

## PS4 Controller am ESP32

Da der ESP32 Bluetooth unterstützt, müssen keine zusätzlichen Module verwendet werden.

Der PS4 Controller ist eine souveräne Lösung für kabellose Steuerung, da er mit dem richtigen Code und ohne weitere Modifikation der Hardware, ein völlig einsatzfähiges Gerät ist und dadurch viel Zeit für Schaltpläne usw. spart.

Die Steuerung kann kinderleicht ausgelegt werden, und ist somit ein enormer Vorteil für gewisse Projekte, da es nahezu jeder steuern kann, oder es innerhalb weniger Minuten beherrscht.

### 1. Library Installation

ZIP Library herunterladen: [github.com/aed3/PS4-esp32/archive/refs/heads/master.zip](https://github.com/aed3/PS4-esp32/archive/refs/heads/master.zip)

#### Platform IO:

New PlatformIO Terminal

```
> pio lib -g install "C:\Users\...\PS4-esp32-master.zip"
```

#### Arduino IDE:

Sketch → Include Library → Add .ZIP Library

### 2. MAC-Adressen Änderung

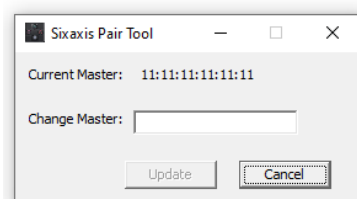
Im Controller wird diejenige MAC-Adresse gespeichert, zu der er sich verbinden möchte.

Um diese zu ändern, braucht man das Programm SixaxisPairTool.

Die MAC im Code und die MAC im Controller müssen übereinstimmen.

Tool herunterladen: [sixaxispairtool.en.lo4d.com/download/mirror-ls1](https://sixaxispairtool.en.lo4d.com/download/mirror-ls1)

Dann Controller anschließen und Programm starten.



Oben steht die derzeitig gespeicherte MAC-Adresse

Unten ist das Eingabefeld zur Änderung der MAC-Adresse

### 3. Wichtige Funktionen

#### PS4.begin()

Diese Funktion ist vergleichbar mit `Serial.begin()`

Sie verbindet den Controller zwar nicht direkt, doch schafft die Möglichkeit für eine Verbindung.

Im Code muss danach nichts mehr eingebaut werden, da sich der Controller automatisch zum ESP32 verbindet, sobald die mittlere PS-Taste gedrückt wird. Es muss nurmehr mit `PS4.isConnected()` abgefragt werden, ob sie schon verbunden sind.

```
#include <PS4Controller.h>

void setup(void)
{
    // Hier wird die MAC-Adresse eingefügt,
    // die zuvor im Controller gespeichert wurde
    PS4.begin("11:11:11:11:11:11");
}
```

Tasten abfragen:

**PS4.Triangle()**                      Grüne Dreiecktaste

**PS4.Left()**                          Steuerkreuz Links

Analoge Inputs:

**PS4.RStickX()**                      X-Koordinate des rechten Joysticks

**PS4.RStickY()**                      Y-Koordinate des rechten Joysticks

Andere wichtige Funktionen:

**PS4.isConnected()**                  Abfragen, ob Controller verbunden ist

**PS4.end()**                              Verbindung zum Controller schließen

**PS4.Battery()**                        Ladestand des Controllers abfragen

**PS4.setLed()**                         Farbe der RGB Led am Controller ändern

Alle restlichen Funktionen sind hier nicht aufgelistet, da fast jeder Funktionsname selbsterklärend ist, und sie durch das Autovervollständigungs-Menü in Visual Studio Code angezeigt werden können.

Einfach „**PS4.**“ eintippen und durchscrollen

In dem folgenden Beispielcode wird eine LED am Pin 5 über die X-Taste am Controller gesteuert

```
#include <Arduino.h>
#include <PS4Controller.h>

#define LED_PIN      5

void setup(void)
{
    PS4.begin("11:11:11:11:11:11");
    pinMode(LED_PIN, OUTPUT);
}

void loop(void)
{
    if(PS4.isConnected())
    {
        if(PS4.Cross())
            digitalWrite(LED_PIN, 1);
        else
            digitalWrite(LED_PIN, 0);
    }
}
```

### Was muss beachtet werden?

- Es funktioniert **ausschließlich mit PS4** Controllern (Keine XBOX, PS3, ... Controller)
- Wenn der Controller gleichzeitig geladen und verwendet wird, muss darauf geachtet werden, dass er nicht mit einem Datenübertragungskabel von einem PC geladen wird, da er sich sonst sofort mit dem PC über USB verbindet, anstatt über Bluetooth zum ESP32.  
Bei einem USB-Kabel ohne Datenleitungen muss dies aber nicht in Betracht gezogen werden.
- Bei der MAC-Adressen Änderung muss zwingend ein **datenübertragungsfähiges** USB-Kabel verwendet werden, ansonsten wird der Controller nicht erkannt.
- Die MAC-Adresse im Code muss gleich sein, wie diejenige die im Controller gespeichert wurde.