Nom : .....

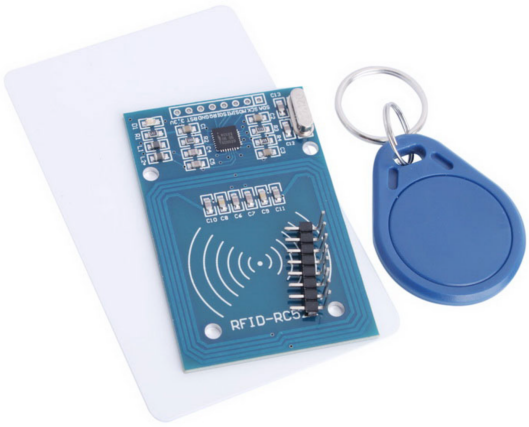
OBJECTIF :
SUPPORT D'IDENTIFICATION, LECTEURS, UNITÉS DE TRAITEMENT, GESTION DE COMMUNICATION

L'utilisation MF RC522 met en oeuvre la modulation et la démodulation d'une onde radio ayant une fréquence de 13.56MHz. Ces deux techniques sont complètement intégrées avec des méthodes de communication sans contact. La partie numérique gère les techniques ISO14443A, la détection d'erreur, l'algorithme de cryptage Quick CRYPTO1, et le terme de vérification série MIFARE.

Le module MFRC522 prend en charge le protocole de communications sans contact à haute vitesse MIFARE, avec des taux de transfert de données bidirectionnels jusqu'à 424 kbit/s. Le module MF522 utilise un composant d'origine Philips MFRC522 comme lecteur. Il est facile à utiliser, faible coût et adapté au développement de syst§mes. Ce module peut être directement adapté à une multitude de lecteurs de différentes formes. Le module utilise une tension de 3.3V. Avec seulement quelques lignes de code simples qui programment l'interface SPI, tout utilisateur connecte directement le CPU au module de communication. Cela offre en outre un travail stable et fiable, quel que soit la distance entre le lecteur et le tag RFID.

MATERIEL

Current:13-26mA / DC 3.3V
Idle Current:10-13mA / DC 3.3V
Sleep current: <80uA
Peak current: <30mA
Operating Frequency: 13.56MHz
Environmental Operating temperature: -20 to 80 degrees Celsius
Environment Storage temperature: -40 to 85 degrees Celsius
Relative Humidity: 5% -95%



tag RFID et son lecteurs

LIRE LE TAG

1 Est-ce que le tag RFID comporte une source d'énergie comme une pile ?

.....

2 Quelle est alors l'origine de l'énergie faisant marcher les composants sur le tag RFID ?

.....

3 A quel composant d'électronique de puissance vous fait penser ce transfert d'énergie entre le lecteur et le tag RFID ?

.....

4 Installer la librairie du RC522 (voir figure 1) et la tester.

5 Réaliser le cablage entre le lecteur et la carte du microcontrolleur.

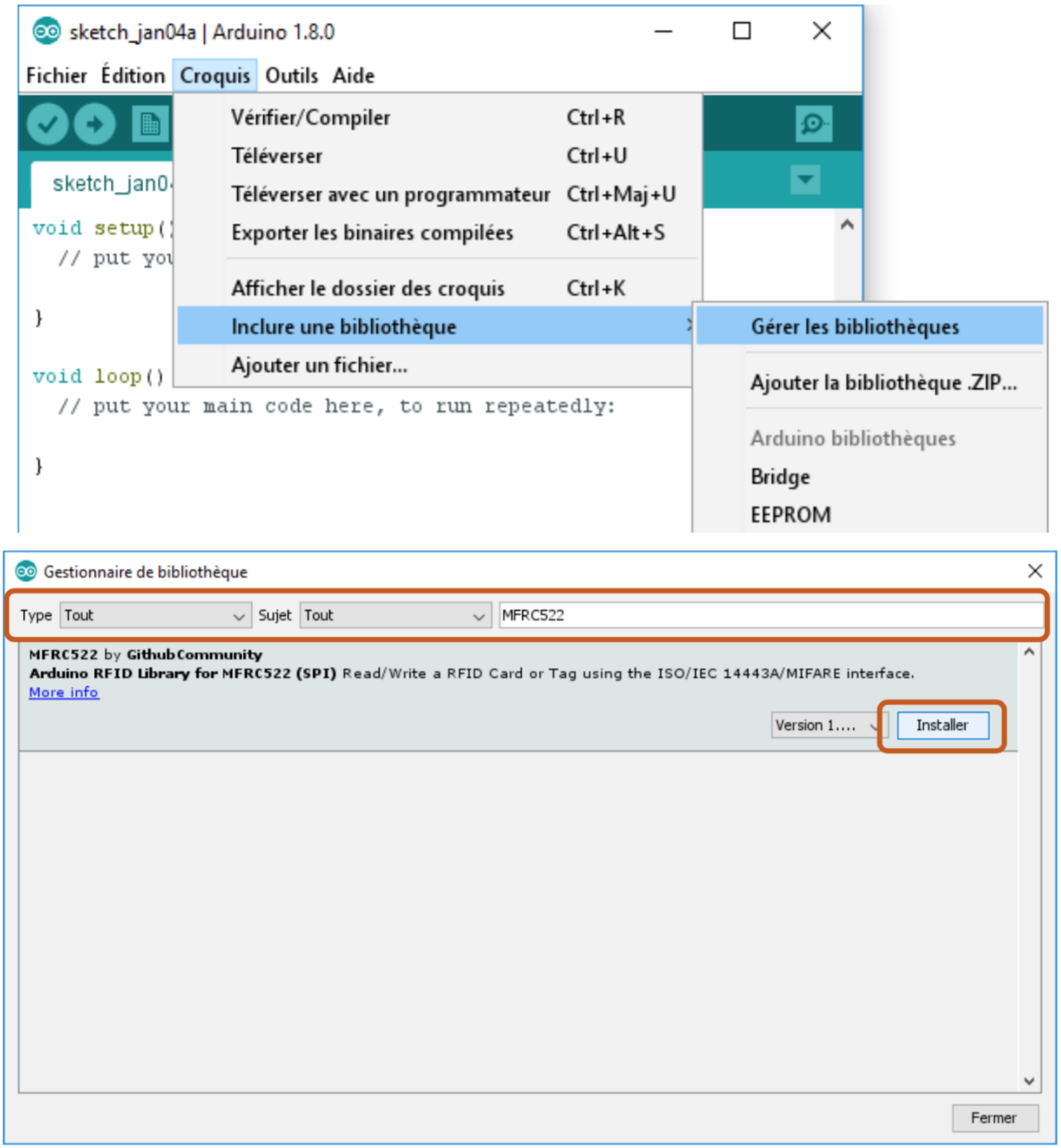
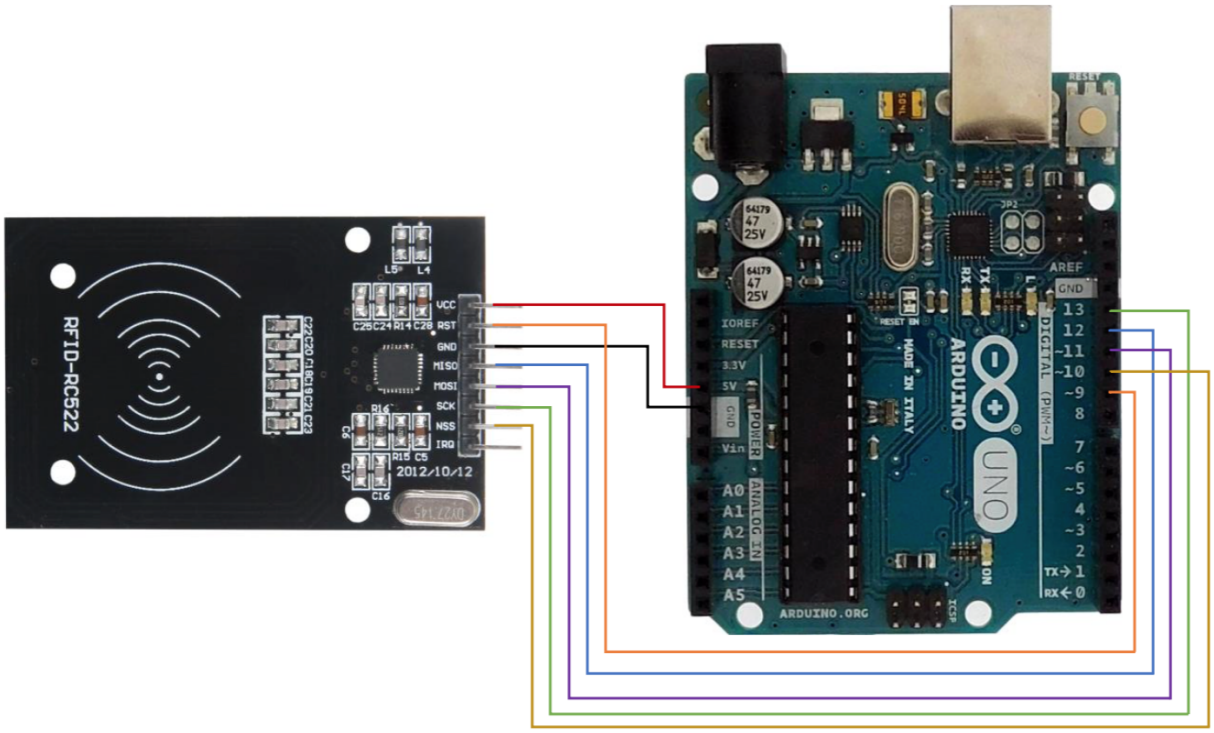


Figure 1: installation de la librairie SPI pour la carte RFID



RFID RC522	Arduino
VCC	+5V
RST	9
GND	GND
MISO	12
MOSI	11
SCK	13
NSS	10
IRC	/

Figure 2: Connection du lecteur RFID à la carte arduino

6 Quel est le type du bus qui relie le microcontrolleur arduino et le lecteur de tag RFID ?

.....

7 Decrire la fonction de chaque fil ?

.....

.....

.....

8 Implémentez le code suivant : [1]. Les commentaires aident à comprendre le code. Il n'est pas nécessaire de les implémenter puisque nous les avons sur le sujet. Habituellement, la documentation du code est très importante.

```
1 /*
2  * MFRC522 - Library to use ARDUINO RFID MODULE KIT 13.56 MHZ WITH TAGS SPI W AND R BY COOQROBOT.
3  * The library file MFRC522.h has a wealth of useful info. Please read it.
```

[1] http://makecourse.weebly.com/week10segment1.html

```
4   * The functions are documented in MFRC522.cpp.
5   *
6   * Based on code Dr.Leong   ( WWW.B2CQSHOP.COM )
7   * Created by Miguel Balboa (circuitito.com), Jan, 2012.
8   * Rewritten by Soren Thing Andersen (access.thing.dk), fall of 2013
9   * (Translation to English, refactored, comments, anti collision, cascade levels.)
10  * Released into the public domain.
11  *
12  * Sample program showing how to read data from a PICC using a MFRC522 reader on the Arduino SPI interface.
13  *---------------------------------------------------------------------------- empty_skull
14  * Aggiunti pin per arduino Mega
15  * add pin configuration for arduino mega
16  * http://mac86project.altervista.org/
17  ---------------------------------------------------------------------------- Nicola Coppola
18  * Pin layout should be as follows:
19  * Signal     Pin              Pin              Pin
20  *            Arduino Uno      Arduino Mega     MFRC522 board
21  *------------------------------------------------------------
22  * Reset      9                5                RST
23  * SPI SS     10               53               SDA
24  * SPI MOSI   11               52               MOSI
25  * SPI MISO   12               51               MISO
26  * SPI SCK    13               50               SCK
27  *
28  * The reader can be found on eBay for around 5 dollars. Search for "mf-rc522" on ebay.com.
29  */
30
31  #include <SPI.h>
32  #include <MFRC522.h>
33
34  #define SS_PIN 10
35  #define RST_PIN 9
36  MFRC522 mfrc522(SS_PIN, RST_PIN);        // Create MFRC522 instance.
37
38  void setup() {
39    Serial.begin(9600);   // Initialize serial communications with the PC
40    SPI.begin();                  // Init SPI bus
41    mfrc522.PCD_Init();   // Init MFRC522 card
42    Serial.println("Scan_PICC_to_see_UID_and_type...");
43  }
44
45  void loop() {
46    // Look for new cards
47    if ( ! mfrc522.PICC_IsNewCardPresent()) {
48      return;//go to start of loop if there is no card present
49    }
50    // Select one of the cards
51    if ( ! mfrc522.PICC_ReadCardSerial()) {
52      //if ReadCardSerial returns 1,
53      // the "uid" struct (see MFRC522.h lines 238-45)) contains the ID of the read card.
54      return;
55    }
56    // Dump debug info about the card. PICC_HaltA() is automatically called.
57    mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
58  }
```

9 Montrer que la lecture de la carte marche

Constatation professeur :

10 A quoi sert l'instruction `#include` ?

.....

11 A quoi sert l'instruction `#define` ?

.....

12 Quel est le type de la variable mfrc522 ?

.....

13 Quel est le type de la variable Serial ?

.....

14 Quel est le type de la variable SPI ?

.....

15 Quelles méthodes de l'objet mfrc522 sont utilisées dans ce code ?

. . . . . .

. . . . . .

. . . . . .

. . . . . .

ECRIRE ET LIRE LE TAG

16 Dans le code source suivant, quelle nouvelle variable globale a t'il été crée ?

. . . . . .

17 Est-ce que la déclaration des connections a changé ?

. . . . . .

18 Quel est le type de la variable block ? Combien d'octets occupe t'elle en mémoire ? Combien de bits ?

. . . . . .

19 Quel est le type de la variable blockcontent ? Combien d'octets occupe t'elle en mémoire ? Combien de bits ?

. . . . . .

20 Quel est le type de la variable readbackblock ? Combien d'octets occupe t'elle en mémoire ? Combien de bits ?

. . . . . .

21 A la ligne 111, la boucle `for` va jusqu'à 16. Pourquoi ?

. . . . . .

22 Que fait l'instruction à la ligne 113 ?

. . . . . .

23 Implémentez le code suivant et faites marcher l'écriture-lecture du tag.

```
1
2  /*****************************************
3          PURPOSE:        Learn to use the MF522-AN RFID card reader
4          Created by      Rudy Schlaf for www.makecourse.com
5          DATE:           2/2014
6  *****************************************/
7
8  /*
9   * This sketch uses the MFRC522 Library to use ARDUINO RFID MODULE KIT 13.56 MHZ WITH TAGS SPI W
10  * AND R BY COOQROBOT.
11  * The library file MFRC522.h has a wealth of useful info. Please read it.
12  * The functions are documented in MFRC522.cpp.
13  *
14  * Based on code Dr.Leong   ( WWW.B2CQSHOP.COM )
15  * Created by Miguel Balboa (circuitito.com), Jan, 2012.
16  * Rewritten by Soren Thing Andersen (access.thing.dk), fall of 2013 (Translation to English,
17  * refactored, comments, anti collision, cascade levels.)
18  *
19  * This library has been released into the public domain.
20  */
21
22
23 #include <SPI.h>//include the SPI bus library
24 #include <MFRC522.h>//include the RFID reader library
25
26 #define SS_PIN 10  //slave select pin
27 #define RST_PIN 5  //reset pin
28 MFRC522 mfrc522(SS_PIN, RST_PIN);        // instatiate a MFRC522 reader object.
29 MFRC522::MIFARE_Key key;//create a MIFARE_Key struct named 'key', which will hold the card information
30
31
```

```
void setup() {
        Serial.begin(9600);         // Initialize serial communications with the PC
        SPI.begin();                // Init SPI bus
        mfrc522.PCD_Init();         // Init MFRC522 card (in case you wonder what PCD means: proximity
                                    // coupling device)
        Serial.println("Scan a MIFARE Classic card");

        // Prepare the security key for the read and write functions - all six key bytes are set to
        //   0xFF at chip delivery from the factory.
        // Since the cards in the kit are new and the keys were never defined, they are 0xFF
        // if we had a card that was programmed by someone else, we would need to know the key to
        //   be able to access it. This key would then need to be stored in 'key' instead.

        for (byte i = 0; i < 6; i++) {
                key.keyByte[i] = 0xFF;//keyByte is defined in the "MIFARE_Key" 'struct' definition in the
                                      //  .h file of the library
        }

}

int block=2; //this is the block number we will write into and then read. Do not write
             //  into 'sector trailer' block, since this can make the block unusable.

//an array with 16 bytes to be written into one of the 64 card blocks is defined
byte blockcontent[16] = {"makecourse_____"};
//byte blockcontent[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};//all zeros.
//This can be used to delete a block.
//This array is used for reading out a block. The MIFARE_Read method
//requires a buffer that is at least 18 bytes to hold the 16 bytes of a block.
byte readbackblock[18];

void loop()
{

  /*********************************establishing contact with a tag/card**********************************************************/

  // Look for new cards (in case you wonder what PICC means: proximity integrated circuit card)
  if ( ! mfrc522.PICC_IsNewCardPresent()) {//if PICC_IsNewCardPresent returns 1, a new card has been found and we continue
    return;//if it did not find a new card is returns a '0' and we return to the start of the loop
  }

  // Select one of the cards
  //if PICC_ReadCardSerial returns 1, the "uid" struct (see MFRC522.h lines 238-45)) contains the ID of the read card.
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    //if it returns a '0' something went wrong and we return to the start of the loop
    return;
  }

  // Among other things, the PICC_ReadCardSerial() method reads the UID and the SAK (Select acknowledge) into the mfrc522.uid struct,
  //   which is also instantiated during this process.
  // The UID is needed during the authentication process
  //The Uid struct:
  //typedef struct {
  //byte               size;                // Number of bytes in the UID. 4, 7 or 10.
  //byte               uidByte[10];         //the user ID in 10 bytes.
  //byte               sak;                 // The SAK (Select acknowledge) byte returned from the PICC after successful selection.
  //} Uid;

  Serial.println("card selected");

  /*********************************writing and reading a block on the card**********************************************************/

  writeBlock(block, blockcontent);//the blockcontent array is written into the card block
  //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

  //The 'PICC_DumpToSerial' method 'dumps' the entire MIFARE data block into the serial monitor.
  //Very useful while programming a sketch with the RFID reader...
  //Notes:
  //(1) MIFARE cards conceal key A in all trailer blocks, and shows 0x00 instead of 0xFF.
  //    This is a secutiry feature. Key B appears to be public by default.
  //(2) The card needs to be on the reader for the entire duration of the dump.
  //    If it is removed prematurely, the dump interrupts and an error message will appear.
  //(3) The dump takes longer than the time alloted for interaction per pairing between reader and card,
  //    i.e. the readBlock function below will produce a timeout if the dump is used.

  //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));//uncomment this if you want to see the entire 1k memory with the block written into it.

  readBlock(block, readbackblock);//read the block back
  Serial.print("read block: ");
  for (int j=0 ; j<16 ; j++)//print the block contents
  {
    Serial.write (readbackblock[j]);//Serial.write() transmits the ASCII numbers as human readable characters to serial monitor
  }
  Serial.println("");
 }
```

Constatation professeur :