

Alternative Messwertanzeige

Die folgende Anleitung ermöglicht die Erweiterung unserer CO₂-Ampel um weitere Anzeigevarianten. Der Zusammenbau der hier aufgeführten Komponenten erfordert einfache Lötfertigkeiten.

Vorbereitung - was brauche ich?

1. Lötstation
2. Benötigte Hardware (wahlweise eine der Anzeigen aus der aufgeführten Liste)
 - a. LED-Matrix in verschiedenen Farben ([Adafruit Feather CharliePlex 15x7](#))
Universal Ausgabe: Text, Messwerte und Trend
 - b. 7-Segment-Anzeige in verschiedenen Farben ([Adafruit 7-Segment FeatherWings](#))
Gut lesbare LED-Ziffern zur Messwertausgabe
 - c. Neopixel-Matrix ([Adafruit NeopixelWing 4x8](#))
Weithin sichtbare Signalampel (grün, gelb, rot)
 - d. NeopixelRing ([Adafruit Neopixel 24](#)) und Grove-Kabel ([Male-Jumper to Grove](#))
Messwertanzeige (Gauge) und Ampel gleichzeitig. Auch Ringe mit weniger Pixel sind möglich, dann muss das Ardublock-Programm angepasst werden.

Die Hardware ist in verschiedenen Maker-Shops verfügbar (Kosten für eine Anzeige ca. 20 €).

Zusammenbau

1. Hardwarekomponenten vorbereiten

Die Anzeigen werden mit den benötigten Komponenten geliefert, erfordern aber einen Zusammenbau gemäß Anleitung auf der jeweiligen Webseite (Links dazu siehe oben).

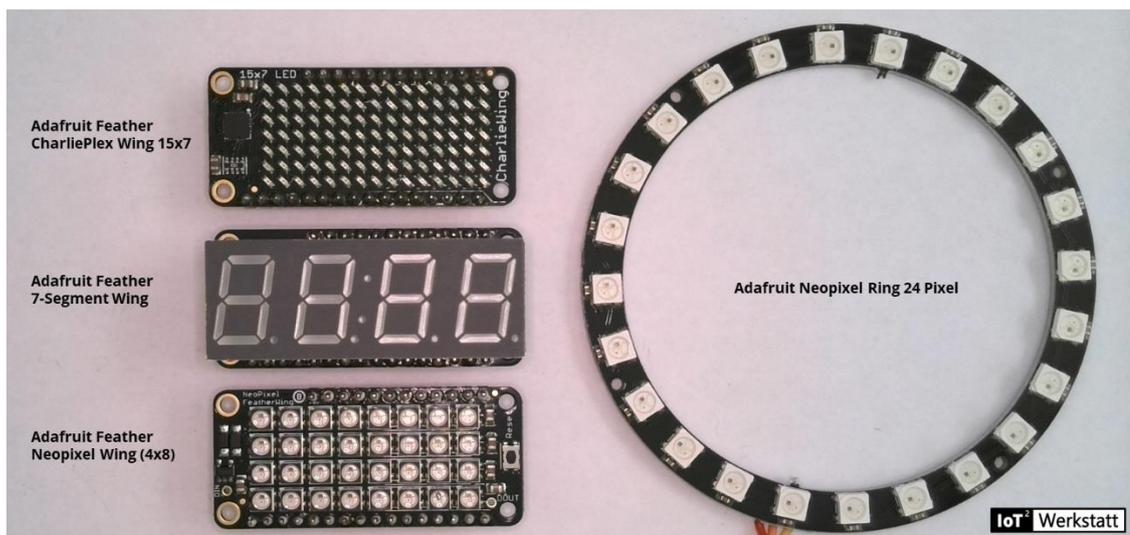


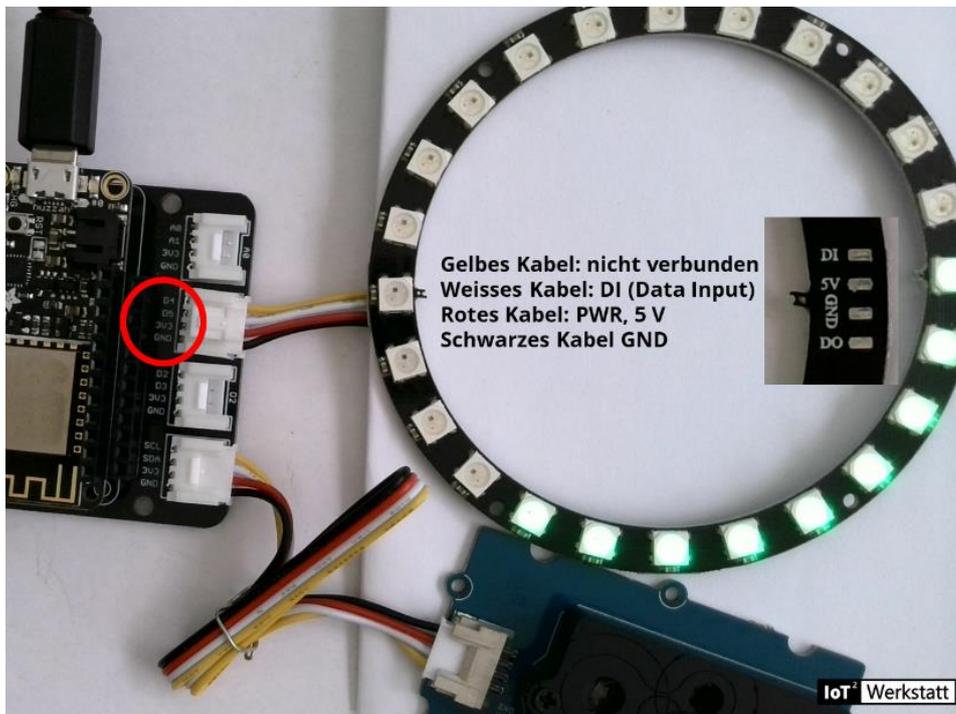
Abb. 1: Mögliche Anzeigeoptionen



Der **NeoPixel-Wing (4x8 Matrix)** bietet beim Zusammenbau verschiedene Optionen. Zur Anpassung an unser Setup müssen wir mit dem Messer vorsichtig eine Verbindung durchtrennen und dafür an anderer Stelle eine Lötverbindung herstellen.

Abb 2: Rückseite NeoPixel-FeatherWing mit Anpassung

Der **NeoPixel-Ring** wird über GPIO15 angesprochen. Auf dem Grove-Shield ist das die Buchse mit der Bezeichnung D5. Je nach Ring-Variante können die Kontakte am Ring etwas anders aussehen. Hinweis: Wir verwenden das Grove Kabel, manche Grove-Stecker haben eine kleine Nase zur Verriegelung. Bei Steckvorgang diese mit dem Finger zusammendrücken.



Der

Abb 4: NeoPixel-Ring mit 24 Pixeln

GPIO15 für den NeoPixel-Ring kann auch ohne Grove-Kabel über die Pins am Feather-Sockel abgegriffen werden.

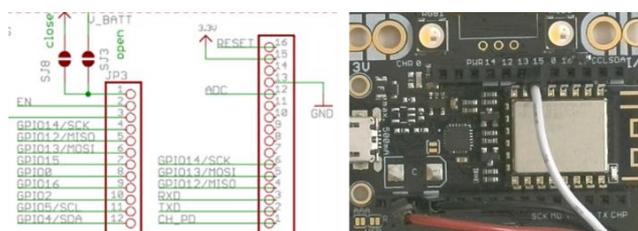


Abb 5: Belegung des Feather-Sockels am Octopus

NeoPixel-Ring: Verbindung am Octopus (über Feather-Sockel)

- Weißes Kabel: GPIO15
- Rotes Kabel: 3 V
- Schwarzes Kabel: GND

2. Programmentwicklung Messwertanzeige vor Ort

Jede Anzeigeoption wird durch die IoT²-Werkstatt in Form eines Programmblöckchens unterstützt. Die CharliePlex Matrix ist Standardausgabe und befindet sich links in der Werkzeugkiste „Anzeigen/Aktoren“. Die anderen Komponenten sind unter „Externe Interfaces“ eingeordnet.

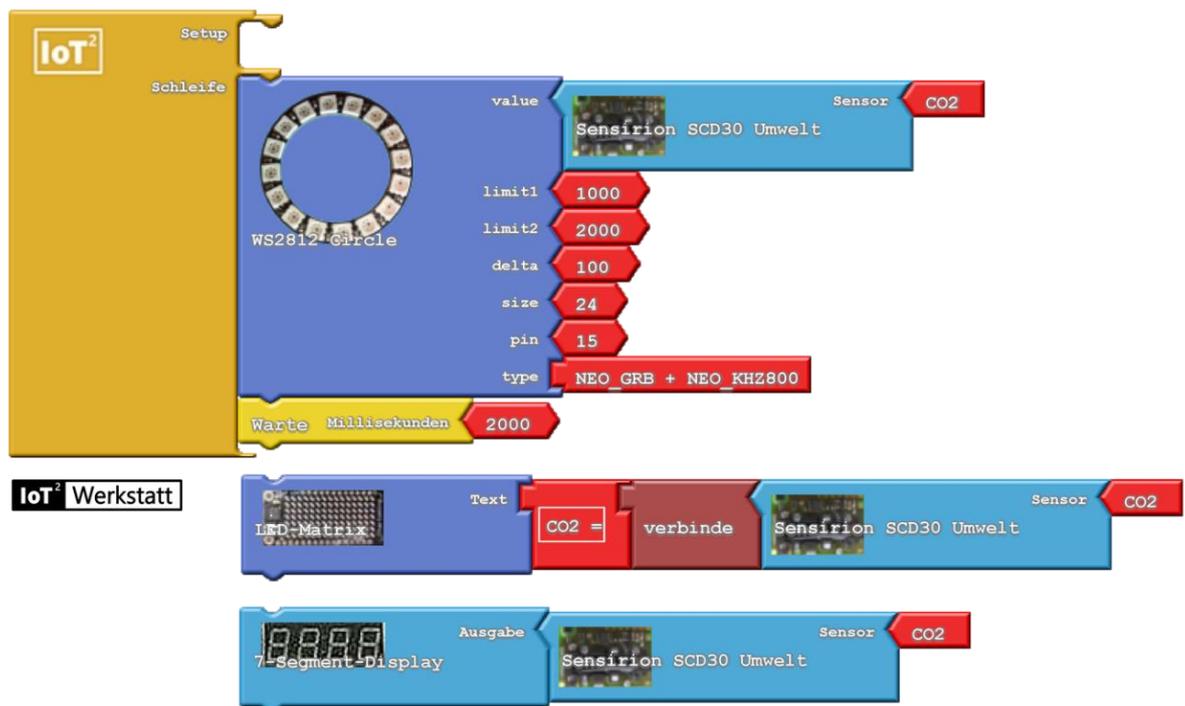
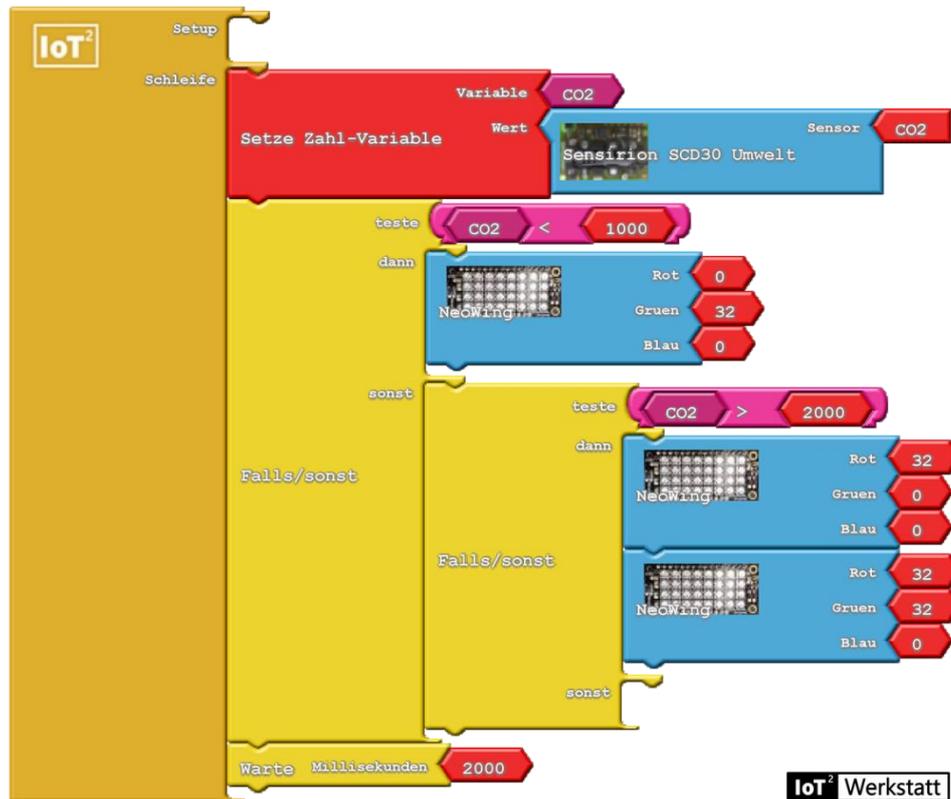


Abb 6: Je nach Anzeigeoption kommt ein korrespondierender Block zum Einsatz (CO2_Ampel_HuzzahLCD_Quick2.abp)

Der **Neopixel-Ring** wird als Gauge-Anzeige betrieben, d.h. je nach CO₂-Konzentration ist eine unterschiedliche Anzahl an Pixeln aktiv. Pro 100 ppm leuchtet ein grünes Pixel (delta). Ab 1000 ppm (limit1) leuchten die Pixel gelb. Ab 2000 ppm (limit2) färben sich die Pixel rot. Der Parameter delta legt fest, wieviel ppm ein Pixel repräsentiert (hier 100 ppm/Pixel, d.h. 4 Pixel = 400 ppm). Size gibt die Anzahl der Pixel auf dem Ring an (max. 24, sonst ist der Strombedarf ggf. zu hoch).

Die **Neopixel-Matrix** wird als Ampel betrieben, d.h. je nach Messwert leuchtet die Matrix grün, gelb, oder rot. Dazu entwerfen wir unseren ersten echten Algorithmus. Zum Einsatz kommt eine geschachtelte Fallunterscheidung. Ist der Messwert kleiner als 1000 ppm, stellen wir die Matrix auf Grün. Sonst schauen wir, ob der Messwert größer 2000 ppm ist, dann stellen wir die Matrix auf Rot. Gelb (Rot und Grün gleichzeitig) signalisiert einen Wert dazwischen.



IoT² Werkstatt

Abb. 7: Ampel-Algorithmus steuert die Neopixel-Matrix (CO2_Ampel_HuzzahLCD_Quick3.abp)

Wer noch etwas mehr Algorithmus vertragen kann, der schaue sich das folgende Programm etwas näher an. Hier nutzen wir Datenfelder und die Matrix als Trendanzeige, d.h. jede Spalte der Matrix steht für einen Zeitpunkt. Die Anzahl der aktiven Pixel in der Spalte steht für die Höhe des Messwertes.

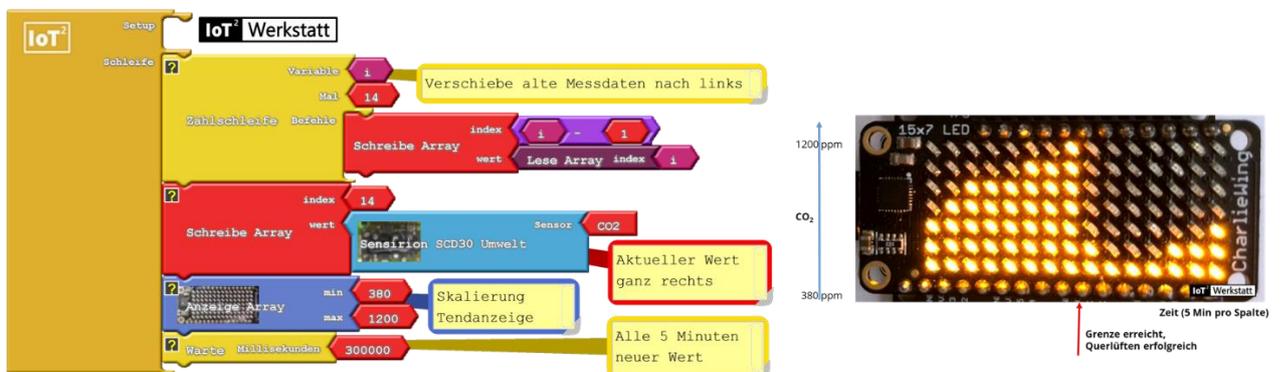


Abb. 8: Ein Datenfeld speichert die 15 letzten Messwerte. Die Trendanzeige visualisiert den Verlauf dieser letzten 75 Minuten auf der Matrix.

Beispielprogramm

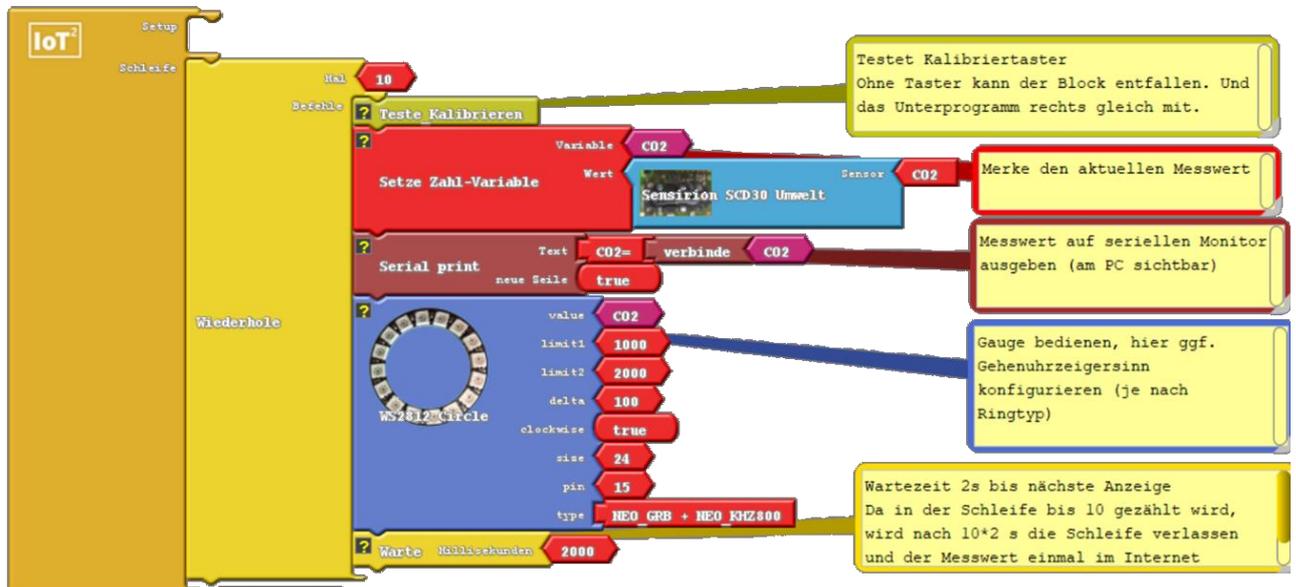


Figure 1: Auch andere Neopixel-Ringe lassen sich nutzen (max. 24 Pixel, delta gibt an, welche physikalische Differenz einem Pixel entspricht. Delta 100 = 100 ppm/Pixel. Alles kleiner limit1 ist grün, alles größer limit2 wird rot, dazwischen gelb).

Jeder Einsatz von IT muss in Zeiten der Klimakrise natürlich auch die Ressourceneffizienz betrachten. Unsere CO₂-Ampel soll das Infektionsrisiko reduzieren, soll die Konzentrationsfähigkeit der Personen im Raum sichern, soll dabei energieeffizientes Lüften ermöglichen, benötigt aber selbst auch Energie zum Betrieb. Im Folgenden wollen wir den Energiebedarf der einzelnen Komponenten kurz abschätzen:

Komponente	Strombedarf (bei 3.3 V)
ESP8266 (Huzzah, Octopus)	ca. 100 mA
Sensor SCD 30	ca. 20 mA
Anzeige Neopixel-Matrix	Je nach Helligkeit, ca. 50 mA (max. 90 mA, alle Pixel weiss)
Anzeige Neopixel-Ring	Gauge, je nach Messwert 15-40 mA (max. 90 mA, alle weiss)

Damit ergibt sich ein mittlerer Leistungsbedarf von ca. 660 mW (3.3 V * 200 mA). Je nach Wirkungsgrad des verwendeten Netzteils (70 %) sind das rund 9 kWh (1 W * 24 h * 365 Tage) im Jahr. Bei aktuellem Strommix entspricht dies einer CO₂-Emission von 3.6 kg CO₂ (bei [400 gCO₂ /kWh in Deutschland](#), Stand 2019) oder Energiekosten von rund 3 € im Jahr.

Um einen Netto-Gewinn zu erzielen und damit auch etwas für den Klimaschutz zu tun, müssen wir die Emissionen der Raumheizung kennen. Annahme: Ölheizung ([318 gCO₂/kWh](#)), Raumgröße 10*5*2.5 = 125 m³. Um die gesamte Luft im Raum um 20 °C zu erwärmen benötigen wir 0.8 kWh (spez. Wärmekapazität Luft, Dichte). Damit erzeugt ein kompletter Lüftungsvorgang (mit Aufwärmen der frischen Luft) CO₂-Emissionen von 318*0.8 = 0.265 kg CO₂.

Wir müssen also ca. 14 (3.6/0.265) Lüftungsvorgänge im Jahr einsparen um unsere Ampel wirklich CO₂-neutral zu betreiben.

