

~

@ >L%GJK;@MF?~

;G*%9 ` cX_~: Th%À~: XW\XahaZfTa_X\ghaZ~

N~) & ~(. '* (**~

~

~

~

~

~

~

Hochschule
für Technik
Stuttgart

@bV[fV[h_X~seLXV[a\`KghgZTeg~

KV[X_`aZfges*,~

/() / ,KghgZTeg~

Vb*fXafbe8 [V%fqgZTegWX~

~

Aa [T_g

Material Bausatz.....	3
„Die Basics“ %J: ThfTgn~ibezX_rgXghaW~UXeX\gmh` ~ = \afgV^Xa#.....	4
„Expert Mode“ – (Löten & Software Spielereien)	8
Schaltplan der CO ₂ -Ampel	8
Vorbereitung der Bauteile / Löten	9
Bezugsliste der Bauteile	11
Treiber & Software Installation (MAC Windows)	12
Ampel Varianten / (LoRaWAN) Erweiterungen.....	18
Hinweise zur Kalibrierung der CO ₂ -Ampel.....	19
CO ₂ -Ampel Quickmanual.....	20

Das Projekt entstand an der HFT-Stuttgart unter Zusammenarbeit von:

Eric Duminil
Tobias Gabriel Erhart
Myriam Guedey
Nathalie Heimsch
Benjamin Hueber
Robert Otto
Jonas Stave

Die aktuellste Version der Projektdokumentation lässt sich unter folgenden Link aufrufen:

<https://transfer.hft-stuttgart.de/gitlab/co2ampel>

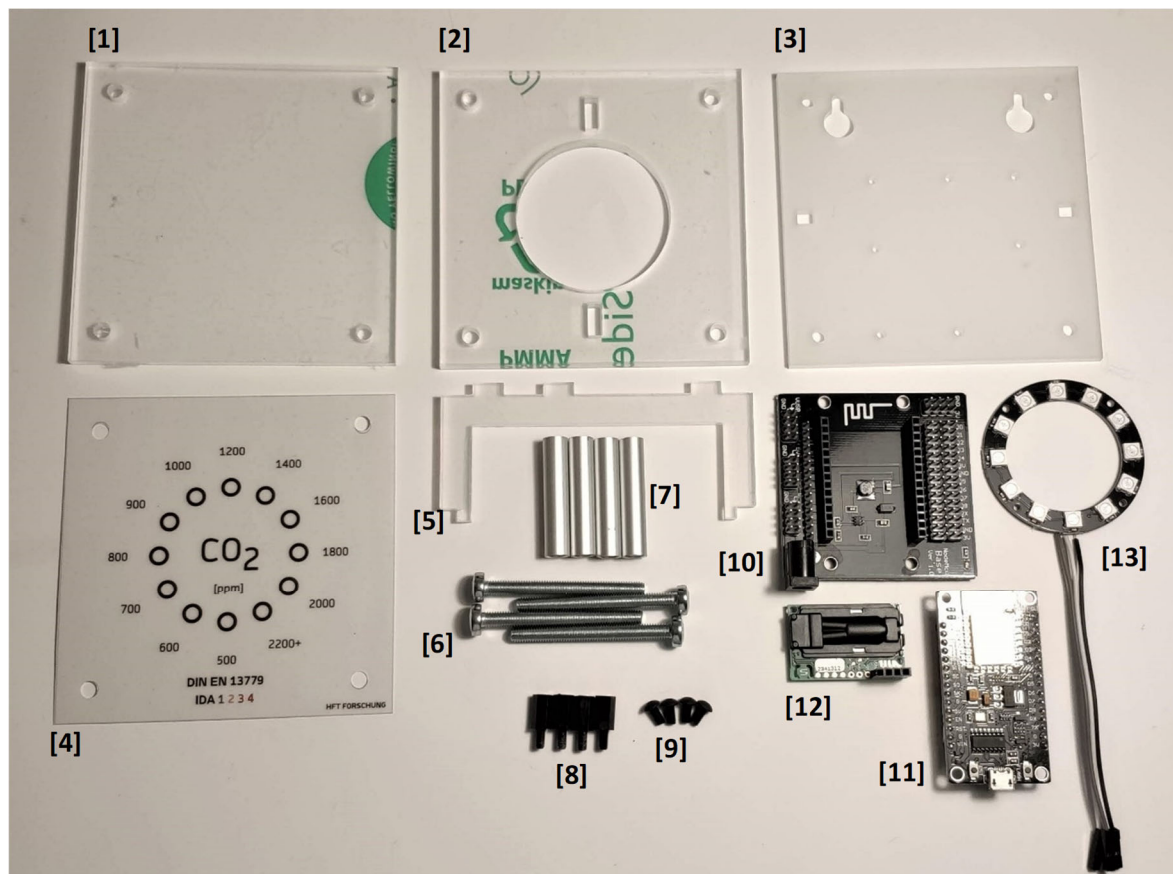
Haftungsausschluss und Nutzungsbedingungen

Die Inhalte dieses Dokuments wurden nach eigenüblicher Sorgfalt erstellt. Wir übernehmen jedoch keine Gewähr für die Richtigkeit, Vollständigkeit und Aktualität der bereitgestellten Inhalte. Es wird insbesondere keine Gewähr dafür übernommen, dass die hier beschriebene CO₂-Ampel die dargestellten Funktionen erfüllt und sich für die dargestellte bzw. beabsichtigte Verwendung eignet. Die Nutzung der Inhalte erfolgt auf eigene Gefahr des Nutzers. Mit der Zusendung dieses Dokumentes und dessen Verwendung kommt keinerlei Vertragsverhältnis oder sonstige Rechtsbeziehung zustande. Die Inhalte dieses Dokuments werden unter der GNU General Public License, Version 3 (<https://www.gnu.org/licenses/gpl-3.0.de.html>) lizenziert. Als Quelle ist die Hochschule für Technik Stuttgart anzugeben.

Der Bausatz besteht aus nachfolgenden Komponenten und kann ohne besonderes Vorwissen einfach zusammengesteckt und geschraubt werden, die hierfür notwendigen Schritte werden im Detail in der Bauanleitung aufgeführt. Wenn die Komponenten selbst zusammengestellt werden, müssen die passenden Verbindungen noch an LED-Ring und CO₂-Sensor angelötet werden. Darauf wird am Ende der Bauanleitung eingegangen.

Viel Spaß beim Bauen und Ausprobieren :)~

E TgeT ~: ThfTgm~



▪ **Gehäuse** bestehend aus:

- Frontplatte [1]
- LED-Halter (Zwischenplatte) [2]
- Rückplatte [3]
- Anzeigen Papier [4]
- Abstandshalter (optional für LoraWan) [5]
- 4 x Gewindeschrauben M5x50mm [6]
- 4 x Alu Distanzhülsen 8x1.0mm | 37 mm [7]
- 4 x Abstandsbolzen M3x10mm +6 [8]
- 4 x Platinen Schrauben M3x8mm [9]

▪ **Sensor Komponenten** bestehend aus:

- Breakout Board (NodeMCU V3 Base Plate) [10]
- ESP8266 (NodeMCU V3) [11]
- CO₂-Sensor (Sensirion SCD 30, opt: Sensair S8) [12]
- LED Ring (NeoPixel 12 x WS2812 RGB LED | 50 mm) [13]

p< \X~: Tf\vfé%": ThfTgm~ibeZX_rgXghaW\UXeX\gmh' ~ = \afgXV^Xa#~

~

~ : Xarg\ZgXfO Xe^mXhZ~

- Kreuzschlitz-Schraubenzieher
- Torx-Schraubenzieher
- Lötzinn (optional)
- Lötkolben (optional)

~



) ~ H_Xk\Z_Tf~ibeUXeX\gXa~

Von allen Gehäuseteilen auf beiden Seiten die Schutzfolie ganzflächig abziehen. ~



*T~ 9UfgTaWfUb_mXa~'O TaWNTe\TagX#~

~ Die vier Abstandsbolzen auf die Rückplatte schrauben. Für diesen Schritt sind zwei Positionierungen möglich (**2a** | **2b**):

Soll der Stromanschluss des CO₂-Sensors von **unten** erfolgen (z.B. Sensor wird an eine Wand gehängt) dann werden die Abstandsbolzen wie abgebildet **linksbündig** eingeschraubt. ~



*U~ 9UfgTaWfUb_mXa~'L \fV[NTe\TagX#~

Soll der Stromanschluss des CO₂-Sensors von **oben** erfolgen (z.B. Sensor steht auf einem Tisch) dann die Abstandsbolzen wie abgebildet **rechtsbündig** einschrauben.

Nachfolgend ist der Aufbau der Tisch Variante (2b) zu sehen. Der Aufbau der Wand Variante erfolgt nach dem gleichen Prinzip. ~

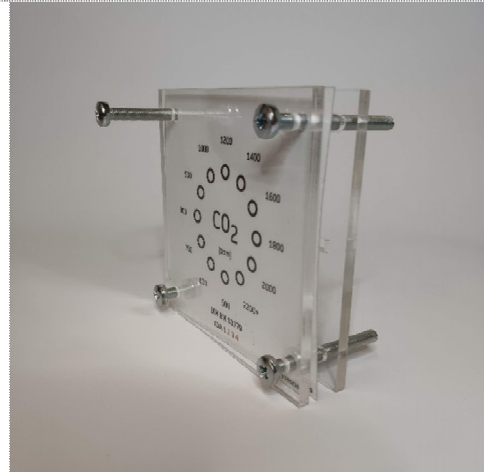


+ ~ ? X [qhfX~>ebaggX\~

Das Gehäuse Frontteil setzt sich in der Reihenfolge aus Frontplatte, Anzeigenpapier und LED-Halter (Zwischenplatte) zusammen. Und wird mit den M5 Gewindeschrauben zusammengesteckt.

Bei der Zwischenplatte ist darauf zu achten, dass sich die rechteckigen Aussparungen horizontal zur Beschriftung der Anzeige befinden.

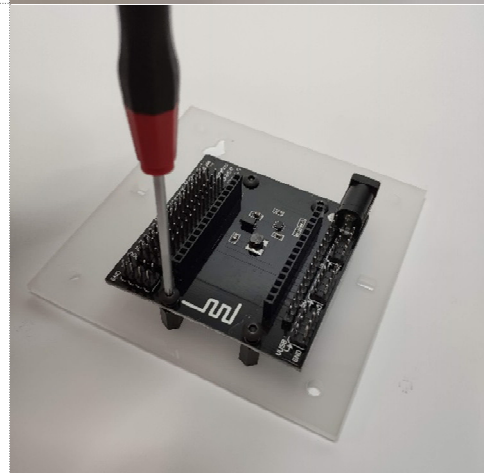
~



, ~ : TfX-H TgX~

Die Base Plate wird mit den vier Platinen Schrauben auf die Abstandsbolzen geschraubt.

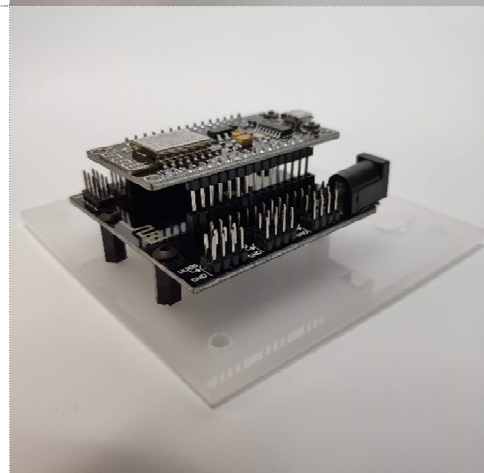
Hier ist auf die Ausrichtung des Stromsteckers zu achten. Für die Wand Variante (2a) zeigt dieser nach unten und ist bündig mit dem Ende der Rückplatte. Für die Tisch Variante (2b) zeigt dieser wie abgebildet nach oben. ~



- ~ F bWXE ; M~

Der ESP8266 / NodeMCU Chip wird nun auf die entsprechenden Pins der Base Plate gesteckt.

Die Ausrichtung kann man an der seitlichen Beschriftung des NodeMCUs und der Base Plate überprüfen, bzw. muss der USB-Anschluss des NodeMCUs auch auf der Seite des Stromsteckers der Base Plate liegen. ~



. ~ KXaf\eba~; G*%KXafbe~

Der Sensirion Sensor wird nun ebenfalls neben dem NodeMCU auf die Base Plate gesteckt.

Dafür werden folgende PINs der Base Plate genutzt:
3V | GND | D5 | D6

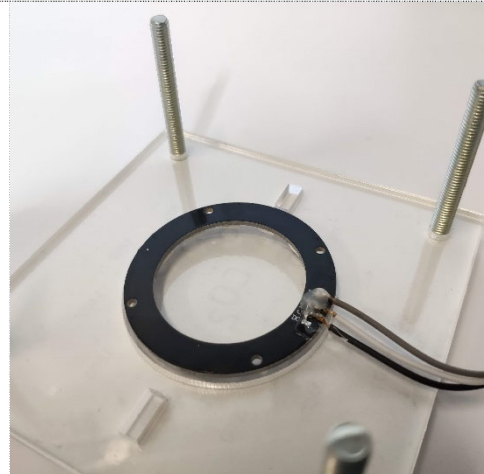
Bei richtiger Belegung liegen die obere Kante des Sensirion Sensors und des NodeMCUs auf gleicher Höhe. ~



/ ~ D = < J \aZ~

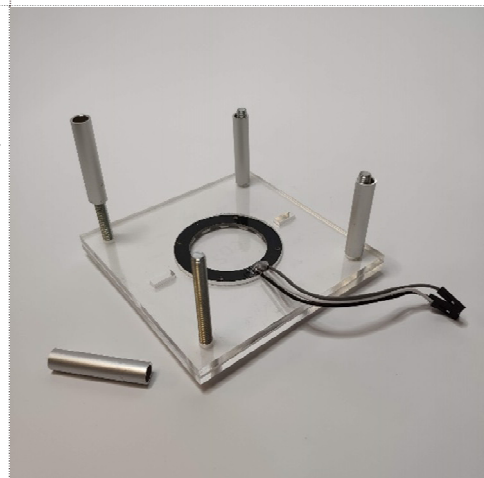
Das Loch für den LED-Ring ist konisch gelasert, d.h. der Durchmesser ist auf der Vorderseite etwas kleiner als auf der Rückseite. Der LED Ring wird von der Rückseite aus in die Platte gesetzt. Durch den Konus wird der Ring festgehalten. Ein evtl. Grad am äußeren Ring der LED, muss vorher noch abgeschnitten werden.

Hierbei ist darauf zu achten, dass die Anschlüsse des LED Rings von der Schrift aus gesehen nach unten Zeigen (6:00 Uhr) und die LEDs anhand der gedruckten Kreise auf dem Anzeigepapier ausgerichtet werden. ~



0 ~ < \fgTam[s_fXa~

Die vier Alu Distanzhülsen werden über die Gewindeschrauben des Gehäuse Frontteils gestülpt. ~

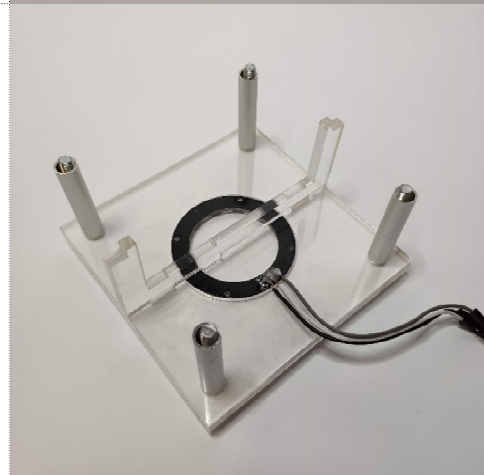


1 ~ 9 UfgTaWf [T_gXe"bcg\baT_~

Der Abstandshalter aus Plexiglas wird in die Aussparungen des Gehäuse Frontteils gesteckt.

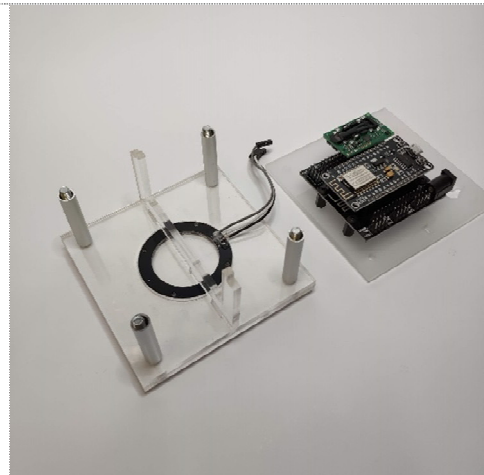
Hinweis:

Es hat sich gezeigt, dass der Abstandshalter nur optional benötigt wird (z.B. für eine Lora-Antenne) und der LED-Ring auch ohne den Halter fest im Gehäuse gehalten wird. ~



) (~ RhfT` ` XaUe\qZXa~

Nun werden das Gehäuse Frontteil mit der LED und die Gehäuse Rückplatte mit der montierten Sensor Einheit wie auf der Abbildung zu erkennen nebeneinandergelegt. ~



)) ~ 9 afV[hffD= < ~

Der LED Ring muss nun noch auf den zugehörigen PINs der Base Plate eingesteckt werden.

Dafür werden folgende PINs der Base Plate genutzt:

GND | 3V | D1

| | |

GND | 5V | D1

Dies ist die entsprechende Belegung des LED Rings. ~



) * ~ RhfT` ` XafXgmXa~

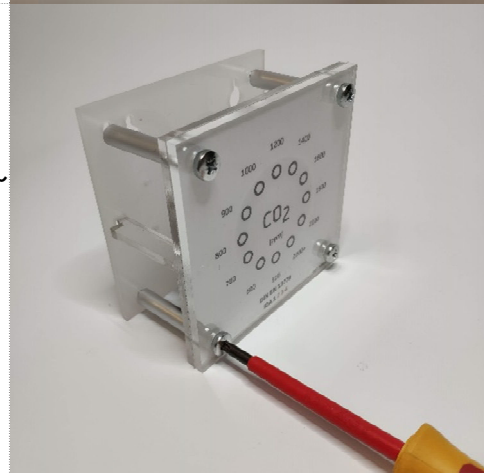
Nun wird die Rückplatte des Gehäuses auf den Frontteil des Gehäuses gestülpt.

Dabei ist darauf zu achten, das Kabel der LED nicht in zwischen dem Abstandshalter einzuklemmen und dass die Enden der Gewindeschrauben anhand der Gewindebohrungen der Rückplatte ausgerichtet werden, sowie dass, der Abstandshalter in die Aussparungen der Rückplatte einrastet. ~



) + ~ ? X [qhfX-ixefV[eThUXa~

Nun kann das gesamte Gehäuse vorsichtig aufgerichtet werden und die vier Gewindeschrauben werden am besten kreuzweise handfest angezogen. ~

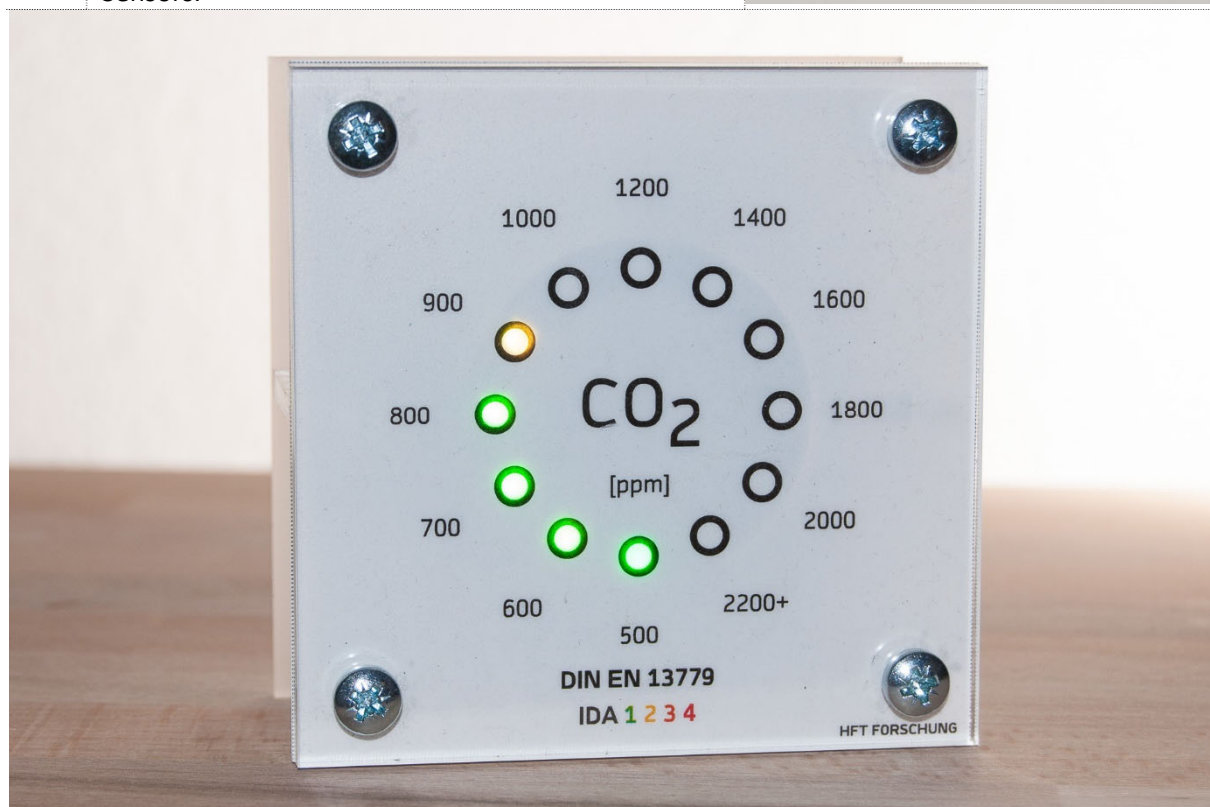
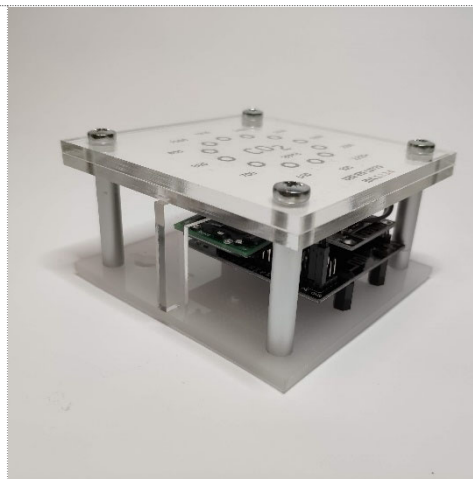


) - ~ >XegZ~i~

Damit ist der CO₂-Sensor vollständig zusammengebaut und kann durch das beiliegende Netzteil direkt angeschlossen werden.

Die Software ist auf dem NodeMCU bereits vorinstalliert und die Messung beginnt direkt nach einstecken des Netzsteckers.

Das nachfolgende Quickmanual gibt noch Auskunft zum Anzeigeverhalten und der Kalibrierung des Sensors. ~

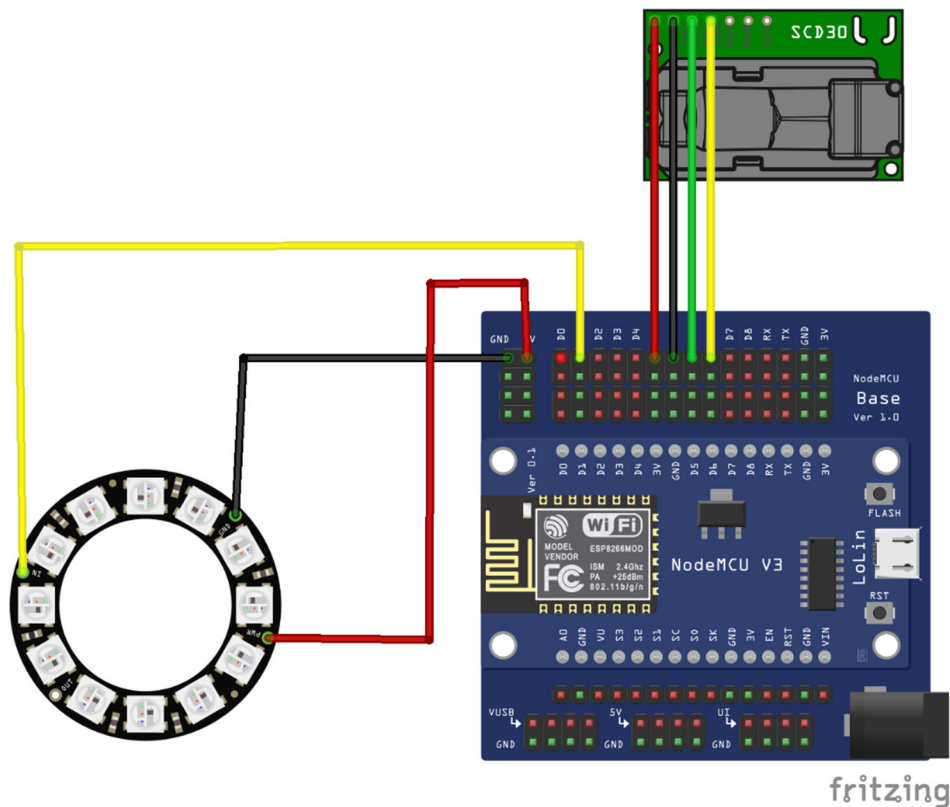


p=kcXegE bWXéÕ~DrgXaÀ~KbYgj TexKc\X_XeX\Xa#~

~

KV[T_gc_Ta~WXe~; G*%9 ` cX_~

Für die CO₂-Ampel ergibt sich folgende Board-Belegung.



Die **Pin-Belegung** ist auch nochmals folgenden beiden Tabellen zu entnehmen.

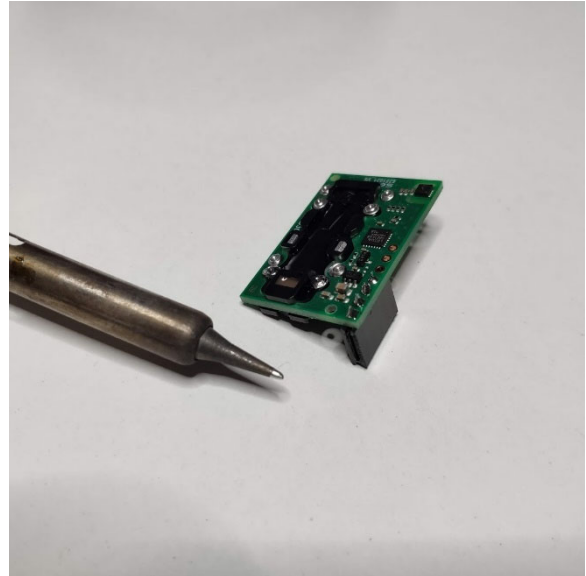
Sensirion SCD30	Sensair S8	NodeMCU Breakout Board
VIN		3V
GND	G0	GND
TX		D5
RX		D6
	G+	5V
	UART_RX	D8
	UART_TX	D7

RGB WS2812 LED	NodeMCU Breakout Board
VIN	3V
GND	GND
DI	D1

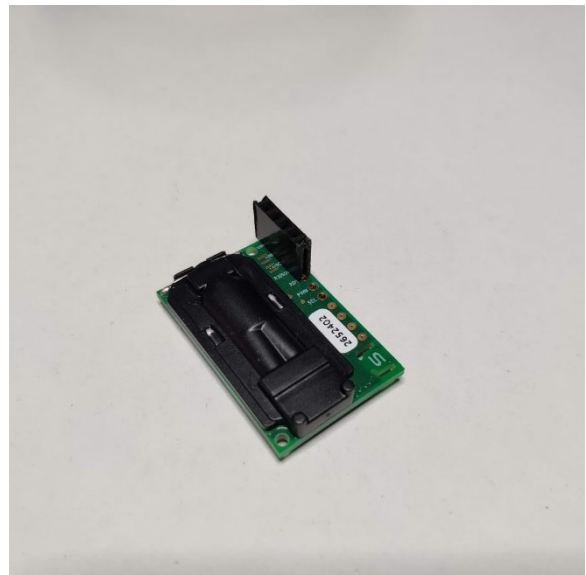
NbeUXe\ghaZ~WXe~: Thg\X~DrqXa~

Der LED RGB Ring WS2812, sowie der Sensirion SCD30 verfügen in der Regel nur über die Kontakte ohne eine Pinleiste oder entsprechende Kabel. Daher müssen diese nach obigem Belegungsschema angelötet werden.

Für die den LED Ring, können die Kabelenden eines (Buchsen-) Jumperkabels direkt an die LED Platine gelötet werden. Im Falle des Sensirion Sensors verwendet man am besten eine Buchsenleiste mit 4 Pins zum Aufstecken auf den NodeMCU Breakout Board.



Beim Löten sollte darauf geachtet werden, dass alle Kontakte fettfrei sind. Hier kann Spiritus oder Fensterreiniger sowie ein Putzlappen/Küchentuch verwendet werden um die Flächen zu reinigen. Um die Kabel bzw. die Pinleiste zu befestigen kann eine dritte Hand behilflich sein. So können die Kabel bzw. die Pinnleiste fixiert werden. Beim Löten sollte die Lötspitzen spitze nicht direkt in Kontakt mit dem Zinn kommen. Am besten ist es, einen Streifen Zinn an den PIN anzulegen und den PIN an einer anderen Position gleichzeitig zu erwärmen. So bleibt die Lötspitze sauber und man hat länger Spaß am Löten.



: XmhZf \fgX-WXe~: ThgX\X~

Die einzelnen Komponenten, lassen sich derzeit leider nicht gesammelt über einen Händler beziehen. Wir haben daher in der Liste verschiedene mögliche Bezugsquellen der einzelnen Komponenten aufgelistet, die einen Versand aus Deutschland anbieten. Etwas günstiger können unter Umständen auch die üblichen Bezugsquellen (Aliexpress & Co.) aus dem Ausland sein. Einzelne Komponenten sind derzeit auch immer mal wieder ausverkauft, daher lohnt sich auch ein direktes Nachfragen bei den Händlern nach Verfügbarkeit.

Die gelisteten Preise beruhen auf einem Mengenrabatt bei der Abnahme von 25 Einheiten (Kleinteile wie Schrauben und Kabel auch entsprechend mehr).

Komponente / Bauteilbezeichnung	Material Kosten	Bezugsquelle Alternative 1	Bezugsquelle Alternative 2
Sensor-Bauteile	55.81 €		
NodeMCU Lua Lolin V3 Module ESP8266	4.00 €	https://www.az-delivery.de/products/copy-of-nodemcu-lua-amica-v2-modul-mit-esp8266-12e?variant=36247507410	https://www.makershop.de/plattformen/nodemcu/nodemcu-esp8266-dev-kit/
NodeMCU Expansion Base Board Erweiterung DevKit Development Shield V3.1	3.89 €	https://www.makershop.de/plattformen/nodemcu/nodemcu-base/	https://www.mechatronics-pro.de/de/www.mechatronics-pro.de/de/nodemcu-v3-erweiterung-devkit-development-expansion-base-board-shield-iot_292.html
CO2-Sensor Sensirion AG SCD30	39.89 €	https://www.digikev.de/product-detail/de/sensirion-ag/SCD30/1649-1098-ND/8445334	https://www.mouser.de/ProductDetail/Sensirion/SCD30?qs=rrS6PyfT74fdywu4FxpYIQ%3D%3D
LED Ring 5V RGB WS2812B 12-Bit 50mm	2.96 €	https://www.az-delivery.de/en/products/kopie-von-rgb-led-ring-ws2812-mit-12-rgb-leds-5v-fuer-arduino?variant=19562617274464	
Kabel 10cm – Jumper Wire weiblich-weiblich	0.11 €	https://www.az-delivery.de/products/40-stk-jumper-wire?pos=27&sid=cefb83077&ss=r	https://www.mechatronics-pro.de/de/40-pin-dupont-jumper-kabel-buchse-stecker-10/20-cm-m-f-f-m-m-2-54-mm-arduino_296.html
Buchsenleisten weiblich-männlich	0.07 €	https://www.az-delivery.de/products/set-buchsenleisten-extra-lang?variant=19860818755680	https://www.makershop.de/zubehoer/kl-einteile/buchsenleiste-weiblich/
Micro-USB Netzteil	4.90 €	https://www.conrad.de/de/p/powery-ladegeraet-netzteil-mit-micro-usb-1a-fuer-lg-accolade-100-250v-805039141.html	https://www.amazon.de/dp/B07VNSTFY5/ref=cm_sw_em_r_mt_dp_1-41FbABCCDRA?_encoding=UTF8&pnc=1
Gehäuse-Teile	4.16 €		
Plexiglas (Materialkosten ohne lasern - für Laser-Vorlage siehe Gitlab – Case Drawings)	2.80 €	Für Schulen und Kindertageseinrichtung besteht die Möglichkeit einzelne fertig gelaserte Gehäuse-Bausätze über die HFT zu einem Unkostenbeitrag von 7€ pro Stk. zzgl. Versand zu beziehen. Anfragen bitte an co2sensor@hft-stuttgart.de senden.	
Alu Rundrohr 8X1mm	0.16 €	https://www.stahl-shop24.de/alu-rundrohr-8x1mm-1000mm	https://www.ebay.de/itm/192336054487
Gewinde Linsenkopfschrauben M5x50	0.60 €	https://ebay.us/rFqS65	
Platinen Abstandshalter M3, männlich/weiblich 10 mm + 6 mm	0.26 €	https://www.amazon.de/dp/B07CG36VYC/ref=cm_sw_em_r_mt_dp_v41FbKVSTGA0	
Platinen Schrauben Domkopf Innensechskant M3 x 6 mm	0.34 €	https://ebay.us/rFqS65	https://www.amazon.de/dp/B00DY2BPIW/ref=cm_sw_em_r_mt_dp_V41FbY1D4NIX
Gesamt	59.97 €		

L eX\UXeÀ KbYgj T eX\afgT_Tg\ba~E 9 ; ~O \aWbj f#

) &F bWXE ; M L eX\UXe\afgT_XeXa~

Das in unserem Bausatz verwendete NodeMCU V3/ESP8266 Board verwendet den CH340/CH341 USB Chipsatz. Der passende Treiber für den Anschluss am MAC/PC ist zum Herunterladen im Git-Repository verlinkt und muss installiert werden, damit der Computer beim Einstecken den NodeMCU richtig erkennt.

Link zu unserem **Git-Repository**:

<https://transfer.hft-stuttgart.de/gitlab/co2ampel>

Wurde der Treiber richtig installiert und das Board wird mit einem Micro-USB Kabel mit dem Computer verbunden, so sollte ein entsprechender Hinweis erscheinen. Unter Windows findet man im Gerätemanager den CH340/CH341 als COM-Port. Bei Mac OS und Linux wird beim Einstecken ein entsprechender Hinweis eingeblendet.

* &9 eWh\abA = \afgT_XeXa~

Zum Aufspielen der Firmware / Skripte auf den NodeMCU kann die Software Arduino IDE genutzt werden. Diese lässt sich unter folgendem Link sowohl für MAC als auch PC herunterladen:

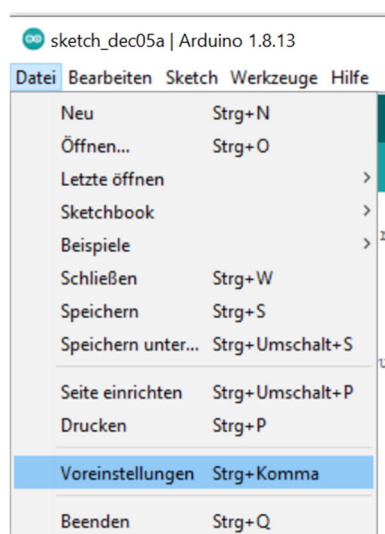
<https://www.arduino.cc/en/software>

Nach der Installation der Arduino IDE muss für die Kommunikation mit dem NodeMCU noch die Arduino IDE Software richtig eingerichtet werden. Dazu startet man Arduino IDE und nimmt die folgende Einstellung vor.

+&: bT eWiXeJ T_gXeMJD_X\agetZXA~

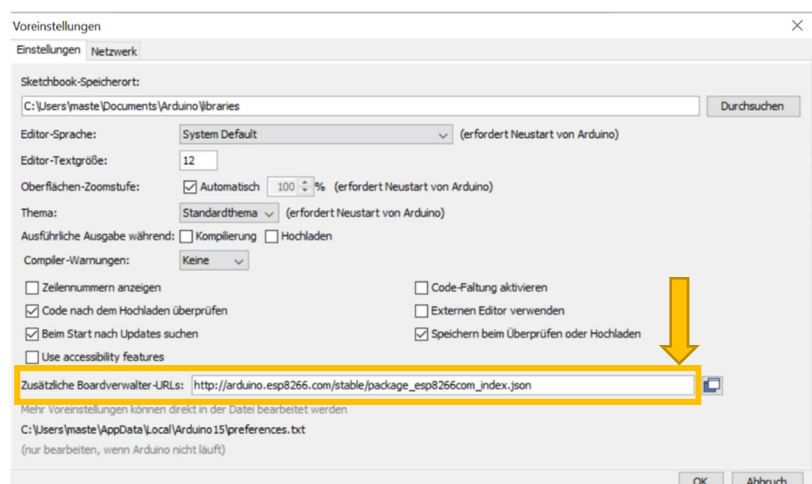
Zuerst muss in den Einstellungen eine entsprechende Boardverwalter-URL eingetragen werden, um damit automatisch die neusten Boarddefinitionen zu installieren. Dazu folgende Adresse kopieren:

Boardverwalter-URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json



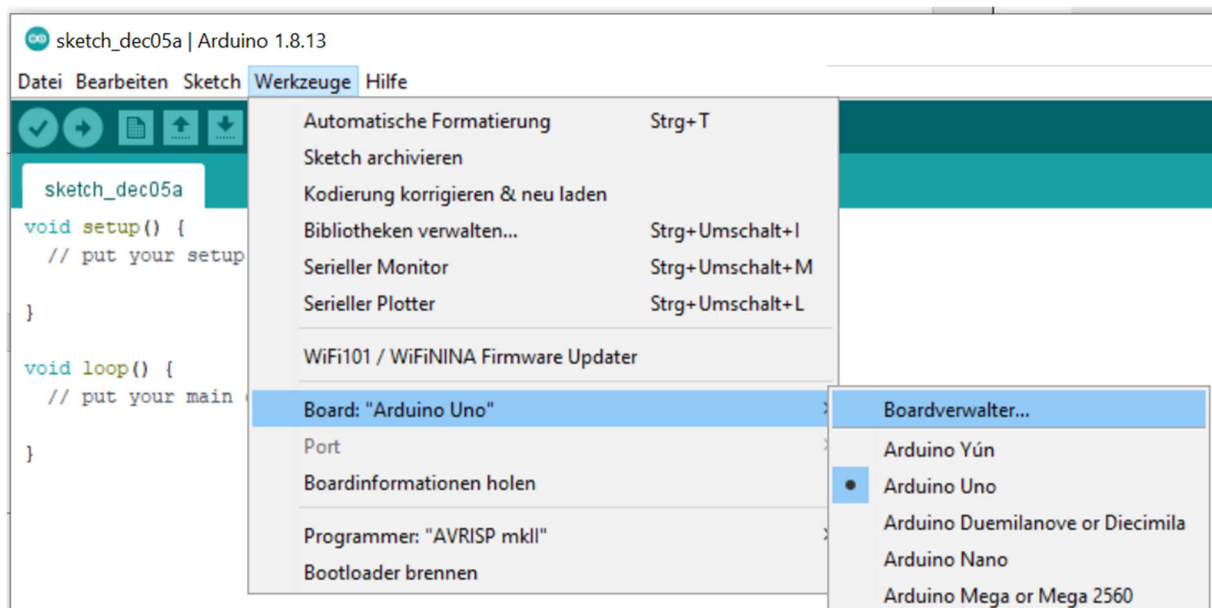
In der Arduino IDE wählt man unter dem Menüpunkt **[Datei]** den Eintrag **[Voreinstellungen]**.

In diesen Einstellungen fügt man unten unter **[Zusätzliche Boardverwalter-URLs]** die kopierte Adresse ein:

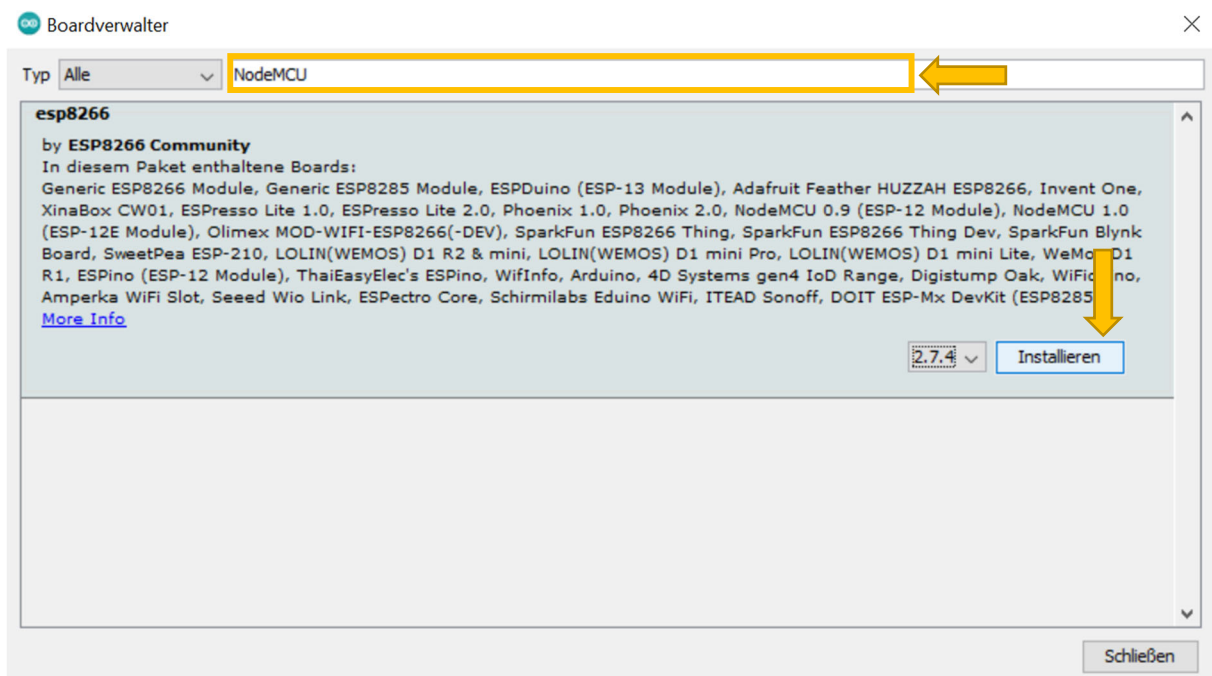


, &: bTēWXYā \g\baXa ~afgT _XeXa ~

Nun kann man die neue Boarddefinitionen in die IDE laden. Dazu wählt man unter dem Menüpunkt **[Werkzeuge]** den Eintrag **[Board]** und dann **[Boardverwalter...]**.



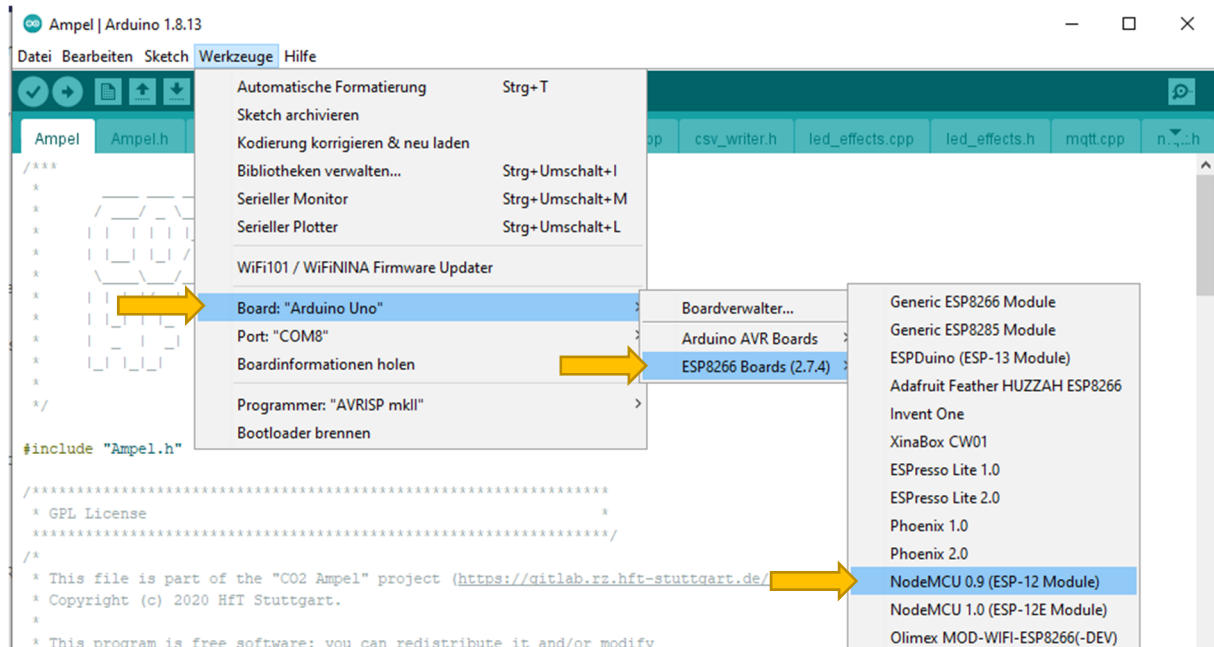
Im sich öffnenden Fenster gibt man in das Suchfeld **"NodeMCU"** ein, darauf sollte der Eintrag **"esp8266 by ESP8266 Community"** erscheinen. Hier nun auf **[Installieren]** klicken, sodass die passenden Boardinformationen der Arduino IDE hinzugefügt werden.



- &: bTewhaW~; GE %HbegThfj q[_Xa~

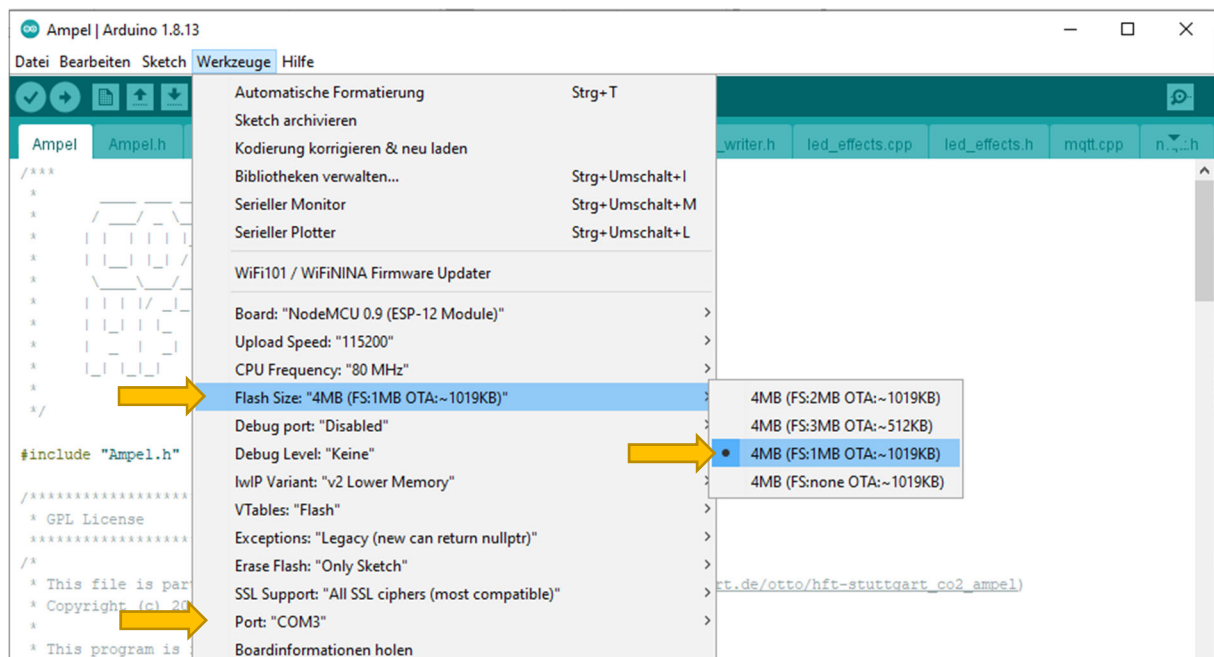
Nun muss noch das richtige Board in der Arduino IDE eingestellt werden, damit die Arduino IDE mit dem NodeMCU richtig kommunizieren kann.

Dafür klickt man unter **[Werkzeuge]** auf **[Board]** und findet dann unter **[ESP8266 Boards]** den passenden Eintrag **[NodeMCU 0.9]** und wählt diesen aus.



Anschließend muss noch der richtige COM-Port der entsprechenden USB-Schnittstelle eingestellt werden, die geschieht unter dem Menüpunkt **[Werkzeuge]** und dann **[Port]**. Wird der NodeMCU per USB nun mit dem PC verbunden, taucht hier der entsprechende Port auf, welchen man dann auswählt.

Dann sollte noch überprüft werden ob die **[Flash Size]** korrekt auf **[4MB (FS:1MB OTA)]** eingestellt ist. Damit sind alle Vorbereitungen für das Aufspielen des Skripts getroffen.



. &K^e\cgThYKXafbeY Tf [Xa À bcg\baT_ ^baWzhe\XeXa~

Nun kann im letzten Schritt das Skript auf den Sensor aufgespielt werden. Vorher können noch die gewünschten individuellen Anpassungen in der Konfiguration des Sensors gemacht werden. Dazu lädt man zunächst den Skripte-Ordner aus dem Git-Repository herunter und öffnet die Datei **ampel-firmware.ino** mit der **Arduino IDE**.

Die Konfiguration des Sensors, wie z.B. die Einstellung der WLAN-Netzwerkverbindung, kann dann im Reiter **[config.h]** vorgenommen werden.



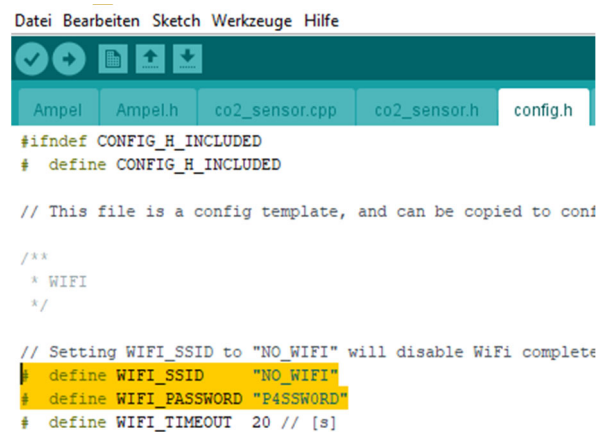
In der **config.h** können neben

```
define WIFI_SSID
define WIFI_PASSWORD
```

der Name des WLAN-Netzwerks und das dazugehörige Passwort **innerhalb der „“** eingegeben werden.

Nachdem hier Eintragungen vorgenommen wurden müssen die Änderungen noch gespeichert werden:

Tastatur: **(Strg+S)** oder **[Datei] -> [Speichern]**



Die weiteren Konfigurationsmöglichkeiten des Sensors können den Kommentaren in der **config.h** Datei entnommen werden.

Danach kann das Skript auf den Sensor hochgeladen werden, dies geschieht mittels des Upload-Buttons **[→]** oder Tastatur: **(Strg+H)**

Das Kompilieren des Codes und das anschließende Hochladen auf den Sensor nimmt etwa 1 bis 2 Minuten in Anspruch. Der Fortschritt kann in der Statusbar der Arduino IDE verfolgt werden. Verläuft der Vorgang erfolgreich, startet der Sensor im Anschluss und verbindet sich mit dem Netzwerk und beginnt darauffolgend mit der Messung.

~

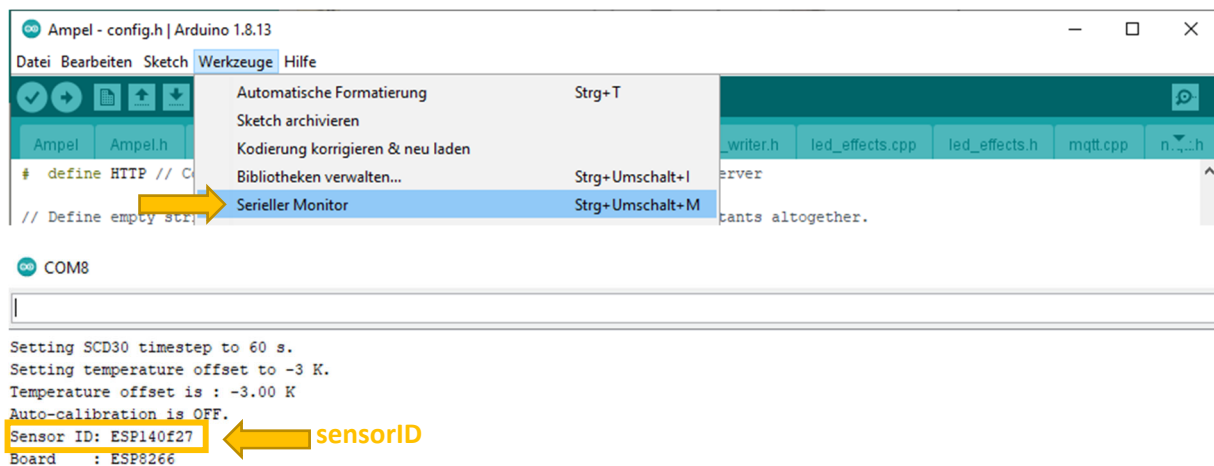
/&E XffWTgKa Õ RhZe\WhaW<bj a_bTW~

Die Messdaten des Sensors können über eine WLAN-Verbindung direkt vom Sensor abgerufen werden, dabei muss der Zugriff auf den Sensor aus demselben WLAN-Netzwerk erfolgen, mit dem sich der Sensor verbunden hat. Der Zugriff ist sowohl per Smartphone als auch Computer möglich.

Der Zugriff auf den Sensor kann dann über folgende Adresse erfolgen:

<http://sensorID.local> (z.B. <http://esp8266EX.local>)

Dabei muss „**sensorID**“ durch die jeweilige Sensor-Adresse ersetzt werden. Die Sensor ID kann direkt nach dem hochladen des Skripts oder dem Neustart des Sensors unter **[Werkzeuge]** über den **[Seriellen-Monitor]** der **Arduino IDE** ausgelesen werden (Tastatur: **Strg+Umsch+M**):



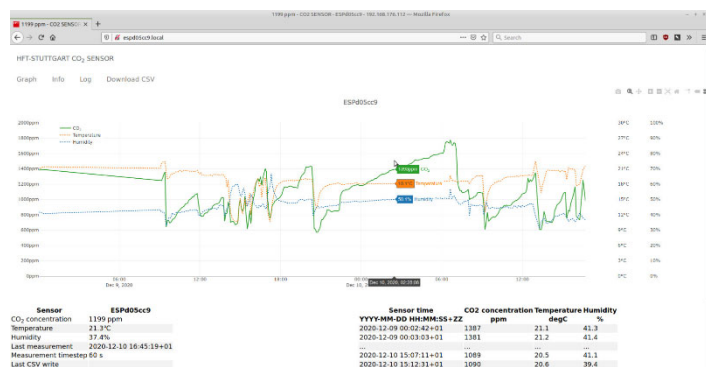
Der Netzwerkzugriff auf den NodeMCU ist standardmäßig mit einem Passwort geschützt, die Zugriffsdaten können in der **config.h** geändert werden.



Über einen Browser kann dann der Zugriff auf den Sensor erfolgen:

Über die Web-Oberfläche des Sensors lassen sich die Daten direkt visualisieren und können auch als CSV-Datei heruntergeladen werden.

Der Speicher des Sensors ermöglicht eine Aufzeichnung der Messdaten bei einem Messintervall von 5 Minuten für ca. 80 Tage. Über die Web-Oberfläche lässt sich der Speicher auch wieder zurücksetzen.



Die Daten des Sensors können außerdem über das **MQTT-Protokoll** an einen Server gesendet werden, die Konfiguration hierfür muss ebenfalls in der **config.h** Datei vorgenommen werden.

0&< XUhZZ\az~CbaYzheTgba~sUXeKXeX_X%KV[a\gpfX_X~

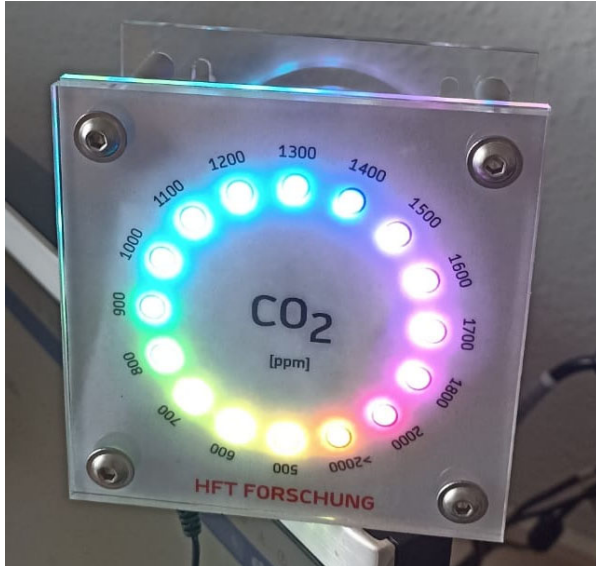
Über die Arduino IDE kann die CO₂-Ampel auch nach dem aufspielen der Software über die Serielle-Schnittstelle konfiguriert werden. Hierfür schließt man die Ampel per USB-Kabel an den PC an und stellt im Arduino IDE wie beschrieben eine Verbindung zur Ampel her. Im Anschluss öffnet man wieder unter **[Werkzeuge]** den **[Seriellen-Monitor]** der **Arduino IDE** (Tastatur: *Strg+Umsch+M*). Folgende Befehle können dann über einfache Tastatureingabe an die Ampel gesendet werden:

- `auto_calibrate 0/1` (Disables/enables autocalibration).
- `calibrate` (Starts calibration process).
- `calibrate 600` (Starts calibration process, to given ppm).
- `calibrate! 600` (Calibrates right now, to given ppm).
- `co2 1500` (Sets co2 level, for debugging purposes).
- `color 0xFF0015` (Shows color, specified as RGB, for debugging).
- `csv 60` (Sets CSV writing interval, in s).
- `format_filesystem` (Deletes the whole filesystem).
- `free` (Displays available heap space).
- `local_ip` (Displays local IP and current SSID).
- `lora 300` (Sets LoRaWAN sending interval, in s).
- `mqtt 60` (Sets MQTT sending interval, in s).
- `night_mode` (Toggles night mode on/off).
- `Reset` (Restarts the ESP).
- `reset_scd` (Resets SCD30).
- `send_local_ip` (Sends local IP and SSID via MQTT. Can be useful to find sensor).
- `set_time 1618829570` (Sets time to the given UNIX time).
- `show_csv` (Displays the complete CSV file on Serial).
- `timer 30` (Sets measurement interval, in s).
- `wifi_scan` (Scans available WiFi networks).

~

~

9 ` cX_NTe\TaQXa~JDbJTO 9 F #-ej X\XehaZXa~



Ampel Variante mit 16er LED-Ring



LoRaWAN-Ampel auf Basis eines TTGO LORA32 ~

Neben der hier in der Anleitung dokumentierten Variante bietet die Software auch die Möglichkeit für verschiedene Anpassungen der CO₂-Ampel. Die Einstellungsparameter hierfür sind jeweils direkt im Softwarecode in den Kommentaren dokumentiert. ~

So kann z.B. auch ein LED-Ring mit 16 LEDs eingesetzt werden. (Hierfür muss der entsprechende Eintrag in der **config.h** angepasst werden.) Außerdem kann anstatt des „Standard“ ESP8266 / NodeMCU V3 Boards auch ein Board mit LoRaWAN Funktion wie zum Beispiel das „TTGO LORA32“ mit SX1276 Chip verwendet werden, um eine drahtlose Übertragung der Messdaten auch außerhalb der Reichweite von WLAN-Netzwerken zu ermöglichen.

Den Anpassungen und Erweiterungen sind im Prinzip keine Grenzen gesetzt, also viel Spaß beim Ausprobieren und Tüfteln. Wir freuen uns, wenn unsere Entwicklung als Basis für weitere tolle Projekte dienen kann.

Beste Grüße,

das CO₂-Ampel Team der HFT Stuttgart...

~

@ \aj X\fxmheCT\Ue\XehaZ-WXe; G*%9 ` cX ~

CO₂ Messung

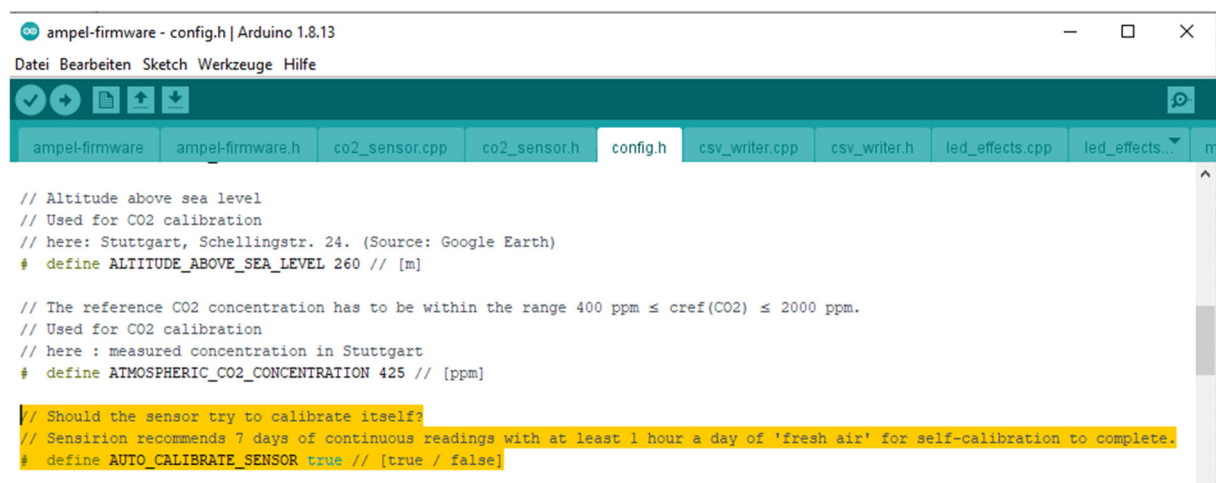
Der Sensirion CO₂-Sensor ist ab Werk vorkalibriert und liefert unseren Erfahrungen nach im Auslieferungszustand überwiegend plausibel CO₂-Messwerte. Dies wurde anhand von Vergleichsmessungen mit einem Kimo C310 Messgerät mit CO₂-Sonde überprüft.

Der SCD30 verfügt über eine Autokalibrierungsfunktion mit der der Sensor seine „Baseline“ findet. Dies setzt eine ununterbrochene Messung von mindestens 7-Tagen voraus, in der der Sensor für eine Stunde pro Tag der „Frischlufte“ ausgesetzt wird.

Wenn dies in der Praxis jedoch nicht eingehalten werden kann, lässt sich die Autokalibrierungsfunktion in der **config.h** Datei ausstellen.

Vergleiche hierzu auch die Praxiserkenntnisse der **IoT-Werkstatt des Umwelt-Campus Birkenfeld**:

<https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/IoT-Werkstatt/octopus/QuickstartKalibrierung.pdf>



```

ampel-firmware - config.h | Arduino 1.8.13
Datei Bearbeiten Sketch Werkzeuge Hilfe

ampel-firmware  ampel-firmware.h  co2_sensor.cpp  co2_sensor.h  config.h  csv_writer.cpp  csv_writer.h  led_effects.cpp  led_effects.h  m

// Altitude above sea level
// Used for CO2 calibration
// here: Stuttgart, Schellingstr. 24. (Source: Google Earth)
# define ALTITUDE_ABOVE_SEA_LEVEL 260 // [m]

// The reference CO2 concentration has to be within the range 400 ppm ≤ cref(CO2) ≤ 2000 ppm.
// Used for CO2 calibration
// here : measured concentration in Stuttgart
# define ATMOSPHERIC_CO2_CONCENTRATION 425 // [ppm]

// Should the sensor try to calibrate itself?
// Sensirion recommends 7 days of continuous readings with at least 1 hour a day of 'fresh air' for self-calibration to complete.
# define AUTO_CALIBRATE_SENSOR true // [true / false]

```

Werden Messwerte unterhalb von 350 ppm CO₂ gemessen, was in der Praxis nicht vorkommen sollte, wechselt die LED-Anzeige in einen **Lila** Leuchtmodus. Dann sollte der Sensor rekaliert werden:

<Tmh-H Tgn\XeXa-K\X-WTf-? XeggX\ a\ZX-E \ahgXa-T' bYXaXa~XafgXesV^Xa-haW-[TgXa-K\X~
 WXa-b\Tf[éCabcYThYWXeH_TgaX&< \X-D=< f mX\ZXa-X\ aXa-U_ThXa-J \aZ-WXe-XeX'a\ZX~
 KX^haWXa-X\ aXa~; bhagWbj a-TU_ghYXo Xaa-T_X-U_ThXa-D=< f-XebfV[Xa-f\ aW\$^Taa-WXeCabcY-
 _bfZX TffXa-j XeWXa&W9 gXaTgi.^Taa-W\X-CT_Ue\XehaZ-ThV[-sUXeW\X-KXe\X_X-KV[a\ gfg_X~
 WheV[-WXa~: XYX[_hVT_UeTgX6-ZXfgTeXg-j XeWXa#<~ \X-F qV[fgX-E XffhaZ-Xeb_Zg\$] Xaa~seVT&
 *E \ahgXa-fgTU\X-E Xffj XegX-Ta-WXe9h^Xa_hYgZX' XffXa-j XeWXa&O TeW\X-CT_Ue\XehaZ~
 Xeb_ZeXV[-j \eW\Xf-WheV[-X\ a-ZesaXf-DThY\V[gf\ZaT_f\Xeg< \X-aqV[fgX\$^T_Ue\XegX~
 E XffhaZ-Xeb_ZgWTaa-aTV[-VT&) E \ahgX&

Wir haben die Erfahrung gemacht, dass insbesondere bei sehr kalten Außentemperaturen unterhalb von 10°C eine erneute Kalibrierung zu tendenziell zu niedrigen Messwerten führt. Um dem entgegen zu wirken, kann der Sensor z.B. in eine Plastik/Zipp-Lock Tüte gepackt werden, welche man mit Frisch/Außenluft füllt, dann verschließt und die Kalibrierung im Innenraum durchführt.

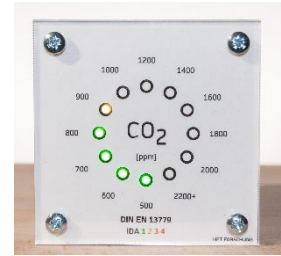
Temperatur Messung

Durch die Abwärme der Sensor-Elektronik sind die Temperatur-Messwerte des SCD30 tendenziell zu hoch, weshalb in unserem Skript ein Temperatur-Offset von -3K eingestellt ist, welcher sich in der **config.h** Datei anpassen lässt.

;G*9 ` cX_~hV^` TahT_~

~

<X@>L%;G*9 ` cX_~eYffg;G*~a~WXeJTh` hYghaWmXVg~
 ` gX\baXeD=<W^K^T_T-Ta-j Taa-Xf-RX\g\fgmh~sYgXa&< X-K^T_T~
 beXagXegfV[-Ta~WXe<F~F~+//1&~



- DhYghT_gggCTgKZbeX~2W\X-D=<f~XhV[gKa-Zesa~U\~0((~cc`#
- DhYghT_gggCTgKZbeX~*2W\X-D=<f~XhV[gKa-ZX_U~1((%)*(~cc`#
- DhYghT_gggCTgKZbeX~+2W\X-D=<f~XhV[gKa-beTaZX~)*(~U\~).((cc`#
- DhYghT_gggCTgKZbeX~,2W\X-D=<f~XhV[gKa~bg~TU~).((cc`#
- DhYghT_gggCTgKZbeX~,2W\X-ZXfT`gX-K^T_T-U_a^g~bg~TU~*(((cc`#

:X\beTaZX~fgDsYgXa-X`cYb[_Xa\$UX\bg~fgDsYgXa-weaZXaW-XebWXe~V[-i~

~

E bagTZX2~

<Tf~?XeggiXeS ZgsUXemj X\DTaZ_rV[Xe~a~WXeJsv^fX\gX&E gmj X\KV[eThUXa~\`~9UfgTaW~
 iba~.(~`~^Taa-Xf-Ta~WXeO TaW~@b~mfV[eThUX+~\$~k,(#bWXe~`g^semXeXa-KV[eThUXa-ThV[~
 Ta-E rUX_a~JXXfgVZgj XeWXa&

~

AaUXg\XUaT[-`X2~

KgKV^Xa-K\X-WTf~`*N%F XgmX\~iba-hagXa~a~WTf~?XeggX\~O TaW~NTe\TagX#&GWXeiba-bUXa~
 UX\WXeL fV[-NTe\TagX&~

~

KgTeg

:X\`KgTegibeZTaZ-f\ZaT_f\XegWTf~?XeggmhXefgW\X-KhV[X-aTV[-X\ax`<eT[gbfaxgmj Xe^~`g~
 X\ax`~jXZXaUbZXa~DThY~V[g~F TV[-Xe~b ZeX~V[Xe~9a`X WhaZ~\`~F Xgmj Xe^~j XV[fX_gW\X~
 >TeUX-WXf~DThY~V[gf~ThY~?esa&~a~bgkf~DThY~V[gf\ZaT_f\XegH~bU_X`X~UX\WXe~
 F Xgmj Xe^~iXeU\~aWhaZ&~a~U_ThXf~DThY~V[gZX[gZZ~aTa\$U\~f~WXeKXafbeW\X-XefgX~
 _\XXeg&~

~

F TV[gE bWhf2

>seWXa~m&~aqV[gV[Xa~Xg~XU~\`~KV[~Tm\`~`Xe\$~TffXa-fV[-W\X-D=<f~Thffv[TgKa~a~WX`~
 WXepYTf[~eCabcYThYWXeH_TgaX&<X-D=<f~mX\ZXa-X\axa~U_ThXa~j~aZ~WXe~YseX\~a~ZX-KX^haWXa~
 O~eW~WXepYTf[~eCabcYXeaXhgZXWes^~gj~eW~W\X-D=<~9amX\ZX~j~XWXeX~aZXfV[TgKa&

~

CT_Ue\XchaZ2

H_Tgm\XeXa-K\X-WTf~?XeggX\~a~ZX-E`ahgKa-T`~bYXaXa~>XafgX&<esV^Xa~haW~[TgKa-K\X-WXa~
 pYTf[~eCabcYThYWXeH_TgaX&<X-D=<f~mX\ZXa-X\axa~U_ThXa~j~aZ~WXe~YseX\~a~ZX-KX^haWXa~
 X\axa~;bhagWbj a~TU_ghYg&O Xaa~T_X~U_ThXa~D=<f~XebfV[Xa~f~aW\$^Taa~WXeCabcY~
 _bfZX_TffXa~j~X~dWXa&<X~F~qV[f~gX-E XffhaZ~Xe~b_ZgaTV[-VT&~E`ahgX&O TeW\X~CT_Ue\XchaZ~
 Xe~b_ZeX~V[-j~eW~W\Xf~WheV[-X~a~ZesaXf~DThY~V[gf\ZaT_f\Xeg&<X~aqV[f~gX\$^T_Ue\XegX~
 E XffhaZ~Xe~b_ZgaTV[-VT&~E`ahgX&

~

O X\gXa~a~be`TgbaXa-hagXe2~

CbagT^g~ Vb*fXafbe8 [YgfhggZTegWX~