```
/**
 * ----------------------------------------------------------------------------------------------------------------
 * Example sketch/program showing how to read data from more than one PICC to serial.
 * ----------------------------------------------------------------------------------------------------------------
 * This is a MFRC522 library example; for further details and other examples see:
https://github.com/miguelbalboa/rfid
 *
 * Example sketch/program showing how to read data from more than one PICC (that is: a RFID
Tag or Card) using a
 * MFRC522 based RFID Reader on the Arduino SPI interface.
 *
 * Warning: This may not work! Multiple devices at one SPI are difficult and cause many trouble!!
Engineering skill
 *          and knowledge are required!
 *
 * @license Released into the public domain.
 *
 * Typical pin layout used:
 * -----------------------------------------------------------------------------------------
 *             MFRC522      Arduino       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 1    SDA(SS)      ** custom, take a unused pin, only HIGH/LOW required **
 * SPI SS 2    SDA(SS)      ** custom, take a unused pin, only HIGH/LOW required **
 * 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
 *
 */

#include <SPI.h>
#include <MFRC522.h>
#include "ssd1306.h"
#include "nano_gfx.h"

#define RST_PIN         9          // Configurable, see typical pin layout above
#define SS_PIN          10         // Configurable, take a unused pin, only HIGH/LOW required, must be
diffrent to SS 2

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance.
byte readCard[4];   // Stores scanned ID read from RFID Module
bool initrfid=true;

double cpt=0;
double cptmax=500;
double retry =0;

/**
 * Initialize.
 */
```

```
void setup() {

  Serial.begin(9600); // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on
ATMEGA32U4)
  SPI.begin();        // Init SPI bus
  Serial.println("Liste des commandes");
  Serial.println ("init[Nouvelle ligne] : affiche les informations firmwares");
  ssd1306_setFixedFont(ssd1306xled_font6x8);
  ssd1306_128x64_i2c_init();
  ssd1306_setFixedFont(ssd1306xled_font6x8);
  ssd1306_clearScreen();
  ssd1306_printFixed(0, 0, "Testeur RFID ", STYLE_NORMAL);
  delay(2000);
}

/**
 * Main loop.
 */
void loop() {
if (initrfid==true)
{
  initrfid=false;
    mfrc522.PCD_Init(SS_PIN, RST_PIN); // Init each MFRC522 card
   // mfrc522.PCD_DumpVersionToSerial();
   ShowReaderDetails();
} //endif initrfid==true

//lecture de l'uid du Tag RFID
getID();

// boucle de lecture du firmware
// une lecture toutes les 10 secondes
cpt++;
if (cpt>=cptmax)
  {
    Serial.println(cpt);
    ShowReaderDetails();
    cpt=0;
  }

}

/*********************************
 * Fonctions
 */

uint8_t getID() {
  // Getting ready for Reading PICCs
  if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed to RFID reader continue
    return 0;
  }
```

```
  if ( ! mfrc522.PICC_ReadCardSerial()) {   //Since a PICC placed get Serial and continue
    return 0;
  }
  // There are Mifare PICCs which have 4 byte or 7 byte UID care if you use 7 byte PICC
  // I think we should assume every PICC as they have 4 byte UID
  // Until we support 7 byte PICCs
  Serial.println(F("Scanned PICC's UID:"));
  for ( uint8_t i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
    //Serial.print(readCard[i], HEX);
  }
  Serial.println(" ");
  //convertir la chaine de Byte en HEXA
  char str[12] = "";
  array_to_string(readCard, 4, str);
  Serial.println(str);
  ssd1306_printFixed(15,  48,"        ", STYLE_NORMAL);
  ssd1306_printFixed(15,  56,str , STYLE_NORMAL);
  Serial.println("");
  mfrc522.PICC_HaltA(); // Stop reading
  return 1;
}

/**** Lecture du firmware ***/
void ShowReaderDetails() {
  // Get the MFRC522 software version
  byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
  Serial.println(F("MFRC522 Software"));
  Serial.print(F("Version: 0x"));
  Serial.print(v, HEX);
  char buffer[7];         //the ASCII of the integer will be stored in this char array
  itoa(retry,buffer,10); //(integer, yourBuffer, base)
  retry++;
  ssd1306_printFixed(0,  8," Lecture numero :  ", STYLE_NORMAL);
  ssd1306_printFixed(0,  16,buffer , STYLE_NORMAL);
  ssd1306_printFixed(0,  24, "MFRC522 :  ", STYLE_NORMAL);
  ssd1306_printFixed(0,  32, "Version:  ", STYLE_NORMAL);
  if (v == 0x91)
    {
    Serial.println(F(" = v1.0"));
    ssd1306_printFixed(0,  40, " v1.0", STYLE_NORMAL);
    }
  else if (v == 0x92)
    {
      Serial.println(F(" = v2.0"));
     ssd1306_printFixed(0,  40, " v2.0", STYLE_NORMAL);
    }
  else
    {
    Serial.print(F(" (unknown),probably a chinese clone?"));
     ssd1306_printFixed(0,  40, " unknown", STYLE_NORMAL);
    Serial.println("");
```

```
  // When 0x00 or 0xFF is returned, communication probably failed
    if ((v == 0x00) || (v == 0xFF))
      {
        Serial.println(F("WARNING: Communication failure, is the MFRC522 properly
connected?"));
        Serial.println(F("SYSTEM HALTED: Check connections."));
        ssd1306_printFixed(0,  40, " Comm Failure", STYLE_NORMAL);
        while (true); // do not go further
      } //end  if ((v == 0x00) || (v == 0xFF))
    }//end else
}

/** convertion Byte[] en chaine de caracteres ***/
void array_to_string(byte array[], unsigned int len, char buffer[])
{
    for (unsigned int i = 0; i < len; i++)
    {
        byte nib1 = (array[i] >> 4) & 0x0F;
        byte nib2 = (array[i] >> 0) & 0x0F;
        buffer[i*2+0] = nib1  < 0xA ? '0' + nib1  : 'A' + nib1  - 0xA;
        buffer[i*2+1] = nib2  < 0xA ? '0' + nib2  : 'A' + nib2  - 0xA;
    }
    buffer[len*2] = '\0';
}
```