Using Smartphont to Read NFC tag

What is NFC?

NFC (Near Field Communication) is a short-range wireless technology that allows two electronic devices to communicate when they are within a few centimeters of each other. It's most commonly used in contactless payments, digital keycards, smart tags, and tap-to-share features. NFC works by using magnetic field induction and doesn't require pairing like Bluetooth — just a simple tap is enough to exchange data.

In this tutorial, we are using an iPhone to tap on the NTAG203 to open up a link. This tutorial is more intensive and will be more useful if you have other interactive components in your project involving Arduino.

If you only want a simple read and write function, we have another **tutorial** which shows you how to read and write an NFC tag using the app `NFC tools` .

Wiring and Library

We will be using DFRobot PN532 module, please refer to **this tutorial**. We will be using `Adafruit_PN532` library, we have a tutorial on **how to install a library** here.

Types of Tag

In this tutorial, we are using **NTAG203**, NTAG215 or NTAG216 should work fine as well. You **cannot** use MIFARE Classic tag or card.

Code for Writing Data

```
#include <Wire.h>
#include <Adafruit_PN532.h>

#define SDA_PIN A4
#define SCL_PIN A5

Adafruit_PN532 nfc(SDA_PIN, SCL_PIN);

//your url
const char *url = "youtube.com"; // Keep short due to 144 byte limit & omit the "https://" part
```

```cpp
uint8_t urlPrefix = 0x01; // 0x01 = http://www.

void setup(void) {
  Serial.begin(115200);
  Serial.println("Starting NFC writer with Adafruit PN532");

  nfc.begin();

  uint32_t versiondata = nfc.getFirmwareVersion();
  if (!versiondata) {
    Serial.println("Didn't find PN532 board");
    while (1);
  }

  nfc.SAMConfig(); // configure board to read RFID
  Serial.println("Waiting for an NFC tag...");
}

void loop(void) {
  uint8_t uid[] = { 0 };
  uint8_t uidLength;

  if (nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength)) {
    Serial.println("Tag detected!");

    // Build NDEF URI record
    uint8_t urlLength = strlen(url);
    uint8_t payloadLength = 1 + urlLength; // Prefix + URL

    uint8_t ndef[] = {
      0xD1,  // MB, ME, SR, TNF=0x01 (well-known)
      0x01,  // Type Length = 1
      payloadLength, // Payload Length
      0x55,  // Type = 'U'
      urlPrefix // URL Prefix
    };

    uint8_t messageLength = sizeof(ndef) + urlLength;
    uint8_t totalLength = messageLength + 3; // TLV: 0x03 len + msg + 0xFE

    uint8_t full[totalLength];
```

```cpp
  full[0] = 0x03;            // NDEF Message TLV tag
  full[1] = messageLength;      // Length of NDEF message
  memcpy(&full[2], ndef, sizeof(ndef));
  memcpy(&full[2 + sizeof(ndef)], url, urlLength);
  full[totalLength - 1] = 0xFE; // Terminator TLV

  // Write to tag starting at page 4
  int page = 4;
  for (int i = 0; i < totalLength; i += 4) {
    uint8_t buffer[4] = {0x00, 0x00, 0x00, 0x00};
    for (int j = 0; j < 4 && (i + j) < totalLength; j++) {
      buffer[j] = full[i + j];
    }

    if (!nfc.ntag2xx_WritePage(page, buffer)) {
      Serial.print("Failed writing to page ");
      Serial.println(page);
      return;
    }
    page++;
  }

  Serial.println("Wrote NDEF URL to NTAG203 successfully!");
  delay(5000); // prevent immediate re-trigger
  }
}
```