

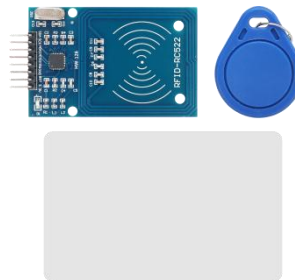
Project 21-RFID access control experiment

1. project description

With this simple Arduino project, you can use the RC522 RFID sensor to control the servo to implement an access control system. You only need a ZY-type-c Nano main control board, a G90 servo motor and an RC522 RFID sensor.

2. Module introduction

2.1 RC522 RFID sensor



The RC522 RFID module based on NXP's MFRC522 IC is one of the cheapest RFID options. It usually comes with RFID card tag and key fob tag with 1KB memory. Best of all, it can write to tags .

Module principle :

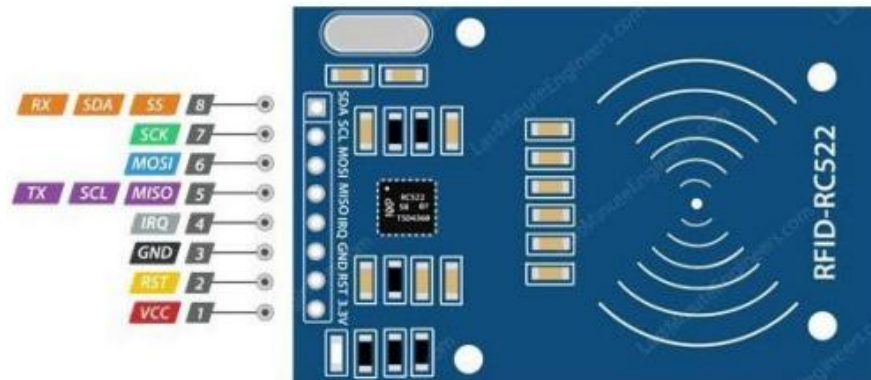
The RC522 RFID reader module is designed to create a 13.56MHz electromagnetic field for communication with RFID tags (ISO 14443A standard tags). The reader can communicate with the microcontroller via a 4-pin serial peripheral interface (SPI) with a maximum data rate of 10Mbps. It also supports communication via I2C and UART protocols.

This module has an interrupt pin. This is convenient because instead of constantly asking the RFID module "Is there a card in view?" the module will alert us when a tag comes into its vicinity.

The module operates from 2.5 to 3.3V, but the logic pins are 5V tolerant, making it easy to connect to an Arduino or any 5V logic microcontroller without using any logic level converters.

RC522 RFID module pinout:

The RC522 module has a total of 8 pins for connecting external devices .



VCC Power the module. This can be anything between 2.5 and 3.3 volts. It can be connected to the Arduino's 3.3V output.

RST is the reset and power-down input. When this pin goes low, hard power-down is enabled. This turns off all internal

current sinks, including the oscillator, and the input pins are disconnected from the outside world. On the rising edge, the module is reset.

GND is the ground pin and needs to be connected to the GND pin on the Arduino.

IRQ is an interrupt pin that alerts the microcontroller when an RFID tag comes into its vicinity.

The MISO/SCL/Tx pin acts as a master input slave output when the SPI interface is enabled, a serial clock when the I2C interface is enabled, and a serial data output when the UART interface is enabled.

MOSI (Master Out Slave In) is the SPI input of the RC522 module.

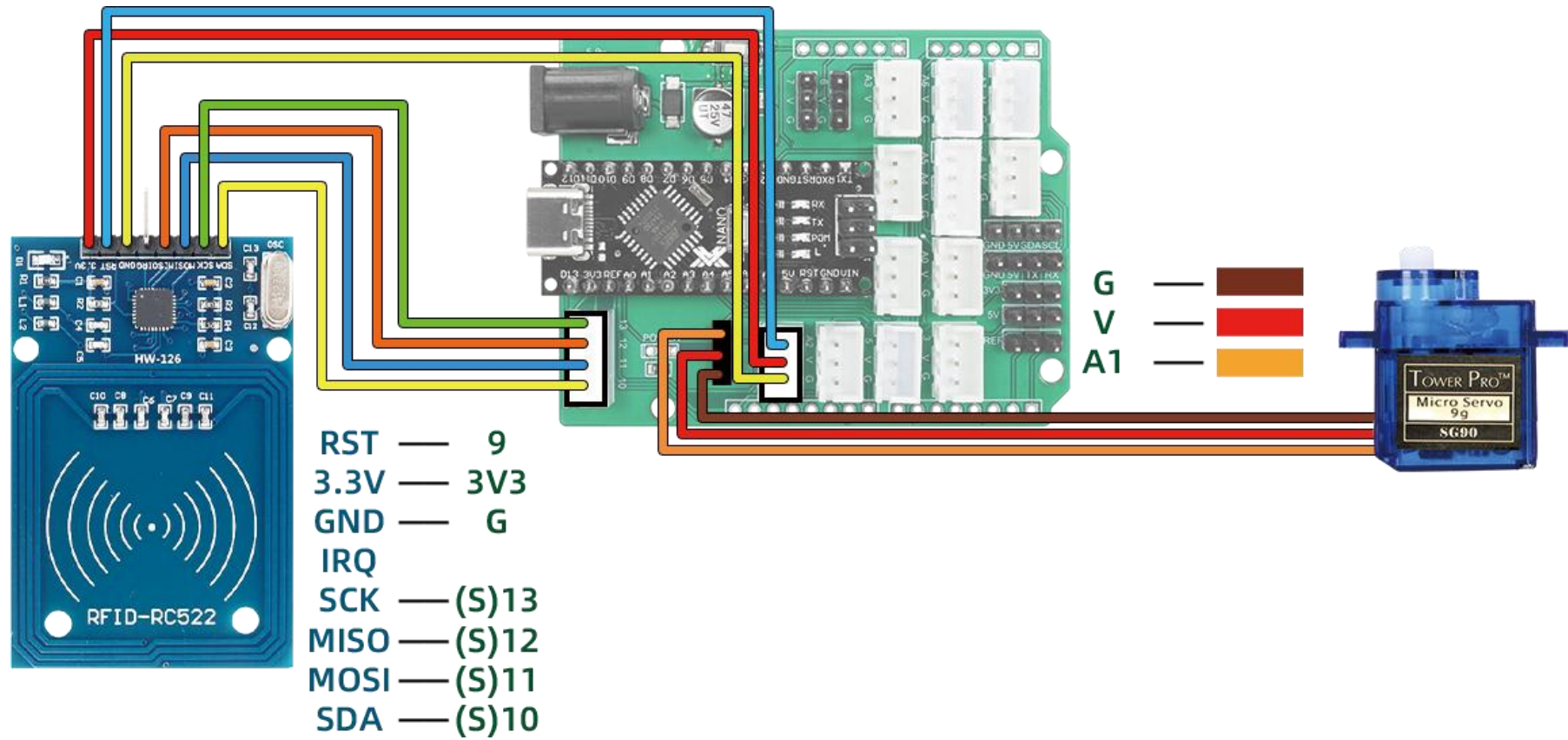
SCK (Serial Clock) accepts clock pulses provided by the SPI bus master (i.e. Arduino).

The SS/SDA/Rx pins act as signal inputs when the SPI interface is enabled, serial data inputs when the I2C interface is enabled, and serial data inputs when the UART interface is enabled. This pin is usually labeled by enclosing the pin in a square so it can be used as a reference to identify other pins.

How to make your RFID door lock ?

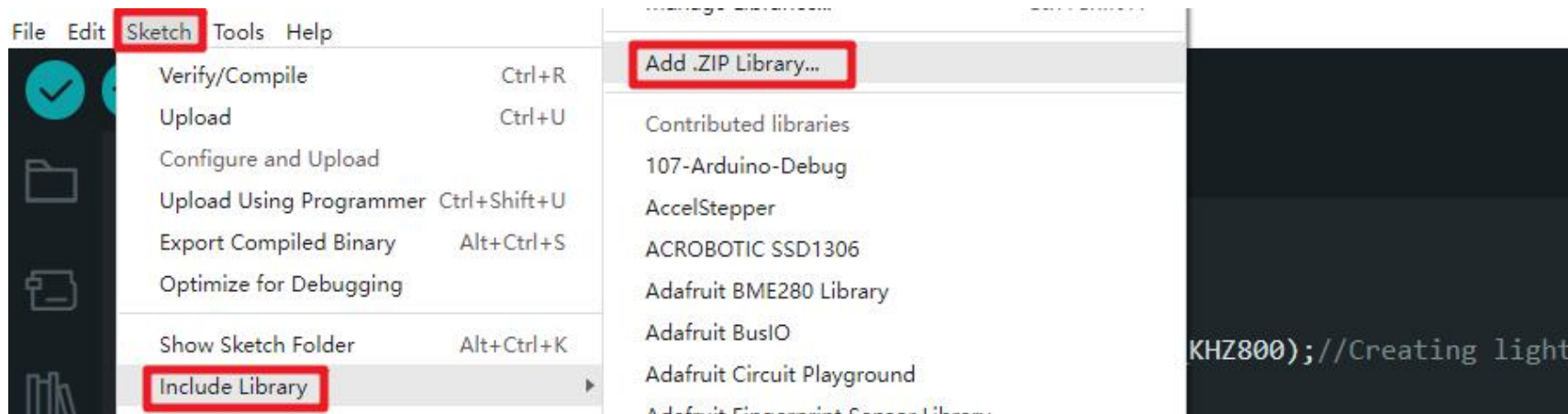
First, let's connect the RFID sensor to your ZY TYPE-C NANO using Dupont wire . This will allow you to test the circuit and read the serial numbers from the tags so that they can be loaded into an array in the code. When the serial number is detected to match the one in the array, access is granted to control the rotation of the servo and unlock the door. Function.

3. Project wiring diagram

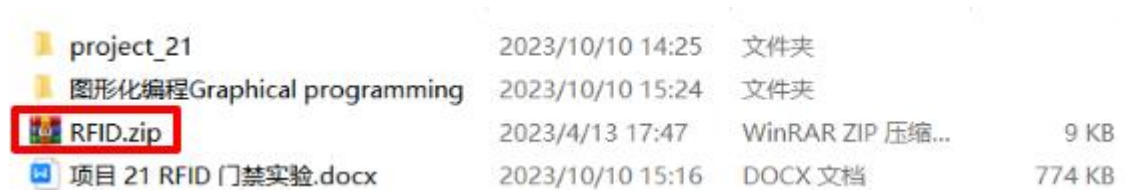


4. Add RFID.zip library

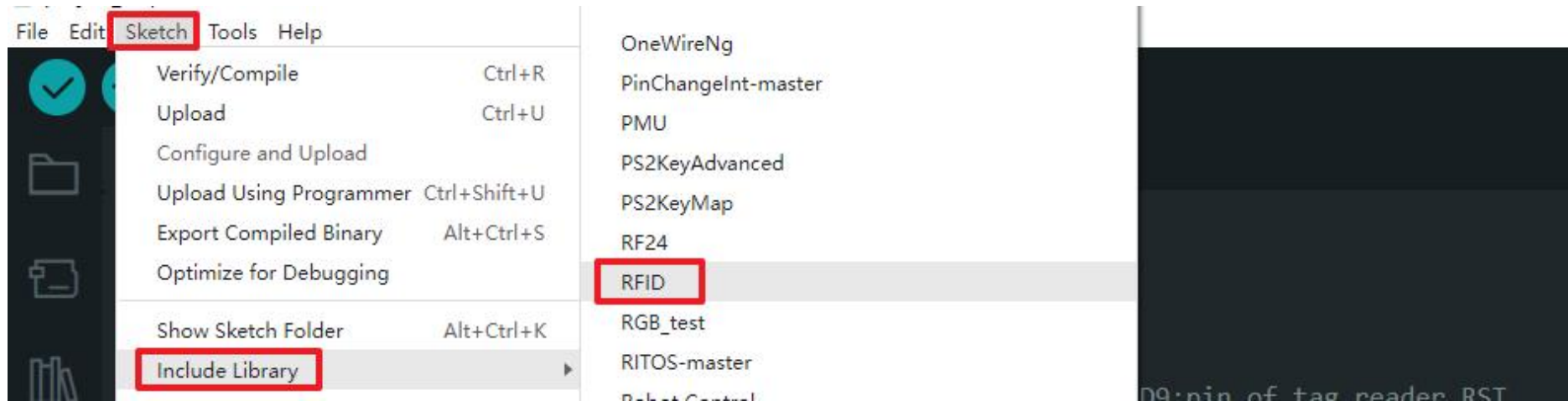
Here, the library used is RFID .zip . In the Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library and at the top of the drop-down list, select the "Add .ZIP Library" option.



The system will prompt you to select the library to be added , as shown below, navigate to the path location of the saved RFID.zip file on your computer (Project 21 RFID Access Control Experiment\RFID.zip) and open it .



Open the Sketch > Include Library menu. You should now see Libraries at the bottom of the drop-down menu. It's ready to use in your sketches .

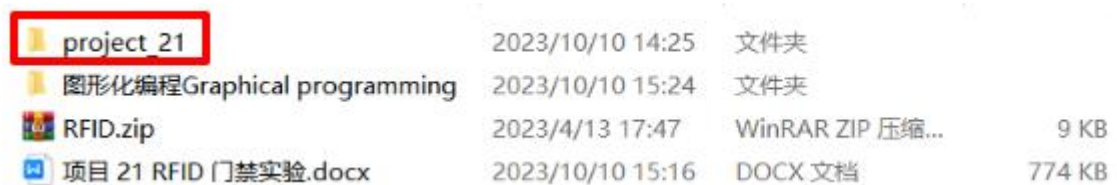


Also confirm that the infrared remote control library file Servo.zip has been successfully added. If it has not been added, please go back to item 19 to see how to add the library.

project19	2023/10/9 15:50	文件夹	
Infrared_control_servo.mp	2023/9/27 10:18	MP 文件	166 KB
Servo.zip	2022/6/15 16:16	WinRAR ZIP 压缩文件	20 KB
项目 19 红外遥控舵机.docx	2023/10/9 17:21	DOCX 文档	577 KB

5. Download Arduino code

Open the project Arduino code file (path: Project 21 RFID Access Control Experiment\project21\project21.ino)



Connect the main control board to the computer using USB, select the board type as Nano, select the newly displayed COM number, click "Download" to start compiling and downloading the program to the main control board.

Code analysis:

```

2  #include <SPI.h>           //SPI通信    SPI communication
3  #include <RFID.h>          //添加读卡器库    Add a card reader library
4  #include <Servo.h>         //添加舵机库    Add servo library
5
6  RFID rfid(10, 9);          //读卡器SDA连接引脚10, 读卡器RST连接引脚9    D10:pin of tag reader SDA. D9:pin of tag reader RST
7  unsigned char str[MAX_LEN]; //字符数    MAX_LEN is 16: size of the array    eg: [42,173,104,179]    101010,10101101,1101000,10110011
8
9  String accessGranted [2] = {"210101368113", "31215123949"}; //用于授权访问的RFID序列号    RFID serial numbers to grant access to
10 int accessGrantedSize = 2; //序列号的数量    The number of serial numbers
11
12 Servo lockServo;           //门锁装置舵机    Servo for locking mechanism
13 int lockPos = 10;          //锁定位置角度    Locked position limit
14 int unlockPos = 90;        //无锁位置角度    Unlocked position limit
15 boolean locked = true;

```



```

17 void setup()
18 {
19     Serial.begin(9600);           //串行监视器只需要获取标签ID号和用于故障排除    Serial monitor
20     SPI.begin();                 //启动与reader的SPI通信    Start SPI communication with reader
21     rfid.init();                 //初始化    initialization
22     lockServo.attach(A1);
23     lockServo.write(lockPos);     //将伺服移至锁定位置    Move servo into locked position
24     Serial.println("Place card/tag near reader...");
25 }

```

```

27 void loop()
28 {
29     if (rfid.findCard(PICC_REQIDL, str) == MI_OK) //等待标签放置在读卡器附近    Wait for a tag to be placed near the reader
30     {
31         Serial.println("Card found");
32         String temp = "";           //存储读取的RFID编号的临时变量    Temporary variable to store the read RFID number
33         if (rfid.anticoll(str) == MI_OK) //防碰撞检测，读取标签序列号    Anti-collision detection, read tag serial number
34         {
35             Serial.print("The card's ID number is : ");
36             for (int i = 0; i < 4; i++) //记录并显示标签序列号    Record and display the tag serial number
37             {
38                 Serial.println(str[i]); //10 1010,1010 1101,110 1000,1011 0011    高、低四位转换十进制：2 10,10 13,6 8,11 3
39                 temp = temp + (0x0F & (str[i] >> 4)); //右移4位后位与运算取高四位再与temp拼接成新的字符串    Shift 4 bits to the right and
40                 temp = temp + (0x0F & str[i]); //直接位与运算取低四位再与temp拼接成新的字符串    The direct AND takes the lower four
41             }
42             Serial.println(temp);
43             checkAccess(temp); //检查识别的标记是否为允许打开的标记    Check if the identified tag is an allowed to
44         }
45         rfid.selectTag(str); //锁卡为防止冗余读，删除线条将使素描卡连续读    Lock card to prevent a redundant read,
46     }
47     rfid.halt();

```



```

50 void checkAccess (String temp)           //函数，检查已识别的标记是否已注册以允许访问  Function to check if an identified tag is
51 {
52     boolean granted = false;
53     for (int i=0; i <= (accessGrantedSize-1); i++) //遍历数组中注册的所有标签ID号  Runs through all tag ID numbers registered in the arra
54     {
55         if(accessGranted[i] == temp)         //如果找到标记，则打开/关闭锁  If a tag is found then open/close the lock
56         {
57             Serial.println ("Access Granted");
58             granted = true;
59             if (locked == true)               //如果锁是关着的，就打开它  If the lock is closed then open it
60             {
61                 lockServo.write(unlockPos);
62                 locked = false;
63             }
64             else if (locked == false)         //如果锁是开着的，就关上它  If the lock is open then close it
65             {
66                 lockServo.write(lockPos);
67                 locked = true;
68             }
69         }
70     }
71     if (granted == false)                   //如果没有找到标签  If the tag is not found
72     {
73         Serial.println ("Access Denied");

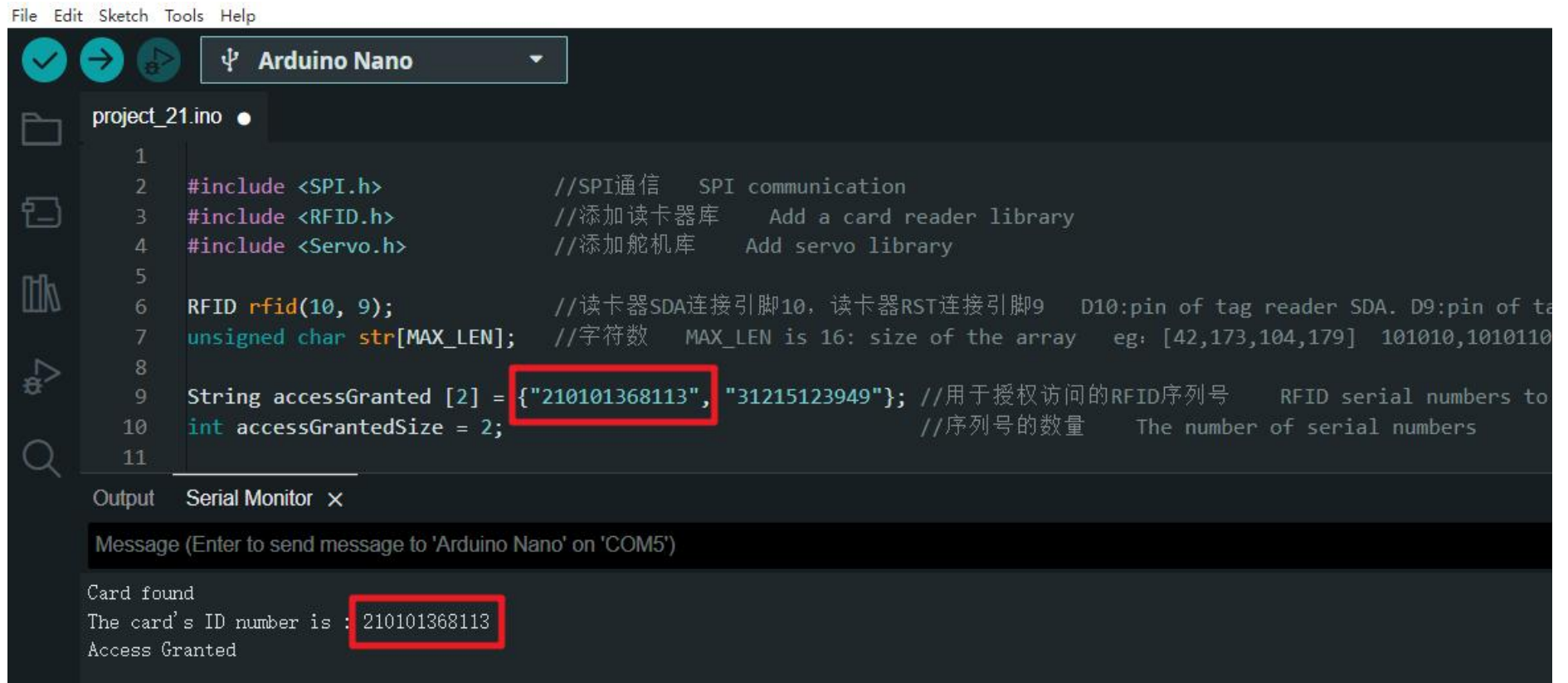
```

Project effect:

After starting the project, after the first detection on the access control card reader, the access control serial number is obtained on the serial port monitor (the serial number obtained by each person will be different, such as "210101368113"), replace the serial number into the code, **and** save code and then re-burn the code. When the same access control serial

number is detected again, the door servo will act. Testing once opens the door, testing again will close it.

There are two elements in the serial number array, corresponding to two "access keys", and each "access key" corresponds to a series of serial numbers.



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar shows icons for checking, running, and uploading code, along with a dropdown menu for the board, currently set to 'Arduino Nano'. The file explorer on the left shows 'project_21.ino'. The main editor displays the code for 'project_21.ino' with line numbers 1 through 11. The code includes headers for SPI, RFID, and Servo, and defines an array of access keys. The serial monitor at the bottom shows the output of the code, including the message 'Card found', the ID number '210101368113', and 'Access Granted'. Red boxes highlight the first access key in the code and the corresponding ID number in the serial monitor output.

```
1
2 #include <SPI.h>           //SPI通信   SPI communication
3 #include <RFID.h>          //添加读卡器库   Add a card reader library
4 #include <Servo.h>         //添加舵机库   Add servo library
5
6 RFID rfid(10, 9);          //读卡器SDA连接引脚10, 读卡器RST连接引脚9   D10:pin of tag reader SDA. D9:pin of ta
7 unsigned char str[MAX_LEN]; //字符数   MAX_LEN is 16: size of the array   eg: [42,173,104,179] 101010,1010110
8
9 String accessGranted [2] = {"210101368113", "31215123949"}; //用于授权访问的RFID序列号   RFID serial numbers to
10 int accessGrantedSize = 2; //序列号的数量   The number of serial numbers
11
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Nano' on 'COM5')

Card found
The card's ID number is : 210101368113
Access Granted

6. Download Mind+ graphical code

Open the project Mind+ code file (path: Project 21 RFID access control experiment\Graphical programming\RFID_access_control.mp)

project 21	2023/10/10 14:25	文件夹	
图形化编程Graphical programming	2023/10/10 15:24	文件夹	
RFID.zip	2023/4/13 17:47	WinRAR ZIP 压缩...	9 KB
项目 21 RFID 门禁实验.docx	2023/10/10 15:16	DOCX 文档	774 KB
RFID_access_control.mp	2023/9/27 10:34	MP 文件	
zhiyi-mfrc522-thirdex-V0.0.2.mpext	2023/10/11 18:30	MPEXT 文件	

Connect the main control board to the computer with a USB cable and select the newly appeared CH340 serial port COM number. Click "Upload to Device" to complete the code upload.

Programming analysis:

In addition to using the servo library (please refer to Project 19 for how to add the servo library), this project also uses a third-party library. The code in the attachment has integrated the library. When you need to customize a new access control system graphical code file you need to add the user library first. The following is the process of adding a library:

Click "Extend" and then select the "User Library" option and click "Import User Library"

← 返回选择用户库

主控板套件扩展板传感器执行器通信模块显示器功能模块网络服务用户库

注意：用户库中的模块由Mind+爱好者制作，[点击这里](#)查看[开发教程](#)以及[用户库列表](#)

未加载：

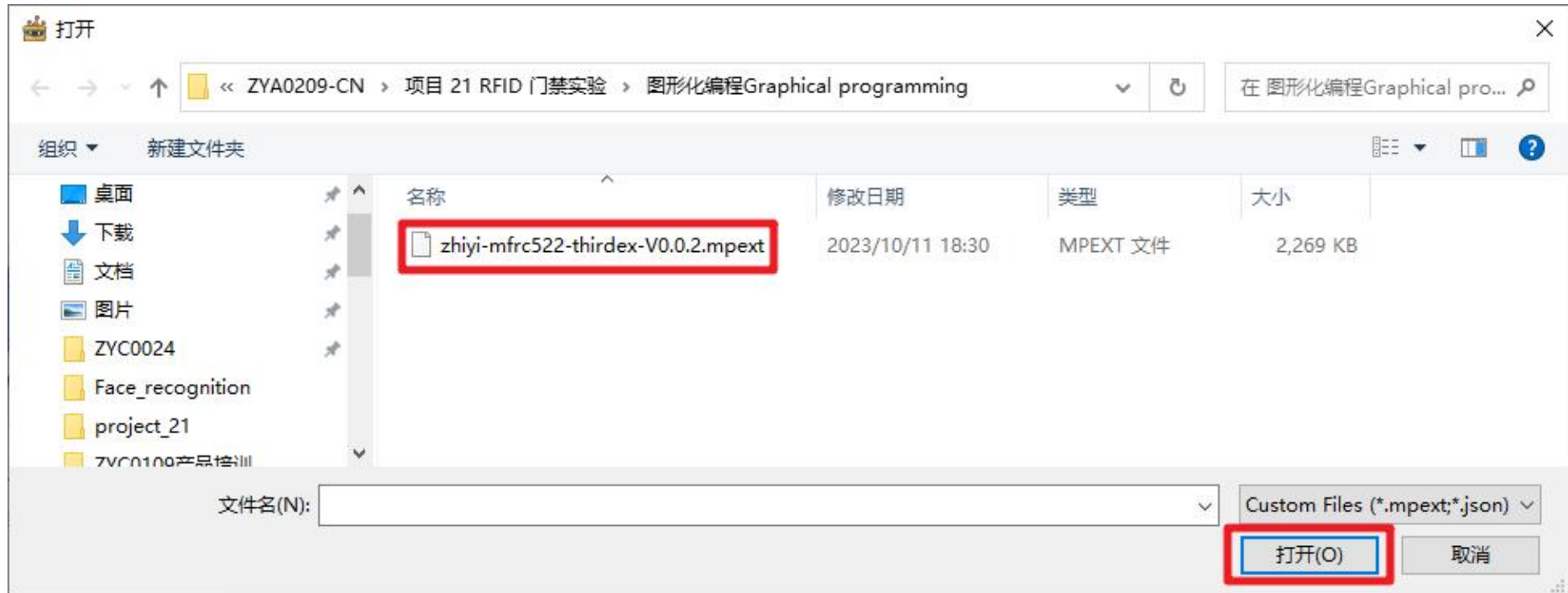
不可用

esp8266 wifi热点
基于esp8266的AP模式下WiFi热
点连接数据交互



导入用户库...
打开你计算机上的用户库文件

The system will pop up a window prompting you to select a locally saved library file:



Click on the added library to complete the import. When you return to the home page, you can see "Executor" and "User Library"

主控板

套件

扩展板

传感器

执行器

通信模块

显示器

功能模块

网络服务

用户库

搜索或输入项目网址...

注意：用户库中的模块由Mind+爱好者制作，[点击这里](#)查看开发教程以及用户库列表

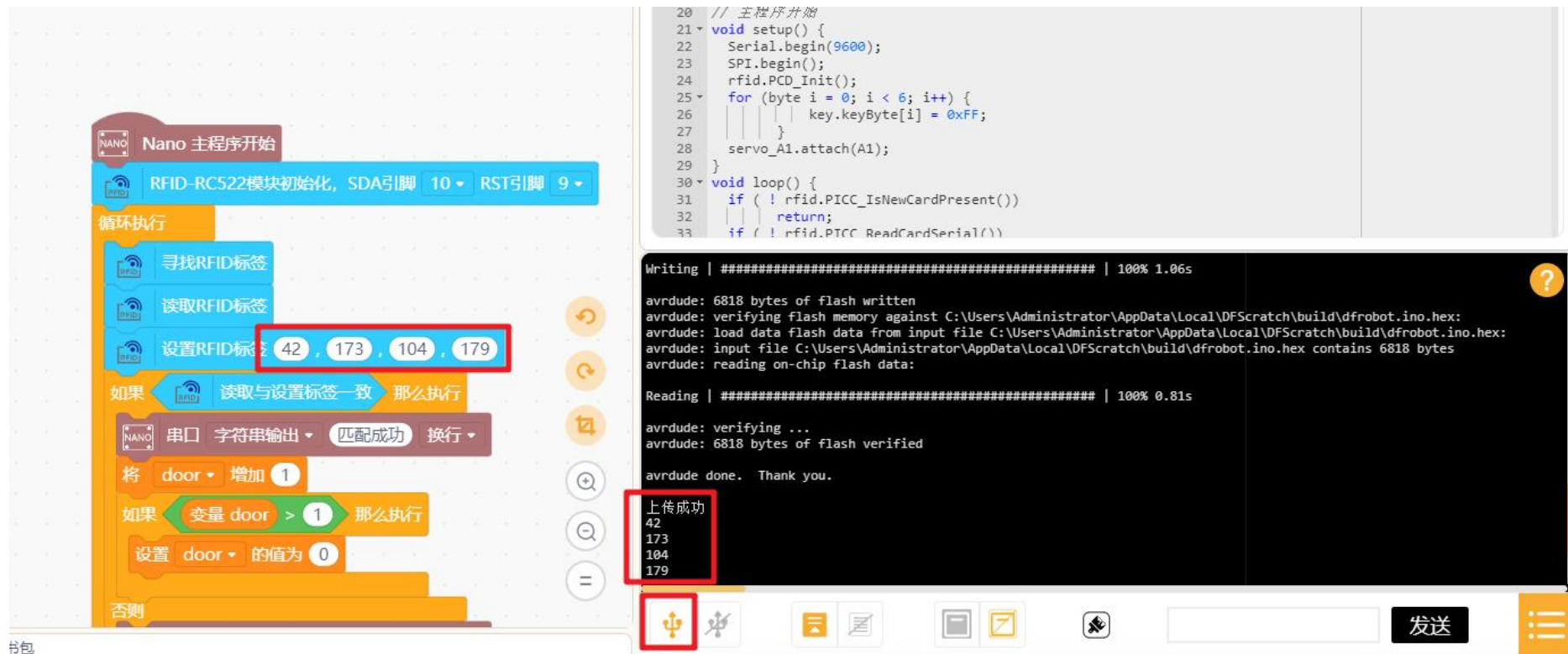
已加载：



RFID-RC522模块
可以简单的读取门禁卡。



After the first burning is successful, read the card on the serial monitor card reader to obtain the serial number, and replace the serial number into the code (for example: 42.173.104.179)



The image shows the Arduino IDE interface with a block-based code editor on the left and a C++ code editor on the right. The block editor shows a program that initializes an RFID module, searches for a tag, reads its serial number (42, 173, 104, 179), and checks if it matches a stored value. The C++ code on the right implements the same logic. The bottom right shows the serial monitor output, which displays the serial number 42, 173, 104, 179.

Block-based code (left):

- Nano 主程序开始
- RFID-RC522模块初始化, SDA引脚 10, RST引脚 9
- 循环执行
 - 寻找RFID标签
 - 读取RFID标签
 - 设置RFID标签: 42, 173, 104, 179
 - 如果 读取与设置标签一致 那么执行
 - 串口 字符串输出 匹配成功 换行
 - 将 door 增加 1
 - 如果 变量 door > 1 那么执行
 - 设置 door 的值为 0
 - 否则

C++ code (right):

```

20 // 主程序开始
21 void setup() {
22   Serial.begin(9600);
23   SPI.begin();
24   rfid.PCD_Init();
25   for (byte i = 0; i < 6; i++) {
26     key.keyByte[i] = 0xFF;
27   }
28   servo_A1.attach(A1);
29 }
30 void loop() {
31   if (!rfid.PICC_IsNewCardPresent())
32     return;
33   if (!rfid.PICC_ReadCardSerial())

```

Serial Monitor Output (bottom right):

```

Writing | ##### | 100% 1.06s
avrdude: 6818 bytes of flash written
avrdude: verifying flash memory against C:\Users\Administrator\AppData\Local\DFScratch\build\dfrobot.ino.hex:
avrdude: load data flash data from input file C:\Users\Administrator\AppData\Local\DFScratch\build\dfrobot.ino.hex:
avrdude: input file C:\Users\Administrator\AppData\Local\DFScratch\build\dfrobot.ino.hex contains 6818 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 0.81s
avrdude: verifying ...
avrdude: 6818 bytes of flash verified
avrdude done. Thank you.

上传成功
42
173
104
179

```

Burn the code again, and the servo will start when the card with the same serial number is read again. The servo opens the door the first time and closes it the next time.

Total code:

