

# Morserino-32 Technischer Überblick

Das Morserino-32 basiert auf dem WiFi-Modul LORA 32 V.2 von Heltec (siehe <http://www.heltec.cn/project/wifi-lora-32/?lang=de>). Dieses Modul verfügt über viele integrierte Funktionen, die den Kern der Morserino-32-Funktionalität ausmachen:

- MCU (ESP32, 240 MHz Taktfrequenz), RAM (520 KB SRAM) und Flash-Speicher (64 MB)
- OLED-Monochrom-Grafikdisplay (128 x 64 Pixel)
- USB, Wifi, LoRa und Bluetooth an Bord (Bluetooth wird derzeit beim Morserino-32 nicht verwendet)
- LiPo-Ladeschaltung an Bord

Wir brauchen also nur etwas Peripherie, um aus diesem Juwel ein multifunktionales Morse-Code-Gerät zu machen ...

## Touch Paddel

Da die ESP32-MCU bereits über einen Touch-Eingang verfügt (wir verwenden dazu Pins 2 und 12), müssen wir nur die "Mechanik" realisieren. Zwei Paddles aus Platinen-Material sind über Platinen-Steckverbinder mit dem Mainboard verbunden.

## Externe Paddel

Eine 3-polige Klinkenbuchse bietet Anschlussmöglichkeiten für zwei ESP32 GPIO-Pins (Pins 32 und 33), die mit Pull-Up Widerständen auf 3,3 V gezogen sind.

## Audio für Lautsprecher oder Kopfhörer

Zwei Ausgangspins des ESP32 (Pins 22 und 23) werden verwendet, um ein Audiosignal mit Software-gesteuerter Lautstärke zu generieren: Ein Pin erzeugt ein Rechtecksignal mit der gewünschten Frequenz und 50 % Tastverhältnis, ein anderer Ausgang erzeugt ein Hochfrequenzsignal (32 kHz) mit variablem Tastverhältnis. Diese beiden Signale werden mit Hilfe eines UND-Gatters aus zwei diskreten Transistoren (T1 und T2) gemischt, wodurch ein pulsbreitenmoduliertes Audiosignal entsteht, das direkt in einen 60-Ohm-Lautsprecher eingespeist wird. Der Lautsprecher fungiert als billiges Tiefpassfilter und ändert so die Pulsbreitenmodulation in Amplitudenmodulation.

Bei Verwendung von Kopfhörern wird der Lautsprecher durch einen Widerstand (R11) und die beiden Lautsprecher des Kopfhörers in Reihe ersetzt, um das Audiosignal auf einen angenehmen Pegel zu bringen.

Dieses Arrangement erzeugt Audio mit geringem Stromverbrauch, aber sicherlich nicht in WiFi-Qualität (das resultierende Audio ist kein Sinuston, sondern hat ziemlich starke 2., 3. und 5. Oberwellen).

## **Audioausgang (Line-Out)**

Eine weitere Audioausgabe wird an Pin 17 des ESP32 erzeugt. Dies wird durch ein RC-Tiefpassfilter 3. Ordnung mit einer Grenzfrequenz um die 1000 Hz (R15 - R18, C8 - C11) geführt, wodurch ein Audiosignal erzeugt wird, das fast ein reiner Sinus-Ton mit 700 Hz ist (Oberwellenunterdrückung mindestens 24 dB). Dies kann für Anwendungen verwendet werden, die einen ziemlich reinen Sinuston erfordern, zum Beispiel Mumble (benutzt vom CW-over-Internet-Protokoll - iCW).

## **Audioeingang**

Zur Dekodierung von Morse-Signalen, die als Audiotöne verfügbar sind, wird das Signal mit einem Single-Rail-Niederspannungs-Operationsverstärker (IC1) ausreichend verstärkt, damit es in den Analog-Digital-Konverter-Eingang des ESP32 (Pin 36) eingespeist werden kann. Die Verstärkung kann mit einem Trimpotentiometer (R21) eingestellt werden.

## **Schnittstelle zum Sender**

Um einen Sender oder Transceiver tasten zu können, steuert das Signal von Pin 25 einen MOSFET-Optokoppler (OC1) als "Halbleiterrelais". Dadurch wird eine vollständige galvanische Trennung vom Sender erreicht und das Schalten von bis zu 60 V positiver oder negativer Spannung ermöglicht, so dass alle modernen Sender oder Transceiver ohne zusätzliches Interface getastet werden können.

## **Drehgeber und Drucktastenschalter**

Als Hauptbedienelement für den Morserino-32 wird ein 24-poliger 24-stelliger Drehgeber mit integriertem Drucktastenschalter verwendet, wobei RC-Kombinationen die ESP32-Eingangspins (Pins 38 und 39) auf 3.3 V hochziehen und gleichzeitig das Prellen reduzieren.

Ein zweiter Tastschalter ist mit Pin 0 des ESP32 verbunden.

## **Batterie EIN / AUS-Schalter und Polyfuse**

Ein LiPo-Akku ist über einen Schiebeschalter und eine selbst-rückstellende 500-mA-Sicherung an den ESP32 angeschlossen, die den Akku abschaltet, falls irgendwo ein Kurzschluss auftritt.

## **Serielle Schnittstelle**

Auf der Platine kann eine 4-polige Stiftleiste angelötet werden (links vom Anschluss für das externe Paddle). Über diesen Anschluss kann man externe Geräte über die serielle Schnittstelle des ESP32 anschließen (die Header sind mit R (Receive), T (Transmit), + (3,3 V) und - (Ground) gekennzeichnet. R und T sind mit den Pins RX und TX des ESP32 verbunden, die positive Spannung wird vom VEXT-Pin genommen.

Dies könnte zum Beispiel verwendet werden, um ein GPS-Modul an den Morserino-32 anzuschließen.

Obwohl dies derzeit nicht von der Software unterstützt wird, ist es relativ einfach, eine LoRa APRS-Bake zu realisieren.