/\*\*  
 \* ----------------------------------------------------------------------------  
 \* This is a MFRC522 library example; see https://github.com/miguelbalboa/rfid  
 \* for further details and other examples.  
 \*  
 \* NOTE: The library file MFRC522.h has a lot of useful info. Please read it.  
 \*  
 \* Released into the public domain.  
 \* ----------------------------------------------------------------------------  
 \* This sample shows how to read and write data blocks on a MIFARE Classic PICC  
 \* (= card/tag).  
 \*  
 \* BEWARE: Data will be written to the PICC, in sector #1 (blocks #4 to #7).  
 \*  
 \*  
 \* Typical pin layout used:  
 \* -----------------------------------------------------------------------------------------  
 \*             MFRC522      Arduino       Arduino   Arduino    Arduino          Arduino  
 \*             Reader/PCD   Uno/101       Mega      Nano v3    Leonardo/Micro   Pro Micro  
 \* Signal      Pin          Pin           Pin       Pin        Pin              Pin  
 \* -----------------------------------------------------------------------------------------  
 \* RST/Reset   RST          9             5         D9         RESET/ICSP-5     RST  
 \* SPI SS 1    SDA(SS)      \*\* custom, take a unused pin, only HIGH/LOW required \*\*  
 \* SPI SS 2    SDA(SS)      \*\* custom, take a unused pin, only HIGH/LOW required \*\*  
 \* SPI MOSI    MOSI         11 / ICSP-4   51        D11        ICSP-4           16  
 \* SPI MISO    MISO         12 / ICSP-1   50        D12        ICSP-1           14  
 \* SPI SCK     SCK          13 / ICSP-3   52        D13        ICSP-3           15  
 \*  
 \*/  
  
#include <SPI.h>  
#include <MFRC522.h>  
  
#define RST\_PIN         9          // Configurable, see typical pin layout above  
#define RST\_PIN1         10  
#define SS\_1\_PIN        8         // Configurable, take a unused pin, only HIGH/LOW required, must be diffrent to SS 2  
#define SS\_2\_PIN        7          // Configurable, take a unused pin, only HIGH/LOW required, must be diffrent to SS 1  
  
#define NR\_OF\_READERS   2  
  
byte ssPins[] = {SS\_1\_PIN, SS\_2\_PIN};  
  
MFRC522 mfrc522[NR\_OF\_READERS];   // Create MFRC522 instance.  
  
/\*\*  
 \* Initialize.  
 \*/  
void setup() {  
  
  Serial.begin(9600); // Initialize serial communications with the PC  
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)  
  
  SPI.begin();        // Init SPI bus  
  
  //for (uint8\_t reader = 0; reader < NR\_OF\_READERS; reader++) {  
    mfrc522[0].PCD\_Init(ssPins[0], RST\_PIN); // Init each MFRC522 card  
    Serial.print(F("Reader "));  
    Serial.print(0);  
    Serial.print(F(": "));  
    mfrc522[0].PCD\_DumpVersionToSerial();  
  
    mfrc522[1].PCD\_Init(ssPins[1], RST\_PIN1); // Init each MFRC522 card  
    Serial.print(F("Reader "));  
    Serial.print(1);  
    Serial.print(F(": "));  
    mfrc522[1].PCD\_DumpVersionToSerial();      
  //}  
}  
  
/\*\*  
 \* Main loop.  
 \*/  
void loop() {  
  
  for (uint8\_t reader = 0; reader < NR\_OF\_READERS; reader++) {  
    // Look for new cards  
  
    if (mfrc522[reader].PICC\_IsNewCardPresent() && mfrc522[reader].PICC\_ReadCardSerial()) {  
      Serial.print(F("Reader "));  
      Serial.print(reader);  
      // Show some details of the PICC (that is: the tag/card)  
      Serial.print(F(": Card UID:"));  
      dump\_byte\_array(mfrc522[reader].uid.uidByte, mfrc522[reader].uid.size);  
      Serial.println();  
      Serial.print(F("PICC type: "));  
      MFRC522::PICC\_Type piccType = mfrc522[reader].PICC\_GetType(mfrc522[reader].uid.sak);  
      Serial.println(mfrc522[reader].PICC\_GetTypeName(piccType));  
  
      // Halt PICC  
      mfrc522[reader].PICC\_HaltA();  
      // Stop encryption on PCD  
      mfrc522[reader].PCD\_StopCrypto1();  
    } //if (mfrc522[reader].PICC\_IsNewC  
  } //for(uint8\_t reader  
}  
  
/\*\*  
 \* Helper routine to dump a byte array as hex values to Serial.  
 \*/  
void dump\_byte\_array(byte \*buffer, byte bufferSize) {  
  for (byte i = 0; i < bufferSize; i++) {  
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");  
    Serial.print(buffer[i], HEX);  
  }  
}