

# LoRa Chat

[Originalartikel](#)

[Backup](#)

```
<html> <section class=„section section-description“ readability=„7“>
```

```
    <div class="description" readability="21">A decentralised PoC chat
    program for the Heltec ESP32 based LoRa capable boards.
```

```
<p>Each board can become a node and join in chatting to other boards! It is an ad-hoc setup, and
will work so long as you have radio line of sight to the other interlocutors!&#13;</p><p>These little
boards are great! They have a fast ESP32 uC with a built in OLED and LoRa chipset for comms.
&#13;</p><p>TODO&#13;<br/>*) Add a limited re-transmission option - so nodes will retransmit
messages to others, to increase the range of the potential 'mesh-net' :P&#13;<br/>*) Add optional
encryption layer&#13;<br/>*) Maybe add some level of message trust/integrity&#13;<br/>*) Check
code doesn't have any low-hanging fruit vulns (BOF could be likely - I'm not that familiar with how
secure String types are in Arduino libs)&#13;<br/>*) Make general/cosmetic
improvements&#13;<br/>*) Support a capacitive touch keyboard or other input
device...&#13;<br/>*) Add a cool logo to start up screen (desirable)</p></div>
```

```
</section><section class="section section-details" readability="22">
    <div class="post-content details-content" readability="72">
        <p>We're making the most of the board's capabilities - we're
using hte LoRa.h file by Sandeep Mistry, as well as the SSD1306.h driver for
the ESP8266 (it's compatible here). This gives us an SPI interface to the
LoRa chip, as well as an interface for the OLED screen.&#160;</p><p>NB - the
setup() code and header choices were taken from here: <a
href="https://hackaday.io/project/27791-esp32-lora-oled-module">https://hack
aday.io/project/27791-esp32-lora-oled-module</a><br/></p>
```

```
<p>Most of the code is actually making stuff look ok... The only processing is to check first if the LoRa
stack has a packet for us, and then process it. Then we read chars from the serial port in, and
assemble them with the user's Nickname. <br/></p> <p>We send the nickname with the message
in the LoRa packet, so no node has to keep track of users at all - they just process data. This keeps it
nicely decentralised - you wanna change your nick? Just reset the board... :P <br/></p> <p>At the
moment - it's a requirement for the boards to be connected to a laptop or some other interface for
sending serial data. You could easily use this code to write a wrapper for XMPP (just dump data to
serial and back again, though you'd need to figure out channels) and you could integrate data from
other sources over WiFi without any issues. With a bigger screen and some sort of input device, you
could build IM-Me style devices, no problem - it's really easy!<br/></p> <p>LoRa is very long range,
if a little slow on the data rate. In theory you could have these dotted around the place - fair warning,
they don't do too well with walls or stuff getting in the way, so line-of-sight is important. </p>
```

```
</div>
</section><section class="section section-files">
    <table class="files-table"><tbody readability="1"><tr
class="component-" readability="4"><td class="file-details"
readability="10">
```

```

        <p>
            <h3><a download="" target="_blank"
href="https://cdn.hackaday.io/files/278171190122528/LoRaChat.ino">LoRaChat.i
no</a></h3>
                First Version - mirrored from the GitHub
(check the git link for latest)
        </p>
        <p>
            x-arduino -
            3.66 kB -
            10/21/2017 at 12:01
        </p>
    </td>
    <td class="file-type file-preview">
    </td>
    <td class="file-type">
        <a download="" target="_blank"
href="https://cdn.hackaday.io/files/278171190122528/LoRaChat.ino">
        <div class="icon icon-download">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
viewbox="0 0 16 16"><title>download-circle</title></svg></div>
            Download
        </a>
    </td>
</tr></tbody></table>
</section><section class="section section-components">
    <ul class="section-component-list"><li class="component-151627">
        1
        &#215;
        Heltec (or clone) ESP32 w/ SX1278 OLED module
        for example: <a target="_blank" rel="noopener noreferrer"
href="https://www.aliexpress.com/item/TTG0-loraSX1278-ESP32-0-960LED-16-Mt-b
ytes-128-Mt-bit-433Mhz-for-
arduino/32824758955.html">https://www.aliexpress.com/item/TTG0-loraSX1278-ES
P32-0-960LED-16-Mt-bytes-128-Mt-bit-433Mhz-for-arduino/32824758955.html</a>
    </li>
    </ul>
</section><section class="section section-instructions"
readability="1">
    <ul class="instructions-list post-content section-
instructions-list j-instructions-list" readability="31"><li readability="3">
        1
        <!-- Default to step number if no title -->
        <p>Install Dependencies for Board</p>
        <div class="instruction-list-item"
readability="7"><p>Arduino IDE has support for ESP32 boards, so add this
using the Board Manager. <br/></p></div>
    </li>
    <li readability="0">
        2
        <!-- Default to step number if no title -->

```

```

        <p>Install Libraries</p>
        <div class="instruction-list-item"
readability="6"><p>You will need the following libraries:</p><ul><li>LoRa
Library by Sandeep Mistry</li><li>ESP8266 oled driver for SSD1306 by Daniel
Eichorn and Fabrice Weinberg</li></ul></div>
        </li>
        <li readability="0">
            3
            <!-- Default to step number if no title -->
            <p>Compile and Upload</p>
            <div class="instruction-list-item"
readability="6"><p>It's a simple as Ctrl + U (or Ctrl + R to check
first...)<br/></p></div>
            </li>
        </ul><p>
            <a href="https://hackaday.io/project/27817/instructions"
class="grey-gold-button medium-button show">View all 3 instructions</a>
        </p>
    </section><section class="section last">

```

```

</section> </html>

```

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

<https://schnipsl.qgelm.de/doku.php?id=wallabag:lora-chat>

Last update: **2021/12/06 15:24**

