Source code:

```
#include "RF24.h"
#include <SPI.h>
#include "esp_bt.h"
#include "esp_wifi.h"

SPIClass *sp = nullptr;
SPIClass *hp = nullptr;

RF24 radio(16, 15, 16000000);   // HSPI
RF24 radio1(22, 21, 16000000);  // VSPI

// HSPI: SCK = 14, MISO = 12, MOSI = 13, CS = 15, CE = 16
// VSPI: SCK = 18, MISO = 19, MOSI = 23, CS = 21, CE = 22

unsigned int flag = 0;
unsigned int flagv = 0;
int ch = 45;
int ch1 = 45;

// Manual button handling
const int buttonPin = 33;
bool lastButtonState = HIGH;
bool buttonState = HIGH;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

void two() {
  if (flagv == 0) {
    ch1 += 4;
  } else {
    ch1 -= 4;
  }

  if (flag == 0) {
    ch += 2;
  } else {
    ch -= 2;
  }

  if ((ch1 > 79) && (flagv == 0)) {
    flagv = 1;
  } else if ((ch1 < 2) && (flagv == 1)) {
    flagv = 0;
  }
  if ((ch > 79) && (flag == 0)) {
    flag = 1;
  } else if ((ch < 2) && (flag == 1)) {
    flag = 0;
  }

  radio.setChannel(ch);
  radio1.setChannel(ch1);
}

void one() {
  radio1.setChannel(random(80));
  radio.setChannel(random(80));
  delayMicroseconds(random(60));
}

void setup() {
  Serial.begin(115200);

  esp_bt_controller_deinit();
  esp_wifi_stop();
  esp_wifi_deinit();
  esp_wifi_disconnect();

  pinMode(buttonPin, INPUT_PULLUP);  // Use internal pull-up

  initHP();
  initSP();
}

void initSP() {
  sp = new SPIClass(VSPI);
  sp->begin();
  if (radio1.begin(sp)) {
    Serial.println("SP Started !!!");
    radio1.setAutoAck(false);
    radio1.stopListening();
    radio1.setRetries(0, 0);
    radio1.setPALevel(RF24_PA_MAX, true);
    radio1.setDataRate(RF24_2MBPS);
    radio1.setCRCLength(RF24_CRC_DISABLED);
    radio1.printPrettyDetails();
    radio1.startConstCarrier(RF24_PA_MAX, ch1);
  } else {
    Serial.println("SP couldn't start !!!");
  }
}
```

```
void initHP() {
  hp = new SPIClass(HSPI);
  hp->begin();
  if (radio.begin(hp)) {
    Serial.println("HP Started !!!");
    radio.setAutoAck(false);
    radio.stopListening();
    radio.setRetries(0, 0);
    radio.setPALevel(RF24_PA_MAX, true);
    radio.setDataRate(RF24_2MBPS);
    radio.setCRCLength(RF24_CRC_DISABLED);
    radio.printPrettyDetails();
    radio.startConstCarrier(RF24_PA_MAX, ch);
  } else {
    Serial.println("HP couldn't start !!!");
  }
}

void loop() {
  int reading = digitalRead(buttonPin);

  if (reading != lastButtonState) {
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
      buttonState = reading;
    }
  }

  lastButtonState = reading;

  if (buttonState == HIGH)
    two();
  else
    one();
}
```