

# Morserino-32 Technical Overview

The Morserino-32 is based on the Heltec WiFi LORA 32 V.2 module (see <http://www.heltec.cn/project/wifi-lora-32/?lang=en>). This module comes with many integrated features, that really build the core of Morserino-32's functionality:

- MCU (ESP32, 240 MHz clock speed), RAM (520 KB SRAM) and Flash Memory (64 MB)
- OLED monochrome graphics display (128x64 pixels)
- USB, Wifi, LoRa and Bluetooth on board (Bluetooth currently not used for the Morserino-32)
- LiPo charging circuit on board

So in fact we only need some periphery to turn this jewel into a multi-functional Morse code device...

## Touch Paddles

As ESP32 MCU already has touch input (we use Pins 2 and 12), we only need to supply the mechanics. Two paddles made from PCB material are connected to the mainboard through PCB edge connectors.

## External Paddles

A 3-pole phone jack offers connectivity to two ESP32 GPIO pins (Pins 32 und 33) that are pulled-up to 3.3 V.

## Audio for Loudspeaker or Headphones

Two output pins of the ESP32 (Pins 22 and 23) are being used to generate an audio signal with the volume controlled by software: one pin generates a square signal with the desired frequency and 50% duty cycle, another output creates a high frequency signal (32 kHz) with a variable duty cycle. These two signals are mixed with the help of an AND gate made from two discrete transistors (T1 and T2), thus creating a pulse-width modulated audio signal, with the resulting signal fed directly into a 60 Ohm loudspeaker. The speaker acts as a cheap low pass filter, thus changing the pulse-width modulation into amplitude modulation.

For headphone use the speaker is replaced with a resistor (R11) and the two speakers of the headphone in series, to bring the audio level down to a comfortable level.

This arrangement creates low power audio, but certainly not in HiFi quality (the resulting audio is not a sinus tone, but has pretty strong 2nd, 3rd and 5th harmonics).

## Audio Output (Line-Out)

Another audio output is generated on Pin 17 of the ESP32. This time this is fed through a 3rd order RTC low pass filter with a cut off frequency around 1000 Hz (R15 - R18, C8 - C11), thus creating an audio signal

that is almost a pure sine tone with 700 Hz output (harmonics are at least 24 dB down). This can be used for applications that require a pretty pure sine tone, for example Mumble (used by the CW over Internet protocol - iCW).

## **Audio Input**

In order to decode Morse signals that are available as audio tones, a single-rail low-voltage operational amplifier (IC1) is being used to amplify the signal sufficiently so that it can be fed into the ESP32's Analog-Digital Converter input (Pin 36). Amplification can be controlled by a trimmer potentiometer (R21).

## **Interface to Transmitter**

In order to be able to key a transmitter or transceiver, the signal of Pin 25 is controlling a MOSFET opto-coupler (OC1) as a "solid state relay". This achieves complete galvanic insulation from the transmitter, and allows to switch up to 60 V of positive or negative voltage, which means that all modern transmitters or transceivers can be keyed without the need for additional circuitry.

## **Rotary Encoder and Push Button Switches**

A 24 indent 24 position rotary encoder with integrated push button switch is being used as the primary user interface for the Morserino-32, with RC combinations pulling up the ESP32 input pins (Pins 38 and 39) and at the same time reducing bouncing.

A second push-button switch is connected to Pin 0 of the ESP32.

## **Battery ON/OFF switch and Polyfuse**

A LiPo battery is connected to the ESP32's battery charging circuitry through a sliding switch and a 500 mA self-resetting fuse that switches the battery off should there be a short circuit anywhere down the line.

## **Serial Port**

On the PCB there is a provision for 4 pin headers (to the left of the external paddles connector). Through this connector you can connect external devices through the serial port of the ESP32 (the headers are marked as **R** (Receive), **T** (Transmit), **+** (3.3V) and **-** (ground). R and T are connected to Pins RX and TX of the ESP32, the positive voltage taken from the VEXT pin.

This could be used, for example, to connect a GPS module to the Morserino-32. While this is currently not supported by software, it would be relatively straightforward to create a LoRa APRS beacon.