

# Implémentation

Le code ci-dessous permet, en utilisant les bibliothèques précédentes, d'émuler un clavier bluetooth pour envoyer les caractères lus sur une carte RFID.

Le programme comporte :

- une fonction qui permet de transmettre également la valeur du niveau de charge de la batterie (comme le fait tout périphérique bluetooth)
- un contrôle de longueur de la chaîne lue et convertie avant son envoi par la fonction `print`, et la prise en charge d'une erreur si la longueur n'est pas exactement celle attendue pour les cartes Izly et cartes d'étudiants.
- un forçage de la fonction `releaseAll` après chaque transmission pour éviter les caractères surnuméraires.
- on a ajouté des signaux sonores pour confirmer la lecture

[BleKeyboardExemple.ino](#)

```
#include <M5Unified.h>
#include "BleKeyboard.h"
#include "MFRC522_I2C.h"

BleKeyboard bleKeyboard("M5CardReader", "M5Stack", 100);
MFRC522 mfrc522(0x28);

void setup() {
  M5.begin();
  Wire.begin();
  Serial.begin(115200);
  bleKeyboard.begin();
  M5.Lcd.println("M5Stack BLE Keyboard");
  mfrc522.PCD_Init();
  M5.Display.setTextSize(2);
  M5.Display.setFont(&fonts::efontCN_12);
  M5.Speaker.begin();
  M5.Speaker.setVolume(50);
}
```

```

void loop() {
  M5.update();
  String uid = "";
  M5.Lcd.clear();

  M5.Display.setTextDatum(MC_DATUM);
  M5.Display.setTextSize(3);
  M5.Display.drawString("Présentez", 160, 100);
  M5.Display.drawString("votre carte", 160, 140);

  while (1) {
    M5.Display.setTextSize(1);
    float batLevel = M5.Power.getBatteryLevel();
    M5.Display.drawString("Batterie : " + String(batLevel), 160, 40);
    bleKeyboard.setBatteryLevel(M5.Power.getBatteryLevel());
    if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        if (mfrc522.uid.uidByte[i] < 0x10) {
          uid += "0";
        }
        uid += String(mfrc522.uid.uidByte[i], HEX);
        if (i < mfrc522.uid.size - 1) {
          uid += ":";
        }
      }
    }
    M5.Lcd.clear();
    int L = uid.length();
    if (L == 20) {
      M5.Display.setTextSize(3);
      M5.Display.drawString("Carte scannée", 160, 120);
      M5.Speaker.tone(440, 250); // Frequence Hz, duree ms
      M5.delay(100);
      M5.Speaker.tone(880, 250);
      //M5.Display.drawString(uid, 160, 120);
      int L = uid.length();
      //M5.Display.drawString(String(L), 160, 160);
      String uidDeb = "";
      String uidFin = "";
      for (int j = 0; j < 10 ; j++) {
        uidDeb += uid[j];

```

```
        uidFin += uid[j+10];
    }
    //M5.Display.drawString(uidDeb, 160, 200);
    //M5.Display.drawString(uidFin, 160, 240);
    bleKeyboard.print(uidDeb);
    M5.delay(100);
    bleKeyboard.print(uidFin);
    M5.delay(2000);
    bleKeyboard.releaseAll();
    break;
}
else {
    M5.Display.setTextSize(3);
    M5.Display.drawString("Erreur de lecture", 160, 80);
    M5.Display.drawString("Réessayez svp", 160, 120);
    M5.Speaker.tone(440, 250); // Frequence Hz, duree ms
    M5.delay(100);
    M5.Speaker.tone(220, 250);
    M5.delay(2000);
    bleKeyboard.releaseAll();
    break;
}
}
}
```

---

Revision #2

Created 22 July 2025 14:09:40 by Christian Simon

Updated 22 July 2025 14:16:35 by Christian Simon