

Part 4

Lesson

7

RFID Time Clock

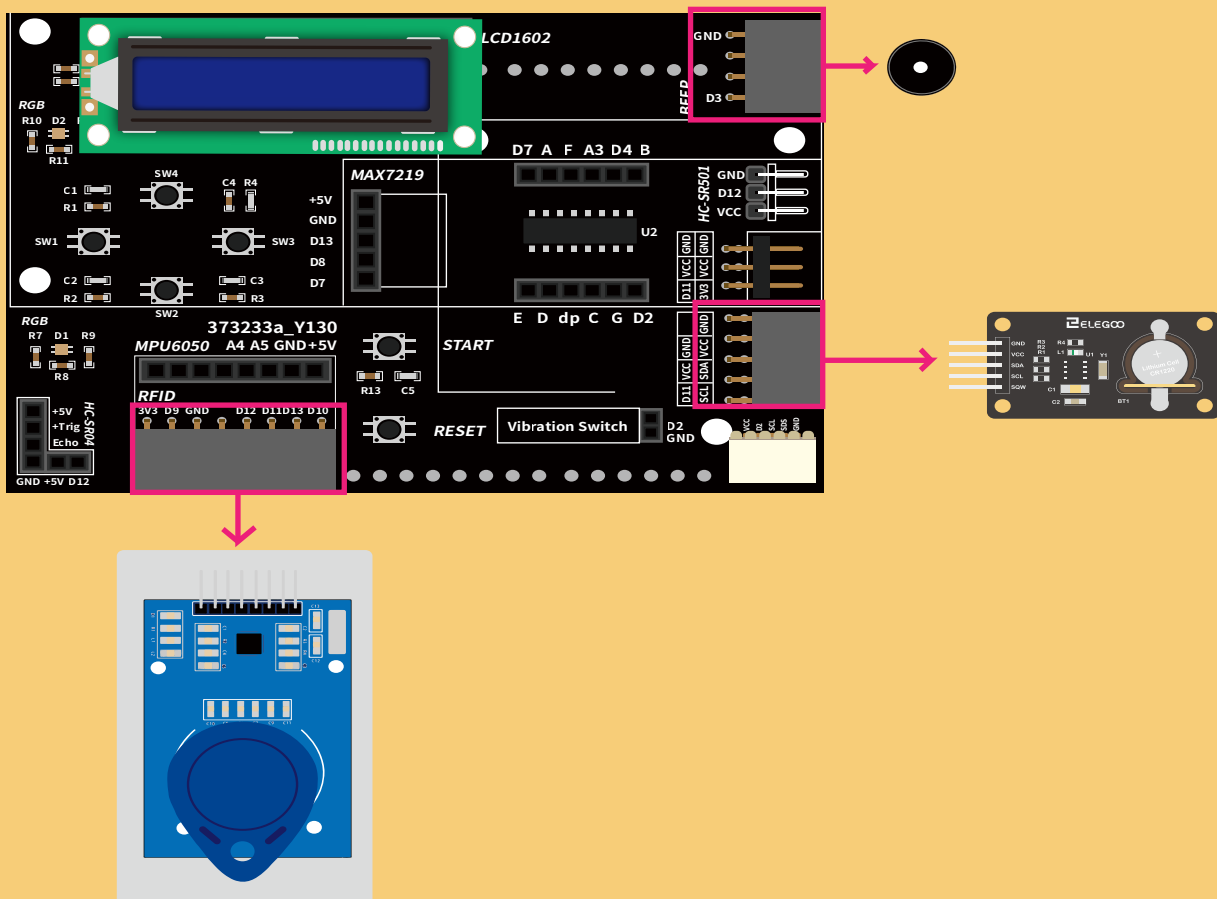
Summary:

Through the study of the program of RFID Time Clock, we will strengthen our understanding of the practical application of LCD1602 display module, RC522 RF card module and buzzer, cultivate our programming ideas, and appreciate the charm of Arduino.

Component Required:

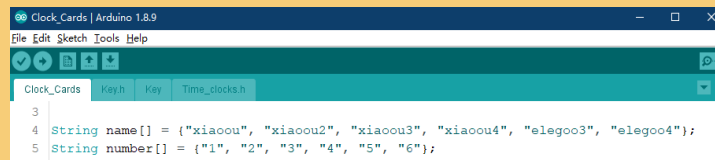
- (1) x Arduino Uno
- (1) x RFID
- (1) x Passive Buzzer
- (1) x ALL IN ONE Sensor Shield
- (1) x LCD1602
- (1) x Clock Module

Wiring Diagram:



Realization of RFID Time Clock

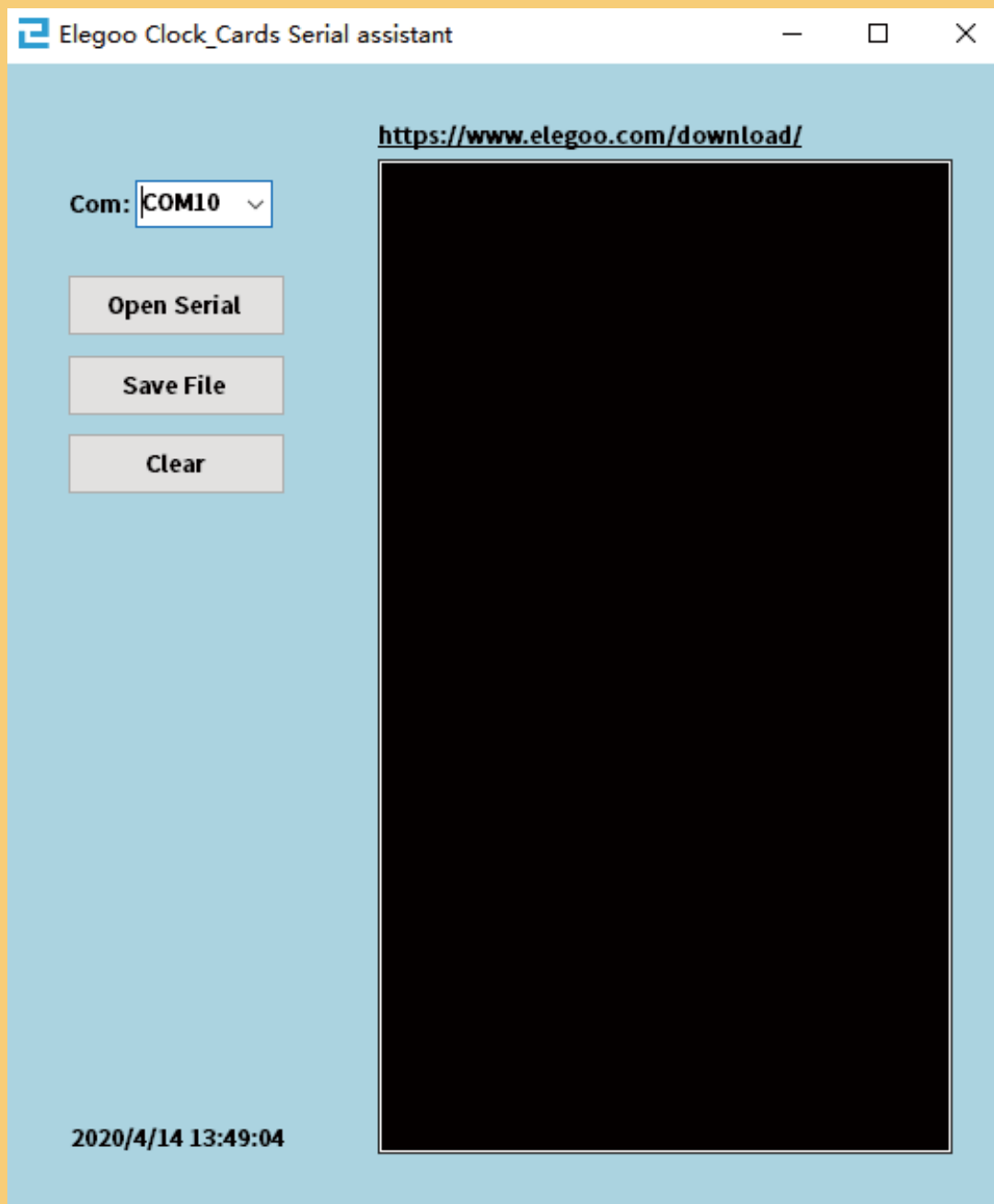
First, enter the employee's name and employee identification number and then download the program.



The screenshot shows the Arduino IDE interface with the 'Clock_Cards' sketch loaded. The code defines arrays for employee names and IDs. The 'Time_clocks.h' file is also visible in the file explorer.

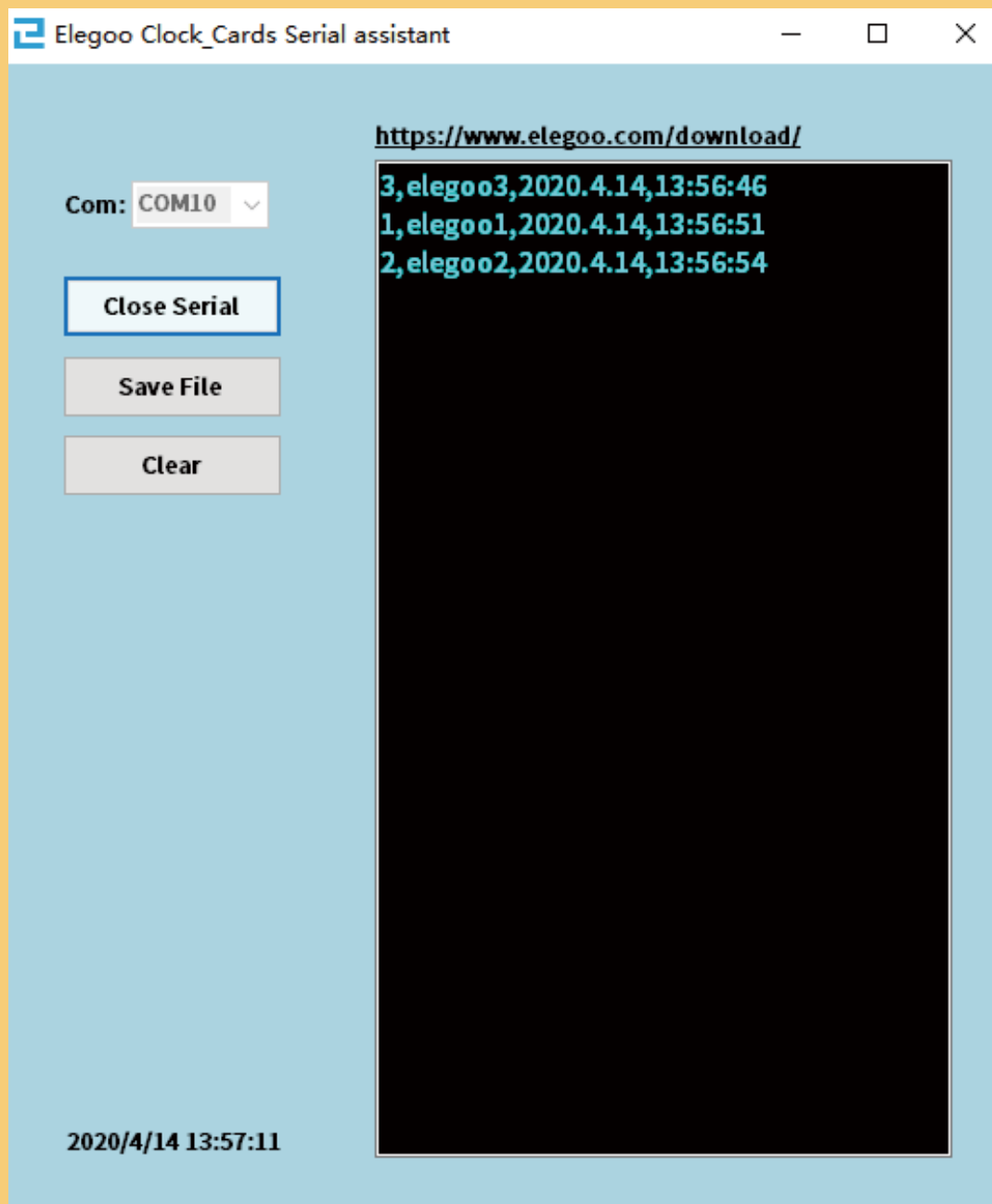
```
3  
4 String name[] = {"xiaou", "xiaou2", "xiaou3", "xiaou4", "elegoo3", "elegoo4"};  
5 String number[] = {"1", "2", "3", "4", "5", "6"};
```

After downloading the program, do not disconnect the USB cable and then open the supporting software and press “Open Serial”.

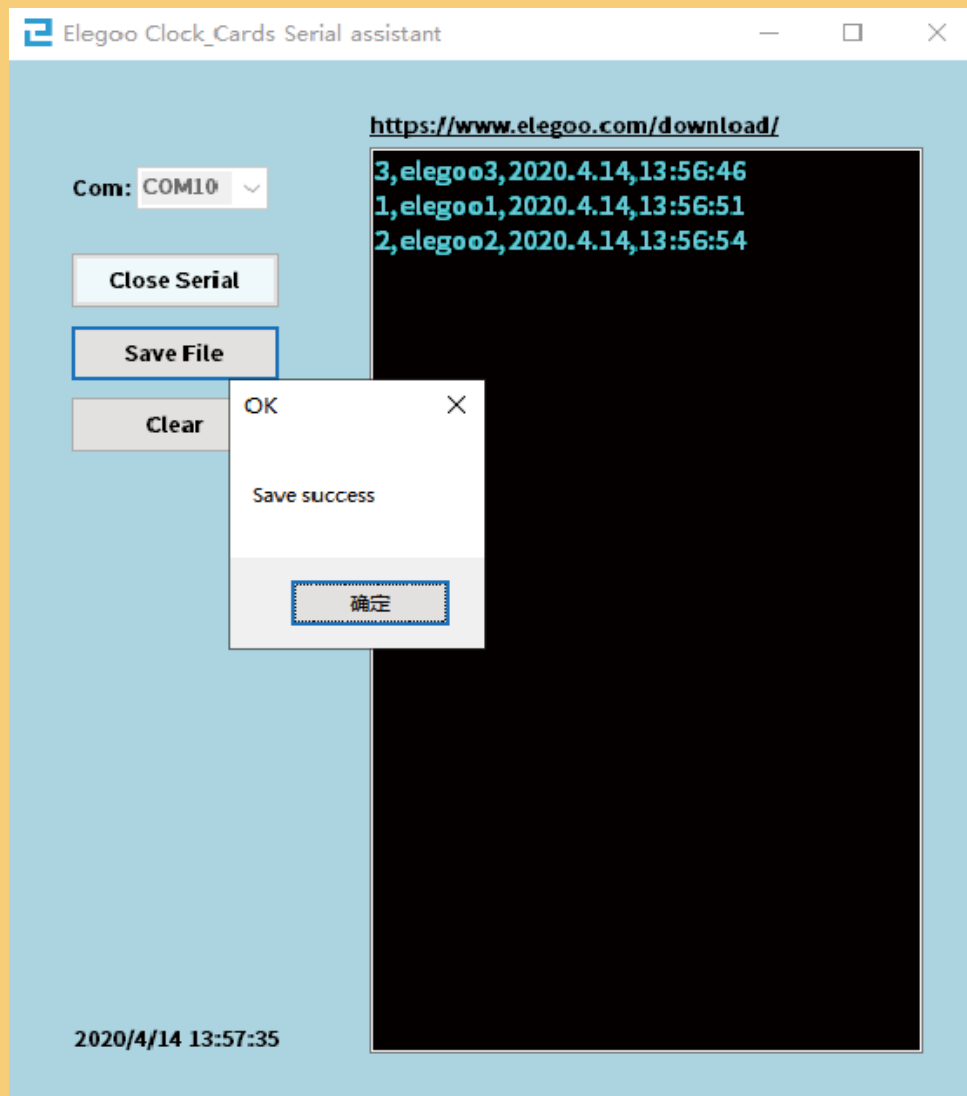




Press the key in the initial interface to enter the input mode, and then use the RF card to enter the information in turn. Wait for about two seconds when entering, and the prompt on the display indicates that the entry is successful.

When the input is completed, press the key again to exit the input mode. At this time, swipe with the recorded RF card, then you can obtain the name, employee identification number and swiping time of the person who swipe the card **on the software**.



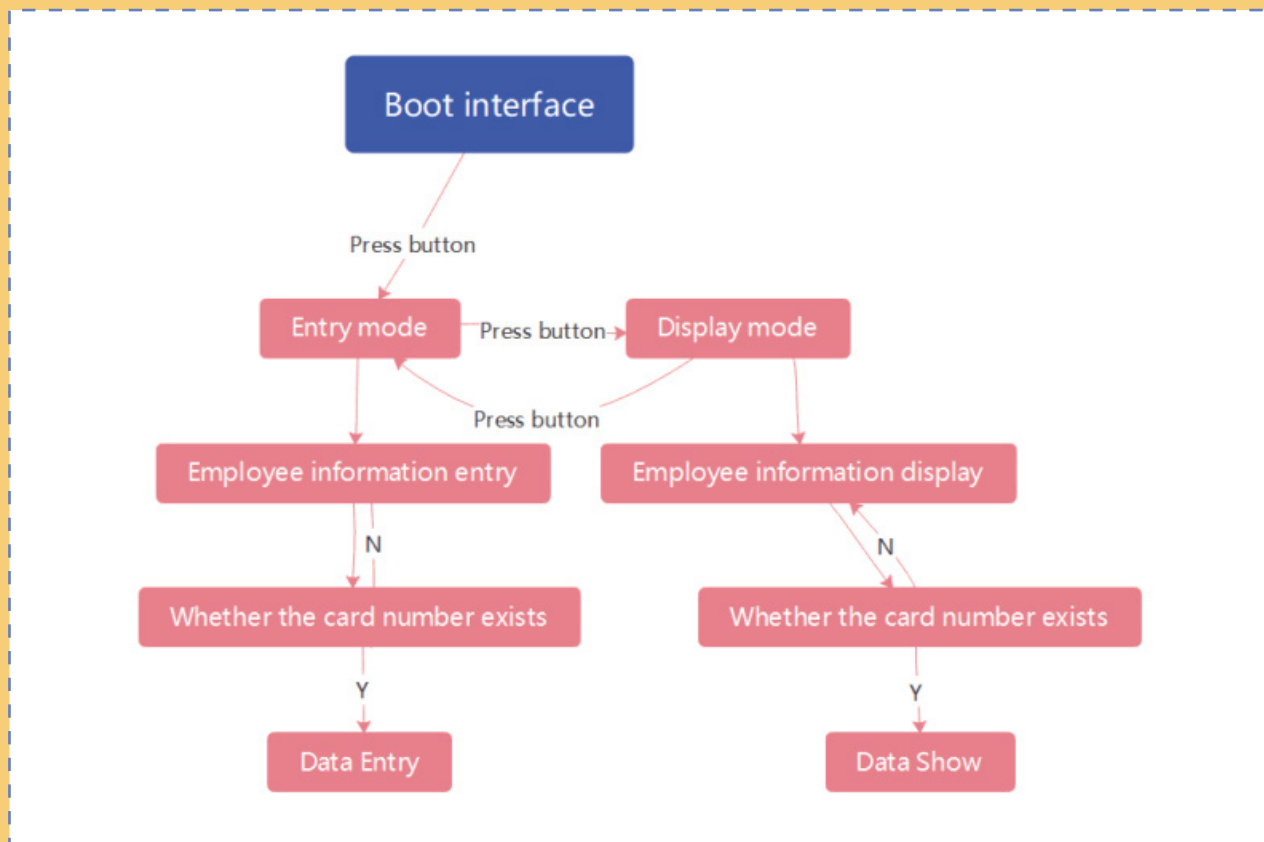
Finally, pressing “Save File” will save the display information in **CSV format**.



 DailyData.csv	2020/4/9 15:07
 MySerial.exe	2020/4/14 13:54

	A	B	C	D	E
1	Number	Name	Date	Time	
2		3 elegoo3	2020.4.14	13:56:46	
3		1 elegoo1	2020.4.14	13:56:51	
4		2 elegoo2	2020.4.14	13:56:54	
5					

Implementation schematic



Creation process:

- 1. First, add the library files that need to be used and the objects that need to be defined for calling the corresponding library functions.

```
//In Time_clocks.h
```

```
#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal.h>
#include <SPI.h>
#include <MFRC522.h>
#include "pitches.h"
```

```
#define RST_PIN  A1
#define SS_PIN  10
```

```
void LCD_ShowDate(RTCDateTime dt);
```

```
DS3231 clock;
RTCDateTime dt;
LiquidCrystal lcd(A3, 2, 4, 7, 8, A2);
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

```
//In Clock_Cards

#include "Time_clocks.h"
#include "PinChangeInt.h"

void setup() {
  lcd.begin(16, 2);
  Serial.begin(9600);
  clock.begin();
  clock.setDateTime(__DATE__, __TIME__);
  SPI.begin();    // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card
  pinMode(A0, INPUT_PULLUP);
  attachPinChangeInterrupt(A0, button_choose_interrupt, FALLING );
}
```

■ 2. Boot interface programming.

```
//In Clock_Cards

void loop() {
  Start_interface();
}

void Start_interface()
{
  if (flag_start == 1)
  {
    lcd.setCursor(3, 0);
    lcd.print("Starting up..");
    delay(2000);
    flag_start = 0;
    lcd.clear();
  }
}
```

- 3. Key definition, key debounce, key event status change, and finally clear the screen every time the event changes.

```
//In Key.h

#ifndef KEY_H
#define KEY_H

int time_button_choose=0;
int last_time_button_choose=0;

#define ELIMINAT_THE_JITTER_OF_BUTTON_CHOOSE
(200 < (abs(time_button_choose - last_time_button_choose)))

#endif
```

```
//In Key

#include "Key.h"

void button_choose_interrupt ()
{
    time_button_choose = millis();

    if(ELIMINAT_THE_JITTER_OF_BUTTON_CHOOSE)
    {
        last_time_button_choose = time_button_choose;
        if(mode==DISPLAY_MODE)
        {
            mode=INPUT_MODE;
            lcd.clear();
        }
        else if(mode==INPUT_MODE)
        {
            mode=DISPLAY_MODE;
            lcd.clear();
        }
    }
}
```


- 4. Switch the mode according to the key status sign.

```
//In Clock_Cards

void Mode_Chose(int mode)
{
    switch (mode)
    {
        case DISPLAY_MODE:
            .....
        case INPUT_MODE:
            .....
        default: break;
    }
}
```

- 5. Create an array to store employee information.

```
//In Clock_Cards

String name[]={"xiaou", "xiaou2", "xiaou3", "xiaou4", "elegoo3", "elegoo4"};
String number[]={"1", "2", "3", "4", "5", "6"};
```

- 6. Under input mode, a linked list of corresponding sizes is created according to the array size for information storage.

```
//In Clock_Cards

void Mode_Chose(int mode)
{
    if (cnt < (sizeof(name) / sizeof(String)))
    {
        Data_Input(name[cnt], number[cnt]);
    }
    delay(100); break;
}
```

- 7. Determine whether the card number exists, create a linked list to store the information if it exists, and skip it if it does not exist.

```
//In Clock_Cards

void Data_Input(String name, String number)
```

The advantages and disadvantages of array and linked list:

Array list:

Advantages: Easy to use, query efficiency is higher than linked list, memory as a continuous area.

Disadvantages: Fixed size, not suitable for dynamic storage, not convenient for dynamic addition.

Linked list:

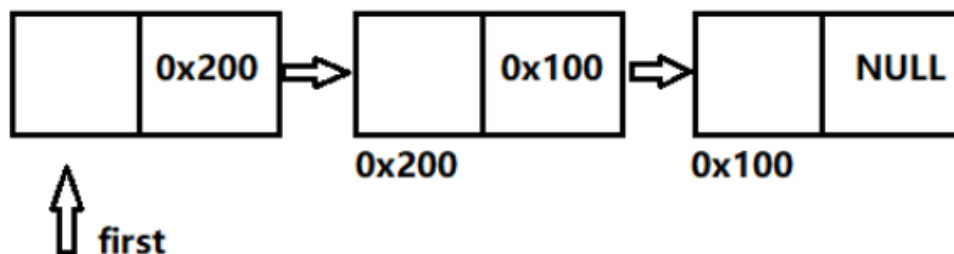
Advantages: Dynamic addition and deletion, variable size.

Disadvantages: It can only be accessed through sequential pointers, and the query efficiency is low.

Linked list: In essence, it is a node to establish a connection with a pointer.

struct node

```
{  
    ElemType data; //Data field  
    struct node *next; //Pointer field  
};
```



The pointer next stores the address of the next node.

- 8.Display mode programming: Traverse the linked list to read the data according to the card number.

```
//In Clock_Cards
```

```
void Data_Display()
```