

RP2040 Arduino

Autpmount RPi Headless

- <https://github.com/fastddy516/pi-usb-automount>

Installation CLI

- Download **TBD**
- CLI an einen Pfad kopieren der im Path liegt
`sudo cp arduino-cli /usr/local/bin/`

Pfade

- `~/.arduino15` → Boards, Cache, etc
- `~/Arduino` → Libs

RP2040 / RP2350

- `nano ~/.arduino15/arduino-cli.yaml`

[arduino-cli.yaml](#)

```
board_manager:  
  additional_urls:  
    -  
    https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```

- Index updaten
`arduino-cli core update-index`
- Core installieren
`arduino-cli core install rp2040:rp2040`
- Board auflisten → `arduino-cli board listall`
Installierte Cores auflisten → `arduino-cli core list`

FQBNs

- Generic RP2040 `rp2040:rp2040:generic rp2040:rp2040`
- Generic RP2350 `rp2040:rp2040:generic_rp2350 rp2040:rp2040`

Testprojekt

- `mkdir ~/rp2040_zero_demo && cd ~/rp2040_zero_demo`

- | [ws2812.ino](#)

```
#include <Arduino.h>

// WS2812 LED an GPIO16 (bit-banging, da keine Lib)
#define LED_PIN 16
#define LED_COUNT 1 // Eine LED

// Funktion zum Senden eines Bits an WS2812 (Timing für RP2040
angepasst)
void sendBit(bool bit) {
  if (bit) {
    digitalWrite(LED_PIN, HIGH);
    delayMicroseconds(800); // High für 0.8us (1-Bit)
    digitalWrite(LED_PIN, LOW);
    delayMicroseconds(450); // Low für 0.45us
  } else {
    digitalWrite(LED_PIN, HIGH);
    delayMicroseconds(400); // High für 0.4us (0-Bit)
    digitalWrite(LED_PIN, LOW);
    delayMicroseconds(850); // Low für 0.85us
  }
}

// Funktion zum Senden einer Farbe (GRB-Format für WS2812)
void sendColor(uint8_t r, uint8_t g, uint8_t b) {
  // Sende G (8 Bits)
  for (int i = 7; i >= 0; i--) sendBit((g >> i) & 1);
  // Sende R
  for (int i = 7; i >= 0; i--) sendBit((r >> i) & 1);
  // Sende B
  for (int i = 7; i >= 0; i--) sendBit((b >> i) & 1);
  delayMicroseconds(50); // Reset-Pause
}

void setup() {
  Serial.begin(115200); // Serieller Port initialisieren
  while (!Serial); // Warte auf Verbindung

  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  Serial.println("RP2040 Zero Demo Started");
  Serial.println("LED an GPIO16 wird blinken (Rot, Grün, Blau)");
}
```

```
void loop() {  
  // LED Rot blinken  
  sendColor(255, 0, 0); // Rot  
  Serial.println("LED: Rot");  
  delay(1000);  
  
  // LED Grün  
  sendColor(0, 255, 0); // Grün  
  Serial.println("LED: Grün");  
  delay(1000);  
  
  // LED Blau  
  sendColor(0, 0, 255); // Blau  
  Serial.println("LED: Blau");  
  delay(1000);  
  
  // LED Aus  
  sendColor(0, 0, 0);  
  Serial.println("LED: Aus");  
  delay(1000);  
}
```

- `arduino-cli compile --fqbn rp2040:rp2040:generic . -v --build-path ./build`
Hinweis: Der *Dateiname.ino* muss gleich dem Pfad sein!
- RP2040 in Bootmodus versetzen ...
- Flashen: `arduino-cli upload --fqbn rp2040:rp2040:generic -v .`

From:
<https://drklipper.de/> - Dr. Klipper Wiki

Permanent link:
https://drklipper.de/doku.php?id=projekte:sekwai:rp2040_arduino&rev=1754076839

Last update: **2025/08/01 21:33**

