Near Field Communication
by Prof.Dr. Raphael Ruf at
Hochschule Ravensburg-Weingarten

Task 1: Data Read NFC Card & Tag

HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Authors:

Sai Charith Sathyanarayana (13945724)
Arvind Chinni Ravi (22145673)

Date of Submission:
November 20th, 2025

Table of Contents:

1. Introduction

The NFC Module V3 (PN532) is a versatile, highly integrated transceiver operating at 13.56 MHz that supports read/write operation, card emulation, and peer-to-peer modes. It delivers high throughput up to 424 kbit/s in both directions, while handling framing, parity, CRC, and modulation internally for robust, error-resilient communication. Its flexible host interfaces (SPI, I²C, and high-speed UART) simplify integration with microcontrollers such as the Arduino Nano. These features make the NFC V3 (PN532) ideal for embedded systems requiring secure, low-latency contactless data exchange and easy hardware interfacing for rapid prototyping and deployment. We designed the system to sequentially scan each of the 16 sectors and every block, ensuring comprehensive data recovery.

2. Methodology

The NFC card reading workflow comprises the following technical steps:

- **Inductive Power Transfer:** Passive NFC transponders are energised via inductive coupling to the reader's alternating magnetic field (13.56 MHz). The card's antenna coil rectifies and supplies power to the onboard IC, eliminating the need for a local battery.
- **Transponder Initialisation(IC):** The harvested energy triggers the NFC IC's power-on sequence and internal clock, bringing the protocol stack and security logic online.
- **RF Link Establishment:** The reader and tag negotiate an RF link using HF modulation schemes; carrier coupling and subcarrier modulation form the physical-layer connection.
- **Frame-level Communication:** The tag and reader exchange protocol frames (including UID and payload) with built-in parity and CRC checking to ensure data integrity.
- **Data Demodulation & Parsing:** The PN532-based receiver demodulates incoming signals and decodes frame payloads, extracting UID and memory block data.
- **Host Interface Transfer (SPI):** Decoded data is forwarded to the Arduino Nano over the Serial Peripheral Interface (SPI), using master/slave clocked transfers for deterministic throughput.
- **Application Processing & Visualisation:** The Arduino validates and parses sector/block contents, performs authentication as required, and outputs human-readable results to the serial monitor for logging and debugging.

3. Algorithm

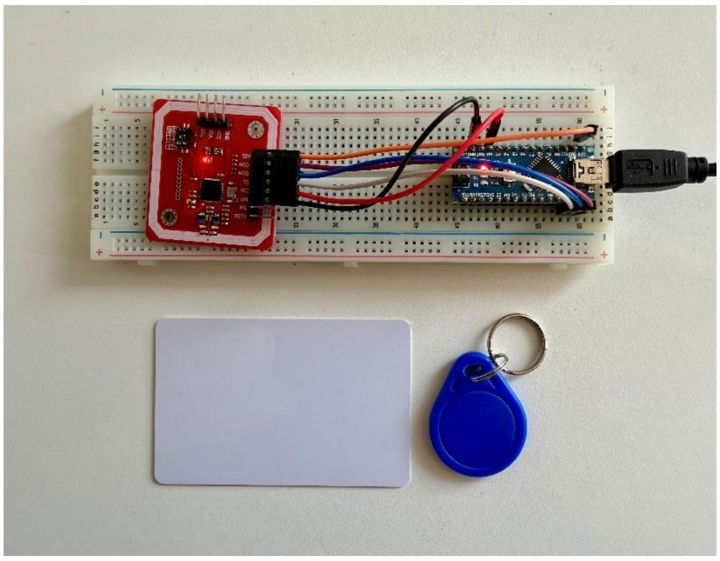
1. **Setup Phase**
    - Include required libraries (SPI, Adafruit_PN532) and define SPI pins for the PN532 module.
    - Create a PN532 object in SPI mode.
    - Define the default Mifare Classic Key A (FF FF FF FF FF FF) for authentication.
    - Initialise Serial communication at 115200 baud for debugging output.
    - Initialize the PN532 module using nfc.begin().
    - Check the PN532 firmware version:
        1. If not detected, then print an error and halt the program.
        2. If detected, then print chip type and firmware version.
    - Configure the PN532 in normal operation mode using SAMConfig().
    - Print a message indicating the system is waiting for a Mifare Classic card.
2. **Loop Phase**
    - Continuously scan for an NFC card (ISO14443A / Mifare).
    - When a card is detected:
        1. Retrieve and print the UID and its length.
        2. Check if the UID length is 4 bytes (indicating a MIFARE Classic card):
            - If not, print "Not a Mifare Classic card" and continue scanning.
3. **Reading Card Data**
    1. Loop through all 64 blocks (16 sectors × 4 blocks each).
    2. For each block:
    - If it is the first block of a sector:
        - Print the sector number.
        - Reset the authentication flag (authenticated = false).
    - If the sector is not authenticated:
        - Authenticate using Key A and the card UID.
        - If authentication fails, then print an error and skip the block.
        - If authentication succeeds, then set authenticated to true.
    - Read the block data (16 bytes):
        - If successful, then print the block number and the data in hexadecimal.
        - If reading fails, then print "Authentication failed for block".

4. Program

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

//SPI Pins for Arduino Nano
#define PN532_SCK   13
#define PN532_MISO  12
#define PN532_MOSI  11
#define PN532_SS    10

// Create PN532 instance in SPI mode
Adafruit_PN532 nfc(PN532_SCK, PN532_MISO, PN532_MOSI, PN532_SS);

// Default key A for Mifare Classic card
uint8_t keyA[6] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };

void setup() {
  Serial.begin(115200);
  while (!Serial);

  Serial.println("Searching for PN532...");

  nfc.begin();

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

  Serial.print("Found  PN5");  Serial.println((versiondata  >>  24)  &  0xFF,
HEX);
  Serial.print("Firmware version: ");
  Serial.print((versiondata >> 16) & 0xFF);
  Serial.print(".");
  Serial.println((versiondata >> 8) & 0xFF);
  nfc.SAMConfig();
  Serial.println("Waiting for a MIFARE Classic card...");
}

void loop() {

  uint8_t uid[7];
  uint8_t uidLength;

  if (!nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength))
    return; // No tag

  Serial.println("\nFound NFC Chip PN532");
  Serial.print("UID Length: ");
  Serial.print(uidLength);
  Serial.println(" bytes");
  Serial.print("UID Value: ");
  nfc.PrintHex(uid, uidLength);
  Serial.println();
  if (uidLength != 4) {
    Serial.println("Not a Mifare classic card!");
    delay(1500);
    return;
  }

  Serial.println("This is a Mifare classic card.");

  uint8_t data[16];
  bool authenticated = false;

  for (uint8_t block = 0; block < 64; block++) {

    // New sector every 4 blocks
    if (nfc.mifareclassic_IsFirstBlock(block)) {
      authenticated = false;
      Serial.print("\nSector ");
      Serial.print(block / 4);
      Serial.println(":");
    }

    // Authenticate sector trailer (first block of sector)
    if (!authenticated) {
      if  (nfc.mifareclassic_AuthenticateBlock(uid,  uidLength,  block,  0,
keyA)) {
        authenticated = true;
      } else {
        Serial.print("Authentication failed for block ");
        Serial.println(block);
        continue;
      }
    }

    // Read block
    if (nfc.mifareclassic_ReadDataBlock(block, data)) {
      Serial.print("Block ");
      Serial.print(block);
      Serial.print(": ");
      nfc.PrintHex(data, 16);
    } else {
      Serial.print("Block ");
      Serial.print(block);
      Serial.println(": READ ERROR");
    }
  }
}
```

5. Result

5.1 Connection



5.2 Output

```
Searching for PN532...
Found PN532
Firmware version: 1.6
Waiting for a MIFARE Classic card...

Found NFC Chip PN532
UID Length: 4 bytes
UID Value: 0x53 0x6C 0xA1 0x15

This is a Mifare Classic card.

Sector 0:
Block 0: 0x53 0x6C 0xA1 0x15 0x8B 0x08 0x04 0x00 0x62 0x63 0x64 0x65 0x66 0x67 0x68
0x69
Block 1: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 2: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 3: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF
```

```
Sector 1:
Block 4: 0x6D 0x6F 0x6E 0x64 0x61 0x79 0x20 0x77 0x65 0x20 0x68 0x61 0x76 0x65 0x20
0x71
Block 5: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 6: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 7: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 2:
Block 8: 0x48 0x61 0x6C 0x6C 0x6F 0x2C 0x20 0x44 0x61 0x6E 0x6B 0x65 0x21 0x00 0x00
0x00
Block 9: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 10: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 11: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 3:
Block 12: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 13: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 14: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 15: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 4:
Block 16: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 17: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 18: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 19: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF
```

```
Sector 5:
Block 20: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 21: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 22: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 23: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 6:
Block 24: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 25: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 26: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 27: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 7:
Block 28: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 29: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 30: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 31: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 8:
Block 32: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 33: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 34: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 35: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF
```

```
Sector 9:
Block 36: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 37: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 38: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 39: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 10:
Block 40: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 41: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 42: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 43: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 11:
Block 44: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 45: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 46: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 47: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 12:
Block 48: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 49: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 50: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 51: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF
```

```
Sector 13:
Block 52: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 53: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 54: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 55: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 14:
Block 56: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 57: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 58: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 59: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF

Sector 15:
Block 60: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 61: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 62: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00
Block 63: 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF


Found NFC Chip PN532
UID Length: 4 bytes
UID Value: 0x77 0xC7 0x8F 0x5F

This is an NFC Tag.

Sector 0:
Authentication failed for block 0
Authentication failed for block 1
Authentication failed for block 2
Authentication failed for block 3

Sector 1:
Authentication failed for block 4
Authentication failed for block 5
Authentication failed for block 6
Authentication failed for block 7

Sector 2:
Authentication failed for block 8
Authentication failed for block 9
Authentication failed for block 10
Authentication failed for block 11

Sector 3:
Authentication failed for block 12
Authentication failed for block 13
Authentication failed for block 14
Authentication failed for block 15

Sector 4:
Authentication failed for block 16
Authentication failed for block 17
Authentication failed for block 18
Authentication failed for block 19

Sector 5:
Authentication failed for block 20
Authentication failed for block 21
Authentication failed for block 22
Authentication failed for block 23

Sector 6:
Authentication failed for block 24
Authentication failed for block 25
Authentication failed for block 26
Authentication failed for block 27

Sector 7:
```

```
Authentication failed for block 28
Authentication failed for block 29
Authentication failed for block 30
Authentication failed for block 31

Sector 8:
Authentication failed for block 32
Authentication failed for block 33
Authentication failed for block 34
Authentication failed for block 35

Sector 9:
Authentication failed for block 36
Authentication failed for block 37
Authentication failed for block 38
Authentication failed for block 39

Sector 10:
Authentication failed for block 40
Authentication failed for block 41
Authentication failed for block 42
Authentication failed for block 43

Sector 11:
Authentication failed for block 44
Authentication failed for block 45
Authentication failed for block 46
Authentication failed for block 47

Sector 12:
Authentication failed for block 48
Authentication failed for block 49
Authentication failed for block 50
Authentication failed for block 51

Sector 13:
Authentication failed for block 52
Authentication failed for block 53
Authentication failed for block 54
Authentication failed for block 55
```

```
Sector 14:
Authentication failed for block 56
Authentication failed for block 57
Authentication failed for block 58
Authentication failed for block 59

Sector 15:
Authentication failed for block 60
Authentication failed for block 61
Authentication failed for block 62
Authentication failed for block 63
```

- Since the Tag has no chip inside, it cannot authenticate or access the data.

6. Conclusion

This project successfully showcased the integration of an Arduino Nano with the PN532 NFC module to read and extract data from NFC tags and cards. By leveraging the Adafruit PN532 library and following a systematic approach, the system efficiently accessed all 16 sectors and their respective blocks, demonstrating accuracy and reliability. This implementation highlights the potential of NFC in embedded systems and opens avenues for applications like secure entry, automated attendance systems, and intelligent data management.