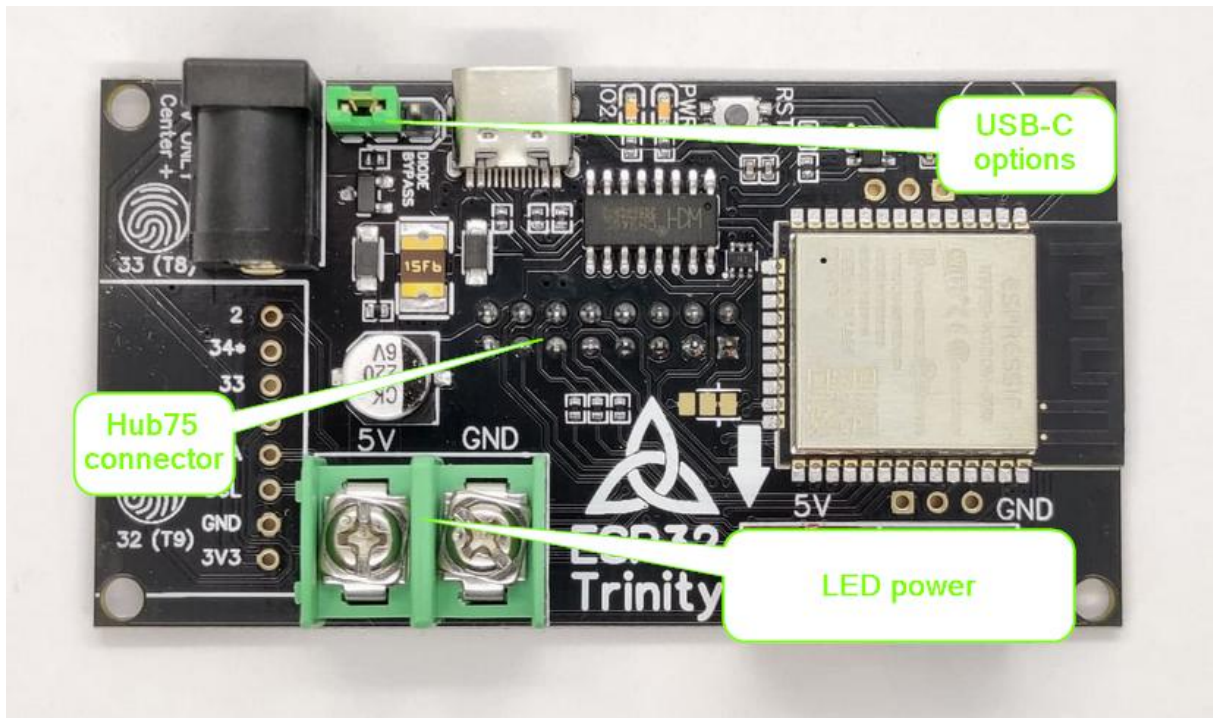


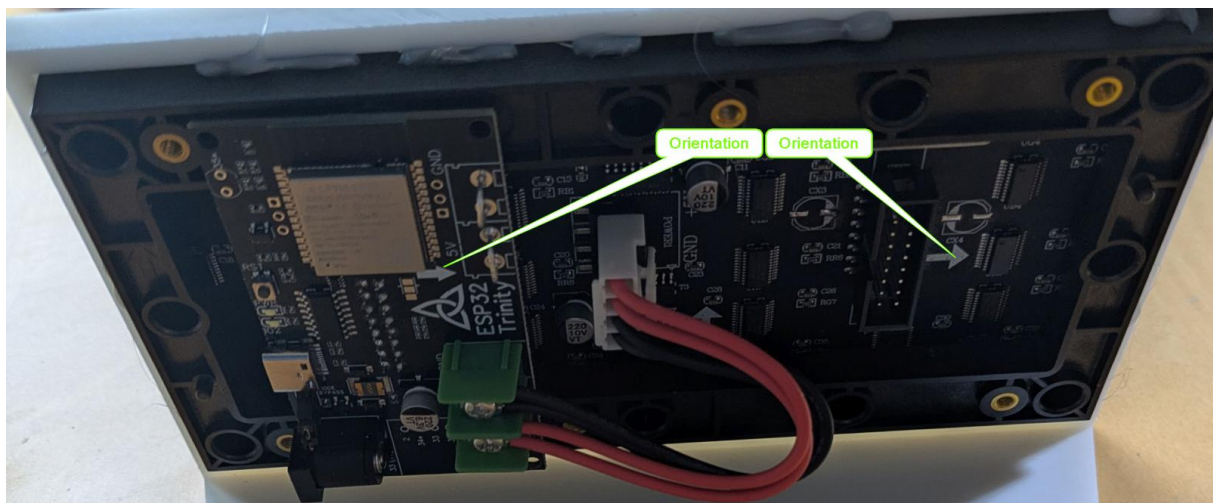
Kurzanleitung ESP32 Trinity WiFi Tetris Uhr

Basierend auf dem Open Source Projekt <https://esp32trinity.com/>

Zusammenbau



- 1) Die HUB75 Buchse auf der Rückseite der Platine anlöten.
- 2) Den 3Pin Anschluss (USB-C options) oben links anlöten.



- 3) Die Platine auf das LED Modul setzen. Dabei beachten dass die Pfeile auf Platine und Modul die gleiche Ausrichtung haben. **Beim Einsetzen sicherstellen dass die Buchse mittig im HUB75 Konnektor landet!**
- 4) Den ‚LED Power‘ Anschluss mit dem LED Modul verbinden.
- 5) Das Led Modul in den Rahmen kleben.

Hardware config

Über die Position des Jumpers (USB-C Options) kann man festlegen ob das LED Modul auch über den USB-C Anschluss mit Spannung versorgt wird. Mit Jumper links wird nur der ESP32 versorgt; Jumper rechts (Diode Bypass) legt fest dass auch das LED Modul versorgt wird.

Alternativ kann man das LED Modul über die Rundbuchse mit 5Volt versorgen.

Bei meinen Tests zog das Modul maximal 300mA, daher sollte die Versorgung über USB-C kein Problem darstellen.

Inbetriebnahme

Beim ersten Start (config Modus) stellt die Uhr ein WiFi Netzwerk mit der Kennung ,WiFiTetris' zur Verfügung. (Kennwort: ,clock123'). Über dieses kann man sein eigenes WiFi Netzwerk eintragen und einige Parameter der Uhr konfigurieren (Zeitzone, Geschwindigkeit der Animation, ..)

Nach erfolgreicher Konfiguration speichert die Uhr die Netzwerkkennung und verbindet sich automatisch nach dem Start.

Über den Reset Knopf und erneueterem Reset in der Anfangsphase kommt man wieder in den config Modus.

<div>In WifiTetris anmelden 192.168.4.1</div>	<div>In WifiTetris anmelden 192.168.4.1</div>
<h1>WiFiManager</h1>	
WifiTetris	
<div>Configure WiFi</div>	
<div>Info</div>	SSID <input type="text"/>
<div>Exit</div>	Password <input type="password"/>
	<input type="checkbox"/> Show Password
	Time Zone <input type="text" value="Europe/Dublin"/>
	24 hour Clock <input checked="" type="checkbox"/> Force every digit reset <input checked="" type="checkbox"/> Animation Delay (smaller == faster)
	<input type="text" value="50"/>
<div>Update</div>	Set clkphase false <input type="checkbox"/> Set Driver to FM6126A <input type="checkbox"/> 64x64 panel <input type="checkbox"/>
<div>No AP set</div>	<div>Save</div>

Programmierung

Mittels der Sourcen auf dem ESP32 Trinity Github und der Arduino IDE mit ,ESP32 Extension' kann die Platine über den USB-C Anschluss programmiert werden. Dabei am besten das Board ,Wemos D1 Min ESP32' auswählen.