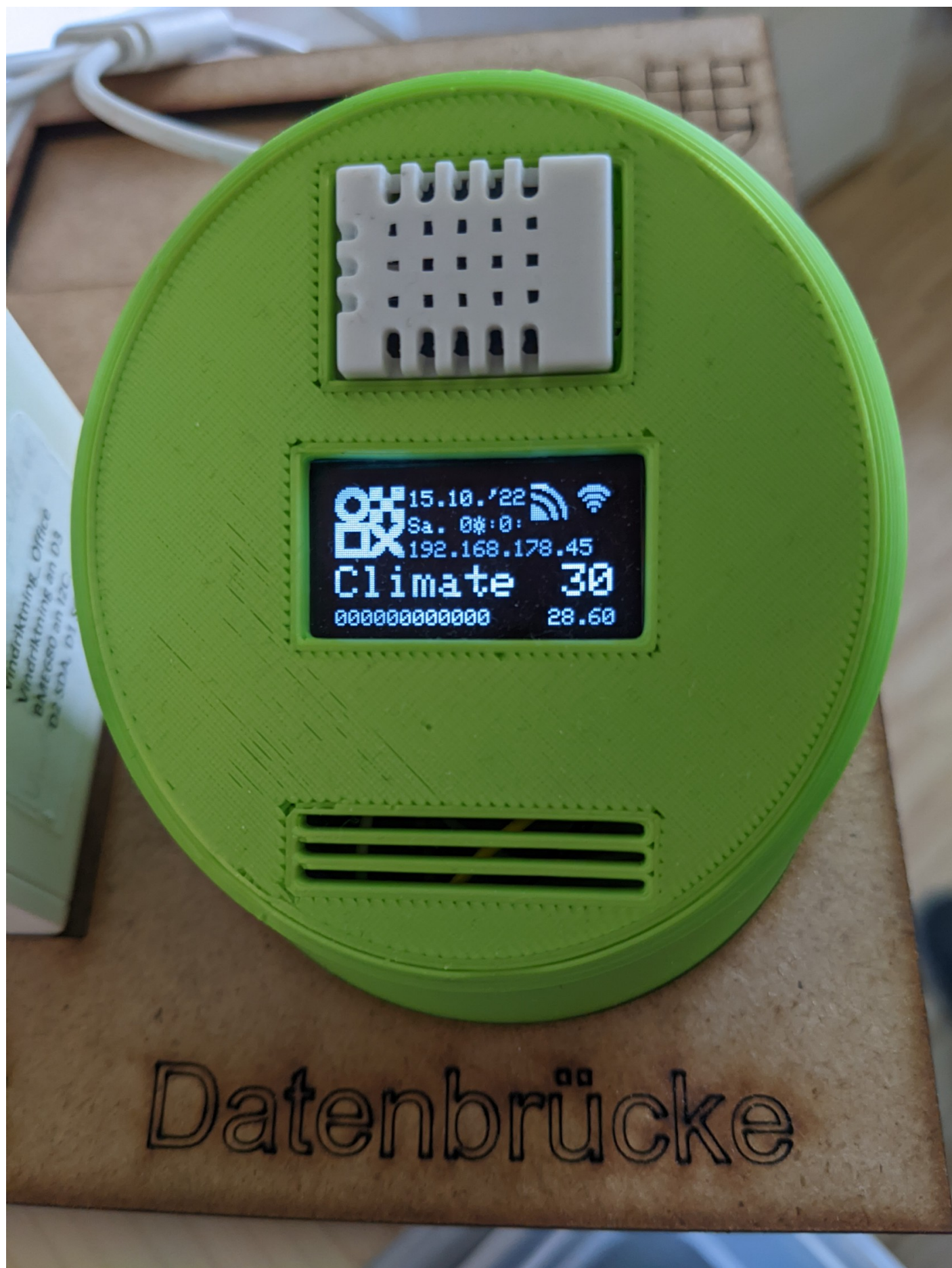


ESPnow-Bridges

Derzeit stehen zwei Datenbrücken zwischen dem ESPnow- und dem WLAN-Protokoll im Haus. Eines davon an der Theke von "The Office", eines in der "X-Zone" im 1. OG.



Datenbrücke

Hardware

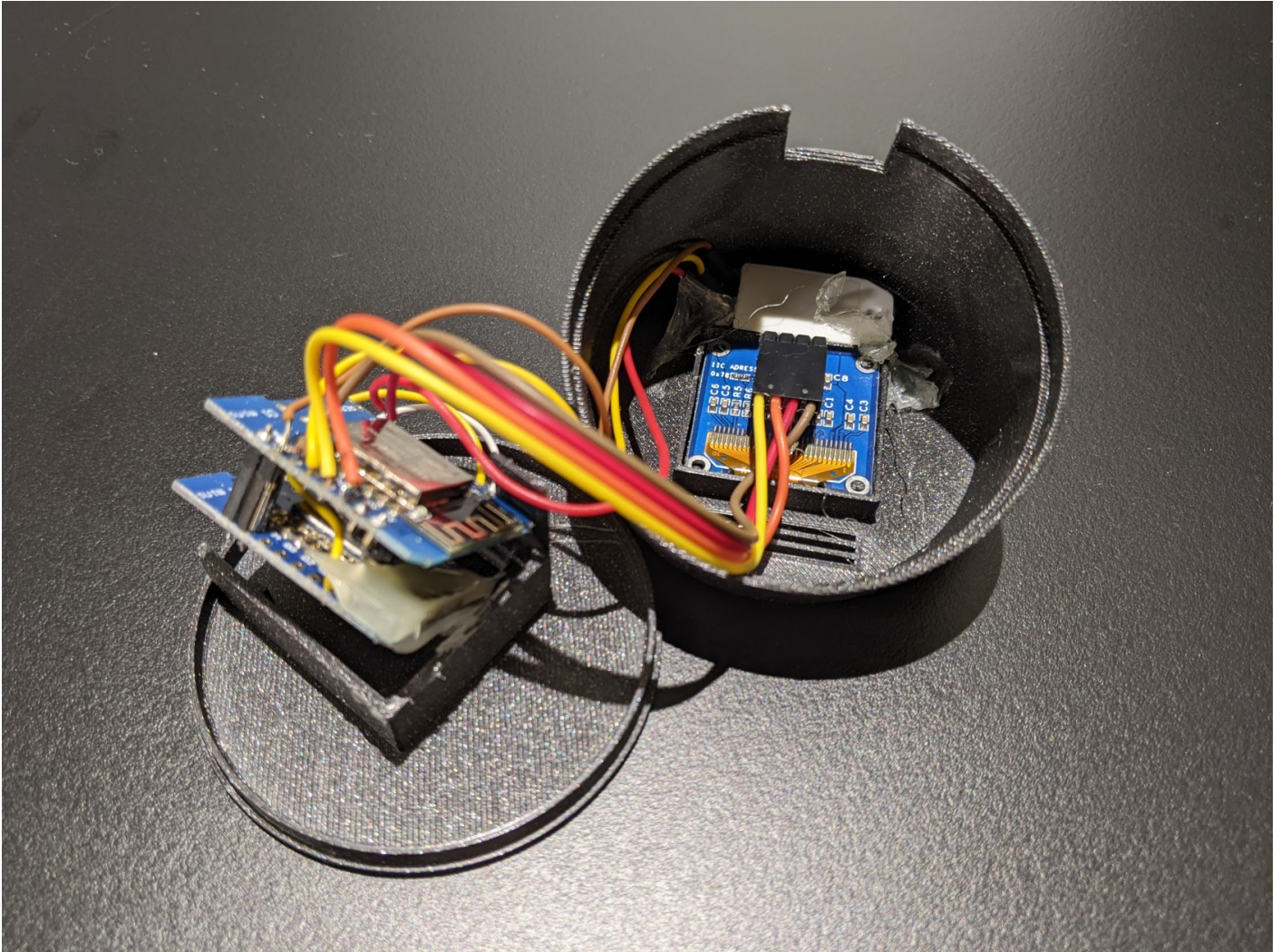
Die Bridges sind in einem kleinen 3D-gedruckten Gehäuse (aus [Thingiverse](#)). Wie auf Thingiverse beschrieben ist hier ein DHT22-Sensor (Luftfeuchte und Temperatur) und ein 0,96" OLED-Display verbaut.

Im Bild sieht man eine solche Datenbrücke mit Datum, Wochentag, nicht funktionierender Uhrzeit (Bug, muss gefixt werden!), MQTT-Verbindung (erstes Icon rechts oben), WLAN-Empfang (zweites Icon rechts oben), der eigenen IP-Adresse (hier 192.168.178.45). Es folgt ein Datenpaket wie es via MQTT weitergegeben wird. Der Sender ist ein Klimasensor "Climate" welcher drei Daten sendet: 30 (Luftfeuchtigkeit in %), 28.6 (Temperatur) sowie 000000000000 (Batteriespannung; es gibt in diesem Gerät keine Batterie). Der Sender ist in diesem Fall das eigene Gerät mit dem eingebauten DHT22-Sender.

Innen werkeln zwei D1-mini ESP8266. Einer ist dabei im "normalen" WLAN, ein weiterer horcht nur auf ESPnow-Nachrichten. Sie sind via UART verbunden. Auf ihnen laufen je ein Arduino-Skript, welches größtenteils von [MrDIY](#) übernommen wurde.

1. Auf dem D1-mini für ESPnow befindet sich "gateway_receiver_MLR.ino"; dieses Programm "horcht" ständig auf ESPnow-Kommunikation und hat die Software-Mac-Adresse 12:34:56:78:9A:BC
2. Auf dem D1-mini für die WLAN-Kommunikation befindet sich "gateway_min_MLR.ino", an ihm sind auch das OLED und der DHT22 angeschlossen.

Der zweite D1-mini sendet alle ESPnow-Daten, die vom ersten D1-mini kommen, mittels MQTT weiter an den hausinternen Server "berta". Auch nimmt er einmal pro Minute die Messwerte des DHT22-Sensors und schickt diese ebenso an "berta". berta kümmert sich dann um die Weiterleitung an den großen externen Server "alfons". Alle Daten werden in die dort installierte influxDB-Datenbank geschrieben.



Warum eigentlich ESPnow???

ESPnow ist ein vom ESP-Hersteller Espressif entwickeltes Protokoll. Bei "normalem" WLAN müssen sich auch IoT-Geräte immer beim Router anmelden und die Verbindung halten; das kostet aber Strom, was nicht jeder IoT-Sensor gerne hat. Etwa bei autark arbeitenden Wetterstationen, die nur via Solar betrieben werden, gilt es immer, Strom zu sparen. Daher schalten sich diese IoT-Sensoren auch ab, wenn sie keine Daten nehmen oder senden. Nun müsste sich so ein Sensor aber bei jedem Start erneut mit dem Router verbinden, was durchaus mehrere Sekunden dauern kann - Zeit, während der er mit voller WLAN-Leistung funkt! Bei einem ESP32 sprechen wir da von bis zu neun Sekunden und einem Stromverbrauch von bis zu 120mA! Dagegen funktioniert das ESPnow-Protokoll viel einfacher. Der Sender "ruft" einmal kurz seine Datenpakete entweder einem ESPnow-Partner zu (dazu muss er aber die MAC-Adresse des Partners kennen), oder er ruft an alle verfügbaren Geräte (sog. Broadcasten). Dieses "Rufen" dauert nur wenige zehn bis einhundert

Millisekunden! Der Stromverbrauch reduziert sich also drastisch.

Revision #11

Created 15 Oktober 2022 13:12:49 by Fabian

Updated 12 Januar 2024 08:02:50 by Fabian