RFID Introduction

RFID, also known as radio frequency identification, is a communication technology. It can identify specific targets by radio signals and read and write data without establishing mechanical or optical contact between the recognition system and specific targets. Rf is generally microwave, 1-100ghz, suitable for short distance identification communication.

RFID readers are also divided into mobile and fixed types. At present, RFID technology is widely used, such as library, access control system, food safety traceability, etc.

Components:

Transponder: it consists of an antenna, a coupling element and a chip. Typically, the sender transponder looks like a tag, and each tag has a unique electronic code that identifies the object by attaching it to it.

Card reader: a device consisting of an antenna, a coupling element, and a chip that reads (and sometimes writes) tag information and can be designed as a handheld RFID reader or a stationary reader.

Working Principle

The basic principle of RFID technology is not complicated: the tag will receive a reader sent when it enters the magnetic field of the RF signal, and then will get the energy stored in the chip (passive tag) of information to send out the induced current, send a particular frequency or active tags, by the reader and decoding information, processing and sent to the central information system.

A complete RFID system consists of three parts: card reader, electronic tag (i.e. transponder) and application system. The principle is that the reader sends out a specific frequency of radio wave energy to

drive the circuit to send out internal data, and then the reader receives the interrupt data in an orderly manner and sends it to the application for processing accordingly.

From the point of view of the communication and energy induction between electronic tags, there are inductive coupling and backscatter coupling. Low frequency RFID usually USES the first type, while high frequency RFID USES the second type.

Depending on the structure and technology, the reader can be a read or read/write device, which is the control and processing center of the RFID system information. Card readers are usually composed of coupling module, transceiver module, control module and interface unit. In general, the half-duplex communication mode is used to exchange information between the reader and the transponder, and the reader provides the power and timing of the passive transponder through coupling.

Experiment Purpose

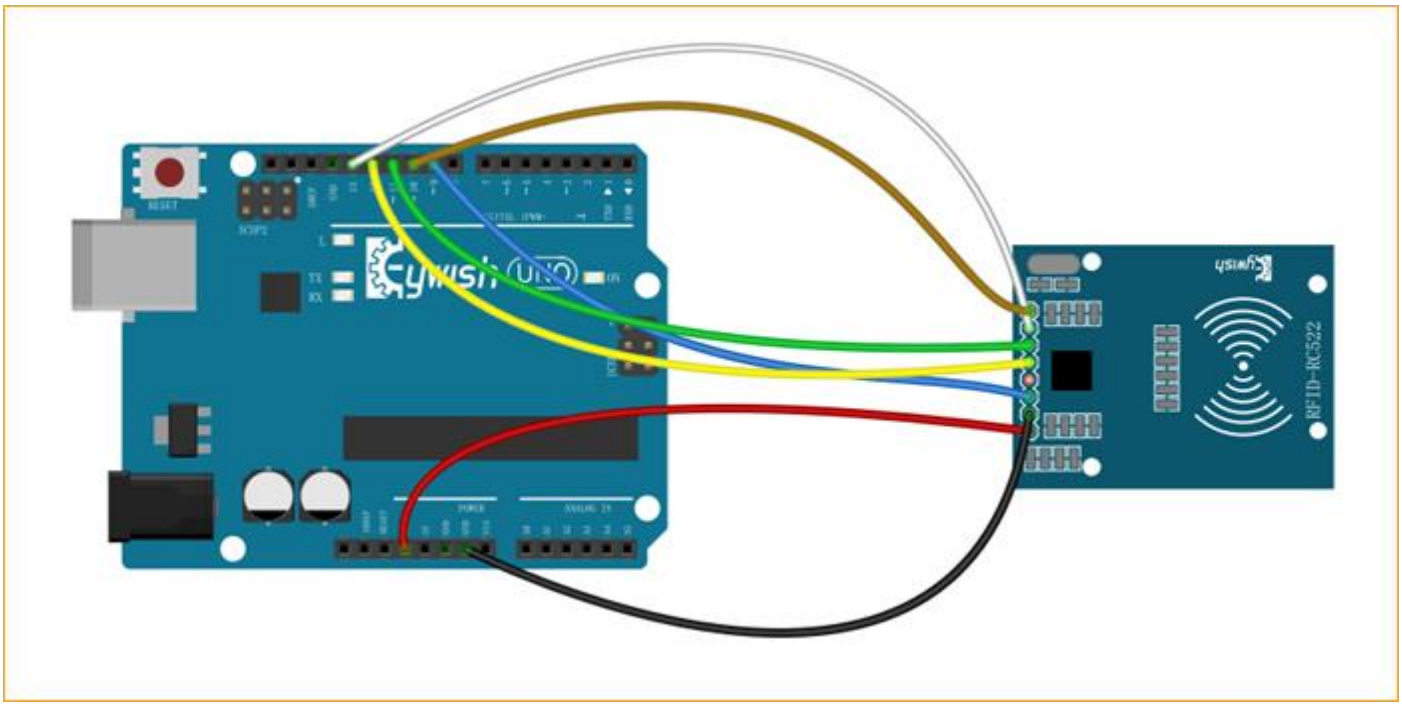
Reading through the card reader, if the tag matches what we use, then the buzzer beeps with a low frequency. Otherwise, a hasty alarm is issued.

Component List

- Keywish Arduino UNO R3 mainboard
- Breadboard
- USB cable
- RFID suite *1
- Passive buzzer *1
- Jumper wires

Wiring of Circuit

RFID	Arduino Uno R3
SDA	10
SCK	13
MOSI	11
MISO	12
IRQ	NALL
GND	GND
RST	9
VCC	3.3V



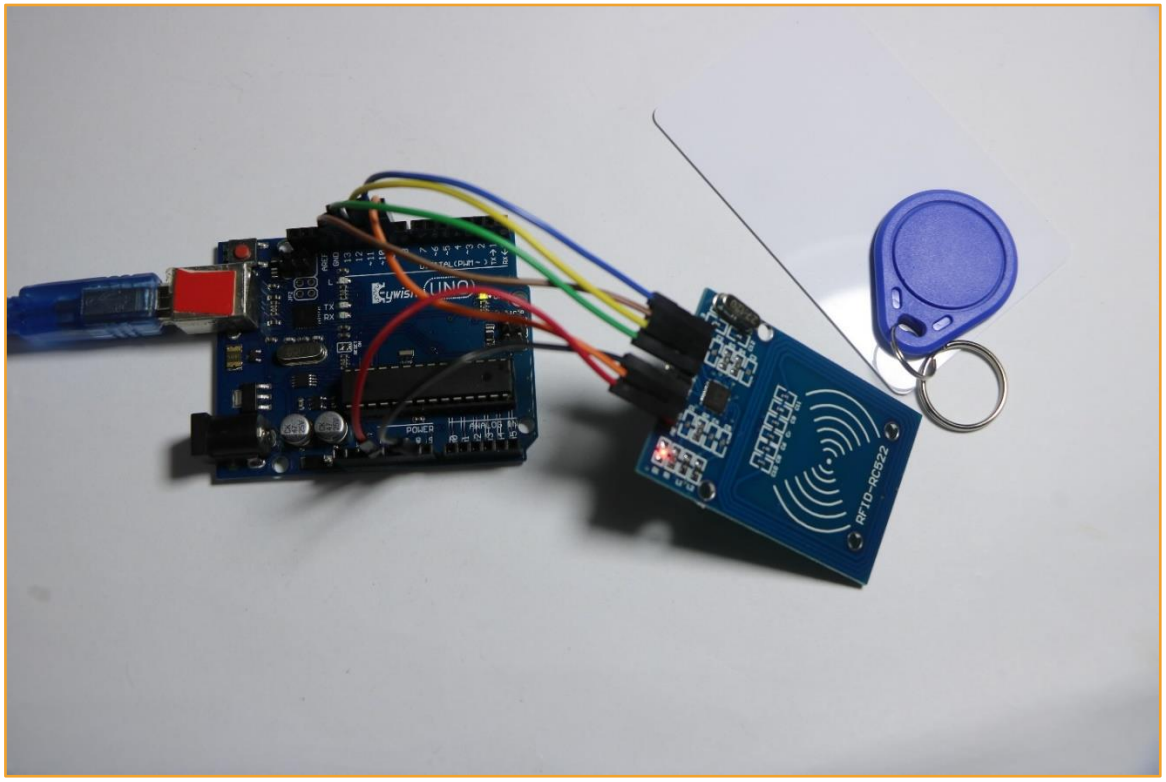
Code

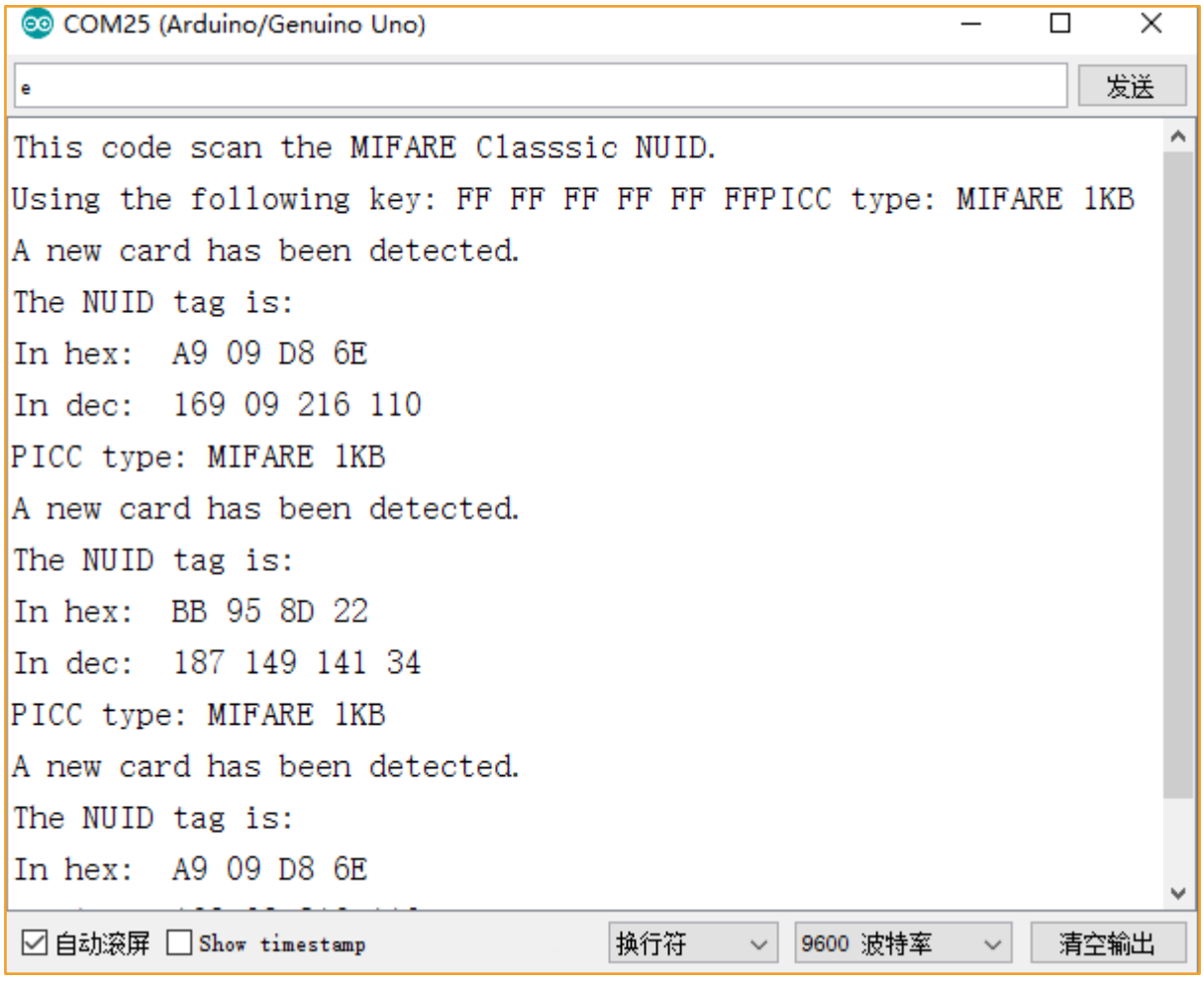
```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9


MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class

MFRC522::MIFARE_Key key;

// Init array that will store new NUID
byte nuidPICC[4];

void setup() {
  Serial.begin(9600);

  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522

  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }

  Serial.println(F("This code scan the MIFARE Classsic NUID."));
  Serial.print(F("Using the following key:"));
  printHex(key.keyByte, MFRC522::MF_KEY_SIZE);
}

void loop() {

  // Reset the loop if no new card present on the sensor/reader. This saves the entire
process when idle.
  if ( ! rfid.PICC_IsNewCardPresent())
    return;

  // Verify if the NUID has been readed
  if ( ! rfid.PICC_ReadCardSerial())
    return;
```

```
Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  Serial.println(rfid.PICC_GetTypeName(piccType));

  // Check is the PICC of Classic MIFARE type
  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
    piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
    piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE Classic."));
    return;
  }

  if (rfid.uid.uidByte[0] != nuidPICC[0] ||
    rfid.uid.uidByte[1] != nuidPICC[1] ||
    rfid.uid.uidByte[2] != nuidPICC[2] ||
    rfid.uid.uidByte[3] != nuidPICC[3] ) {
    Serial.println(F("A new card has been detected."));

    // Store NUID into nuidPICC array
    for (byte i = 0; i < 4; i++) {
      nuidPICC[i] = rfid.uid.uidByte[i];
    }

    Serial.println(F("The NUID tag is:"));
    Serial.print(F("In hex: "));
    printHex(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
    Serial.print(F("In dec: "));
    printDec(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
  }
  else Serial.println(F("Card read previously."));

  // Halt PICC
  rfid.PICC_HaltA();

  // Stop encryption on PCD
  rfid.PCD_StopCrypto1();
}


/**
 * Helper routine to dump a byte array as hex values to Serial.
```

```
*/
void printHex(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    Serial.print(buffer[i], HEX);
  }
}

/**
 * Helper routine to dump a byte array as dec values to Serial.
 */
void printDec(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    Serial.print(buffer[i], DEC);
  }
}
```

Experiment Result





MixlyBlock programming program

The program prepared by mixly is shown in the figure below: