

Sans titre

*

*

* Example to change UID of changeable MIFARE card.

*

* This is a MFRC522 library example; for further details and other examples see: <https://github.com/miguelbalboa/rfid>

*

* This sample shows how to set the UID on a UID changeable MIFARE card.

* NOTE: for more informations read the README.rst

*

* @author Tom Clement

* @license Released into the public domain.

*

* Typical pin layout used:

*

*
Arduino MFRC522 Arduino Arduino Arduino Arduino
* Reader/PCD Uno/101 Mega Nano v3 Leonardo/Micro
Pro Micro
* Signal Pin Pin Pin Pin Pin
Pin
*

* RST/Reset RST 9 5 D9 RESET/ICSP-5
RST
* SPI SS SDA(SS) 10 53 D10 10
10
* SPI MOSI MOSI 11 / ICSP-4 51 D11 ICSP-4
16
* SPI MISO MISO 12 / ICSP-1 50 D12 ICSP-1
14
* SPI SCK SCK 13 / ICSP-3 52 D13 ICSP-3
15
*/

/* V2.0 du 05/10

* Modifié par JCS

* - Traduction des commentaires en francais

* - Ajout d'un LED en pin 5

* V2.1 du 5/10

* Modifié par JCS

* Flush pour detecter si la carte est toujours présente

*/

Sans titre

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

#define RST_PIN 9 // Valable pour un Arduino Uno
#define SS_PIN 10 // Valable pour un Arduino Uno
#define FBAdr 1 //feedback adress in Rocrail
boolean debug = true; //false: prepare output for RR, true:huaman readable
outputp for terminal window

MFRC522 mfrc522(SS_PIN, RST_PIN); // Cr ation d'un instance MF RC5200 (MiFaire
RC522)
MFRC522::MIFARE_Key key;

int pinLed= 5; //D finition du port pour la LED
byte GainDB;
String version="2.1";
String date_modif="05/10/2017";

String oldRFID = "";
String RFID = "";
int sendEndCount = 3;
int Intervall = 3000;

void setup() {
  Serial.begin(9600); // Initialisation de la communication serie
  Serial.println("http://smartpoker.jimdo.com - Test du MFRC522");
  Serial.print ("version : " );
  Serial.println(version);
  Serial.print("Date derniere modification : ");
  Serial.println(date_modif);
  while (!Serial); // NE RIEN FAIRE tant que le port Serie n'est pas ouvert
  pinMode(pinLed,OUTPUT); // initialisation du port
  // Test de la LEDn allumage pendant 2 secondes
  Serial.println("Test de la LED - Allumage pendant 5 secondes");
  digitalWrite(pinLed, HIGH);
  delay(5000);
  digitalWrite(pinLed, LOW);
  // Fin test de la LED
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card

  // Prepare key - all keys are set to FFFFFFFFh at chip delivery from the
factory.
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  } //EndWhile
```

Sans titre

```
//Affichage du Gain de l'antenne
GainDB=0;
GainDB=mfr522.PCD_GetAntennaGain();
Serial.println("L'affichage du GainDB permet de connaitre la distance de
lecture");
Serial.println("La valeur : 40 correspond à 0x04, c'est la valeur par
defaut");

/* Valeurs extraites de la bibliotheque
* RxGain_18dB      = 0x00 << 4, // 000b - 18 dB, minimum
RxGain_23dB      = 0x01 << 4, // 001b - 23 dB
RxGain_18dB_2    = 0x02 << 4, // 010b - 18 dB, it seems 010b is a
duplicate for 000b
RxGain_23dB_2    = 0x03 << 4, // 011b - 23 dB, it seems 011b is a
duplicate for 001b
RxGain_33dB      = 0x04 << 4, // 100b - 33 dB, average, and typical
default
RxGain_38dB      = 0x05 << 4, // 101b - 38 dB
RxGain_43dB      = 0x06 << 4, // 110b - 43 dB
RxGain_48dB      = 0x07 << 4, // 111b - 48 dB, maximum
RxGain_min       = 0x00 << 4, // 000b - 18 dB, minimum, convenience for
RxGain_18dB
RxGain_avg       = 0x04 << 4, // 100b - 33 dB, average, convenience for
RxGain_33dB
RxGain_max = 0x07 << 4 // 111b - 48 dB, maximum, convenience for RxGain_48dB
*/
Serial.print ("GainDB=");
Serial.println(GainDB,HEX);
Serial.println("Augmentation du Gain au Max");
mfr522.PCD_SetAntennaGain(0x07<<4);
GainDB=mfr522.PCD_GetAntennaGain();
Serial.println("Affichage du Nouveau Gain");
GainDB=mfr522.PCD_GetAntennaGain();
Serial.print ("GainDB=");
Serial.println(GainDB,HEX);

/* La modification du GAIN doit se faire apres l'appel à PCD_INIT()
* Sinon la valeur du Gain est remise à sa valeur par defaut
*/

Serial.println("SMARTPOKER : Detection du changement d'UID ou du retrait de la
carte"); // Affichage sur le port Serie du message
// Le TAG est pres à etre Lu

} //EndSetup

void loop() {

sendEndCount++;
Serial.print("SenEndCount=");
Serial.println(sendEndCount);
```

Sans titre

```
if (sendEndCount >10000 )
{
  //Eviter un depassement de pile
  sendEndCount=5;
}

if (!mfr522.PICC_IsNewCardPresent()) {
  return;
  Serial.println("Pas de carte détectée");
}

if (mfr522.PICC_ReadCardSerial()) {
  digitalWrite(pinLed,HIGH);
  RFID = mkString(mfr522.uid.uidByte, mfr522.uid.size);
  Serial.println(RFID);
}

if (RFID == oldRFID)
{
  Serial.println("detection identique");
}
else //Else1
{
  Serial.println("detection d'un changement");
  // Cas des TAG ayant un UID sur 4Octets
  if (mfr522.uid.size = 4)
  {
    oldRFID = RFID;
  } //End test taille=4

  // cas des tags ayant un UID sur 7 Octet, dans ce cas enlever le LSB
  if (mfr522.uid.size > 4) {
    oldRFID = RFID;

  } //end test taille >4

  sendEndCount = 0;
} //End Else1

digitalWrite(pinLed,LOW);
Serial.println("Controle");
Serial.print("OLD =");
Serial.println(oldRFID);
Serial.print("NEW =");
Serial.println(RFID);
  delay(Intervall);
}

// Assemble l'UID et le converti en Hexadécimal
String mkString(byte *buffer, byte bufferSize) {
String UID = "";
```

Sans titre

```
for (byte i = 0; i < bufferSize; i++) {  
    if (buffer[i] == 0) {  
        UID = UID + "00";  
  
    }  
    else {  
        UID = UID + String(buffer[i], HEX);  
  
    }  
}  
return UID;  
} // End mkString
```