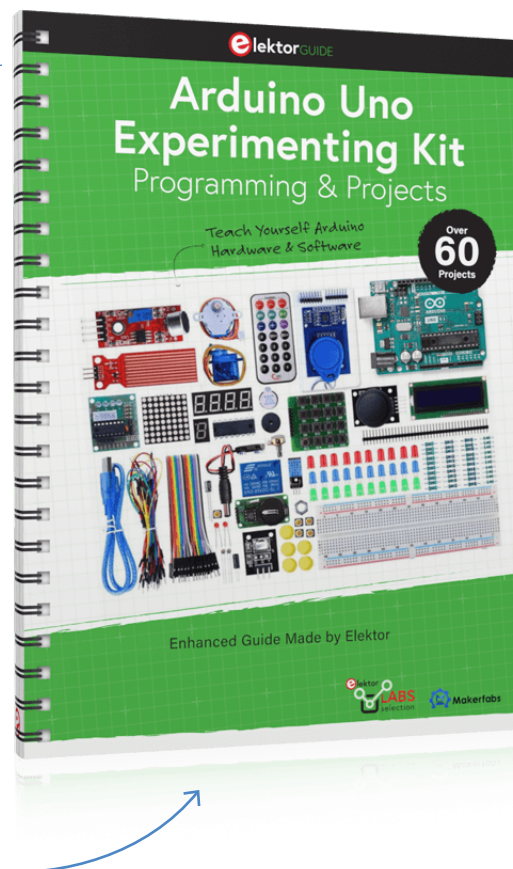


lecture d'étiquettes RFID et serrure de porte RFID

exemples de projets tirés du
guide d'expérimentation pour
Arduino d'Elektor

RFID est l'abréviation de *Radio-Frequency Identification*, une technologie sans fil qui couvre des dispositifs à des fins de sécurité, d'accès et de suivi des marchandises. En fait, un système RFID comprend un lecteur RFID et une ou plusieurs étiquettes. Les deux dispositifs matériels sont contenus dans un kit polyvalent de composants destiné à l'expérimentation avec Arduino et accompagné d'un guide spécialement rédigé à cet effet. Dans cet article, Dogan Ibrahim, auteur d'Elektor, explore la technologie RFID et s'appuie sur le fameux Arduino UNO, pour illustrer le développement d'un projet embarqué à partir de zéro.

Note de la rédaction. Cet article est un extrait du livre de 238 pages *Arduino Uno Experimenting Kit - Programming & Projects*. Ce guide fait partie de l'offre groupée d'Elektor Arduino Experimenting Bundle. L'extrait a été formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. Étant donné qu'il s'agit d'un extrait d'une publication plus importante, certains termes de cet article peuvent faire référence à des discussions se trouvant ailleurs dans le livre. L'auteur et l'éditeur ont fait de leur mieux pour éviter une telle situation et répondront volontiers aux questions. Les coordonnées des personnes à contacter sont indiquées au paragraphe « Des questions, des commentaires ? ».



La technologie RFID utilise des champs électromagnétiques pour transférer des données sur de courtes distances. Les systèmes RFID sont principalement utilisés dans les applications de sécurité. Par exemple, ils peuvent être utilisés pour ouvrir une porte exclusivement par une personne possédant l'étiquette RFID correspondante.

Le lecteur RFID fourni avec l'offre groupée d'Elektor mentionnée ci-dessus (comprenant un livre et un kit de composants) est le « module RC522 » (**figure 1**). Il présente les caractéristiques suivantes :

- Fréquence de fonctionnement : 13,56 MHz
- Tension de fonctionnement : +3.3 V
- Fonctionnement avec le bus SPI et le bus I²C



Figure 1. Lecteur et étiquette RFID.



Le lecteur RFID est livré avec des connecteurs que vous devez souder au lecteur avant de l'utiliser.

Projet 1 : Trouver l'identifiant de l'étiquette

Dans ce projet, vous apprendrez à afficher l'identifiant de l'étiquette dans le moniteur série d'Arduino.

Schéma de principe : la **figure 2** est une illustration du schéma de principe du projet.

Circuit : les connexions entre les ports de la carte de développement UNO et le module de lecture RFID sont comme suit :

Broche du lecteur RFID	Port de la carte de développement
SDA	10
SCK	13
MOSI	11
MISO	12
IRQ	non utilisé
GND	GND
RST	9
3.3 V	3.3 V

La **figure 3** montre le schéma du projet. **Attention à ne pas connecter la broche d'alimentation à +5 V.**

Listage du programme : avant d'utiliser le lecteur RFID, vous devez ajouter la bibliothèque RFID à votre EDI. La bibliothèque est appelée **MFRC522**, et les étapes pour l'ajouter sont les suivantes :

- Téléchargez le fichier *rfid-master.zip* et placez-le dans un dossier. Vous pouvez le trouver dans les téléchargements associés au livre sur l'e-choppe Elektor [1].

- Lancer l'EDI.
- Cliquez sur *Sketch* → *Include Library* → *Add .ZIP Library*.
- Cherchez le fichier *.zip* sauvegardé et cliquez sur *Open*
- Vous pouvez maintenant commencer à utiliser la bibliothèque.

La bibliothèque du lecteur RFID offre de nombreuses fonctions qui sont accessibles en décompressant le fichier de la bibliothèque. Voici les fonctions les plus importantes de cette bibliothèque :

<code>mfr522.PCD_Init()</code>	Initialisation du lecteur RFID
<code>mfr522.PICC_IsNewCardPresent()</code>	Recherche d'un module de lecture RFID
<code>mfr522.PICC_ReadCardSerial()</code>	Sélection du lecteur RFID à utiliser
<code>mfr522.uid.uidByte[]</code>	Renvoie l'identifiant de l'étiquette dans un tableau
<code>mfr522.PICC_HaltA()</code>	Arrêt de lecture (Halt PICC)

Vous pouvez utiliser le programme *DumpInfo* de l'EDI Arduino pour déterminer l'identifiant (Tag ID) de votre carte. Voici les étapes à suivre :

- Lancer l'EDI
- Cliquez sur *File* → *Examples* → *MFRC522* → *DumpInfo*.
- Compilez et téléversez le programme sur la carte de développement.
- Lancez le moniteur série.
- Placez l'étiquette blanche sur le lecteur et maintenez-la en place jusqu'à ce que l'affichage des données s'arrête dans le moniteur série.

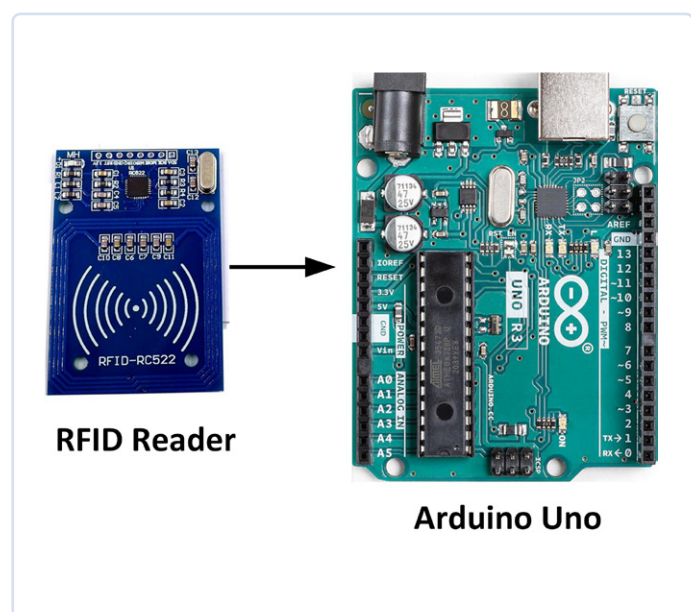


Figure 2. Schéma fonctionnel du projet de lecture d'étiquettes.

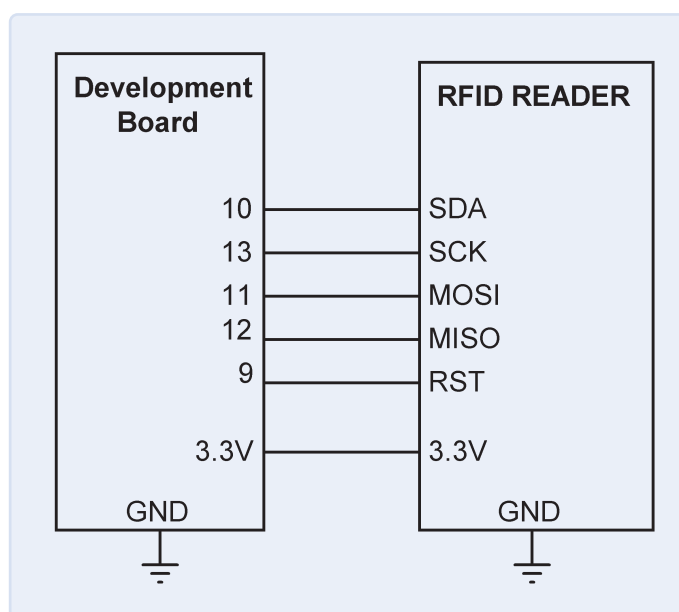


Figure 3. Circuit du projet de lecture d'étiquettes.

Firmware Version: 0x92 = v2.0
 Scan PICC to see UID, SAK, type, and data blocks...

Card UID: 23 F0 58 A7
 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	00	[0 0 0]
	61	00	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	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	00	[0 0 0]
	57	00	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	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	00	[0 0 0]
	53	00	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	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	00	[0 0 0]
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	48	00	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

Figure 4. Dédoublement des données de la mémoire des étiquettes.

Des données similaires à celles de la **figure 4** devraient s'afficher. Il s'agit de 1 KB de mémoire de la carte et de son identifiant. La mémoire de 1 KB est organisée en 16 secteurs (0 à 15), où chaque secteur est divisé en quatre blocs (0, 1, 2, 3). Chaque bloc peut stocker 16 octets de données (0 à 15). Par conséquent :

16 secteurs × 4 blocs × 16 octets =
 1,024 octets de données sur la carte (c.-à-d., 1 KB)

Le troisième bloc de chaque segment (c'est-à-dire le bloc supérieur) est appelé « queue du secteur » (*Sector Trailer*) et contient les bits d'accès. Ceux-ci contrôlent l'accès en lecture/écriture aux blocs restants du secteur. Par conséquent, seuls les trois blocs inférieurs (c'est-à-dire les blocs 0, 1 et 2) de chaque secteur sont disponibles pour le stockage des données utilisateur. Cela signifie que vous disposez de 48 octets (3 × 16 octets) par secteur de 64 octets.

Le bloc 0 du secteur 0 est appelé *bloc fabricant/données* de fabricant,

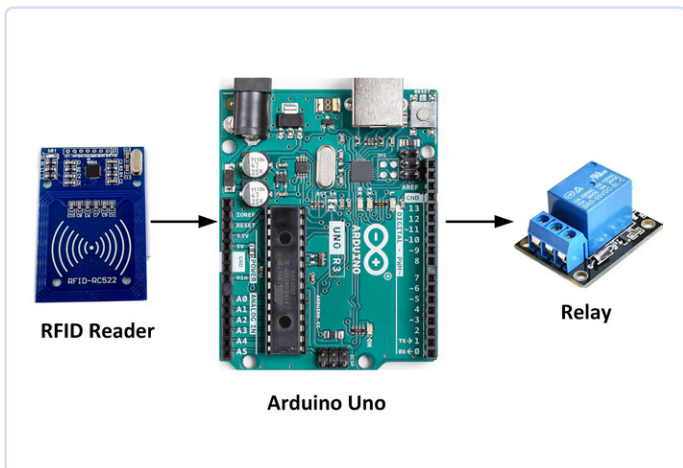


Figure 5. Schéma fonctionnel du projet de contrôle d'accès.

et il comprend les données du fabricant et l'identifiant de l'étiquette. L'ID de l'étiquette est également affiché sous le titre *Card UID* dans la **figure 4**. Dans ce projet, l'ID de l'étiquette est : **23 F0 58 A7**.

Projet 2 : Serrure de porte RFID et contrôle d'accès avec relais

Dans ce projet, le lecteur RFID et le relais fourni (également contenu dans le kit - NDLR) sont tous deux connectés à la carte de développement. On suppose qu'une porte d'entrée sécurisée est commandée par un relais protégé par un système RFID. Pour ouvrir la porte, il suffit de tenir une carte d'identification autorisée à proximité du lecteur RFID. Le relais est activé pendant 15 secondes, puis se désactive, de sorte que la porte se ferme.

Schéma de principe : la **figure 5** présente le schéma de principe du projet. Là encore, des images de cartes réelles sont utilisées - plutôt que des illustrations - avec des blocs fonctionnels.

Circuit : le schéma du circuit est présenté à la **figure 6**. Contrairement aux autres projets de « commutation » proposés dans le livre (non abordés ici), un relais est ajouté au port 2.

Listage du programme : le **listage 1** montre le programme développé pour la serrure de porte commandée par RFID, nommée *RFIDLock*. Au début du programme, les bibliothèques *SPI* et *MFRC522* sont incluses. L'ID de la carte valide est stockée dans la chaîne *ValidCard*, et *RELAY* est assigné au port 2. Dans la fonction *setup()*, *RELAY* est configuré comme sortie et est désactivé. Le bus SPI et le MFRC522 sont également initialisés.

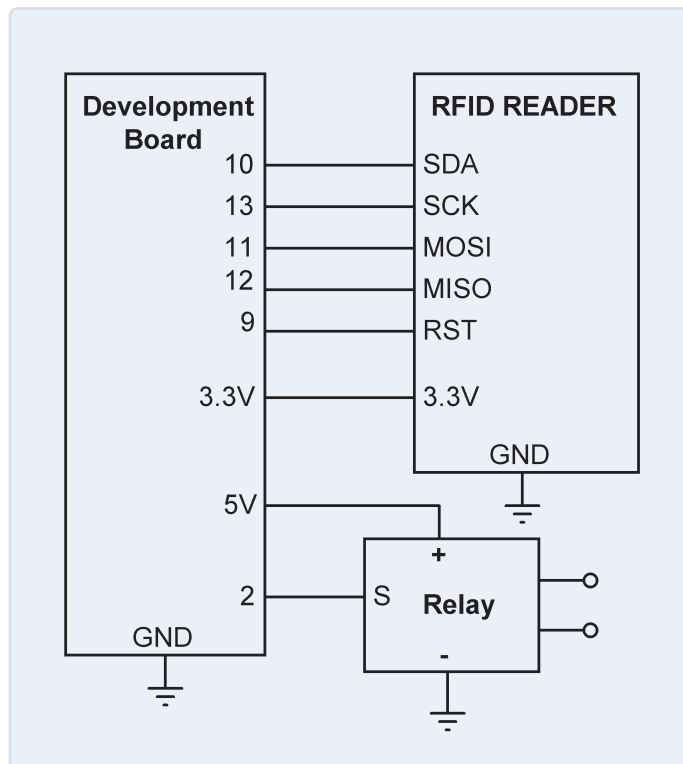


Figure 6. Circuit du projet de contrôle d'accès.



Listage 1. Le programme RFIDLock développé pour le projet.

```
//-----  
//                               RFID LOCK SYSTEM  
//                               =====  
//  
// In this program, the RFID card reader is used with a relay. The relay  
// is only activated if an authorized tag is placed near the reader. The  
// relay stays ON for 15 seconds and then turns OFF. The Tag ID of the  
// authorized valid card in this example is: 23 F0 58 A7  
//  
// Author: Dogan Ibrahim  
// File  : RFIDLock  
// Date  : May, 2022  
//-----  
#include <SPI.h>  
#include <MFRC522.h>  
  
#define SS_PIN 10  
#define RST_PIN 9  
MFRC522 mfrc522(SS_PIN, RST_PIN);    // Create MFRC522 instance  
String ValidCard = "23F058A7";        // Valid Tag ID  
String TagID = "";  
int RELAY = 2;                        // RELAY at port 2  
byte i;  
  
void setup()  
{  
  pinMode(RELAY, OUTPUT);              // RELAY is output  
  digitalWrite(RELAY, LOW);            // Deactivate RELAY  
  SPI.begin();                        // Initiate SPI bus  
  mfrc522.PCD_Init();                 // Initiate MFRC522  
}  
void loop()  
{  
  if (!mfrc522.PICC_IsNewCardPresent()) // Look for card  
  {  
    return;  
  }  
  
  if (!mfrc522.PICC_ReadCardSerial())    // Select the card  
  {  
    return;  
  }  
  TagID = "";  
  
  for (i = 0; i < 4; i++)                 // Read 4 byte Tag ID  
  {  
    TagID.concat(String(mfrc522.uid.uidByte[i], HEX));  
  }  
  
  TagID.toUpperCase();                   // Convert to upper case  
  mfrc522.PICC_HaltA();                  // Stop reading  
  if(TagID == ValidCard)                 // Valid card?  
  {  
    digitalWrite(RELAY, HIGH);            // RELAY ON  
    delay(15000);                         // Wait 15 seconds  
    digitalWrite(RELAY, LOW);             // RELAY OFF  
  }  
  else  
    digitalWrite(RELAY, LOW);             // RELAY OFF  
}
```