

```

#include "rfid1.h"
#include <SPI.h>
#include <SD.h>
#define readers 5

File myFile;
RFID1 rfid;

uchar serNum[5];    // raw rfid ID
char cardName[30]; // User Defined Name
byte cardID[10];   // ID pulled from SD card
char pos[10];      // Location of card scan
byte cardNum[8];   // Refined value of raw ID
int pass[5]= {0,0,0,0,0};
int empty[5]= {0,0,0,0,0};

void setup() {
  Serial.begin(9600);    //initialize the serial
  Serial.setTimeout(100000);
  Serial.print("Initializing SD card...");
  if (!SD.begin(10)) {  //Initialize the SD reader
    Serial.println("initialization failed!");
    while (1);
  }
  Serial.println("initialization done.");
}

void loop()
{
  for (int i=0;i<readers;i++)
  {
    rfidswap(i);    //Poll all readers
    if(pass[i] <= 0){
    uchar status;
    uchar str[MAX_LEN];    //Search for cards
    status = rfid.request(PICC_REQIDL, str);
    status = rfid.anticoll(str); //Returns ID to process
    if(status == MI_OK){
      memcpy(serNum, str, 5);
      for(int j=0,k=0;j<8;j++,k++){
        cardNum[j]=(0x0f & (serNum[k]>>4)),HEX; //Convert 2bit hex ID
        j++;
        cardNum[j]=(0x0f & serNum[k]),HEX; //to 1bit hex ID
      }
      dataGrab_v2(i); //Pull card info version 2
    }
  }
}

```

```

rfid.halt(); //Put reader in sleep mode
pass[i]= 5;
empty[i]=0;
}else {
  if(empty[i] == 0){
Serial.print("pos");
Serial.print(i); //helps to keep serial line from being flooded
Serial.print(" ");
Serial.print("#");
  }
  empty[i]=1;
}
}else{
  pass[i]--; //allows for a short down time given a card is slightly moved
}
}
}

```

```

void rfidswap(int i){ //initialize each reader
  if(i==0){
    rfid.begin(8, 7, 6, 4, 5, 9);
  }//rfid.begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN)
  if(i==1){
    rfid.begin(8, 7, 6, 3, 5, 9);
  }//rfid.begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN)
  if(i==2){
    rfid.begin(8, 7, 6, 2, 5, 9);
  }//rfid.begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN)
  if(i==3){
    rfid.begin(8, 7, 6, A0, 5, 9);
  }//rfid.begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN)
  if(i==4){
    rfid.begin(8, 7, 6, A1, 5, 9);
  }//rfid.begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,NSS_PIN,RST_PIN)
  rfid.init(); //initialize the RFID
}

```

```

void dataGrab_v2(int i){
  myFile = SD.open("DATA.csv", FILE_READ);
  while(myFile){ //Open data file
    memset(cardID, 0, sizeof(cardID));
    memset(pos, 0, sizeof(pos)); //Clean buffer arrays
    memset(cardName, 0, 30);
    myFile.readBytesUntil(',',cardID,30);
    myFile.readBytesUntil(',',pos,10); //Data into arrays
    myFile.readBytesUntil('\n',cardName,30);
  }
}

```

```

for(int j=0;j<8;j++){
    if('A' <= cardID[j] && cardID[j] <= 'F'){
        cardID[j]=cardID[j]-55;           //Ascii char value into decimal value
    }else if('0' <= cardID[j] && cardID[j] <= '9'){
        cardID[j]=cardID[j]-'0';
    }
}
//If cardID matches pulled ID

if((cardID[0]==cardNum[0])&&(cardID[1]==cardNum[1])&&(cardID[2]==cardNum[2])&&(cardID[3]==cardNum[3])&&(cardID[4]==cardNum[4])){
    for(int j=0;j<8;j++){
        // Serial.print(cardID[j],HEX);
    }
    // Serial.println();           //Print pulled card info
    // Serial.println(pos);
    Serial.print("pos");
    Serial.print(i);
    Serial.print(cardName);
    Serial.print("#");
    myFile.close(); //File close resets reading position to begining, remains open to
scan whole file for match
}else {
// Serial.println("pass");
// myFile.close();
}
}
}

```