Steps to Execute the NodeMCU ESP8266 with Blynk and NTP

Follow these steps to successfully set up and execute the **Smart Motion Detection System** using the **ESP8266** module, which integrates motion detection, time display, servo control, buzzer feedback, and cloud interaction via **Blynk**.

---

1. Gather the Required Components

Ensure you have the following components ready:

- **ESP8266 NodeMCU** (or any compatible ESP8266 board)
- **IR Motion Sensor** (for motion detection)
- **Servo Motor** (to move based on motion detection)
- **Buzzer** (for feedback on motion detection)
- **LCD Display** (16x2 LCD with I2C interface to display time and motion status)
- **Wi-Fi Network** (for connecting the ESP8266 to the internet)
- **Blynk App** (for remote control and monitoring)

---

2. Install Required Libraries

Ensure you have the following libraries installed in your Arduino IDE:

1. **NTPClient**: For fetching time from NTP servers.
2. **ESP8266WiFi**: For Wi-Fi communication with the ESP8266 module.
3. **LiquidCrystal_I2C**: For controlling the LCD display.
4. **Servo**: For controlling the servo motor.
5. **BlynkSimpleEsp8266**: To interface with the Blynk app.
6. **TimeLib**: For easier date and time handling.
7. **WiFiUdp**: For handling UDP communication with the NTP server.

To install libraries, go to **Sketch** > **Include Library** > **Manage Libraries** and search for the libraries above.

---

3. Set Up the Blynk App

1. **Download and Install the Blynk App**: Install the Blynk app on your smartphone from the **Google Play Store** or **Apple App Store**.
2. **Create a New Project**:
    - Open the Blynk app and create a new project.
    - Select **ESP8266** as your hardware model.
    - You will receive an **Authentication Token** via email. Copy this token and replace it in your code where you see BLYNK_AUTH_TOKEN.
3. **Add Widgets**: In the Blynk app, add widgets to control and monitor your devices:
    - **Button Widget (V2)**: To control the Buzzer.
    - **Slider Widget (V3)**: To control the Servo motor position.
    - **LED Widget (V4)**: To indicate the state of the motion sensor.
    - **LCD Widget**: To display motion detection status, time, and date.

---

4. Wiring the Components

1. **IR Motion Sensor (PIR)**:
    - Connect the **VCC** to **5V** on the ESP8266.
    - Connect the **GND** to **GND** on the ESP8266.
    - Connect the **OUT** to **GPIO Pin 12** (MOTION_PIN).
2. **Servo Motor**:
    - Connect the **VCC** of the servo to **5V**.
    - Connect the **GND** to **GND**.
    - Connect the **Control Pin** to **GPIO Pin 13** (SERVO_PIN).
3. **Buzzer**:
    - Connect the **VCC** to **5V**.
    - Connect the **GND** to **GND**.
    - Connect the **Control Pin** to **GPIO Pin 14** (BUZZER_PIN2).
4. **LCD Display**:
    - **VCC** to **5V**.
    - **GND** to **GND**.

- **SDA** to **GPIO Pin 4**.
- **SCL** to **GPIO Pin 5**.

---

5. Upload the Code to the ESP8266

1. Open the **Arduino IDE** and write or paste the provided code into the editor.
2. Set the **Board** to **NodeMCU 1.0 (ESP-12E)** and select the correct **Port**.
3. **Replace the Wi-Fi Credentials**:
   - Update the Wi-Fi SSID and password with your network credentials.
   - Replace the **Blynk Auth Token** in the code with the token you received from the Blynk app.
4. **Upload the Code**: Click the **Upload** button to upload the code to your ESP8266.

---

6. Open the Serial Monitor

Once the code is uploaded:

- Open the **Serial Monitor** in the Arduino IDE (**Tools** > **Serial Monitor**) and set the baud rate to **115200**.
- You should see the following messages:
  - "Connecting to Wi-Fi..." followed by "Connected to Wi-Fi".
  - "Blynk connected!" when the connection to the Blynk app is successful.

---

7. Test the System

Motion Detection:

- **Activate the IR sensor**: When the PIR sensor detects motion, the system will:
  - Trigger the **buzzer** to alert about motion.
  - Move the **servo motor** based on the number of motion detections (1st detection → 90°, 2nd → 180°, 3rd → 270°).
  - Update the **LCD** to display the "Motion Detected" message.
  - Send a notification to the **Blynk app** to indicate motion detected.

Control via Blynk:

- **Buzzer Control**: In the Blynk app, press the **Button Widget (V2)** to toggle the buzzer on or off.

- **Servo Control**: Use the **Slider Widget (V3)** in the Blynk app to move the servo motor to a specific position (0° to 180°).
- **IR Sensor Status**: The **LED Widget (V4)** in the Blynk app will show if the motion sensor is active (on when motion is detected).

Time and Date Display:

- The **LCD** will display the current **time** and **date**, fetched from the NTP server. It will show:
    - Time in **12-hour format** with AM/PM.
    - The current **day of the week** and **date** (DD/MM/YYYY).

---

8. (Optional) Adjust Code and Features

You can further customize the system:

- Modify the number of motion detections before resetting the servo position.
- Adjust the **debounce delay** for the motion sensor if needed.
- Add more control features in the **Blynk app**, such as controlling other devices or updating the LCD with additional information.

---

9. Troubleshooting

- **Problem**: The ESP8266 is not connecting to Wi-Fi.
    - **Solution**: Ensure the correct SSID and password are entered in the code. Double-check your Wi-Fi network settings.
- **Problem**: Motion detection is not triggering.
    - **Solution**: Verify the PIR sensor is wired correctly. Ensure it’s powered properly and is positioned in an area with adequate motion.
- **Problem**: The servo is not moving.
    - **Solution**: Check the servo wiring and ensure the correct pin is defined in the code. Adjust the servo range if necessary.
- **Problem**: The Blynk app is not controlling the devices.
    - **Solution**: Check the **Blynk Auth Token** and ensure the ESP8266 is connected to the internet.

---

10. Final Test

After completing all the setup steps:

1. Test **motion detection** to see the servo movement and buzzer feedback.
2. Use the **Blynk app** to control the **servo** and **buzzer** remotely.
3. Verify the **time** and **date** are accurately displayed on the **LCD**.

You have now successfully set up and executed the **Smart Motion Detection System** with cloud control via **Blynk** and real-time data from the **NTP server**.

Conclusion

This project combines motion detection, cloud control, real-time time and date display, and physical feedback (via buzzer and servo) to create a smart system that can be controlled remotely through the Blynk app. You can expand the project by adding more features, such as controlling more devices or integrating with other IoT platforms.

Steps to Execute the RFID Access Control System

Follow these steps to successfully set up and execute the RFID access control system on your Arduino.

---

1. Gather the Required Components

Ensure you have the following components ready:

- **Arduino Board** (e.g., Arduino Uno)
- **MFRC522 RFID Reader** (with pins for SPI communication)
- **ISD1820 Voice Recorder Module** (for playback of audio messages)
- **Buzzer** (for feedback on access)
- **RFID Tags** (authorized and unauthorized)
- **Wires and Breadboard** (for connections)

---

2. Install Required Libraries

You will need the following libraries for the project to work:

1. **MFRC522 library** for RFID communication.
2. **SPI library** for SPI protocol communication with the MFRC522 module.

Follow these steps to install the libraries:

- Open the Arduino IDE.
- Go to **Sketch** > **Include Library** > **Manage Libraries**.
- In the **Library Manager**, search for the **MFRC522** library and install it.
- The **SPI** library should already be included with the Arduino IDE, but if it's missing, search for and install it as well.

---

3. Wiring the Components

1. **MFRC522 RFID Reader Wiring:**
    - **SS (Slave Select)** -> Pin 10 on Arduino.
    - **RST (Reset)** -> Pin 9 on Arduino.
    - **SCK (Serial Clock)** -> Pin 13 on Arduino.
    - **MISO (Master In Slave Out)** -> Pin 12 on Arduino.

- **MOSI (Master Out Slave In)** -> Pin 11 on Arduino.
- **GND** -> GND on Arduino.
- **VCC** -> 5V on Arduino.

2. **ISD1820 Voice Recorder Module Wiring:**
   - **REC** -> Pin 2 on Arduino (used to start recording).
   - **PLAY_E** -> Pin 3 on Arduino (edge-triggered playback).
   - **PLAY_L** -> Pin 4 on Arduino (looped playback).
   - **FT (Feed-Through)** -> Pin 5 on Arduino (disables mic during playback).
   - **VCC** -> 5V on Arduino.
   - **GND** -> GND on Arduino.
3. **Buzzer Wiring:**
   - **Buzzer** -> Pin A0 on Arduino (controls feedback sound).
   - **VCC** -> 5V on Arduino.
   - **GND** -> GND on Arduino.

---

4. Upload the Code to Arduino

1. Open the **Arduino IDE** and write the code or copy-paste the code provided in the previous response.
2. Select the correct board type and port from **Tools** > **Board** (e.g., Arduino Uno) and **Tools** > **Port** (the port your Arduino is connected to).
3. Click the **Upload** button (the arrow pointing right) to upload the code to your Arduino board.

---

5. Open the Serial Monitor

Once the code is uploaded:

- Open the **Serial Monitor** from **Tools** > **Serial Monitor** or press **Ctrl + Shift + M**.
- Set the baud rate to **9600** to match the one specified in the code (Serial.begin(9600)).
- You should see the prompt: "Approximate your card to the reader...".

---

6. Testing the RFID System

Step 1: Present an RFID Card

- Bring an authorized RFID tag near the MFRC522 reader.
- The system will attempt to read the card.
- If the UID matches the authorized UID ("E9 21 0F E3"), the system will:
    - Blink the **buzzer** to indicate access granted.
    - Optionally, play an "Access Granted" message if the playback function is activated.
- If the UID does not match, the system will:
    - Print "Access Denied" in the serial monitor.
    - Play a "Denied" message using the **ISD1820**.

Step 2: Test Unauthorized Card

- Present an **unauthorized RFID card**.
- The system will deny access and provide the appropriate feedback as described.

---

7. (Optional) Adjust Code and Features

If you want to:

- Add more authorized UIDs, simply extend the if condition for checking UIDs. For example:

cpp

Copy code

```
if (content.substring(1) == " " || content.substring(1) == "XX XX XX XX") {
  // Authorized access code
}
```

- Enable/Disable voice playback: Uncomment the lines for digitalWrite(PLAY_E, HIGH) and digitalWrite(PLAY_E, LOW) for authorized/denied messages.

---

8. Troubleshooting

- **Problem:** The RFID reader is not detecting the card.
    - **Solution:** Check the wiring connections and ensure the MFRC522 module is powered correctly. Also, verify that the pins in the code match the physical connections.
- **Problem:** The serial monitor doesn't display the UID of the card.
    - **Solution:** Double-check the connection of the MFRC522 reader and ensure the SPI.begin() and mfrc522.PCD_Init() functions are called in the setup.
- **Problem:** The ISD1820 doesn't play audio.
    - **Solution:** Ensure the **REC**, **PLAY_E**, **PLAY_L**, and **FT** pins are connected correctly. Verify that the ISD1820 has an audio file recorded and is properly powered.
- **Problem:** Buzzer or feedback is not working.
    - **Solution:** Ensure the buzzer is connected to the correct pin (A0 in this case). Check if the buzzer's VCC and GND are correctly wired.

---

9. Final Test

After testing all the components and ensuring everything works as expected:

- Place an authorized RFID card near the reader.
- The system should grant access (buzzer blinks and/or voice playback).
- Place an unauthorized card near the reader.
- The system should deny access (buzzer and voice feedback for denial).

Congratulations! You've successfully set up and executed the RFID Access Control System.

Conclusion

By following the above steps, you can build and execute an RFID-based access control system with audio feedback. This project can be further expanded to include more features such as a display, multiple authorized users, or integration with other security systems.