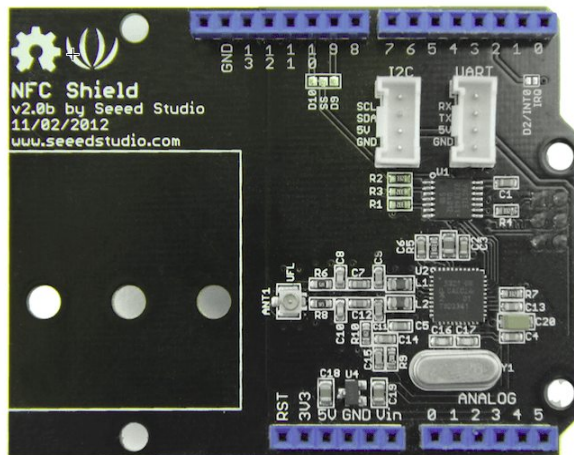


# NFC Shield V2.0

## Présentation

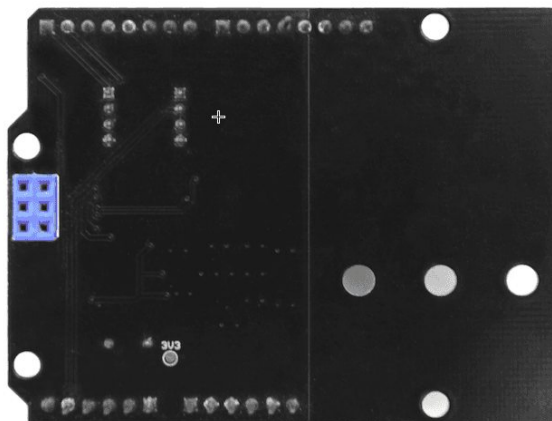


NFC (Near Field Communication) est une technologie largement utilisée. Certaines applications de la NFC comprennent des systèmes de contrôle d'accès sans fil (p. Ex. Portes sans clé et verrous) et des paiements pour appareils mobiles (par exemple, les registres de magasins qui reçoivent des informations de paiement via une application téléphonique).

Le NFC Shield est équipé d'un module émetteur-récepteur PN532 qui gère la communication sans fil à 13,56 MHz, ce qui signifie que vous pouvez lire et écrire une étiquette de 13,56 MHz avec ce blindage ou mettre en œuvre un échange de données point à point (P2P) entre le blindage et un téléphone intelligent .

Pour cette nouvelle version du bouclier, nous avons créé une zone d'antenne PCB distincte, indépendante, qui vous permet

d'extraire plus facilement l'interface NFC à l'extérieur de votre enceinte de circuit principal.



## Idées d'application

Si vous voulez faire quelques projets impressionnants par NFC Shield V2.0, voici quelques projets à titre de référence.

### NFC Shield Demo

*Paper Man, un objet intéressant pour interagir avec Android*



[Faites-le maintenant!](#)

[Plus de projets impressionnants par NFC Shield V2.0](#)

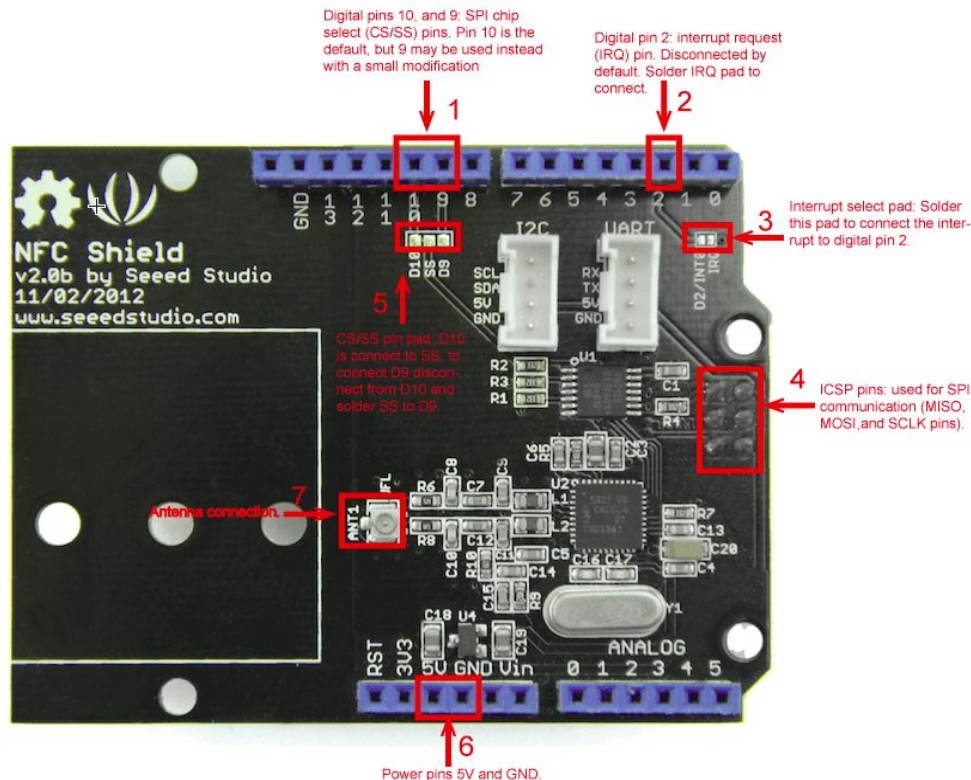
## Caractéristiques

- Utilisation de l'en-tête ICSP pour SPI. Cela signifie que le bouclier fonctionne avec les cartes de développement Arduino suivantes: Uno, Mega, Leonardo
- Communication NFC sans fil à 13,56 MHz
- Protocole SPI - interface d'enregistrement de broches qui ne nécessite que 4 broches
- Tension d'entrée: 5V de la broche 5V de l'Arduino
- Courant typique: 100mA
- 5cm de portée maximale
- Prise en charge de la communication P2P
- Prise en charge des protocoles ISO14443 de type A et B

## Présentation du matériel

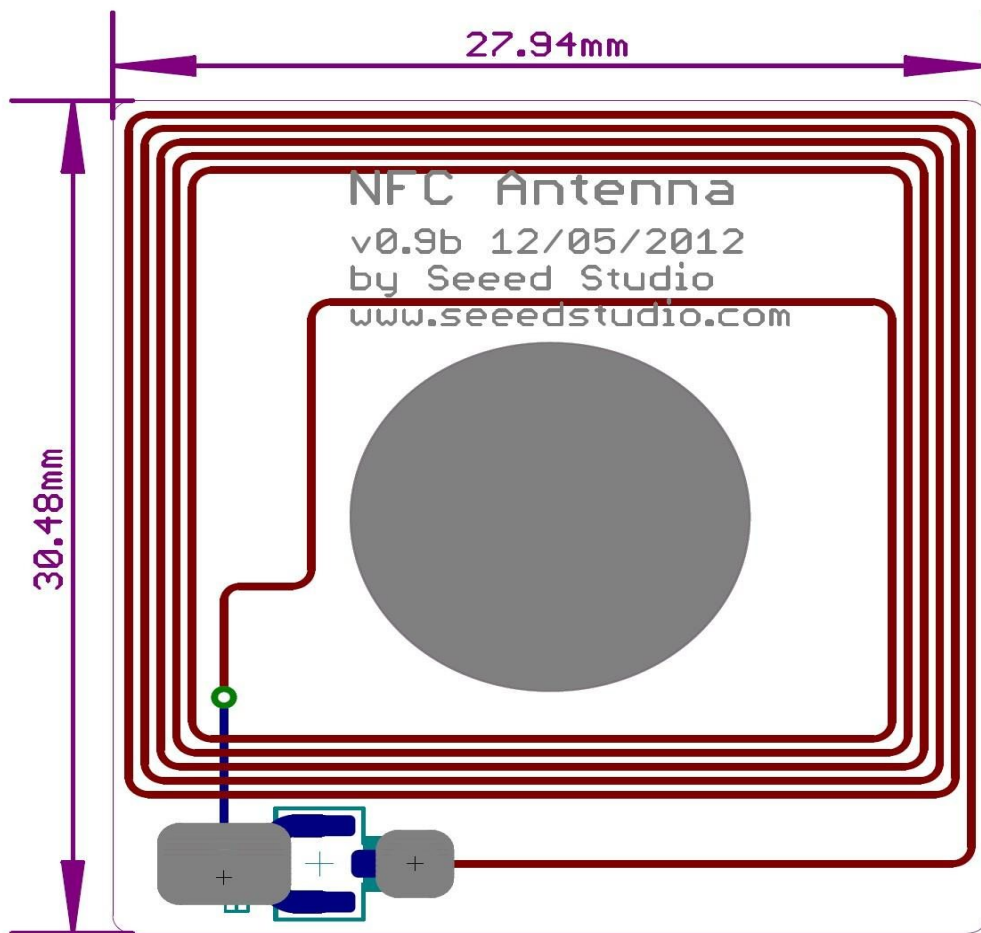
Les broches du bouclier NFC et les autres bornes sont décrits ci-dessous.

### Interface de blindage NFC



- 
- 
- D10 et D9 sont utilisés pour la sélection de puce SPI (CS / SS). D10 est connecté par défaut, pour connecter le soudage D9 au pad D9 et le branchement entre SS et D10 est nécessaire.
- D2 peut être utilisé pour recevoir le signal de broche de demande d'interruption de l'écran (IRQ). L'interruption n'est pas connectée par défaut, la soudure des plots "D2 / INT0" et "IRQ" est requise.
- Le bouclier obtient son interface SPI (SPI MOSI, MISO, et broches SCK) de l'Arduino ICSP en-tête directement, ce qui signifie que le bouclier fonctionne les suivants Arduinos: Uno, Mega et Leonardo.
- La borne ANT1 est l'endroit où l'antenne NFC (fournie avec le blindage) est connectée.
- Le blindage est alimenté par 5V de la carte Arduino.

L'antenne du bouclier NFC, fournie avec le blindage, est un module de circuit imprimé séparé qui est fixé au blindage par un câble. L'antenne est la zone utilisée pour recevoir et transmettre l'information.



**Pièce jointe PCB antenne NFC**

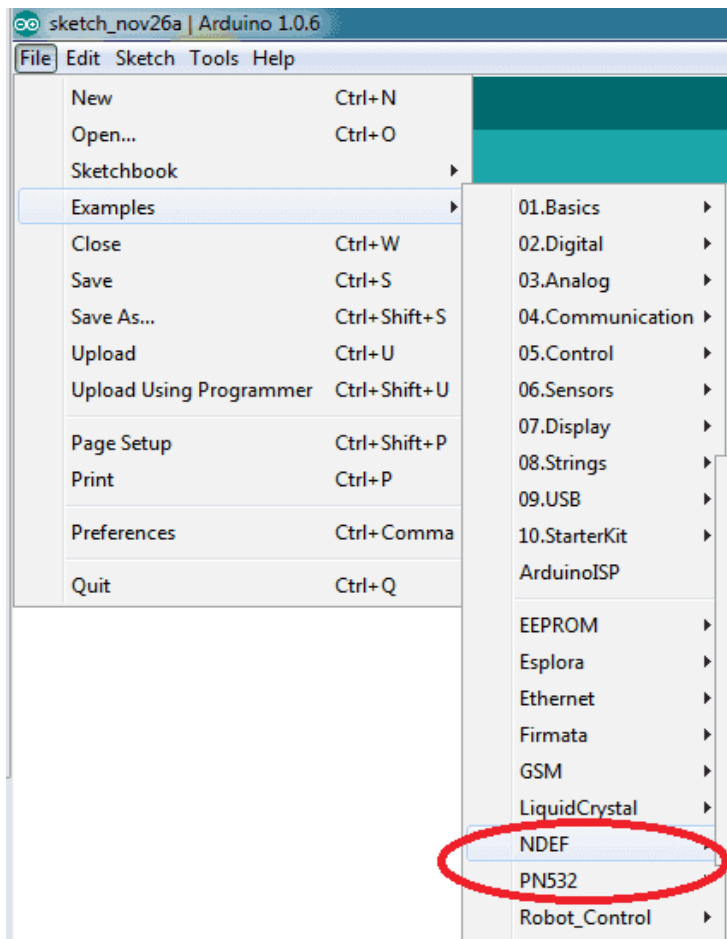
## NFC Shield Setup

### Installation du matériel

1. Fixez l'antenne NFC sur le blindage.
2. Empilez le bouclier NFC sur votre carte de développement Arduino et connectez la carte à un PC à l'aide d'un câble USB.

### Logiciels Bibliothèques Installation

1. Fermez l'IDE Arduino si vous l'avez ouvert.
2. Télécharger la [bibliothèque de PN532](#) dossier ZIP et extraire les fichiers.
3. Copiez les dossiers PN532, PN532\_HSU, PN532\_SPI et PN532\_I2C dans le dossier "libraries" d'Arduino.
4. Télécharger [NDEF bibliothèque de Don](#) dossier ZIP et extraire les fichiers.
5. Ouvrez le dossier extrait et renommez le dossier "NDEF-master" en "NDEF".
6. Copiez le dossier "NDEF" dans le dossier "libraries" d'Arduino.
7. Redémarrez l'IDE Arduino. Vous devriez maintenant pouvoir voir "NDEF" et "PN532" comme options dans le sous-menu Arduino "Exemples" (voir figure ci-dessous).



**Menu des bibliothèques disponibles Arduino**

## **Exemples NFC Bouclier / Applications**

### **Exemple n ° 1: Tag NFC scan**

Cet exemple vous montrera comment utiliser le bouclier NFC pour numériser une balise NFC et afficher ses informations / données.

Dans la copie IDE Arduino, collez, puis téléchargez le code ci-dessous à votre conseil.

## code de

```
#include <SPI.h>
#include "PN532_SPI.h"
#include "PN532.h"
#include "NfcAdapter.h"

PN532_SPI interface(SPI, 10); // create a PN532 SPI interface with the SPI
CS terminal located at digital pin 10
NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

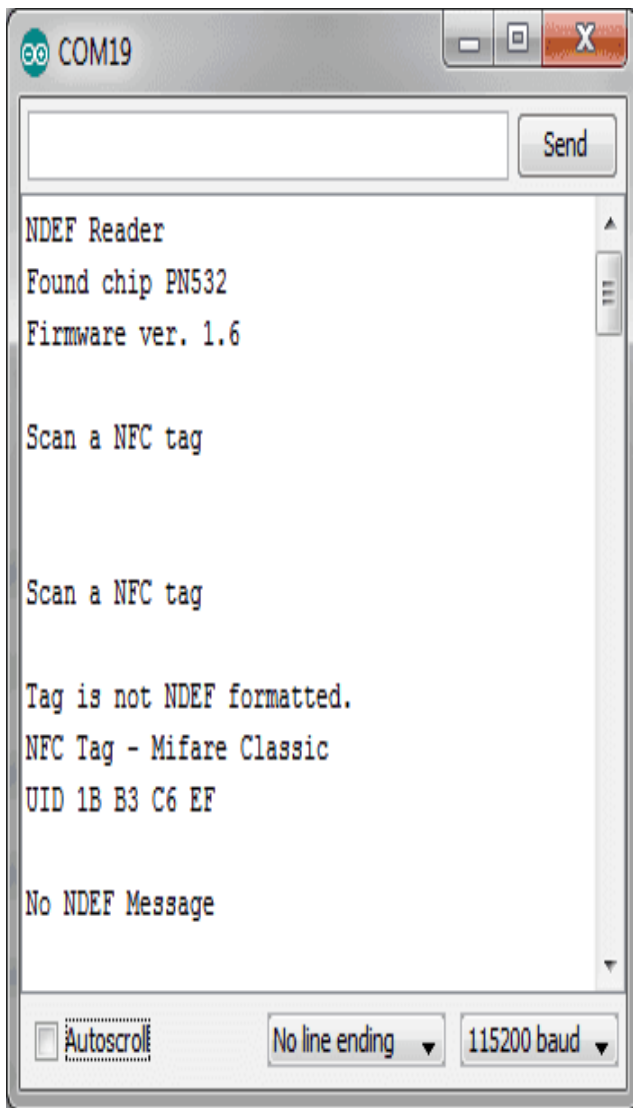
void setup(void) {
  Serial.begin(115200); // begin serial communication
  Serial.println("NDEF Reader");
  nfc.begin(); // begin NFC communication
}

void loop(void) {

  Serial.println("\nScan an NFC tag\n");
  if (nfc.tagPresent()) // Do an NFC scan to see if an NFC tag is present
  {
    NfcTag tag = nfc.read(); // read the NFC tag into an object,
nfc.read() returns an NfcTag object.
    tag.print(); // prints the NFC tags type, UID, and NDEF message (if
available)
  }
  delay(500); // wait half a second (500ms) before scanning again (you may
increment or decrement the wait time)
}
```

Pour tester le code:

1. Ouvrez la fenêtre du moniteur Arduino Serial
2. Définir le débit en bauds à 115200
3. Tenir un tag NFC sur la zone d'antenne NFC
4. Le bouclier NFC va scanner l'étiquette et vous devriez pouvoir voir l'UID, le type d'étiquette et le message (si disponibles) de la balise NFC dans la fenêtre du moniteur série. Voir la figure ci-dessous.

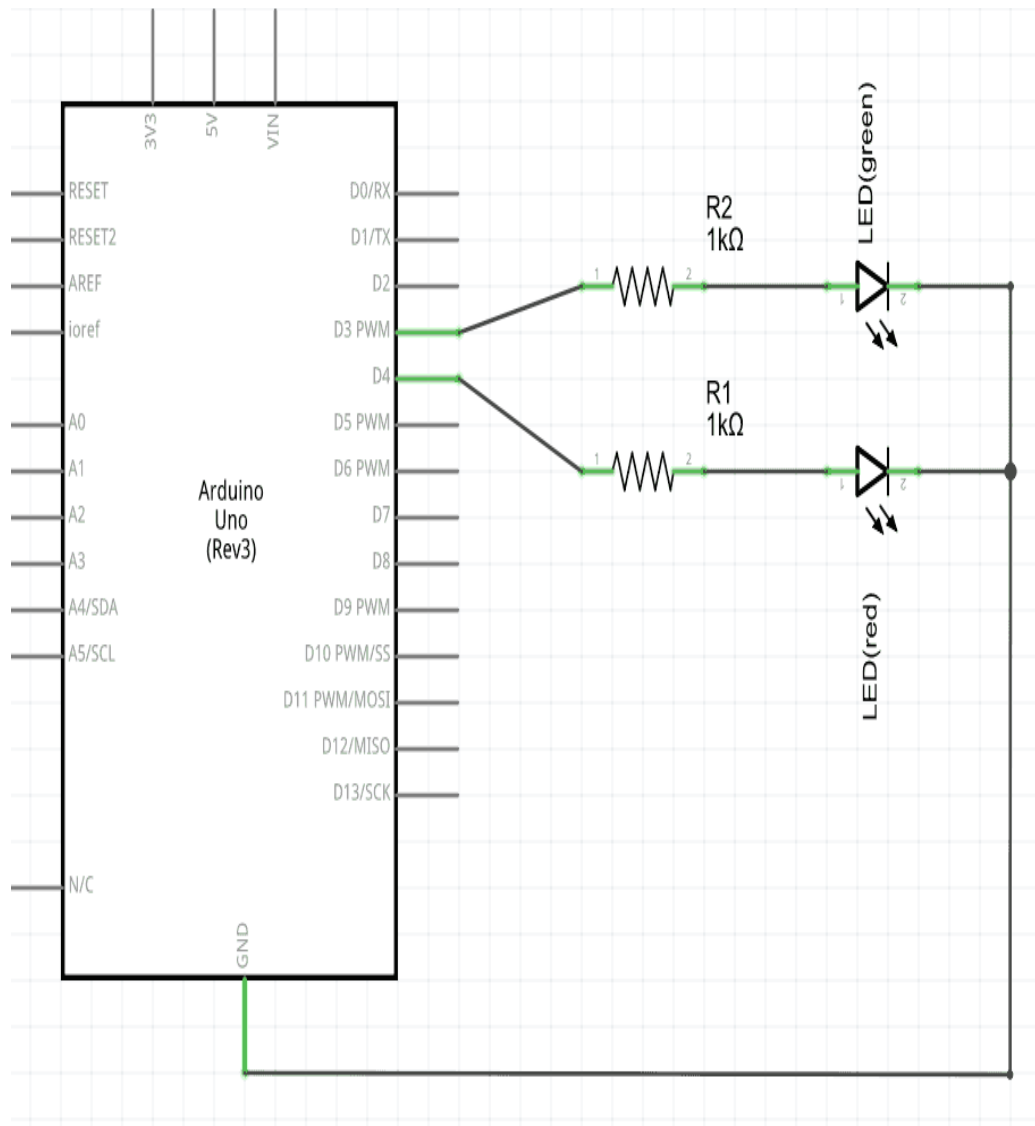


**Exemple # 1** sortie de fenêtre de communication série lors de la numérisation d'une balise NFC.

### **Exemple # 2: NFC (sans clé) de verrouillage de porte**

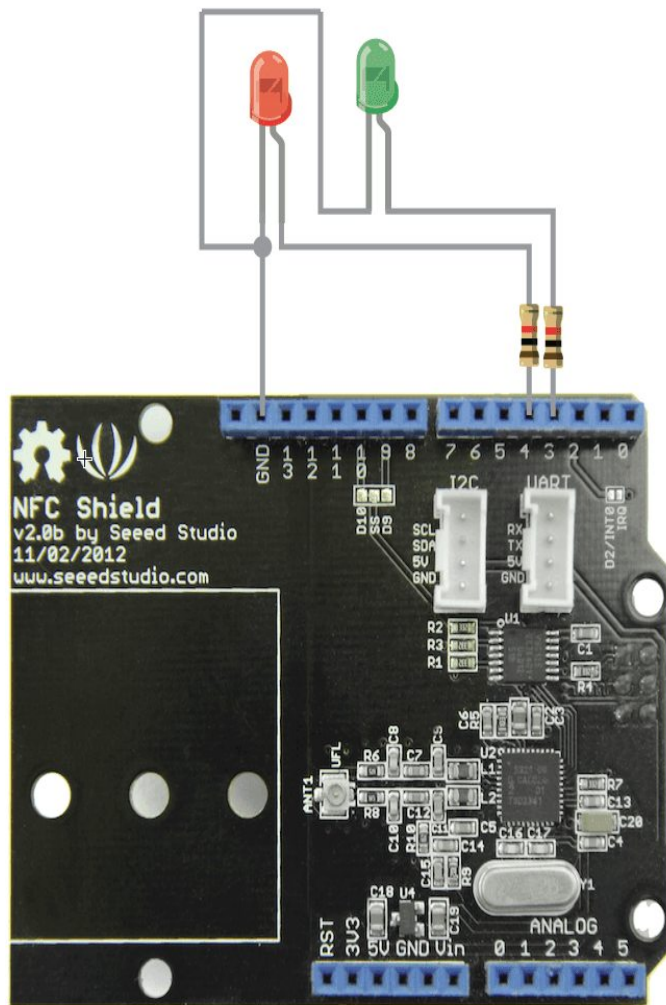
Cet exemple vous montrera comment utiliser une balise NFC comme clé pour déverrouiller une porte ou une serrure. Le mécanisme de la porte / de la serrure sera laissé à votre imagination, nous couvrirons seulement la partie NFC du code.

1. Ne Exemple # 1: NFC Tag Scan, ci-dessus, pour obtenir l'UID de votre tag NFC.
2. Étape optionnelle - branchez une diode verte sur la broche 3, comme indiqué sur la figure / schéma ci-dessous. Nous allons utiliser cette LED pour signaler une correspondance réussie dans les touches.
3. Étape optionnelle - branchez une LED rouge sur la broche 4 comme indiqué sur la figure / schéma ci-dessous. Nous allons utiliser cette LED pour signaler une touche mal adaptée.



**Circuit de verrouillage NFC**





### Circuit de verrouillage NFC

4. Dans l'IDE Arduino, créez un nouveau croquis et copiez, collez et téléchargez le code ci-dessous sur votre carte Arduino en remplaçant la constante de chaîne myUID par l'UID de votre tag obtenu à partir de l'exemple # 1.

#### code de

```
#include <SPI.h>

#include "PN532_SPI.h"
#include "PN532.h"
#include "NfcAdapter.h"

String const myUID = "1B B3 C6 EF"; // replace this UID with your NFC tag's
UID
int const greenLedPin = 3; // green led used for correct key notification
int const redLedPin = 4; // red led used for incorrect key notification

PN532_SPI interface(SPI, 10); // create a SPI interface for the shield with
the SPI CS terminal at digital pin 10
NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

void setup(void) {
  Serial.begin(115200); // start serial comm
  Serial.println("NDEF Reader");
  nfc.begin(); // begin NFC comm
```

```

    // make LED pins outputs
    pinMode(greenLedPin, OUTPUT);
    pinMode(redLedPin, OUTPUT);

    // turn off the LEDs
    digitalWrite(greenLedPin, LOW);
    digitalWrite(redLedPin, LOW);
}

void loop(void) {

    Serial.println("Scanning...");
    if (nfc.tagPresent()) // check if an NFC tag is present on the antenna
area
    {
        NfcTag tag = nfc.read(); // read the NFC tag
        String scannedUID = tag.getUidString(); // get the NFC tag's UID

        if( myUID.compareTo(scannedUID) == 0) // compare the NFC tag's UID
with the correct tag's UID (a match exists when compareTo returns 0)
        {
            // The correct NFC tag was used
            Serial.println("Correct Key");
            // Blink the green LED and make sure the RED led is off
            digitalWrite(greenLedPin, HIGH);
            digitalWrite(redLedPin, LOW);

            delay(500);
            digitalWrite(greenLedPin, LOW);
            delay(500);
            digitalWrite(greenLedPin, HIGH);
            delay(500);
            digitalWrite(greenLedPin, LOW);
            // put your here to trigger the unlocking mechanism (e.g. motor,
transducer)
        }else{
            // an incorrect NFC tag was used
            Serial.println("Incorrect key");
            // blink the red LED and make sure the green LED is off
            digitalWrite(greenLedPin, LOW);
            digitalWrite(redLedPin, HIGH);

            delay(500);
            digitalWrite(redLedPin, LOW);
            delay(500);
            digitalWrite(redLedPin, HIGH);
            delay(500);
            digitalWrite(redLedPin, LOW);
            // DO NOT UNLOCK! an incorrect NFC tag was used.
            // put your code here to trigger an alarm (e.g. buzzard, speaker)
or do something else
        }
    }
    delay(2000);
}

```

Pour tester le code / l'application:

1. Ouvrir la fenêtre du moniteur série d'Arduino

2. Maintenez l'étiquette NFC avec la clé correcte sur la zone de l'antenne.
3. La LED verte doit s'allumer et la fenêtre série doit imprimer "Correct Key"
4. Maintenez maintenant un NFC différent sur la zone d'antenne
5. La LED rouge doit s'allumer et la fenêtre série doit imprimer "Incorrect Key"

### Exemple # 3: Comment utiliser le Pin Interromp (Exemple n ° 2: Revisited)

Bien que le code dans l'exemple # 2 ci-dessus ne ce dont nous avons besoin il ya une approche plus élégante à la manipulation de marques NFC détections. Dans cet exemple, nous allons vous montrer comment faire pour utiliser la broche d'interruption dans le bouclier NFC de sorte que au lieu de sonder le bouclier (en demandant "y at-il une étiquette présente?") Nous attendons le bouclier pour dire à l'Arduino qu'une étiquette Est disponible pour être lu. Pourquoi voudriez-vous faire ceci? Il ya beaucoup de raisons et les interruptions sont un tout autre sujet, mais une raison qui peut vous convaincre est que votre projet / circuit permettra d'économiser de la batterie car nous ne déclenchons pas le bouclier circuit en continu.

#### Modification du matériel

La broche d'interruption du blindage NFC (IRQ) est débranchée de la broche numérique 2 (D2) de l'Arduino, pour connecter l'IRQ et la broche D2 ensemble, allez de l'avant et soudez le tampon sur le blindage «D2 / INT0 IRQ».

#### code de

Téléchargez le code suivant sur votre carte Arduino:

```
#include <SPI.h>
#include "PN532_SPI.h"
#include "PN532.h"
#include "NfcAdapter.h"

// FLAG_NONE used to signal nothing needs to be done
#define FLAG_NONE 0
// FLAG_IRQ_TRIGGERED used to signal an interrupt trigger
#define FLAG_IRQ_TRIGGERED 1
// FLAG_RESET_IRQ used to signal that the interrupt needs to be reset
#define FLAG_RESET_IRQ 2
// flags variable used to store the present flag
volatile int flags = FLAG_NONE;

String const myUID = "1B B3 C6 EF"; // replace this UID with your NFC tag's
UID
// LED pins
int const greenLedPin = 3; // green led used for correct key notification
int const redLedPin = 4; // red led used for incorrect key notification

// the interrupt we'll be using (interrupt 0) is located at digital pin 2
int const irqPin = 2; // interrupt pin

PN532_SPI interface(SPI, 10); // create a SPI interface for the shield with
the SPI CS terminal at digital pin 10

NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

String scannedUID = ""; // this is where we'll store the scanned tag's UID

void setup(void) {
```

```

// make LED pins outputs
pinMode(greenLedPin,OUTPUT);
pinMode(redLedPin,OUTPUT);

Serial.begin(115200); // start serial comm
Serial.println("NDEF Reader");
nfc.begin(); // begin NFC comm

// turn off the LEDs
digitalWrite(greenLedPin,LOW);
digitalWrite(redLedPin,LOW);
// attach the function "irq" to interrupt 0 on the falling edges
attachInterrupt(0,irq,FALLING); // digital pin 2 is interrupt 0, we'll
call the irq function (below) on the falling edge of this pin
}

void loop(void) {
    int flag = getFlag(); // get the present flag

    switch(flag) // check which flag/signal we are on
    {
        case FLAG_NONE:
            // nothing needs to be done
            break;
        case FLAG_IRQ_TRIGGERED: // the interrupt pin has been triggered
            Serial.println("Interrupt Triggered");
            if (nfc.tagPresent())
            {
                // an NFC tag is present
                NfcTag tag = nfc.read(); // read the NFC tag
                scannedUID = tag.getUidString(); // get the NFC tag's UID
                if(myUID.compareTo(scannedUID) == 0) // compare the NFC tag's
UID with the correct tag's UID (a match exists when compareTo returns 0)
                {
                    // the scanned NFC tag matches the saved myUID value
                    Serial.println("Correct tag/key");
                    blinkLed(greenLedPin,200,4); // blink the green led
                    // put your here to trigger the unlocking mechanism (e.g.
motor, transducer)
                }else{
                    // the scanned NFC tag's UDI does not match the myUID value
                    Serial.println("Incorrect tag/key");
                    blinkLed(redLedPin,200,4); // blink the red led
                    // DO NOT UNLOCK! an incorrect NFC tag was used.
                    // put your code here to trigger an alarm (e.g. buzzard,
speaker) or do something else
                }
                // return to the original state
                setFlag(FLAG_NONE);
                reset_PN532_IRQ_pin();
            }else{
                // a tag was not present (the IRQ was triggered by some other
action)
                setFlag(FLAG_NONE);
            }
            break;
        default:
            // do any other stuff for flags not handled above
            break;
    }
}

/*
* Name: setFlat

```

```

    * Description: used to set actions/flags to be executed in the loop(void)
function
    * Parameters:
    *     int flag - the action/flag to store
    * Returns: void
    */
void setFlag(int flag)
{
    flags = flag;
}

/*
* Name: getFlag
* Description: used to get the present flag/action
* Parameters: void
* Returns: int - the flags variable. The action/flag set by setFlag
*/
int getFlag()
{
    return flags;
}

/*
* Name: irq
* Description: Interrupt service routine (ISR). This function will be
executed whenever there is a falling edge on digital pin 2 (the interrupt 0 pin)
* Parameters: void
* Returns: void
*/
void irq()
{
    if(getFlag()==FLAG_NONE){
        setFlag(FLAG_IRQ_TRIGGERED);
    }
}

/*
* Name: reset_PN532_IRQ_pin
* Description: used to reset the PN532 interrupt request (IRQ) pin
* Parameters: void
* Returns: void
*/
void reset_PN532_IRQ_pin()
{
    nfc.tagPresent();
}

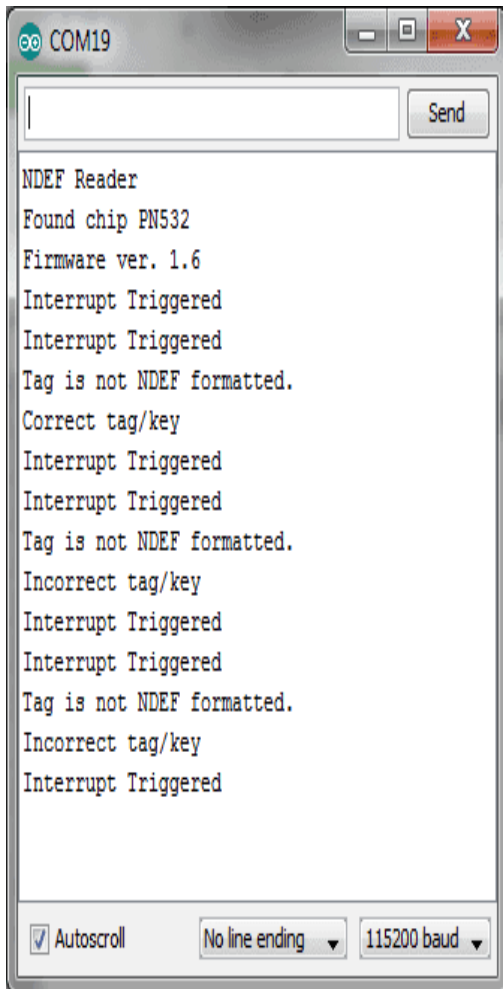
/*
* Name: blinkLed
* Description: used to toggle a pin to blink an LED attached to the pin
* Parameters:
*     ledPin - the pin where the led is connected to
*     delayTime - the time in milliseconds between HIGH and LOW
*     times - the number of times to toggle the pin
* Returns: void
*/
void blinkLed(int ledPin,int delayTime,int times)
{
    for(int i=0;i<times;i++){
        digitalWrite(ledPin,HIGH);
        delay(delayTime);
        digitalWrite(ledPin,LOW);
        delay(delayTime);
    }
}

```

Pour tester le code / l'application:

1. Si désiré, connectez les voyants comme indiqué dans l'exemple # 2 ci-dessus.
2. Ouvrir la fenêtre du moniteur série d'Arduino
3. Maintenez l'étiquette NFC avec la clé correcte sur la zone de l'antenne.
4. La LED verte doit s'allumer et la fenêtre série doit imprimer "Correct Key"
5. Maintenez maintenant un NFC différent sur la zone d'antenne
6. La LED rouge doit s'allumer et la fenêtre série doit imprimer "Incorrect Key"

La fenêtre sérielle de notre test de ce code est affichée ci-dessous, la vôtre devrait être similaire.



**Serial comm window output de l'exemple 3.**

#### **Exemple # 4: Ecrire un message NDEF à une marque**

Les balises NFC sont capables de stocker des données, la quantité de données dépend de chaque balise. Dans cet exemple , nous allons stocker deux chaînes / messages sur un tag, vous serez alors en mesure de lire ce message avec le code dans l' *exemple n ° 6: lire un message NDEF Du Tag.*

Téléchargez le code suivant dans votre carte de développement Arduino.

Remarque

Si votre tag NFC est pas correctement formaté ( "message écriture a échoué" sera affiché dans la fenêtre de série de comm ) , vous aurez besoin de voir si vous tag peut être formaté avec le code

dans l' exemple n ° 5: *Formater une marque comme NDEF*

#### code de

```
#include <SPI.h>

#include "PN532_SPI.h"
#include "PN532.h"
#include "NfcAdapter.h"

PN532_SPI interface(SPI, 10); // create a SPI interface for the shield with
the SPI CS terminal at digital pin 10

NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

void setup(void)
{
    Serial.begin(115200); // start serial comm
    Serial.println("NDEF Reader");
    nfc.begin(); // begin NFC comm
}

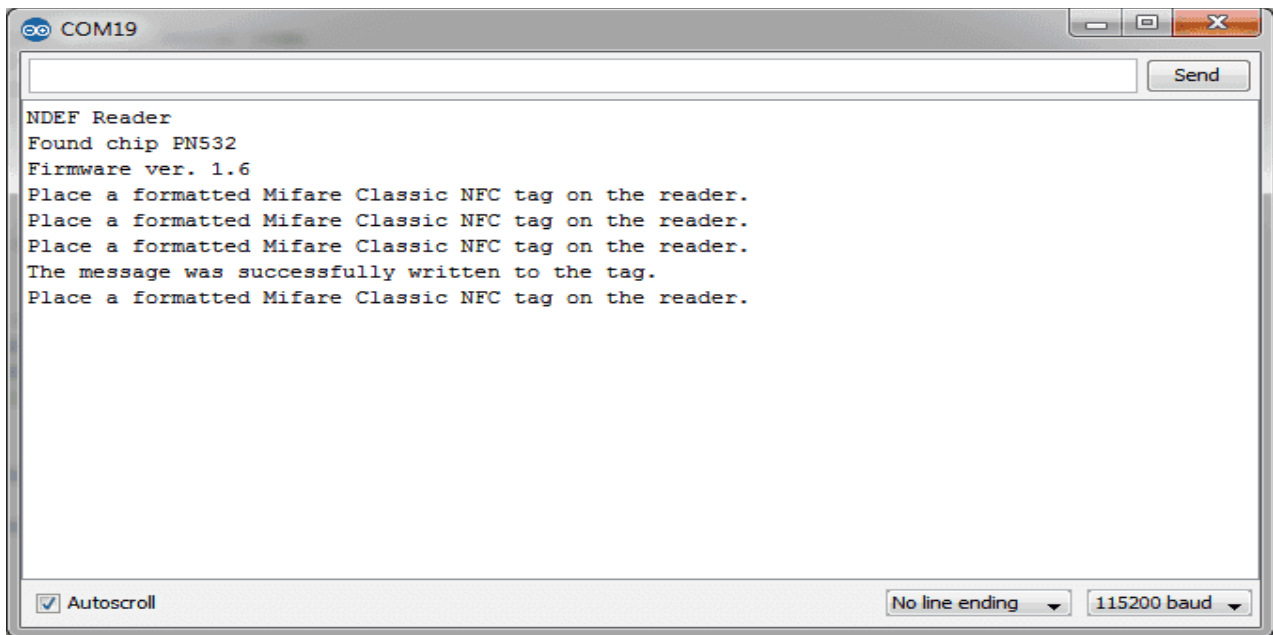
void loop(void)
{
    Serial.println("Place a formatted Mifare Classic NFC tag on the reader.");
    if(nfc.tagPresent())
    {
        NdefMessage message = NdefMessage();
        message.addUriRecord("Hello, world!");
        message.addUriRecord("How are you today?");

        bool success = nfc.write(message);
        if(success)
        {
            Serial.println("The message was successfully written to the tag.");Ho
        }else{
            Serial.println("Message write failed.");
        }
    }

    delay(5000);
}
```

Pour tester le code ci-dessus:

1. Ouvrir une fenêtre de communication série Arduino
2. Maintenez la balise NFC sur la zone de l'antenne de protection NFC et attendez que le message de succès ou d'échec apparaisse comme indiqué sur la figure ci-dessous.
3. Retirez l'étiquette NFC de la zone de l'antenne dès que le message de succès est affiché pour empêcher une réécriture.



**Serial comm fenêtre pour NDEF message écrit à la carte exemple.**

### Exemple n ° 5: Mettre en forme une marque comme NDEF

Votre balise NFC nouvelle marque peut-être pas NDEF au format initial. Pour formater une balise comme NDEF, téléchargez le code suivant sur votre carte de développement Arduino:

code de

```
#include <SPI.h>

#include "PN532_SPI.h"
#include "PN532.h"
#include "NfcAdapter.h"

PN532_SPI interface(SPI, 10); // create a SPI interface for the shield with
the SPI CS terminal at digital pin 10

NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

void setup(void)
{
    Serial.begin(115200); // start serial comm
    Serial.println("NDEF Reader");
    nfc.begin(); // begin NFC comm
}

void loop(void)
{
    Serial.println("Place an unformatted Mifare Classic tag on the
reader.");
    if (nfc.tagPresent()) {
        bool success = nfc.format();
        if (success) {
            Serial.println("Success, tag formatted as NDEF.");
        } else {
            Serial.println("Format failed.");
        }
    }
}
```



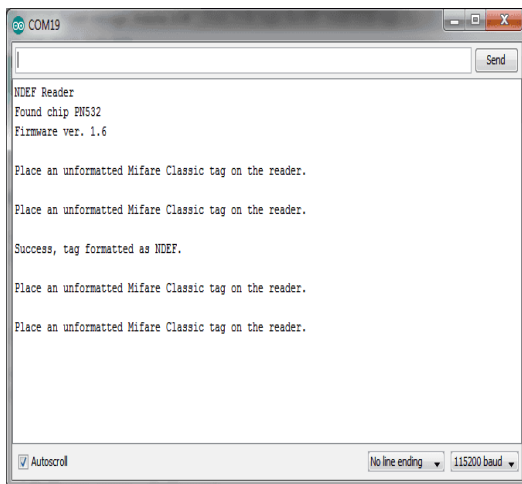
```
}  
  delay(5000);  
}
```

Pour tester / exécuter le code:

1. Ouvrez la fenêtre de communication série Arduino.
2. Maintenez la balise NFC que vous souhaitez formater sur la zone de l'antenne de protection NFC.
3. Attendez que le message de réussite ou d'échec s'affiche comme indiqué dans la figure ci-dessous.
4. Retirez l'étiquette NFC de la zone de l'antenne pour éviter un nouveau format.

Remarque

Si votre balise ne parvient pas à être formatée, réessayez. S'il échoue, votre balise ne peut pas être formatée comme NDEF.



**Serial comm sortie de fenêtre lors du formatage d'une balise NFC à NDEF.**

## Exemple # 6: lire un message NDEF D'un Tag

Comme vous l'avez vu dans l'exemple ci-dessus, le bouclier NFC est capable d'écrire des messages sur les balises NFC. Le NFC est également capable de lire des messages NDEF à partir de balises, dans cet exemple nous allons vous montrer comment.

code de

Téléchargez le code suivant dans votre carte de développement Arduino.

```
#include <SPI.h>  
  
#include "PN532_SPI.h"  
#include "PN532.h"  
#include "NfcAdapter.h"  
  
PN532_SPI interface(SPI, 10); // create a SPI interface for the shield with  
the SPI CS terminal at digital pin 10
```

```

NfcAdapter nfc = NfcAdapter(interface); // create an NFC adapter object

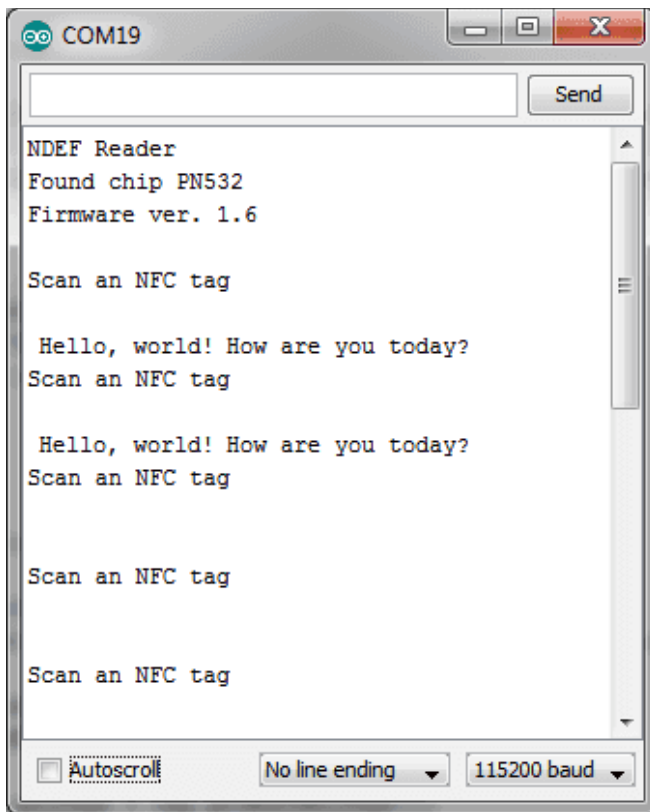
void setup(void)
{
    Serial.begin(115200); // start serial comm
    Serial.println("NDEF Reader");
    nfc.begin(); // begin NFC comm
}

void loop(void)
{
    Serial.println("\nScan an NFC tag\n");
    if (nfc.tagPresent()) // Do an NFC scan to see if an NFC tag is present
    {
        NfcTag tag = nfc.read(); // read the NFC tag
        if(tag.hasNdefMessage())
        {
            NdefMessage message = tag.getNdefMessage();
            for(int i=0;i<message.getRecordCount();i++)
            {
                NdefRecord record = message.getRecord(i);
                int payloadLength = record.getPayloadLength();
                byte payload[payloadLength];
                record.getPayload(payload);
                Serial.write(payload,payloadLength);
            }
        }
        delay(500); // wait half a second (500ms) before scanning again (you may
increment or decrement the wait time)
    }
}

```

Pour tester le code ci-dessus:

1. Ouvrir une fenêtre de communication série Arduino
2. Maintenez la balise NFC avec un message NDEF sur la zone de l'antenne de protection NFC.
3. Le message NDEF écrit sur l'étiquette doit être affiché comme indiqué dans la figure ci-dessous.



**Sortie de la fenêtre de communication série pour le message NDEF lu**

## Exemple # 7: Comment changer le Chip Select Pin De D10 à D9

### Modification du matériel

1. Gratter la connexion à partir des patins étiquetés "SS" et "D10" sur le bouclier
2. Raccorder / souder les patins "SS" et "D9" sur le bouclier.

Vous pouvez alors utiliser le même code dans les exemples ci-dessus mais avec la broche 9 au lieu de 10 pour l'interface PN532:

### code de

```
PN532_SPI interface(SPI, 9); // create a SPI interface for the shield with the  
SPI CS terminal at digital pin 9
```

## Exemple # 8: Utilisez deux NFC Shields avec un Arduino

### Modification du matériel

1. Effectuez la modification matérielle décrite dans l'exemple # 7 sur l'un des deux boucliers.
2. Empiler les deux boucliers sur la carte Arduino.

Vous pouvez maintenant créer deux objets NFC distincts, un pour chaque blindage, comme suit:

### code de

```
PN532_SPI interface_shield_1(SPI, 10); // create a SPI interface for the shield  
with the SPI CS terminal at digital pin 10  
  
PN532_SPI interface_shield_2(SPI, 9); // create a SPI interface for the
```

```
shield with the SPI CS terminal at digital pin 9
```

```
NfcAdapter nfc_shield_1 = NfcAdapter(interface_shield_1); // create an NFC  
adapter object for shield one  
NfcAdapter nfc_shield_2 = NfcAdapter(interface_shield_2); // create an NFC  
adapter object for shield two
```

## Ressources

- [Schéma NFC Shield v2.0](#)
- [Fichier Eagle NFC Shield v2.0](#)
- [NFC Shield v2.1 Schéma](#)
- [Fichier Eagle NFC Shield v2.1](#)
- [Bibliothèque PN532 SPI pour NFC Shield v2.0](#)
- [Fiche technique PN532](#)
- [FAQ du bouclier NFC](#)

## Aidez - nous à mieux

Merci d'avoir choisi [Seeed](#) . Il y a quelques mois, nous avons lancé un projet visant à améliorer notre système de documentation. Ce que vous regardez maintenant est la première édition du nouveau système de documentation. Comparé à l'ancien, voici les progrès que nous avons faits:

- Remplacé l'ancien système de documentation par un nouveau qui a été développé à partir de Mkdocs, un outil plus largement utilisé et plus frais pour développer le système de documentation.
- Intégré le système de documentation avec notre site officiel, maintenant vous pouvez aller à Bazaar et d'autres sections comme Forum et la Communauté plus facilement.
- Révisé et réécrit des documents pour des centaines de produits pour la première édition du système, et continuera à migrer des documents du vieux wiki vers le nouveau.

Une instruction facile à utiliser est aussi importante que le produit lui-même. Nous nous attendons à ce que ce nouveau système améliore votre expérience lors de l'utilisation des produits de Seeed. Cependant , puisque ceci est la première édition, il y a encore beaucoup de choses ont besoin d'améliorer, si vous avez des suggestions ou conclusions, vous êtes les bienvenus pour soumettre la version modifiée comme notre contributeur (S'il vous plaît se référer à la tutotrial [Comment modifier un document](#) ) ou Nous donner des suggestions dans le sondage ci-dessous, S'il vous plaît n'oubliez pas de laisser votre adresse e-mail afin que nous puissions répondre.