RFID Reader Interface Using Raspberry Pi

Introduction:

What is RFID?

Radio Frequency Identification (RFID) is a wireless technology that allows data to be transmitted between an RFID reader and an RFID tag using radio waves. It is commonly used for authentication, access control, and tracking applications.

Working of RFID:

RFID technology consists of two main components:

1. RFID Reader – A device that sends out radio signals to detect and read data from RFID tags.
2. RFID Tag – A small chip with an antenna that stores data and transmits it back to the reader when it comes into range.

When an RFID tag comes near the reader, it absorbs energy from the reader's signal and transmits back its unique identification number (UID) along with any stored data. The reader then processes this data and can take further actions, such as granting access or recording attendance.

Components Required:

- Raspberry Pi (any model with GPIO support)
- RFID Reader Module (e.g., RC522 or RDM6300)
- RFID Tags
- Jumper Wires
- Breadboard (optional)
- Power Supply

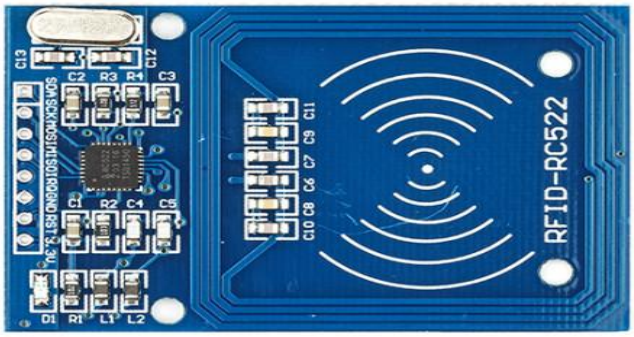

Fig 1. RFID Reader

Circuit Connection:

RFID RC522	Raspberry Pi (BCM GPIO)
VCC	3.3V
GND	GND
RST	GPIO 25
MISO	GPIO 9
MOSI	GPIO 10
SCK	GPIO 11
SDA	GPIO 8

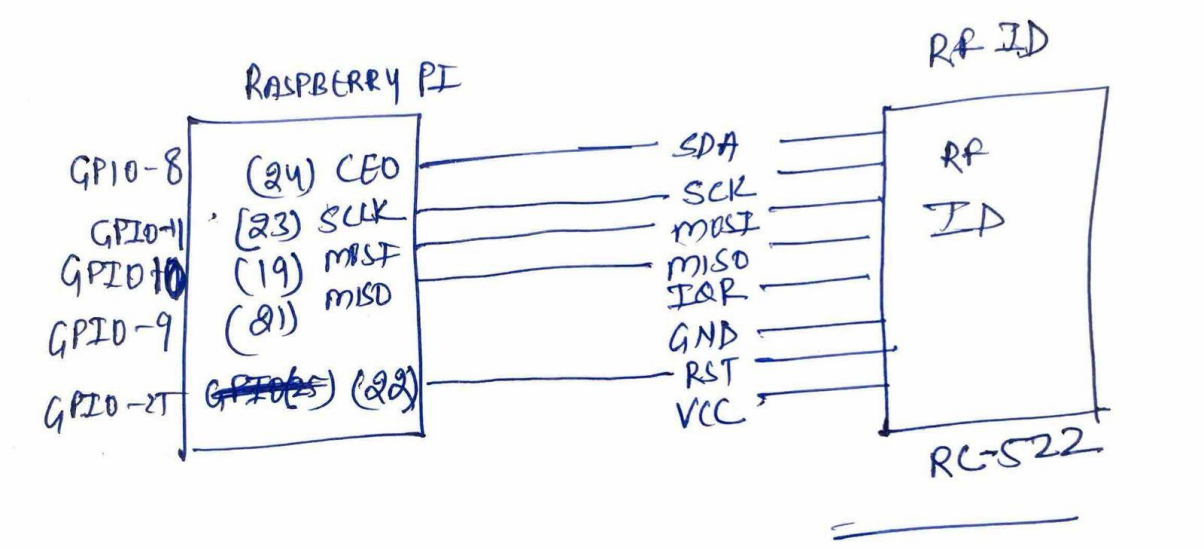

Fig 2. Circuit Connection with Raspberry Pi

Applications:

- Smart Access Control – RFID-based door locks and security systems.
- Attendance Tracking – Automated check-in/check-out systems.
- Inventory Management – Efficient tracking of products and assets.
- Cashless Payments – RFID-based payment solutions.

Learnings:

- Understanding RFID technology and its working principle.
- How to interface an RFID reader with a Raspberry Pi.
- Writing and executing Python scripts for RFID data processing.
- Real-world applications of RFID in security and automation.

Conclusion:

This provides a guide to interface an RFID reader with a Raspberry Pi, covering hardware setup, software installation, and data reading using Python. RFID technology is widely used in **security, access control, and automation**. With this knowledge, you can now build projects like **smart door access, attendance systems, or inventory tracking**. Keep exploring and expanding its applications.

Outcome:

Program:

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
from RPLCD.i2c import CharLCD
import time
lcd = CharLCD('PCF8574', 0x27)

GPIO.setwarnings(False)
LED_Pin = 26
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_Pin, GPIO.OUT)
```

```
reader = SimpleMFRC522()  # Initialize RFID reader
try:
    print("Place your RFID card/tag near the reader...")
    while True:
        lcd.clear()
        lcd.cursor_pos = (0, 0)  # First row
        lcd.write_string(f'RFID Authentication')
        lcd.cursor_pos = (1, 0)  # First row
        lcd.write_string(f'System')

        lcd.cursor_pos = (3, 0)  # First row
        lcd.write_string(f'Please scan ID')

        id, text = reader.read()  # Read RFID tag
        print(f"RFID UID: {id}")
        print(f"Card Data: {text.strip()}")  # Print stored text if any
        if id == 166835356500:
            lcd.clear()
            lcd.cursor_pos = (0, 0)  # First row
            lcd.write_string(f'RFID Authentication')
            lcd.cursor_pos = (1, 0)  # First row
            lcd.write_string(f'System')
            lcd.cursor_pos = (2, 0)  # First row
            lcd.write_string(f'Authorised: ')
            lcd.cursor_pos = (3, 0)  # First row
            lcd.write_string(f'Door Open')
            GPIO.output(LED_Pin, GPIO.HIGH)
            time.sleep(3)
            GPIO.output(LED_Pin, GPIO.LOW)
```

```
        else:
            lcd.clear()
            lcd.cursor_pos = (0, 0)  # First row
            lcd.write_string(f'RFID Authentication')
            lcd.cursor_pos = (1, 0)  # First row
            lcd.write_string(f'System')
            lcd.cursor_pos = (2, 0)  # First row
            lcd.write_string(f'Not Authorised')
            time.sleep(2)



except KeyboardInterrupt:
    print("\nStopping...")
    GPIO.cleanup()  # Clean up GPIO on exit
```