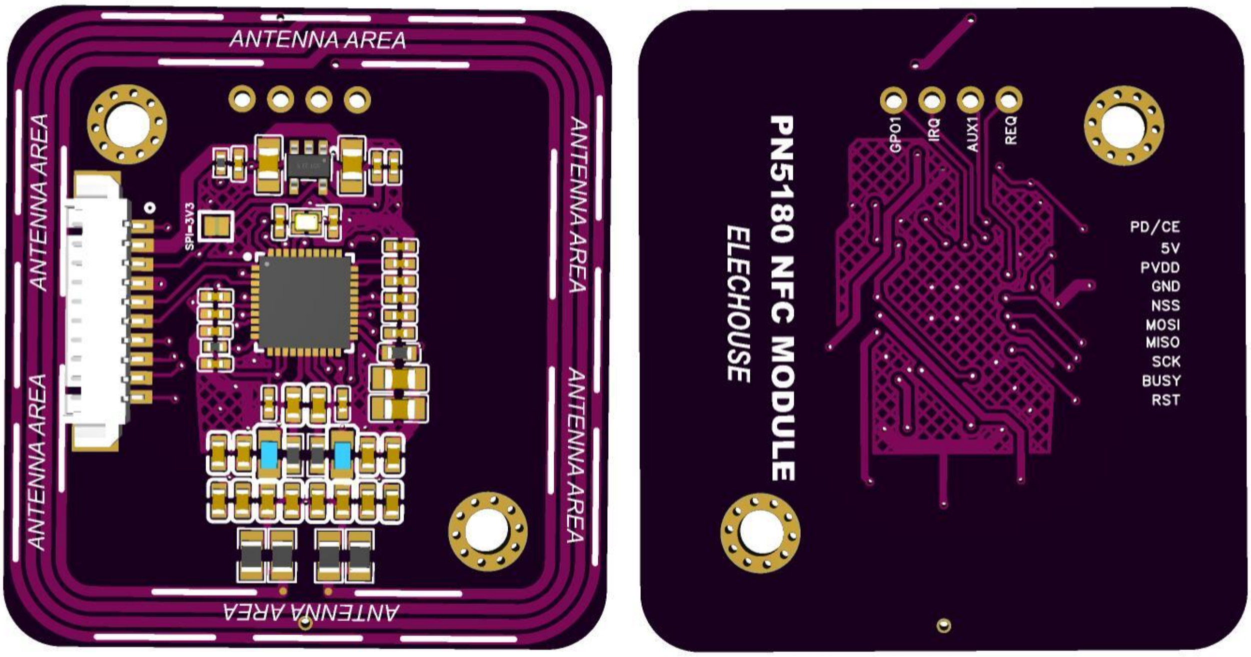
PRODUCT MANUAL: PN5180 INDUSTRIAL NFC FRONTEND MODULE

--HIGH-PERFORMANCE ISO 15693 & ISO 14443 READER



1. PRODUCT OVERVIEW

The **PN5180 NFC Module** is a robust NFC frontend module designed to overcome the stability issues found in generic consumer-grade readers. Built for industrial environments, it features an enhanced power architecture, superior thermal stability, and a dedicated hardware control interface.

It offers best-in-class read range for **ISO 15693 (ICODE)** tags and full support for **ISO 14443 Type A/B**, **FeliCa**, and **NFC P2P (ISO 18092)** standards.

Key Features

- **Industrial Power Architecture:** Features a **80uF capacitor** reservoir and a high-speed **ME6211 LDO**. This design absorbs RF current surges (up to 250mA), preventing voltage dips during continuous high-power transmission.
- **Fail-Safe "Hard Reboot" (PD/CE):** A dedicated **Chip Enable pin** allows the host MCU to physically power-cycle the PN5180 logic core. This ensures 100% recovery from state-machine lockups without manual intervention.
- **Logic Level Flexibility (1.8V - 3.3V):** The **PVDD** reference input allows direct interfacing with modern 1.8V MCUs (STM32, ESP32-S3) or standard 3.3V systems via an onboard jumper.

- **Optimized RF Tuning:** Thermally stable components eliminate "hot-dead" drift, ensuring consistent read range across temperature variations.

2. TECHNICAL SPECIFICATIONS

Parameter	Value	Note
Main Chip	NXP PN5180	High-performance Frontend
Input Voltage (5V Pin)	**4.5V – 5.5V DC**	Required for RF TVDD supply
Logic Voltage (PVDD)	**1.65V – 3.6V DC**	Matches MCU IO voltage
Rated Power	**~0.95 W**	Peak RF transmission power
Supported Protocols	ISO 15693, ISO 14443 A/B, FeliCa, P2P	
Host Interface	SPI (up to 7 Mbps)	Requires BUSY line for flow control
Connector	**HX1.25-10P**	10-Pin, 1.25mm pitch
Dimensions	40.02mm $\times$ 42.36mm	(1575.4mil $\times$ 1667.8mil)

3. HARDWARE INTERFACE

3.1 Pinout Description

Based on the physical PCB layout, the 10-pin interface is defined as follows (Top to Bottom):

#	Pin Name	Type	Description
1	**PD/CE**	Input	**Power Down / Chip Enable.** High (1) = Enable; Low (0) = Hard Power Down. Used for hard rebooting.
2	**5V**	Power	**RF Power Supply.** Must connect to 5V.
3	**PVDD**	Ref	**Logic Level Reference.** Connect to MCU VCC (1.8V or 3.3V). *(See Section 3.2)*
4	**GND**	Power	Ground.
5	**NSS**	Input	SPI Chip Select (Active Low).

6	**MOSI**	Input	SPI Data In (Master Out Slave In).
7	**MISO**	Output	SPI Data Out (Master In Slave Out).
8	**SCK**	Input	SPI Clock.
9	**BUSY**	Output	**Flow Control (Crucial).** High = Chip Busy. Host must wait.
10	**RST**	Input	Soft Reset.

3.2 Logic Level Configuration

The module features a solder jumper labeled **SPI=3V3** on the board:

- **Option A: Standard 3.3V MCUs (ESP32, Arduino)**
  - **Action:** Bridge/Solder the SPI-3V3 jumper.
  - **Wiring:** Leave Pin 3 (PVDD) disconnected. The module uses its internal 3.3V for logic.
- **Option B: 1.8V MCUs (New STM32, ESP32-S3)**
  - **Action:** Leave SPI-3V3 jumper **OPEN** (Default).
  - **Wiring:** Connect Pin 3 (PVDD) to your MCU's 1.8V power rail.

3.3 Extension Interface

Located at the top of the module, the header (Standard 2.54mm pitch) breaks out advanced GPIOs for interrupt-driven applications and low-power modes.

Pin Label (PCB)	Function Description	Application Note
REQ	Request / Wakeup	Input. Used to wake up the PN5180 from certain sleep states. Used to enter firmware upgrade mode.
AUX1	Auxiliary Pin 1	Analog/Digital test signal or auxiliary control. Often left unconnected.
IRQ	Interrupt Request	Crucial for Low Power Mode. Output signal that goes HIGH when a tag is detected (LPCD) or data is ready, allowing the MCU to sleep instead of polling.
GPO1	General Purpose Out	Configurable output pin for status indication.

Tip: If you plan to use the Low Power Card Detection (LPCD) feature provided in the software library, you MUST connect the IRQ pin to an interrupt-capable GPIO on your MCU.

4. SOFTWARE INTEGRATION GUIDE

4.1 Library Installation

For ESP32 and Arduino platforms, use the open-source driver library maintained by Elechouse.

- **GitHub Repository:** wilson-elechouse/PN5180_ELECHOUSE
- **Download:** https://github.com/wilson-elechouse/PN5180_ELECHOUSE

4.2 ESP32 Connection Example

Connect the module to an ESP32 development board using the standard VSPI bus:

PN5180 Pin	ESP32 GPIO
5V	5V / VIN
GND	GND
PVDD	3.3V (or Bridge Jumper)
NSS	GPIO 5
MOSI	GPIO 23
MISO	GPIO 19
SCK	GPIO 18
BUSY	GPIO 16
RST	GPIO 17
PD/CE	**GPIO 22**

4.3 Code Example (ISO 14443A / Mifare)

The following code demonstrates how to initialize the module, perform a "Hard Reboot" using the CE pin for stability, and scan for tags.

```
/*
 * PN5180 Industrial Module - ESP32 Integration Example
 * Protocol: ISO 14443 Type A (Mifare)
 * Requirements: 'PN5180 Library' by Wilson-Elechouse
 */


#include <PN5180.h>
```

```
#include <PN5180ISO14443.h>

// Pin Definitions for ESP32
#define PN5180_NSS    5
#define PN5180_BUSY 16
#define PN5180_RST    17
#define PN5180_CE      22    // Connected to PD/CE pin for Hard Reboot control

// Create PN5180 instance
PN5180ISO14443 nfc(PN5180_NSS, PN5180_BUSY, PN5180_RST);

void setup() {
   Serial.begin(115200);
   Serial.println("SYSTEM START: PN5180 Industrial Module");

   // --- INDUSTRIAL STABILITY: HARD REBOOT SEQUENCE ---
   // This ensures the PN5180 is in a clean state even if the MCU just crashed/reset.
   pinMode(PN5180_CE, OUTPUT);
   Serial.println("Performing Hard Reboot...");
   digitalWrite(PN5180_CE, LOW);    // Physically cut logic power
   delay(50);
   digitalWrite(PN5180_CE, HIGH); // Restore logic power
   delay(50);                                  // Allow boot time

   // --- LIBRARY INITIALIZATION ---
   Serial.println("Initializing SPI & RF Field...");
   nfc.begin();      // Initialize SPI
```

```
    nfc.reset();      // Soft reset command
    nfc.setupRF(); // Enable RF field (High Power Mode)

    // Verify Connection
    uint8_t productVersion;
    nfc.readEEprom(PRODUCT_VERSION, productVersion, 2);
    Serial.printf("Hardware Found - Product Version: %d.%d\n", productVersion, productVersion);
    Serial.println("Scanning for ISO14443A Tags...");
}

void loop() {
    // Buffer to store UID
    uint8_t uid;

    // Check for card presence and read UID
    // readCardSerial returns the UID length (0 if no card)
    uint8_t uidLength = nfc.readCardSerial(uid);

    if (uidLength > 0) {
        Serial.print("Tag Detected! UID: ");
        for (int i = 0; i < uidLength; i++) {
            if (uid[i] < 0x10) Serial.print("0");
            Serial.print(uid[i], HEX);
            Serial.print(" ");
        }
        Serial.println();
        // Add delay to prevent serial flooding
        delay(1000);
    }

    // Minimal polling delay
    delay(20);
}
```

5. MECHANICAL DATA & RESOURCES

- **PCB Dimensions:** 40.02 mm x 42.36 mm
- **Mounting Holes:** 3x Standard M3 equivalent (See 3D model)
- **3D Step File:** Download Here
- **GitHub Library:** https://github.com/wilson-elechouse/PN5180_ELECHOUSE
- **Product page:** https://www.elechouse.com/product/pn5180-nfc-module/