# Bauanleitung WS-WordClock

Achtung: Die Uhr kann im Moment nur mit Sommer-/Winterzeit bis zum Jahr 2022 umgehen (Wechseldaten sind hart kodiert).

Das Projekt am Besten mit TrueStudio bauen.

## Material

1x Light Dependent Resistor: Reichelt LDR 03

1x IR-Empfänger: Reichelt IR-Empfänger-Module TSOP31236 36kHz

1x Controller-Platine: NUCLEO F411RE

1x Connector-Platine: Fertigen lassen. Ist optional, geht auch ohne diese Platine.

LEDs: WS2812B strip, 30 LED/m, mind. 115 LEDS ⬄ mind. 3,80m strip

1x IKEA Ribba Bilderrahmen

1x WLAN-Modul: ESP8266 (bei ebay, oder amazon), ACHTUNG: Die Version, mit den Pfostenverbindern kaufen (siehe Bild)  

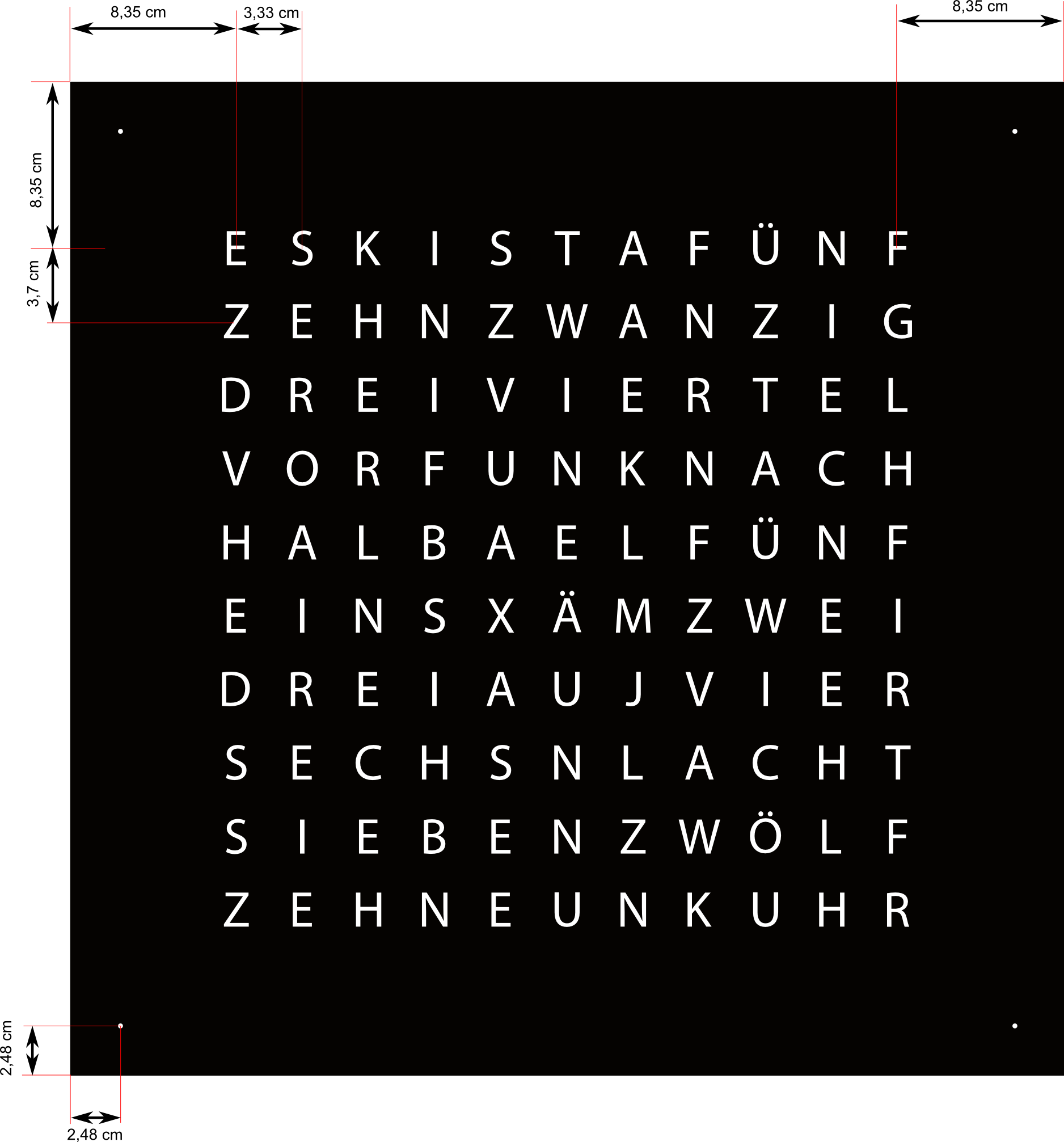

Frontplatte: Drucker, der die Buchstabenmaske auf eine Folie druckt und dann auf die Plexiglasscheibe ziehen kann.

Inlay: Lasercutter, der aus dünnem Holz (o.ä. Material) die Stege lasern kann.

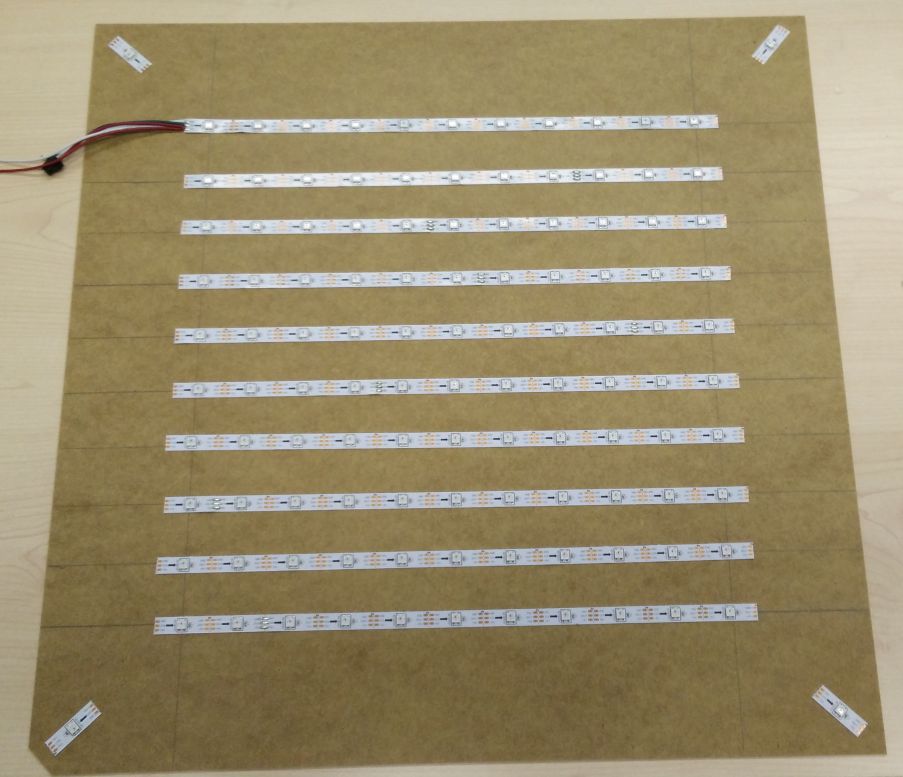
Kabelmaterial, Lötmaterial, etc.

## LED Strips aufkleben

LED-Strips zuschneiden und auf RIBBA-Rückplatte kleben. Hierzu am besten die Positionen der LED Strips zuerst anzeichnen. Bemaßungen siehe Abbildung.



Ergebnis:

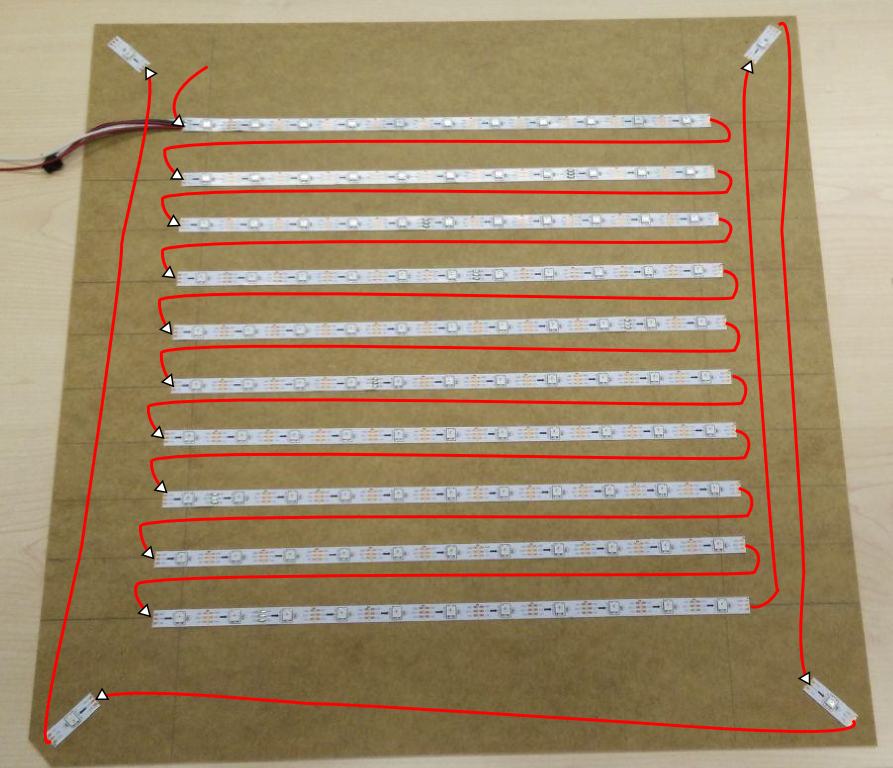


## LED Strips verbinden

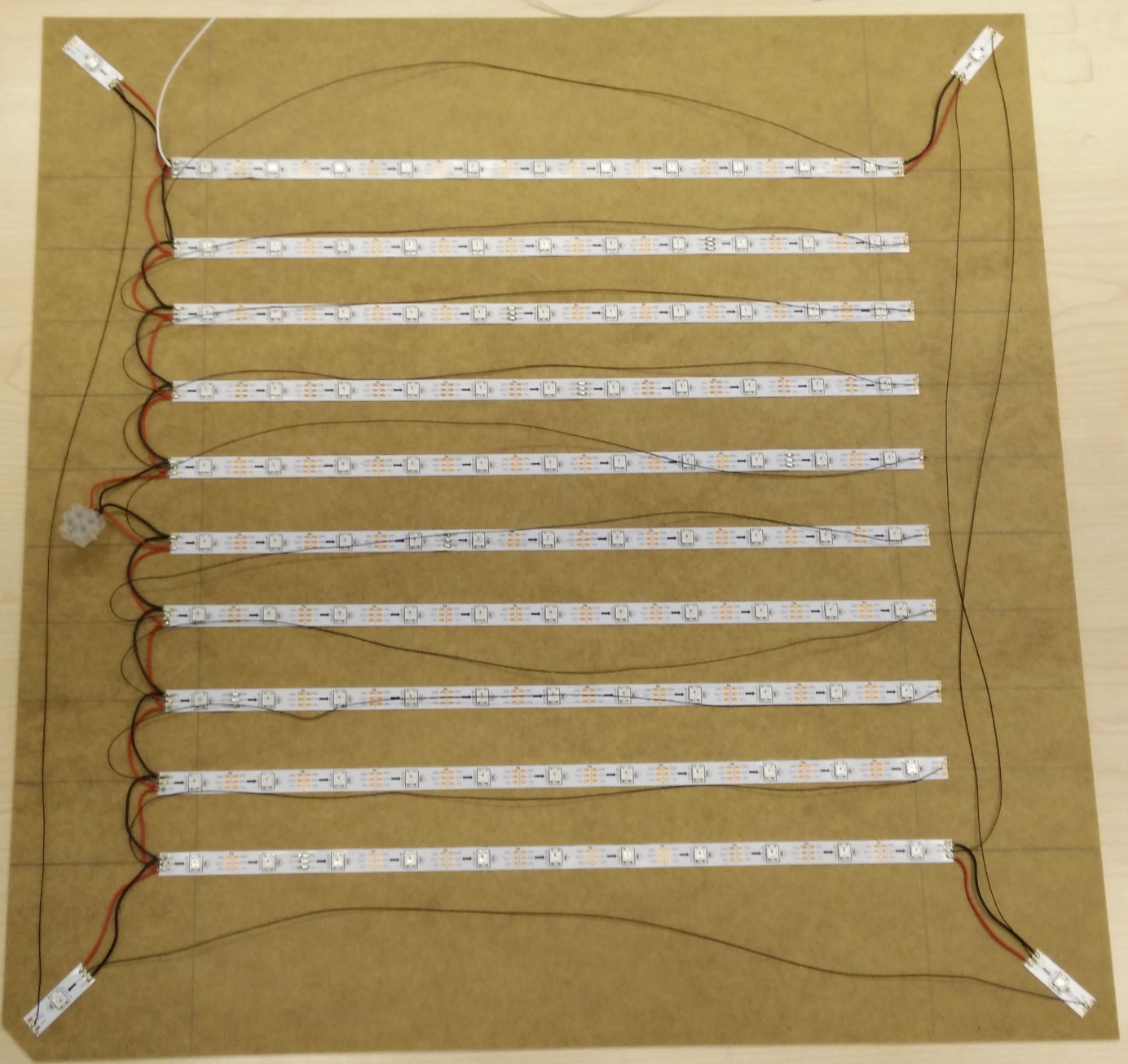
Die LED Strips müssen nun elektrisch verbunden werden. Hierzu gehört +5V, GND und DigIn bzw. DigOut.

Die +5V und GND Pads sollten auf kürzestem Weg parallel verbunden werden. Für ein Beispiel siehe Ergebnisbild dieses Schritts.

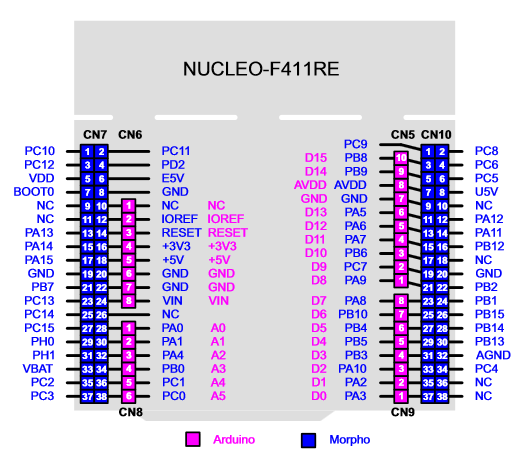
Die Verkabelung der DigIn und DigOut Signale muss nach folgendem Schema vorgenommen werden. Der jeweilige DigOut Ausgang muss mit dem folgenden DigIn Eingang verbunden (durch weiße Dreiecke gekennzeichnet) werden.



Ergebnis:



# Verbindungen herstellen



|  |  |  |  |
| --- | --- | --- | --- |
| **Funktion** | **Pin (Nucleo-F411RE)** | **Pin-Function** | **ESP32** |
| DCF-77 | | |  |
| Data-In | PB8 / D15 | GPIO (Out) | D34 |
| Power on | PB9 / D14 | GPIO (Out) | D35 |
| +3.3V | AVDD (CN10, Pin7) | 3V3 (Out) | OK |
| GND | GND (CN10, Pin9) | GND | OK |
| Light Dependent Resistor (LDR) | | |  |
| Analog-In (ebenfalls zu PC13 / PA4 verbinden s.u.) | PA1 / A1 | Analog (IN) | D32 |
| Ground for LDR | PB7 | GPIO (Out) | D33 |
| interner Pull-Up: PA1 (CN7, Pin30) / PA4 | PC13 (CN7, Pin 23) | GPIO (Out) | D25 |
| IR Remote | | |  |
| Digital-In | PA0 / A0 | GPIO (In) | D26 |
| +3.3V | VDD (CN7, Pin5) | 3V3 (Out) | OK |
| GND | GND (CN7, Pin19) | GND | OK |
| WS2812 LED strip | | |  |
| Digital-Out (ebenfalls zu E5V verbinden s.u.) | PB0 (CN7, Pin34) – PB0 / A3 | GPIO (Out) |  |
| Externer Pull-Up: PB0 (CN7, Pin23) | E5V (CN7, Pin6) | GPIO (Out) |  |
| WLAN Modul (Achtung: Solder bridges auf Nucleo umlöten!) | | |  |
| +3.3V |  | 3V3 (Out) |  |
| GND |  | GND |  |
| TX | PA3 (USART2\_RX) | USART\_RX |  |
| RX | PA2 (USART2\_TX) | USART\_TX |  |
| Power-Down | PA10 (GPIO) – PB3 / D3 | GPIO (Out) |  |
| GPIO0 | PC4 (GPIO) – PB4 / D5 | GPIO (Out) |  |
| GPIO1 | PB3 (GPIO) – PA10 / D2 | GPIO (Out) |  |
| /Reset | PB5 (GPIO) – PB5 / D4 | GPIO (Out) |  |

# WLAN Modul

<http://playground.boxtec.ch/doku.php/wireless/esp8266>

ACHTUNG: SolderBridges auf Nucleo umlöten:

SB13 & SB14 öffnen und SB62 & SB63 schließen

Um das WLAN Modul direkt anzusprechen (via COM-Port), kann in der main.h ein define gsetzt werden, welches den UART zwischen STM32 und ESP8266 deaktiviert, aber das Modul trotzdem powered.

# ESP32 Version

Input:

<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

<https://www.reichelt.de/ac-dc-wandler-85-240-v-ac-5-v-dc-modul-irm-10-5-p157852.html?PROVID=2788&gclid=EAIaIQobChMInaq3gaGn5wIVDPlRCh36ZA7BEAQYASABEgL-7fD_BwE&&r=1>

<https://www.conrad.de/de/p/entwickler-platine-sbc-nodemcu-esp32-1656367.html>