10) {
        Serial.println(" - No pressure");
        state2 = 0;    // the situation of no force on the desk
    } else if (secReading >= 10 && secReading < 900) {
        Serial.println(" - Object");
        state2 = 1;    // the situation of objects on the desk
    } else {
        Serial.println(" - Warning! Too heavy!");
        // the situation of overweight thing on the desk
    }
    // Determine the state of this seat
    if ((state1 == 0 && state2 == 0) || overtime && state1 != 2) {
        color(0,255,0);
        lcd.clear();
        lcd.print("Empty");
    } else if (state1 == 2) {
        color(255,0,0);
        lcd.clear();
        lcd.print("Studying");
```

```
        overtime=0;
    } else if (state1 == 3) {
        color(255,255,255);
        lcd.clear();
        lcd.print("Too heavy!");
    } else {
        color(0,0,255);
        startTimer = true;
// Temporarily unoccupied
    }
    delay(1000);

    // Loop for the situation of temporarily unoccupied seat
    if(startTimer == true){
      start_timer();
      if(secs == 0 && mins == 0 && hrs == 0) {
        startTimer == false;
      }
    }
}
// Function for calculating the time
void start_timer(){
if(secs ==0){
    secs = 60;
    mins = mins - 1;
    }
if(mins <0 && hrs){
    mins = 59;
    if(hrs>0)
    hrs = hrs - 1;
    }    // To check whether the time is normal
    get_time = true;
    if (state1==0&&state2==0) {
        color(0,255,0);
        lcd.clear();
        lcd.print("Empty");
        hrs = sethrs;
        mins = setmins;
        secs = setsecs;    // Empty seat
    }
    else if(state1 == 2) {
        color(255,0,0);
        lcd.clear();
        lcd.print("Studying");
        hrs = sethrs;
        mins = setmins;
```

```
      secs = setsecs;    // Empty seat
    }
    else if(state1 == 3) {
      color(255,255,255);
      lcd.clear();
      lcd.print("Too heavy!");
      hrs = sethrs;
      mins = setmins;
      secs = setsecs;    // Empty seat
    }
    else if(!overtime) {
    counter();

    lcd.setCursor(0, 0);
    lcd.print("Time remaining");

    lcd.setCursor(0, 1);
    lcd.print(hrs);
    lcd.print(":");

    lcd.setCursor(4, 1);
    lcd.print(mins);
    lcd.print(":");

    lcd.setCursor(8, 1);
    lcd.print(secs);

    // When the time is run out, the seat becomes "Empty"
    if(secs == 0 && mins == 0 && hrs == 0) {
      color(0,255,0);
      lcd.clear();
      lcd.print("Empty");
      overtime = 1;
      hrs = sethrs;
      mins = setmins;
      secs = setsecs;    // Empty seat
     }
    }
}
void counter() {
    unsigned long currentMillis = millis(); // Get the current time
    // Check whether the "interval" time has exceeded (1000 milliseconds)
    if ((unsigned long)(currentMillis - previousMillis) >= interval) {
    lcd.clear();
    if(get_time == true){
      secs--;
```

```
      get_time = false;
      }
    previousMillis = millis();
    }
}
```