

Arduino IDE sin IoT: Coche a control remoto

Más adelante verás un coche teledirigido con el móvil y usando internet.

Pero en este caso vamos a usar **DOS ARDUINOS ALVIKS** uno como receptor y otro con envío de órdenes

La fuente de este programa lo puedes encontrar en https://github.com/arduino-libraries/Arduino_Alvik/blob/main/examples/remote_control/remote_control.ino

Se graban los dos programas en los dos alviks, y al apretar OK uno es el que envía la órdenes y el otro le apretamos CANCEL y es el receptor

Es muy curioso que la comunicación lo hace via Wifi pero sin usar ningún punto de acceso exterior, sino entre ellos comunicación pareada (peer)

```
/*  
  This file is part of the Arduino_Alvik library.  
  
  Copyright (c) 2024 Arduino SA  
  
  This Source Code Form is subject to the terms of the Mozilla Public  
  License, v. 2.0. If a copy of the MPL was not distributed with this  
  file, You can obtain one at http://mozilla.org/MPL/2.0/.  
  
*/  
  
// This example shows how to interface 2 Alvik robots via ESPnow.  
// At startup, you can select if an Alvik is a trasmitter by pressing the "check button" or a  
// receiver by pressing "cancel button". Use arrows to move the robot.  
  
#include "Arduino_Alvik.h"  
#include <esp_now.h>
```

```
#include <WiFi.h>

Arduino_Alvik alvik;

uint8_t broadcastAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
uint8_t myData;
esp_now_peer_info_t peerInfo;

int alvik_mode = -1; // 0 is receiver, 1 is sender

bool led_blink = false;

void setup() {
  Serial.begin(115200);
  while((!Serial)&&(millis())>3000));

  alvik.begin();

  WiFi.mode(WIFI_STA);
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }

  while (alvik_mode == -1){
    if (alvik.get_touch_cancel()){
      alvik_mode = 0;
    }
    if (alvik.get_touch_ok()){
      alvik_mode = 1;
    }
  }
  if (alvik_mode == 0){
    esp_now_register_recv_cb(OnDataRecv);
```

```
}  
else{  
    memcpy(peerInfo.peer_addr, broadcastAddress, 6);  
    peerInfo.channel = 0;  
    peerInfo.encrypt = false;  
  
    if (esp_now_add_peer(&peerInfo) != ESP_OK){  
        Serial.println("Failed to add peer");  
        return;  
    }  
}  
}  
  
void loop() {  
    if (alvik_mode==0){  
        alvik.left_led.set_color(led_blink, !led_blink, 0);  
        alvik.right_led.set_color(!led_blink, led_blink, 0);  
        delay(500);  
    }  
    else{  
        if (alvik.get_touch_any()){  
            if (alvik.get_touch_up()){  
                myData = 'F';  
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));  
            }  
            if (alvik.get_touch_down()){  
                myData = 'B';  
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));  
            }  
            if (alvik.get_touch_left()){  
                myData = 'L';  
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));  
            }  
            if (alvik.get_touch_right()){  
                myData = 'R';  
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));  
            }  
        }  
    }  
}
```

```

    }
    if (alvik.get_touch_center()){
        myData = 'S';
        esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
    }
}
alvik.left_led.set_color(0, 0, led_blink);
alvik.right_led.set_color(0, 0, led_blink);
delay(100);
}
led_blink = !led_blink;
}

void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len){
    Serial.print(incomingData[0]);
    switch (incomingData[0]){
        case 'F':
            alvik.drive(7, 0);
            break;
        case 'B':
            alvik.drive(-7, 0);
            break;
        case 'L':
            alvik.drive(0, 45);
            break;
        case 'R':
            alvik.drive(0, -45);
            break;
        case 'S':
            alvik.brake();
            break;
    }
    Serial.println();
}

```

El resultado es espectacular

<https://www.youtube.com/embed/yzlXUp9fais>

Revision #2

Created 28 March 2025 11:01:09 by Javier Quintana

Updated 28 March 2025 11:10:11 by Javier Quintana