






## معرفی و راه‌اندازی ماژول RFID با آردوینو



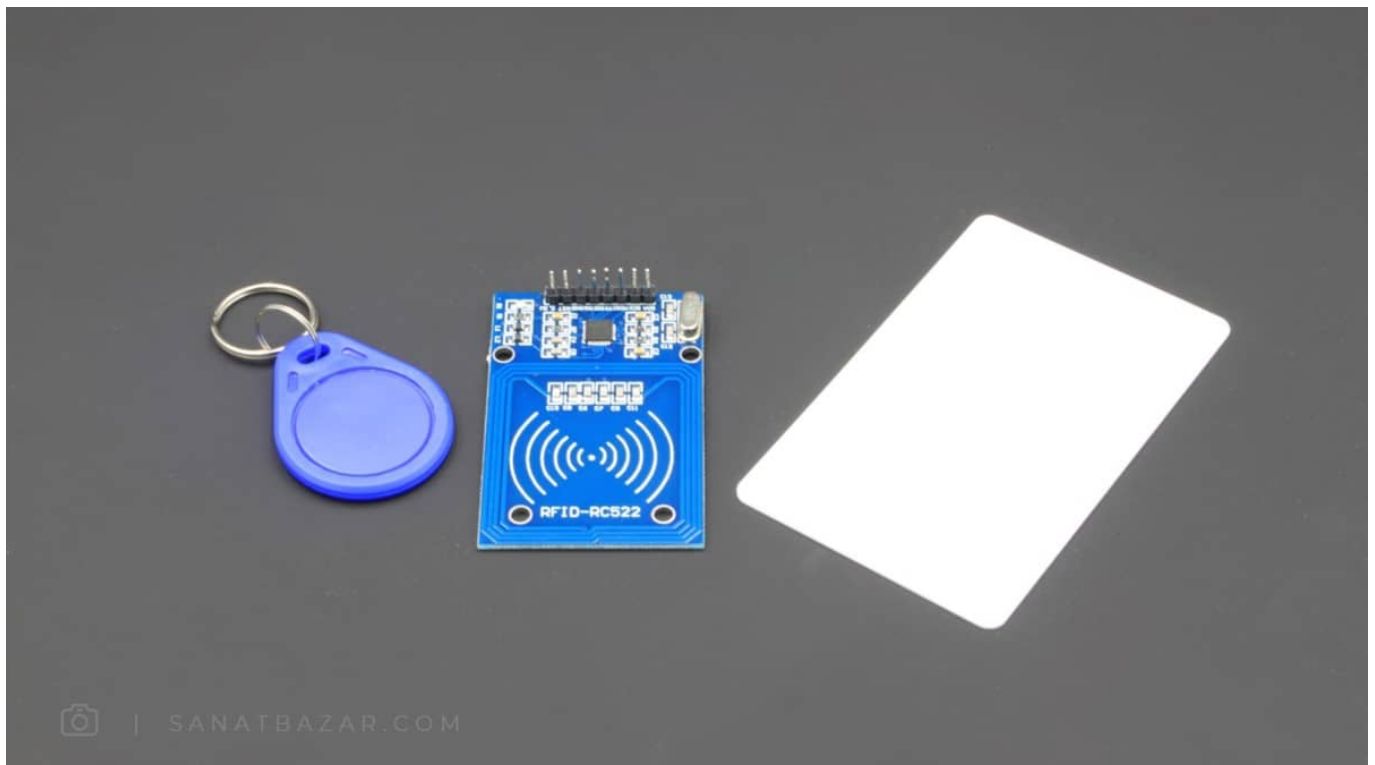
این مطلب قسمت ششم از مجموعه **آموزش جامع اینترنت اشیا با آردوینو** است. در این آموزش می‌خواهیم به معرفی ماژول RFID بپردازیم. شاید به خاطر داشته باشید که تا چند سال پیش برای استفاده از اتوبوس‌های شهری باید بلیط کاغذی به همراه می‌داشتید. بدتر از آن زمانی بود که هزینه اتوبوس و موارد مشابه به صورت نقدی پرداخت می‌شد و شما باید زمان زیادی را صرف خرید بلیط می‌کردید یا منتظر پرداخت پول می‌شدید. چه چیزی اینگونه پرداخت‌ها را در حال حاضر ساده و کم‌دردسر کرده است؟ یک فناوری ارتباطی بی‌سیم به نام RFID. ردپای RFID را می‌توانید در دستگاه‌های پرداخت هوشمند، وسایل ورزشی، دستگاه‌های ثبت ورود و خروج، دربازکن‌های هوشمند و غیره ببینید. در پایان این آموزش، خواهید توانست با استفاده از RFID یک کلید هوشمند بسازید.

## قطعات مورد نیاز

	1 عدد	<a href="#">برد آردوینو و کابل رابط</a>
	1 عدد	<a href="#">ماژول RC522</a>
	1 عدد	<a href="#">تگ RFID</a>
	1 عدد	<a href="#">بردبرد</a>
	1 بسته	<a href="#">سیم جامپر</a>

## آشنایی با نحوه کار RFID

RFID یا Radio frequency identification همان طور که از نامش مشخص است یک روش احراز هویت است که بر اساس امواج رادیویی (الکترومغناطیسی) کار می‌کند. این فناوری برای شناسایی اشیا یا اشخاص استفاده می‌شود. برای مثال از RFID می‌توان در شناسایی اجناس در یک فروشگاه، تشخیص کالاهای موجود در یک انبار، شناسایی خودرو در گیت آزادراه‌ها یا ثبت ورود و خروج افراد یک شرکت استفاده کرد. سیستم RFID از دو بخش تشکیل شده است: ریدر (Reader) و تگ (Tag). تگ به جسم شناسایی شونده متصل می‌شود. ریدر به طور پیوسته سیگنالی را منتشر می‌کند. هرگاه که تگ در محدوده ریدر قرار بگیرد، سیگنالی را در پاسخ به ریدر می‌فرستد. به این ترتیب ریدر می‌تواند هویت تگ را شناسایی کند. با یک مثال معنای ریدر و تگ را بهتر متوجه خواهید شد. به عنوان نمونه، کارت مترو یک تگ RFID و دستگاهی که آن را می‌خواند یک ریدر RFID است. دستگاه همیشه در حال فرستادن سیگنال است. وقتی که کارت را به آن نزدیک می‌کنید، کارت یک سیگنال شامل کد شناسایی خود را به دستگاه می‌فرستد. دستگاه آن را تشخیص داده و تصمیم می‌گیرد که اجازه عبور داده شود یا خیر.



تگ‌های RFID چند نوع مختلف دارند: فعال (Active)، نیمه‌فعال (Semi-Passive) و غیرفعال (Passive). تگ غیرفعال انرژی خود را از سیگنال دریافتی از ریدر می‌گیرد. تگ‌های نیمه‌فعال دارای منبع تغذیه مستقل هستند اما باز هم برای ارسال سیگنال پاسخ از انرژی سیگنال ریدر استفاده می‌کند. تگ فعال دارای منبع تغذیه مستقل بوده و برای ارسال سیگنال نیز از منبع تغذیه خودش استفاده می‌کند. طبیعی است که سیگنال ارسالی با تگ فعال از تگ نیمه‌فعال از تگ غیرفعال قوی‌تر بوده و بنابراین در فاصله بیشتری می‌تواند عمل کند. کاربردترین نوع تگ‌های RFID، تگ غیرفعال است چرا که این نوع ارزان‌قیمت و کم‌حجم بوده و نیازی به منبع تغذیه یا باتری ندارد. این تگ‌ها معمولاً به شکل کارت اعتباری یا جاکلیدی هستند.

درون هر ریدر RFID سه بخش کلی وجود دارد. بخش اول یک المان فرستنده است که سیگنال رادیویی را ارسال می‌کند. بخش دوم یک گیرنده RFID است که سیگنال تگ را دریافت می‌کند. بخش سوم معمولاً یک میکروکنترلر است که سیگنال‌ها را پردازش کرده و دستور متناسب را ارسال می‌کند. ریدر RFID ممکن است مستقیماً به یک کامپیوتر متصل باشد.

درون تگ RFID یک المان گیرنده/فرستنده وجود دارد که کار دریافت و ارسال سیگنال از طریق امواج رادیویی را بر عهده دارد. بخش دیگر تگ، یک پردازنده و یک حافظه است که داده مورد نظر را نگه‌داشته و می‌فرستد. همچنین درون تگ غیرفعال یک مدار الکتریکی وجود دارد که انرژی سیگنال دریافتی را در خود نگه‌داشته و در اختیار پردازنده و حافظه می‌گذارد.

دستگاه‌های RFID با فرکانس‌های مختلفی کار می‌کنند. فرکانس کاری RFID در کشورهای مختلف متفاوت است اما معمولاً در سه دسته قرار می‌گیرند: فرکانس پائین (محدوده کیلوهرتز)، فرکانس بالا (محدوده چند مگاهرتز) و فرکانس خیلی بالا (محدوده چند صد مگاهرتز). هرچه فرکانس سیگنال بالاتر باشد، مسافت موثر بیشتری را می‌تواند طی کند. سیستم فرکانس پائین تا محدوده چند سانتی‌متر، فرکانس بالا در حد ۱ متر و فرکانس خیلی بالا تا محدوده ۱۰-۱۵ متر عملکرد مناسبی دارند.

محدوده کاری	فرکانس کاری	نوع مازول
-------------	-------------	-----------

تا 10cm	125kHz یا 134kHz	فرکانس پائین (LF)
تا 1m	13.56MHz	فرکانس بالا (HF)
تا 10m تا 15m	860MHz تا 960MHz	فرکانس خیلی بالا (UHF)

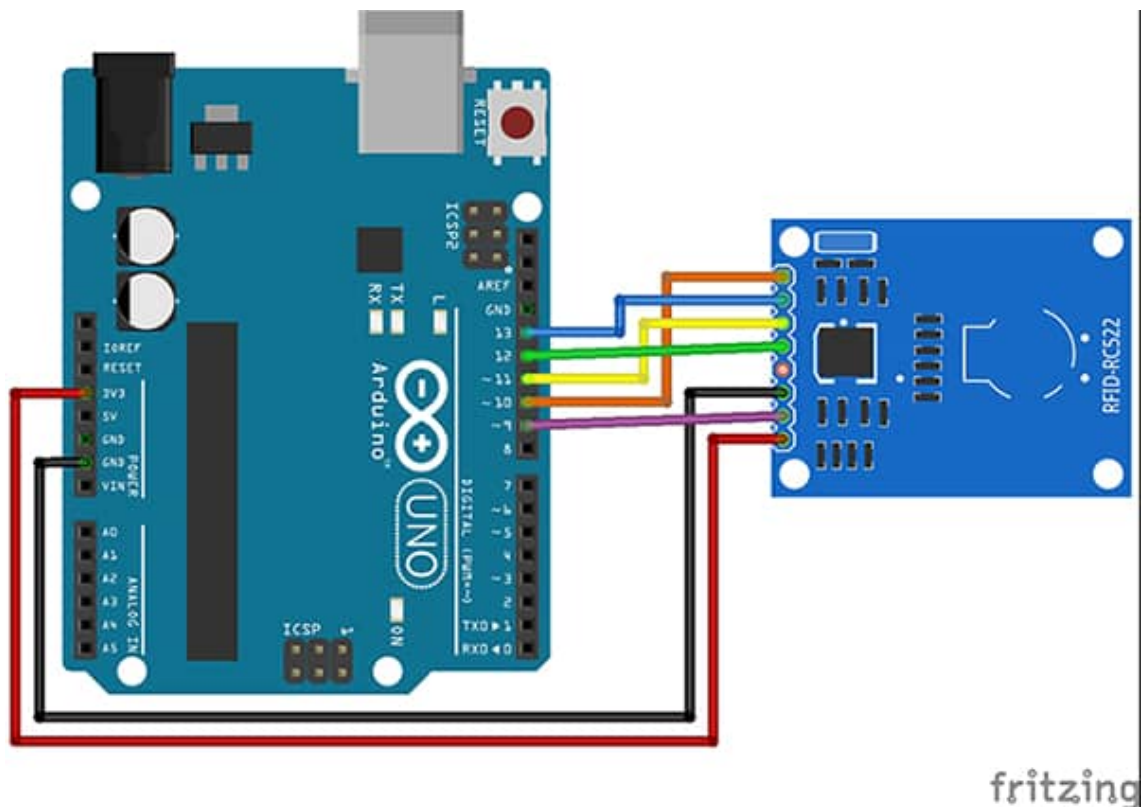
## معرفی ماژول RFID

**ماژول RC522** یک ریدر RFID ارزان قیمت است که در بیشتر پروژه‌ها از آن استفاده می‌شود. این ماژول معمولاً به همراه یک ست تگ به فرم جاکلیدی و کارت اعتباری ارائه می‌شود. تگ ماژول دارای یک حافظه 1KB یا 1024B است. ماژول با فرکانس استاندارد 13.56MHz کار می‌کند و بنابراین یک ماژول فرکانس بالا با دامنه کاری متوسط است. با استفاده از این ماژول می‌توانید اطلاعات تگ را خوانده یا اطلاعات مورد نظران را بر روی آن بنویسید. یکی از ویژگی‌های خوب این ماژول پشتیبانی از پروتکل‌های SPI و I2C و UART است. ماژول طوری طراحی شده که با استفاده از پین‌های موجود بر روی آن می‌توانید از هر کدام از این پروتکل‌ها بدون نیاز به پایه جدید استفاده کنید.

تغذیه این ماژول با ولتاژ 3.3V است. بنابراین وصل کردن ولتاژ 5V به آن باعث سوختن ماژول می‌شود. با این وجود پایه‌های سیگنال ماژول تحمل ولتاژ 5V را دارند. پس می‌توانید آنها را مستقیماً به پایه‌های دیجیتال آردوینو وصل کنید.

کتابخانه معرفی که برای ماژول RFID استفاده می‌شود، MFRC522 نام دارد. این کتابخانه با پروتکل SPI با ماژول ارتباط برقرار می‌کند. پایه‌های MISO، MOSI، SS و SCK را به ترتیب به پایه‌های ۱۰، ۱۱، ۱۲ و ۱۳ آردوینو وصل کنید. پایه‌های VCC و GND را به 3.3V و GND آردوینو و پایه RST را به پایه ۹ آردوینو وصل کنید. کتابخانه زیر را دانلود کرده و در نرم‌افزار آردوینو نصب کنید. نحوه نصب کتابخانه را می‌توانید در مطلب [شروع کار با نرم‌افزار آردوینو](#) مطالعه کنید.

### [دانلود کتابخانه RFID](#)



fritzing

## خواندن اطلاعات RFID

اولین کاری که با یک تگ RFID می‌توانید انجام دهید این است که اطلاعات موجود بر روی حافظه آن را بخوانید. برای این کار کد زیر را بر روی آردوینو آپلود کنید. سپس پنجره سریال مانیتور را باز کرده و تگ را به ماژول نزدیک کنید.

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

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

#define ResetPin 9
#define CSPin 10
MFRC522 RFID(CSPin, ResetPin);

void setup()
{
    Serial.begin(9600);
    SPI.begin();
    RFID.PCD_Init();
    delay(10);
    RFID.PCD_DumpVersionToSerial();
    Serial.println("Approximate your RFID tag...");
}

void loop()
{
    if (!RFID.PICC_IsNewCardPresent())
    {
        return;
    }
    if (!RFID.PICC_ReadCardSerial())
    {
        return;
    }
    RFID.PICC_DumpToSerial(&(RFID.uid));
}

```

در این برنامه ابتدا کتابخانه‌های مورد نظر اضافه شده و پایه‌های Reset و CS تعریف شده‌اند.

```

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

#define ResetPin 9
#define CSPin 10
MFRC522 RFID(CSPin, ResetPin);

```

در قسمت setup توابع زیر باید فراخوانی شود تا ماژول شروع به کار کند.

```

Serial.begin(9600);
SPI.begin();
RFID.PCD_Init();
delay(10);
RFID.PCD_DumpVersionToSerial();
Serial.println("Approximate your RFID tag...");

```

در بخش loop برنامه چک می‌کند که آیا تگ RFID در دسترس است یا خیر و در این صورت اطلاعات آن را می‌خواند.

```

if (!RFID.PICC_IsNewCardPresent())
{
    return;
}

```

```

if (!RFID.PICC_ReadCardSerial())
{
    return;
}
RFID.PICC_DumpToSerial(&(RFID.uid));

```

اگر همه چیز درست باشد چیزی شبیه به عکس زیر را در سریال مانیتور خواهید دید. این اطلاعات شامل مشخصات تگ به همراه تمام 1KB اطلاعات موجود بر روی آن است. برای اینکه بتوانید اطلاعات مناسبی را از ماژول دریافت کنید یا بر روی تگ اطلاعاتی ذخیره کنید باید ساختار حافظه تگ را تا حدی بشناسید. این حافظه از ۱۶ بخش (Sector) تشکیل شده که شماره آنها از ۰ تا ۱۵ است. هر بخش شامل ۴ بلوک (Block) به شماره ۰ تا ۳ است و هر بلوک، ۱۶ داده (Data) به شماره ۰ تا ۱۵ را در خود نگه می‌دارد.

```

COM4
Firmware Version: 0x92 = v2.0
Bring your RPID card or keychain near the module...
Card UID: F4 2C 2C 83
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  AccessBits
  15   63  00 00 00 00 00 00 FF 07 80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
      62  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      61  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
  14   59  00 00 00 00 00 00 FF 07 80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
      58  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      57  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      56  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
  13   55  00 00 00 00 00 00 FF 07 80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
      54  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      53  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
      52  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
  12   51  00 00 00 00 00 00 FF 07 80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
      50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [ 0 0 0 ]
Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

```

نکته مهم این است که بلوک 3 از هر بخش، کد مجوز دسترسی به نوشتن و خواندن را در خود نگه می‌دارد. بنابراین عملاً ۴۸ بلوک از ۶۴ بلوک حافظه برای نوشتن و خواندن قابل استفاده است. این قسمت را تغییر ندهید مگر اینکه اطلاعات کافی در مورد تغییر کد مجوز تگ داشته باشید. علاوه بر این، بلوک ۰ از بخش ۰ حافظه نیز شامل اطلاعات مربوط به سازنده است. این قسمت به هیچ عنوان نباید تغییر داده شوند. در غیر این صورت احتمالاً تگ برای همیشه غیر قابل استفاده خواهد شد.

## نوشتن اطلاعات بر روی RFID

در این بخش به نحوه نوشتن اطلاعات بر روی تگ می‌پردازیم. نوشتن اطلاعات بر روی تگ زمانی به کار می‌آید که مثلاً بخواهید کارتهای ثبت ورود و خروج برای افراد تعریف کنید. برای تغییر اطلاعات تگ نیاز به دانستن رمز آن دارید. اگر یک تگ نو دارید، رمز آن شش کاراکتر 0xFF است. اگر رمز قبلاً تغییر کرده باشد، بدون دانستن آن نمی‌توانید اطلاعاتی روی آن بنویسید.

برای نوشتن اطلاعات روی تگ، معمولاً از یک بلوک آن استفاده می‌کنیم. همان طور که می‌دانید هر بلوک شامل ۱۶ کاراکتر است که به صورت hex نمایش داده می‌شوند. برنامه زیر عبارت [Sanatbazar.com](http://Sanatbazar.com) را بر روی کارت حافظه می‌نویسد:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

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

#define CSPin 10

```

```

#define ResetPin 9
MFRC522 RFIDmodule(CSPin, ResetPin);
MFRC522::MIFARE_Key key;

int blockNumber = 61;
byte myData[16] = {" Sanatbazar.com "};
byte empty[16] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
byte writtenData[18];

int writeBlock(int blockNumber, byte dataAddress[])
{
    int largestModulo4Number = blockNumber / 4 * 4;
    int trailerBlock = largestModulo4Number + 3;
    if (blockNumber > 2 && (blockNumber + 1) % 4 == 0)
    {
        Serial.print(blockNumber);
        Serial.println(" is a trailer block number:");
        return 2;
    }
    Serial.print(blockNumber);
    Serial.println(" is a data block number:");

    byte status = RFIDmodule.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
trailerBlock, &key, &(RFIDmodule.uid));
    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("PCD_Authenticate() failed: ");
        Serial.println(RFIDmodule.GetStatusCodeName(status));
        return 3;
    }

    status = RFIDmodule.MIFARE_Write(blockNumber, dataAddress, 16);

    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("MIFARE_Write() failed: ");
        Serial.println(RFIDmodule.GetStatusCodeName(status));
        return 4;
    }
    Serial.println("Data was written to selected block");
}

int readBlock(int blockNumber, byte dataAddress[])
{
    int largestModulo4Number = blockNumber / 4 * 4;
    int trailerBlock = largestModulo4Number + 3;

    byte status = RFIDmodule.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
trailerBlock, &key, &(RFIDmodule.uid));

    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("PCD_Authenticate() failed (read): ");
        Serial.println(RFIDmodule.GetStatusCodeName(status));
        return 3;
    }

    byte buffersize = 18;
    status = RFIDmodule.MIFARE_Read(blockNumber, dataAddress, &buffersize);
    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("MIFARE_read() failed: ");
        Serial.println(RFIDmodule.GetStatusCodeName(status));
        return 4;
    }
    Serial.println("Data was successfully read");
}

void setup()
{
    Serial.begin(9600);
    SPI.begin();
    RFIDmodule.PCD_Init();
    Serial.println("Approximate your RFID tag...");

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

```

```
void loop()
{
  if (!RFIDmodule.PICC_IsNewCardPresent())
  {
    return;
  }
  if (!RFIDmodule.PICC_ReadCardSerial())
  {
    return;
  }
  Serial.println("Found a tag");
  writeBlock(blockNumber, myData);
  readBlock(blockNumber, writtenData);
  Serial.print("Your written data is: ");
  Serial.println(String((char*)writtenData));
  Serial.println("");
}
}
```

بخش‌های ابتدایی این برنامه مانند برنامه قبل است. پس از تعریف اولیه، اولین کار وارد کردن رمز عبور تگ است:

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

مانند حالت قبل در اینجا نیز در حلقه loop، هر بار بررسی می‌کنیم که آیا تگ جدیدی در دسترس قرار گرفته است یا خیر.

```
if (!RFIDmodule.PICC_IsNewCardPresent())
{
  return;
}
if (!RFIDmodule.PICC_ReadCardSerial())
{
  return;
}
Serial.println("Found a tag");
```

در صورتی که یک تگ RFID در دسترس باشد، با استفاده از تابع writeBlock() اطلاعات مورد نظر شما در بلوکی که تعریف کرده‌اید نوشته می‌شود. پس از آن برای اطمینان از اینکه داده‌ها درست نوشته شده‌اند، داده همان بلوک توسط تابع readBlock() خوانده شده و در سریال مانیتور نمایش داده می‌شود:

```
Serial.println("Found a tag");
writeBlock(blockNumber, myData);
readBlock(blockNumber, writtenData);
Serial.print("Your written data is: ");
Serial.println(String((char*)writtenData));
Serial.println("");
```

دو تابع writeBlock() و readBlock() شباهت زیادی به هم دارند. در ابتدای هر دو تابع باید بررسی کنیم که بلوک انتخاب شده، بلوک سوم هیچ بخشی نباشد.

```
int largestModulo4Number = blockNumber / 4 * 4;
int trailerBlock = largestModulo4Number + 3;
if (blockNumber > 2 && (blockNumber + 1) % 4 == 0)
{
  Serial.print(blockNumber);
  Serial.println(" is a trailer block number:");
  return 2;
}
Serial.print(blockNumber);
Serial.println(" is a data block number:");
```

در صورتی که بلوک انتخاب شده برای نوشتن اطلاعات مناسب باشد، رمز عبور تگ با استفاده از دستور PCD\_Authenticate() بررسی می‌شود:

```
byte status = RFIDmodule.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
trailerBlock, &key, &(RFIDmodule.uid));
if (status != MFRC522::STATUS_OK)
```

```
{
  Serial.print("PCD_Authenticate() failed: ");
  Serial.println(RFIDmodule.GetStatusCodeName(status));
  return 3;
}
```

سپس اطلاعات مورد نظر در بلوک انتخاب شده به کمک تابع `MIFARE_Write()` نوشته می‌شود:

```
status = RFIDmodule.MIFARE_Write(blockNumber, dataAddress, 16);
```

در تابع `readBlock()` بلوکی که برای نوشتن انتخاب کرده بودید خوانده می‌شود. این کار توسط دستور `MIFARE_Read()` صورت می‌گیرد. این دستور برای خواندن اطلاعات یک بلوک ۱۶ کاراکتری نیاز به یک فضای ۱۸ کاراکتری دارد. به همین دلیل متغیر `writtenData` به طول ۱۸ تعریف شده است. در نهایت اطلاعات خوانده شده در سریال مانیتور نمایش داده می‌شود:

```
byte buffersize = 18;
status = RFIDmodule.MIFARE_Read(blockNumber, dataAddress, &buffersize);
```

## سیستم دربارکن هوشمند آردوینویی

اگر دقت کرده باشید در ابتدای نمایش اطلاعات موجود در تگ RFID چند داده دیگر نیز نمایش داده می‌شود که نشان دهنده نوع و شناسه تگ است. شناسه یا ID تگ یک بخش ۴ کاراکتری به صورت hex است که می‌توانید از آن برای کاربرد مورد نظران استفاده کنید. برای مثال می‌توانید از تگ RFID به عنوان کلید راه‌اندازی یک دستگاه یا برای باز کردن درب منزل به همراه یک رله و سولنوئید استفاده کنید.

برای استفاده از RFID به عنوان کلید، کد زیر را بر روی آردوینو آپلود کنید. می‌توانید درون برنامه دستور راه‌اندازی وسیله مورد استفاده در پروژه‌تان را اضافه کنید. اگر می‌خواهید از رله در پروژه استفاده کنید، می‌توانید مطلب [آموزش راه‌اندازی رله با آردوینو](#) را مطالعه کنید.

```
#include <Arduino.h>
/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com
*/

#include <SPI.h>
#include <MFRC522.h>
#define ResetPin 9
#define CSPin 10

String memberID = "81F7C42F";
String tagID = "";
MFRC522 RFIDmodule(CSPin, ResetPin);

boolean readTagID()
{
  if (!RFIDmodule.PICC_IsNewCardPresent())
    return false;
  if (!RFIDmodule.PICC_ReadCardSerial())
    return false;
  tagID = "";
  for (uint8_t i = 0; i < 4; i++)
    tagID.concat(String(RFIDmodule.uid.uidByte[i], HEX));
  tagID.toUpperCase();
  RFIDmodule.PICC_HaltA();
  return true;
}

void setup()
{
  Serial.begin(9600);
  SPI.begin();
  RFIDmodule.PCD_Init();
}
```



```

        Serial.println("Approximate your RFID tag ...");
        Serial.println("");
    }

    void loop()
    {
        while (readTagID())
        {
            if (tagID == memberID)
            {
                Serial.println("You're Welcome!");
            }
            else
            {
                Serial.println("You're not registered!");
            }
            Serial.print(" ID : ");
            Serial.println(tagID);
            Serial.println("");
            delay(2000);
            Serial.println("Approximate your RFID tag ...");
            Serial.println("");
        }
    }
}

```

در این برنامه پس از تعریف کتابخانه‌ها و پایه‌های مورد نظر، شناسه کارت‌ها که مجوز عبور دارد و کلید شما به حساب می‌آید را وارد کنید:

```

#include <SPI.h>
#include <MFRC522.h>
#define ResetPin 9
#define CSPin 10

String memberID = "81F7C42F";
String tagID = "";
MFRC522 RFIDmodule(CSPin, ResetPin);

```

در قسمت setup طبق معمول خروجی سریال و ماژول RFID را راهاندازی می‌کنیم:

```

Serial.begin(9600);
SPI.begin();
RFIDmodule.PCD_Init();
Serial.println("Approximate your RFID tag ...");
Serial.println("");

```

در حلقه loop در صورتی که یک تگ RFID در دسترس قرار بگیرد، شناسه آن خوانده شده و در صورتی که با شناسه کلید یکی باشد، سیستم اجازه ورود داده و در غیر این صورت اجازه نمی‌دهد.

```

while (readTagID())
{
    if (tagID == memberID)
    {
        Serial.println("You're Welcome!");
    }
    else
    {
        Serial.println("You're not registered!");
    }
    Serial.print(" ID : ");
    Serial.println(tagID);
    Serial.println("");
    delay(2000);
    Serial.println("Approximate your RFID tag ...");
    Serial.println("");
}

```

خواندن شناسه تگ به کمک تابع readTagID() انجام می‌شود. در این تابع بایت‌های شناسه یکی یکی به یکدیگر چسبانده شده و در متغیر tagID ذخیره می‌شود. خواندن داده از روی RFID با دستور PICC\_HaltA() متوقف شده و برای خواندن داده بعدی ماژول آماده می‌شود.

```

COM4
Approximate your RFID tag ...

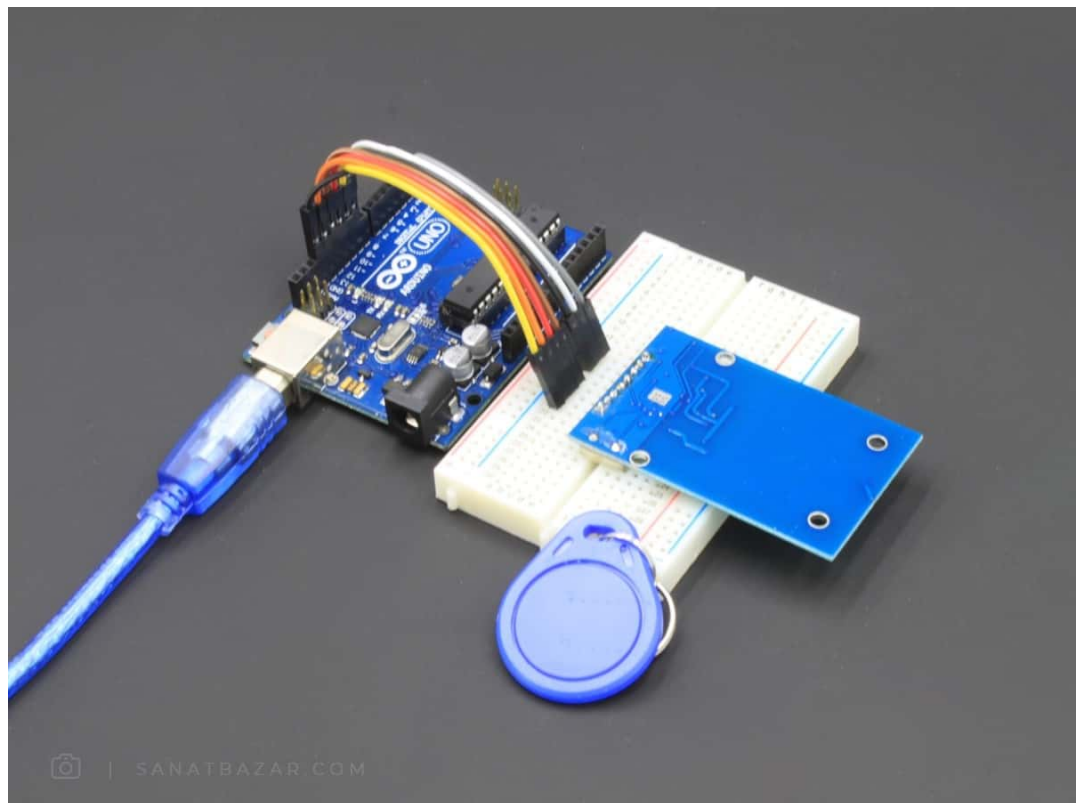
You're not registered!
ID : F42C2C83

Approximate your RFID tag ...

You're Welcome!
ID : 81F7C42F

Approximate your RFID tag ...

```



## نتیجه‌گیری

در این آموزش با ماژول RFID آشنا شده و نحوه راهاندازی آن را یاد گرفتید.

در آموزش بعدی، نحوه [راهاندازی ماژول GSM](#) را خواهید آموخت.

نظرات شما باعث بهبود محتوای آموزشی ما می‌شود. اگر این آموزش را دوست داشتید، همین‌طور اگر سوالی در مورد آن دارید، از شنیدن نظراتتان خوشحال خواهیم شد.