📡 Workshop: Building an Arduino Nano RFID Reader

🎯 Target audience

This workshop is designed for **beginners in electronics and IoT** with **basic Arduino knowledge**. No prior knowledge of SPI or RFID is required.

🚀 Workshop Goal

At the end of this workshop you will:

- Build a fully functional RFID reader module.
- Understand the basics of **RFID** technology.
- Understand the basics of **SPI communication**.
- Have a reusable module for future projects like access control, smart homes, or presence detection.

Author: Boutsman

🧰 Required Materials

Hardware:

- Arduino Nano (clone or original)
- RC522 RFID module
- Dupont jumper wires (male-female or male-male + headers)
- 3D-printed enclosure (download: Makerworld link)
- USB cable for Arduino Nano
- (Optional) Soldering iron & solder (for permanent build)

Software:

- Arduino IDE (latest version)
- MFRC522 Arduino library

🔬 Quick Theory

How RFID works

RFID = **Radio Frequency Identification**

- The reader (RC522 module) creates an electromagnetic field.
- When a tag (card or keyfob) enters this field, it powers up and transmits its unique ID.
- The Arduino reads this ID and processes it.

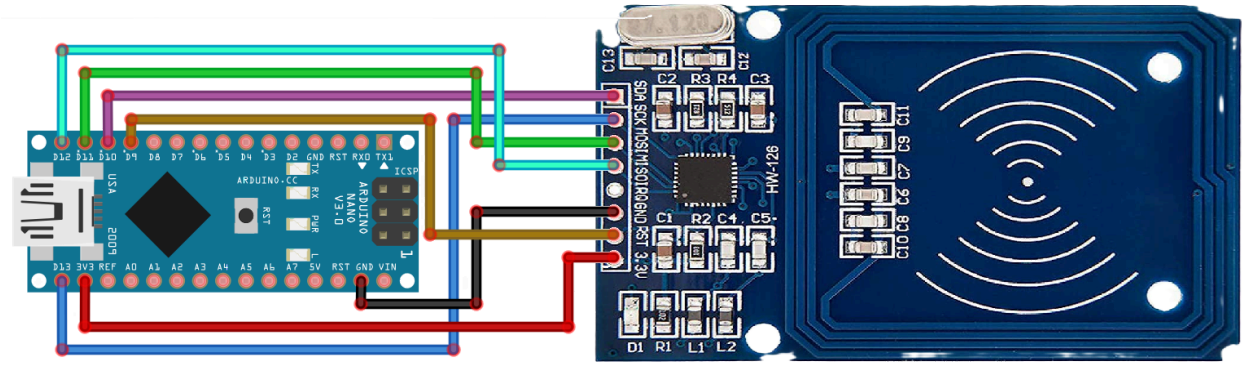
Why we use SPI

SPI = **Serial Peripheral Interface**
It's a fast and efficient protocol for short-distance communication between microcontrollers and peripherals.

Signal	Full Name	Function
MISO	Master In Slave Out	RC522 → Arduino
MOSI	Master Out Slave In	Arduino → RC522
SCK	Serial Clock	Synchronizes communication
SS (SDA on RC522)	Slave Select	Activates the RC522

Author: Boutsman

🖥️ Wiring Instructions



📊 Wiring Table

RC522 Pin	Arduino Nano Pin
SDA	D10
SCK	D13
MOSI	D11
MISO	D12
IRQ	Not used
GND	GND
RST	D9
3.3V	3.3V (⚠ Do not connect to 5V!)

⚠ **Important Safety Note:**

- The RC522 operates at 3.3V. Supplying 5V can permanently damage the module.
- Double-check your wiring before powering up. Wrong connections may destroy components.

🔧 Assembly Steps

Author: Boutsman

An animation can be found on Instagram

1️⃣ **Prepare the 3D-printed enclosure.**
Ensure all parts fit properly before wiring.

2️⃣ **Insert the Arduino Nano.**
Place it into the bracket.

3️⃣ **Mount the RC522 module.**
Place it into the bracket.

4️⃣ **Connect the wires as per the wiring table.**
Use jumper wires for temporary builds, or solder wires for permanent builds.

5️⃣ **Assemble the enclosure.**
Tuck wires neatly inside to avoid pinching during assembly.

🖥️ Software Installation

1️⃣ **Install the Arduino IDE** (if not installed yet).

2️⃣ **Install the MFRC522 Library:**

- Go to: *Sketch → Include Library → Manage Libraries*
- Search: `MFRC522`
- Install library by "Miguel Balboa"

🧪 Test Code

Author: Boutsman

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Scan an RFID tag...");
}

void loop() {
  if (!mfrc522.PICC_IsNewCardPresent()) return;
  if (!mfrc522.PICC_ReadCardSerial()) return;

  Serial.print("UID tag:");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
  Serial.println();
  delay(1000);
}
```

Troubleshooting

Author: Boutsman

Symptom	Possible Issue
No serial output	Wrong COM port selected
Always “Scan an RFID tag...”	Check RST or SDA wiring
Garbage data	Baudrate mismatch (should be 9600)
Still not working	Recheck wiring, especially MISO/MOSI/SCK

🚀 Extension Ideas

- Communicate with Raspberry Pi via UART.
- Control door locks with relays.
- Store valid tag IDs in EEPROM or external database.
- Integrate into Node-RED for home automation.
- Link into Home Assistant.

🔎 Advanced Note

The `IRQ` pin can be used for interrupt-driven reads in advanced projects. In this beginner build, it's not required.

✅ Wrap-Up

You now have:

- A fully functional RFID reader module.
- Clean SPI communication experience.
- A neat 3D-printed case that makes your project feel like a product.

💡 **Pro tip:**
"Solder once, debug less."
Take your time with the wiring — good solder joints prevent 90% of all electronic issues.

🎓 Finished!

Author: Boutsman