

Complete ESP-NOW Workflow (XIAO ESP32C3 → Button → LED)

Using two Seeed Studio XIAO ESP32C3 boards

STEP 1 — Get the Receiver MAC Address

 Upload this to the receiver (LED board):

```
#include <WiFi.h>
```

```
void setup() {  
  Serial.begin(115200);  
  WiFi.mode(WIFI_STA);  
  Serial.println(WiFi.macAddress());  
}
```

```
void loop() {}
```

 Then:

1. Open **Serial Monitor**
2. Set baud rate to **115200**
3. Press reset if needed

 You'll get:

E4:B3:23:C5:91:5C

 Convert it (important):

E4:B3:23:C5:91:5C

↓

{0xE4, 0xB3, 0x23, 0xC5, 0x91, 0x5C}

STEP 2 — Upload Receiver Code (LED Board)

Now replace the MAC code with your actual receiver program.

 **Upload this to the same board:**

```
#include <WiFi.h>
#include <esp_now.h>

#define LED_PIN 2

typedef struct {
  bool buttonState;
} Message;

Message msg;

void onReceive(const uint8_t * mac, const uint8_t *incomingData, int len) {
  memcpy(&msg, incomingData, sizeof(msg));
  digitalWrite(LED_PIN, msg.buttonState ? HIGH : LOW);
}

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  WiFi.mode(WIFI_STA);

  if (esp_now_init() != ESP_OK) {
    Serial.println("ESP-NOW init failed");
    return;
  }

  esp_now_register_recv_cb(onReceive);
}

void loop() {}
```

STEP 3 — Upload Sender Code (Button Board)

 Upload this to the second board:

```
#include <WiFi.h>
#include <esp_now.h>

#define BUTTON_PIN 4

uint8_t receiverMAC[] = {0xE4, 0xB3, 0x23, 0xC5, 0x91, 0x5C};

typedef struct {
  bool buttonState;
} Message;

Message msg;

void setup() {
  Serial.begin(115200);
  pinMode(BUTTON_PIN, INPUT_PULLUP);

  WiFi.mode(WIFI_STA);

  if (esp_now_init() != ESP_OK) {
    Serial.println("ESP-NOW init failed");
    return;
  }

  esp_now_peer_info_t peerInfo = {};
  memcpy(peerInfo.peer_addr, receiverMAC, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;


  if (esp_now_add_peer(&peerInfo) != ESP_OK) {
    Serial.println("Failed to add peer");
    return;
  }
}

void loop() {
  msg.buttonState = (digitalRead(BUTTON_PIN) == LOW);
  esp_now_send(receiverMAC, (uint8_t *)&msg, sizeof(msg));
```

```
delay(100);  
}
```

STEP 4 — Wiring

Button board:

- GPIO4 (D2) → button → GND
- Uses internal pull-up 

LED board:

- GPIO2 (D0) → resistor → LED → GND
-

STEP 5 — Run It

1. Power both boards
2. Press the button

Result:

- Press → LED ON
 - Release → LED OFF
-

Mental Model (what's happening)

- Sender reads button
 - Sends `true/false` via ESP-NOW
 - Receiver gets it instantly
 - LED mirrors the state
-

Pro tip

You only need to:

- Get MAC **once per board**
- Reuse it forever unless you switch hardware