

Anleitung Multi-Monitoring Gadget

**Siemens
Professional Education KHE
Karlsruhe 76187
Siemensallee 75**

„Wenn du sprichst, wiederholst du nur, was du eh schon weißt; wenn du aber zuhörst, kannst du unter Umständen etwas Neues lernen.“

- Dalei Lama

#SPE
#Zukunft
#Technik
#Influencer
#Gestaltung
#Individualität
#WollenundKönnen
#WissenundUmsetzung
#Problemeerkennenundlösen
#LösungenstattKopffindenSandstecken

Inhaltsverzeichnis

1	Vorwort	2
2	Einführung.....	2
3	Aufbau.....	4
	Programmierung	8
4	Anleitung – Programmierung - Multi-MonG.....	11
5	Inbetriebnahme.....	18
6	Inbetriebnahme des Sensors für die Luftqualität.....	20
7	Schlusswort	23
	Abbildungsverzeichnis.....	24
	Tabellenverzeichnis	25
8	Quellenverzeichnis	26
9	Änderungsliste – Version 1.1	27

1 Vorwort

Dieses Dokument ist in mehreren Kapiteln gegliedert und soll dir das Gadget (Multi-Monitoring Gadget) näherbringen. Darum wird im ersten Kapitel das Gadget im Allgemeinen vorgestellt. Hierzu gehören die verschiedenen Funktionen und der Mikrocontroller. Im nächsten Kapitel wird auf die Module, den Verdrahtungsplan, und die Bauteile des Gadgets eingegangen. Damit im darauffolgenden Kapitel, mit der Inbetriebnahme des Gadgets angefangen werden kann. Im vorletzten Kapitel wird die Programmierung vorgestellt und schematisch an einem Beispiel erklärt. Daraufhin folgt eine Anleitung, in der beschrieben wird, wie du dein Gadget programmieren kannst. Zum Schluss folgt das letzte Kapitel, indem der Ablauf der Konstruktion und die Planung deines Gadgets vorgestellt wird.

2 Einführung

Das Gadget ist eine Station mit insgesamt sieben Funktionen. Diese Funktionen werden mit einem Mikrocontroller und verschiedenen Modulen realisiert.

Diese Funktionen sind:

Nr.	Funktion
1	Anzeigen der aktuellen Uhrzeit
2	Anzeigen der Temperatur
3	Anzeigen der Luftfeuchte
4	Anzeigen des Timers
5	Anzeigen einer eigenen Nachricht
6	Anzeigen der Luftqualität
7	Zyklisches wechseln (5 Sekunden), zwischen Funktion eins, zwei, drei und sechs

Tabelle 2.1: Gadget: Funktionen im Überblick

Auf die Funktionen wird in den nächsten Kapiteln ausführlicher eingegangen.

Ein Mikrocontroller ist ein „kleiner mini Computer“. Sie sind im Alltag eher unbekannt und sind die Helden hinter den Dingen [1]. Viele Geräte wären ohne einen Mikrocontroller undenkbar [1]. Meistens werden sie für Steuerungs- und Kommunikationsaufgabe eingesetzt [1]. So finden sie z.B. in den folgenden Geräten und Anwendungen ihren Platz [3]:

- Radio
- Roboter
- Fernseher
- Flugzeuge
- Fernbedienungen
- Waschmaschinen
- Intelligente Systeme z.B. Airbag
- Intelligente Systeme in der Automatisierung
- Erfassung von Messwerten z.B. Drehzahl

Die Aufgaben sind vielfältig und so auch die Möglichkeiten, die sich dadurch ergeben. Ein Mikrocontroller muss für seine zu bewerkstellende Aufgabe programmiert werden. Diesen Programmcode durchläuft der Mikrocontroller immer wieder „unendlich oft“.

So führt er präzise seine Aufgabe aus und kommt dabei nicht durcheinander, denn er führt Zeile für Zeile in seinem programmierten Code aus. Dadurch wird auch deutlich, dass die

Programmierung ein wesentlicher Bestandteil der Elektrotechnik ist und immer mehr Einzug erhält.

In Abbildung 2.1 ist unser verwendeter Mikrocontroller ESP8266 zu sehen.

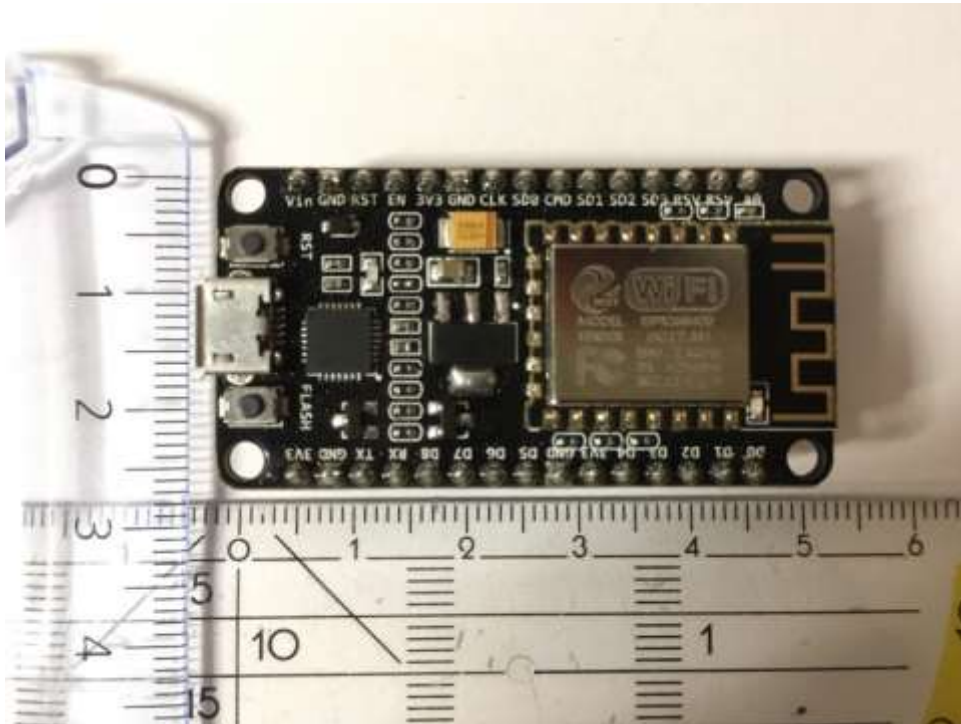


Abbildung 2.1: Darstellung des ESP8266

Wie zu erkennen ist, hat unser ESP8266 eine Länge von ca. 5 cm und eine Breite von ca. 2,7 cm. Aufgrund der Größe kann dieser nicht die gleiche Rechenleistung wie ein Laptop bzw. Computer haben [2]. Da sie eine geringere Leistung haben, benötigen sie in der Regel keine aktive Kühlung wie ein Computer oder Laptop [2].

3 Aufbau

Insgesamt besteht das Gadget aus acht Bauteilen, diese sind in der Tabelle 3.1 aufgelistet.

Nr.	Bauteile	Nr.	Bauteile
1	Platine	6	ESP8266
2	Holzbrett	7	DHT11 (Sensor)
3	Taster	8	Konstruktionsteilen
4	Lautsprecher	9	Verbinder
5	Steckleiste	10	MQ135 (Luftqualitätssensor)

Tabelle 3.1: Gadget: Übersicht über die Bauteile

Diese Bauteile müssen korrekt verdrahtet bzw. miteinander verbunden werden, damit die entsprechenden Funktionen gegeben sind. Diese Aufgabe übernimmt die Platine für uns. Auf ihr befinden sich Leiterbahnen, die die Verbindungen zwischen den Modulen und dem Mikrocontroller herstellen. Die Bauteile ihres Gadgets sind in der Abbildung 3.1 abgebildet. In der oberen Reihe ist in chronologischer Reihenfolge; die Platine, die Steckleisten und die gewinkelte Steckleiste zu erkennen. Darauf folgt in der zweiten Reihe; der Lautsprecher, Taster, Abstandshalter und DHT11 Sensor. Als letztes folgt die dritte Reihe mit der Led-Matrix und dem MQ135 Sensor.

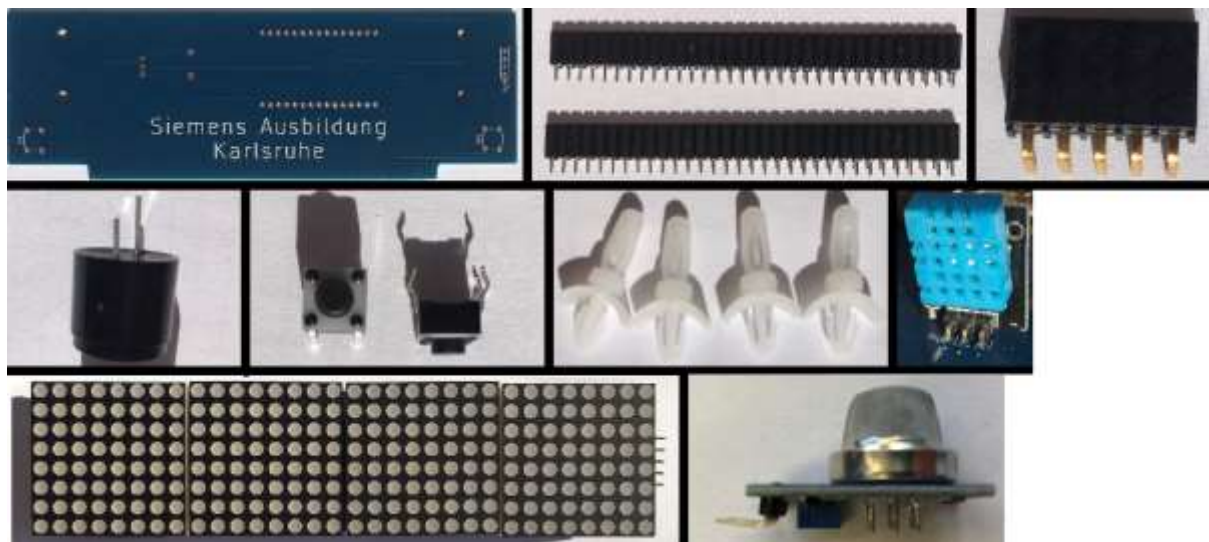


Abbildung 3.1: Aufbau: Bauteile

In Abbildung 3.2 ist der Schaltplan eures Gadgets zu sehen. Es fällt auf, dass jedes Bauteil seinen eigenen Steckplatz hat. Die Größe der Bauteile sind auf der Platine angedeutet und müssen korrekt gesteckt, sowie verlötet werden.

Hierbei ist es wichtig auf die Polarität der Bauteile zu achten, und auf eine angemessene Verlötlung Wert zu legen, damit jede Verbindung korrekt verbunden ist.

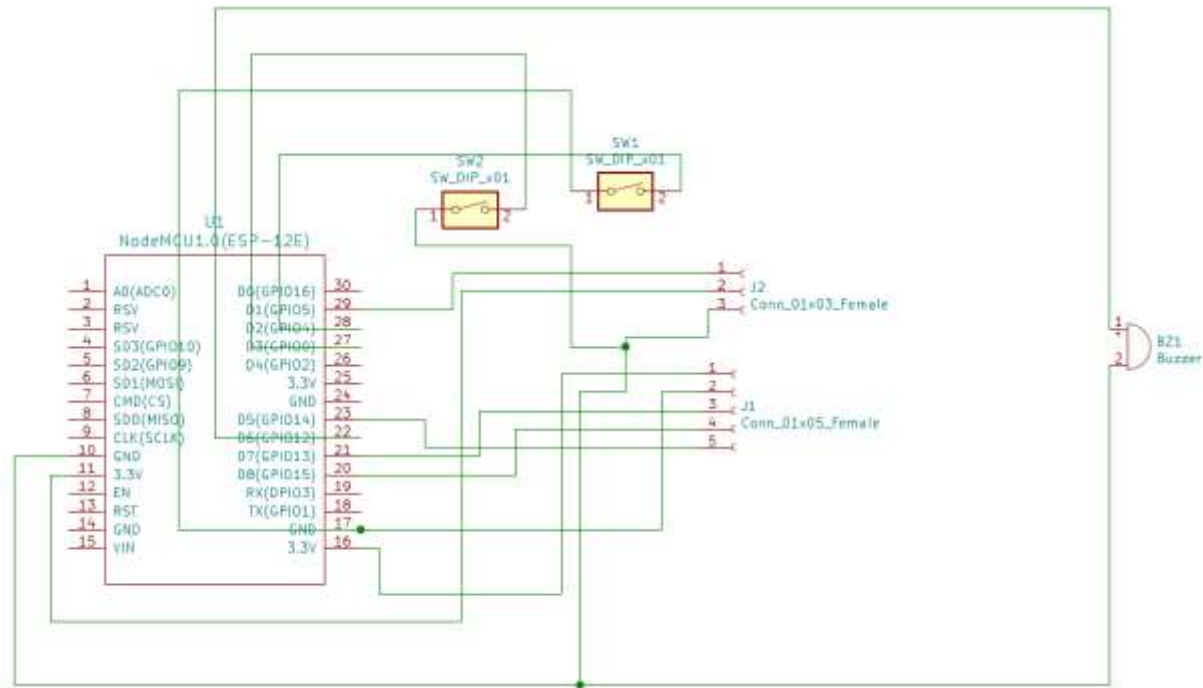


Abbildung 3.2: Gadget: Schaltplan

Als erstes wollen wir die Steckverbinder verlöten. Diese müssen zu Beginn gekürzt werden, um sie dann zu verbinden. Dabei muss darauf geachtet werden, dass diese relativ gerade verlötet werden, damit die Bauteile, und insbesondere der ESP8266 eingesteckt werden kann. In Abbildung 3.3 sind die Steckverbinder für den Mikrokontroller und der Led-Matrix zu sehen.



Abbildung 3.3: Aufbau: Steckverbinder

Als nächstes kann der Piezzo-Lautsprecher verlötet werden, hierbei ist auf die Polarität zu achten. Nach dem Verlöten vom Piezzo-Lautsprecher, müssen seine langen Beine gekürzt werden. Anschließend sind die Taster dran, diese können hineingeklickt werden, und müssen ebenfalls verlötet werden. Somit fehlt noch der DHT11 Sensor, der mithilfe einer Steckleiste, die verlötet werden muss, eingesteckt wird. Aufgepasst, denn hier muss auf die Polarität geachtet werden. Daraufhin darf die Steckleiste für den MQ135 verlötet werden, auf die dann der MQ135 gesteckt werden darf. In Abbildung 3.6 ist der DHT11 Sensor und die Steckleiste für den MQ135 dargestellt. Im Anschluss werden die Abstandshalter in die Platine gesteckt. In Summe sieht es anschließend wie in Abbildung 3.4 und 3.5 aus.



Abbildung 3.4: Aufbau: Vorderseite



Abbildung 3.5: Aufbau: Rückseite

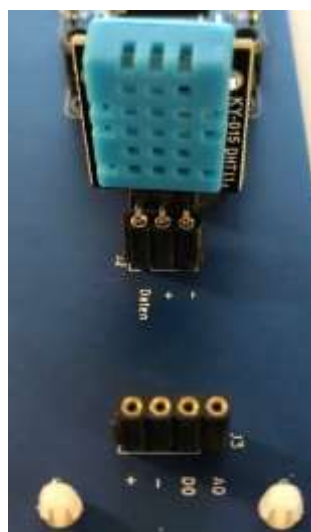


Abbildung 3.6: Aufbau: DHT11 Sensor

Daraufhin folgt der vorletzte Arbeitsschritt, welcher darin besteht, die LED-Matrix auf die Abstandshalter zu stecken und eine Verbindung mit der Platine herzustellen, wie in Abbildung 3.7 und 3.8 dargestellt. Tipp: Stelle eine sichere Verbindung der Steckleiste und der schrägen Steckleiste der LED-Matrix her (darf feste gedrückt oder auch zusätzlich verlötet werden). Der letzte Schritt besteht darin, den ESP8266 in die Steckleiste zu drücken und die Platine schräg mit einem tropfen Sekundenkleber in das Holzbrett zu stecken. Anschließend muss das Gadget Multi-MonG programmiert werden, damit es mit der Erklärung in Kapitel 4 gesteuert werden kann.

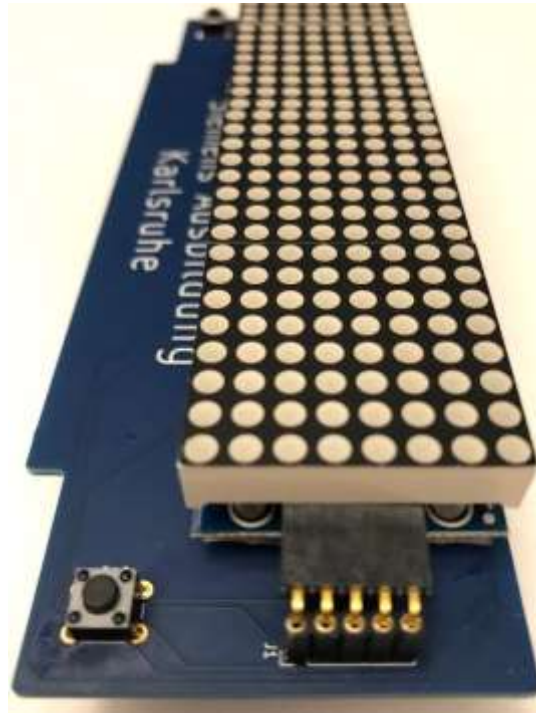


Abbildung 3.7: Aufbau: Verbindung der LED-Matrix mit der Platine



Abbildung 3.8: Aufbau: Gadget

4 Programmierung

Vorab muss erwähnt werden, dass die Programmierung so einfach wie möglich gehalten wurde, wodurch auf gewisse Konventionen bewusst verzichtet wurde.

Betrachtet man einen beliebigen Programmcode, dann besteht dieser aus mehreren Elementen bzw. Abschnitten.

Die wesentlichen Abschnitte sind die folgenden:

```
// Beschreibung von Ihrem Programm

#include <...>          // Eintragung der benötigten Biobiotheken

#define A 5            // Eigene Definitionen


void Setup()           // Dieser Programmteil wird einmal, beim Start, ausgeführt
{
    Ihre Anweisungen;  // Anweisungen die nur einmalig ausgeführt werden müssen
}

void Loop()            // Dieser Programmteil wird immer wieder ausgeführt
{
    Ihre Anweisungen;  // Anweisungen die wiederholt benutzt werden müssen
}
```

Man kann sich leicht vorstellen, dass in der void Loop beliebig viele komplexe Funktionen aufgerufen werden können. Hierdurch kann ein Programmcode übersichtlich, sowie leicht zu verstehen, programmiert werden.

Insgesamt gibt es sechs Funktionen: Uhrzeit, Temperatur, Luftfeuchte, Timer, eigene Nachricht, zyklischer Wechsel und Internetseite. Dabei darf man nicht das Hauptprogramm, das vorhin vorgestellt wurde, vergessen. Somit würde man daraus schließen, dass es sieben Funktionen und das Hauptprogramm gibt. Leider stimmt dies nicht ganz, da es noch die Module, Taster, Led-Matrix und Lautsprecher gibt. Diese Ein- und Ausgabe von Signalen muss natürlich auch gehandhabt werden. Aus diesem Grund kommen demnach noch drei zusätzliche Funktionen hinzu.

Unter dem Strich bedeutet dies, dass es 11 Funktionen und ein Hauptprogramm gibt. Schematisch dargestellt, sieht dies wie in Abbildung 5.1 aus.

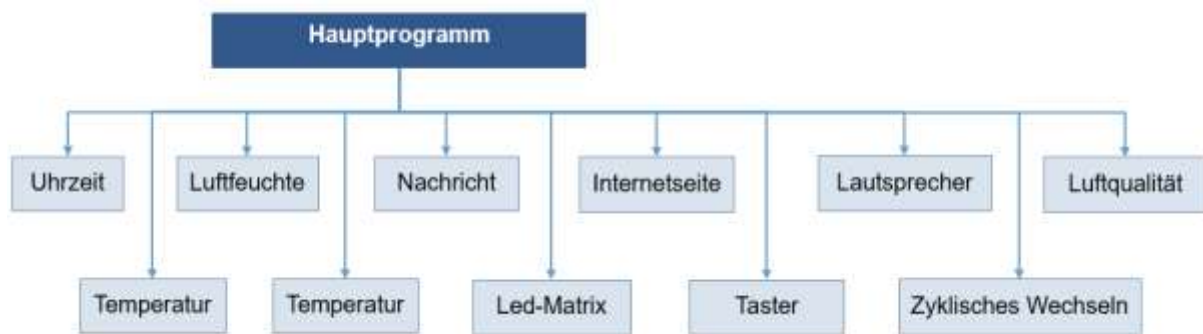


Abbildung 4.1: Programmierung - Programmaufbau

Weitergehend mit Interrupts, welche dafür da sind, einen bestimmten Programmcode in Abhängigkeit von einem Ereignis abzuarbeiten. In unserem Fall haben wir Interrupts dafür verwendet, um die Sekunden zu zählen. Dies bedeutet, dass wir durch das Zählen der Sekunden, in der Lage sind, eine Uhr zu programmieren. Zusätzlich wurde, das Zählen der Sekunden für das zyklische Wechseln der Modis genutzt.

Leider lernt man das Programmieren, weder über Nacht noch über eine Eingebung. Programmieren ist ein logischer Prozess, bei dem Zeile für Zeile programmiert und vom Mikrocontroller abgearbeitet wird. In unserem Fall wurde der Mikrocontroller in der Programmiersprache C programmiert.

Als Beispiel soll die Uhrzeit bzw. deren Programmcode vorgestellt werden.

Für diese benötigt man im Wesentlichen drei Variablen:

Nr.	Name der Variable
1	Sekunden
2	Minuten
3	Stunden

Tabelle 4.1: Programmierung - Variablen für die Uhrzeit

Für diese Funktion wird im Wesentlichen die If-Else-Verzweigung genutzt. Diese Verzweigung überprüft, ob eine Bedingung erfüllt ist und führt bei erfüllter Bedingung einen Programmcode aus. Zu erwähnen ist, dass die Funktion, Internetseite, eine Variable auf „1“ setzt, wenn der Nutzer die Uhrzeit stellen möchte. Dadurch werden die Variablen, die in der Tabelle 5.2 gelistet sind, benötigt.

Nr.	Name der Variable
1	Sekunden_stellen
2	Minuten_stellen
3	Stunden_stellen

Tabelle 4.2: Programmierung - Variablen für die Uhrzeit

Diese Variablen sind dafür da, damit die Uhrzeit neu gesetzt werden kann. Zusätzlich kann man sich vorstellen, dass es unterschiedliche Variablen gibt. Die verwendeten Variablen sind vom Typ >Integer<, abgekürzt int. Variablen vom Typ int sind ganze Zahlen.

Wir stellen uns vor, dass wir eine Variable >test< hätten. Als Beispiel wollen wir abfragen, ob die Variable test größer als die Zahl 10 ist. Wenn dieser Fall eintritt, dann wird die Variable test um 1 erhöht.

```
If ( test < 10 ) // Bedingung ob test
{
    test = test + 1; // Das ist eine Anweisung
}
```

Anmerkung: Anweisungen müssen mit einem > ; <, Semikolon abgeschlossen werden.

Als nächstes wird der Programmcode als Flussdiagramm dargestellt. Dies hat den Grund, dass man mit diesem einfach und schnell ein Programm analysieren, sowie verstehen kann. Doch davor wollen wir die Symbole kurz in der Tabelle 5.3 erläutern [3].




Nr.	Symbol	Erläuterung
1.		Signalisiert den Start und das Ende
2.		Entscheidungen
3.		Anweisungen

Tabelle 4.3: Programmierung - Symbole

Das Fluss-Diagramm ist relativ leicht zu verstehen und ist in Pfeilrichtung zu lesen.

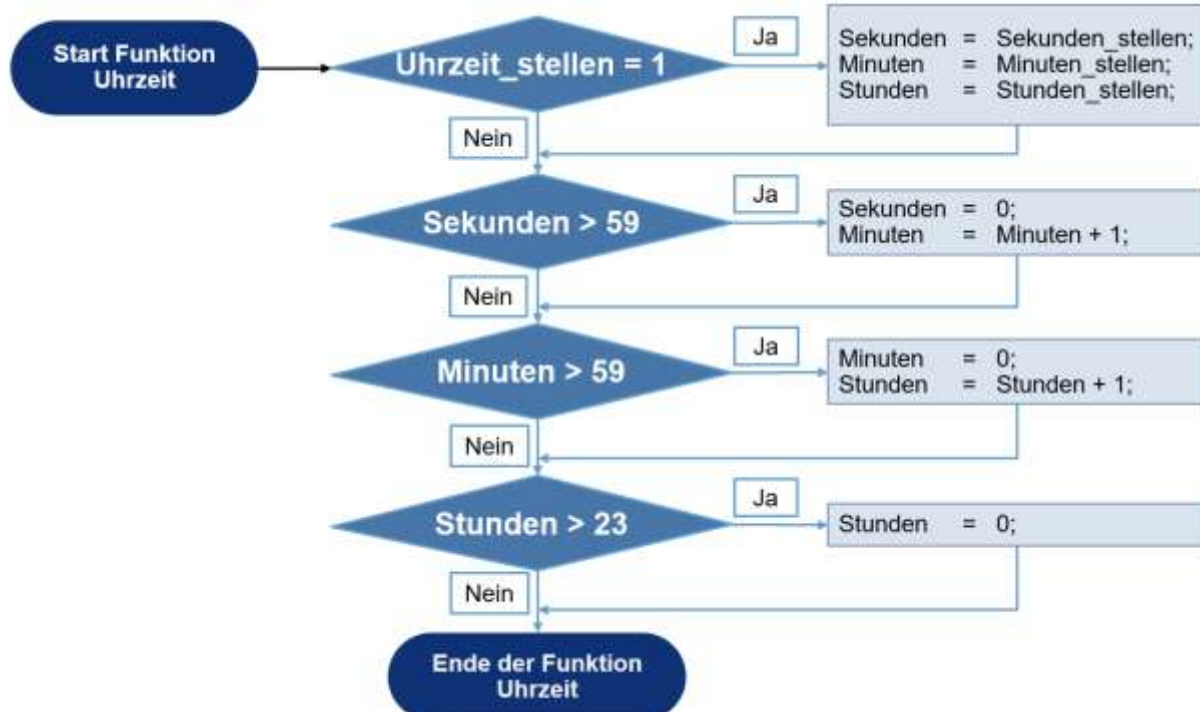


Abbildung 4.2: Programmierung - Uhrzeit Flussdiagramm

Nochmal als Wiederholung, es gibt ein Interrupt, dieser wird jede Sekunde ausgelöst und erhöht den Wert der Variablen - Sekunde - um eins.

5 Anleitung – Programmierung - Multi-MonG

Die Programmierung kannst du in sechs Schritten durchführen. Im Folgenden werden diese sechs Teile erläutert. In den verschiedenen Bildern ist die Abfolge der Auswahlen mit Kreisen umrandet. In Abbildung 6.1 ist die Abfolge dargestellt. Zu Beginn wird auf die Konfiguration eingegangen, um anschließend ein Programm, welches als Beispiel dienen soll, in den ESP8266 zu laden. Daraufhin wird dann das eigentliche Programm verändert und in das Gadget – Multi-MonG geladen.

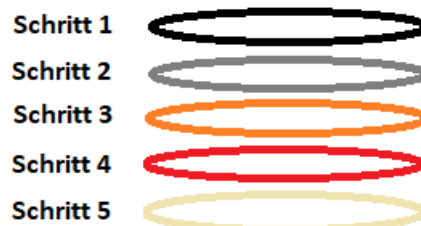


Abbildung 5.1: Anleitung: Folge von Schritten

Teil 1 - Konfiguration zum Hinzufügen vom Board (URL)

- 1.1 Der erste Schritt besteht darin, die Arduino Software zu starten.
- 1.2 Anschließend muss Datei (oben links) und die Auswahl Voreinstellungen ausgewählt werden.
- 1.3 Unter dem Punkt - zusätzliche Boardverwalter URL - muss die folgende Adresse eingegeben werden: http://arduino.esp8266.com/stable/package_esp8266com_index.json

INFO - Anschließend kann das Board über die Arduino Umgebung gedownloadet werden.

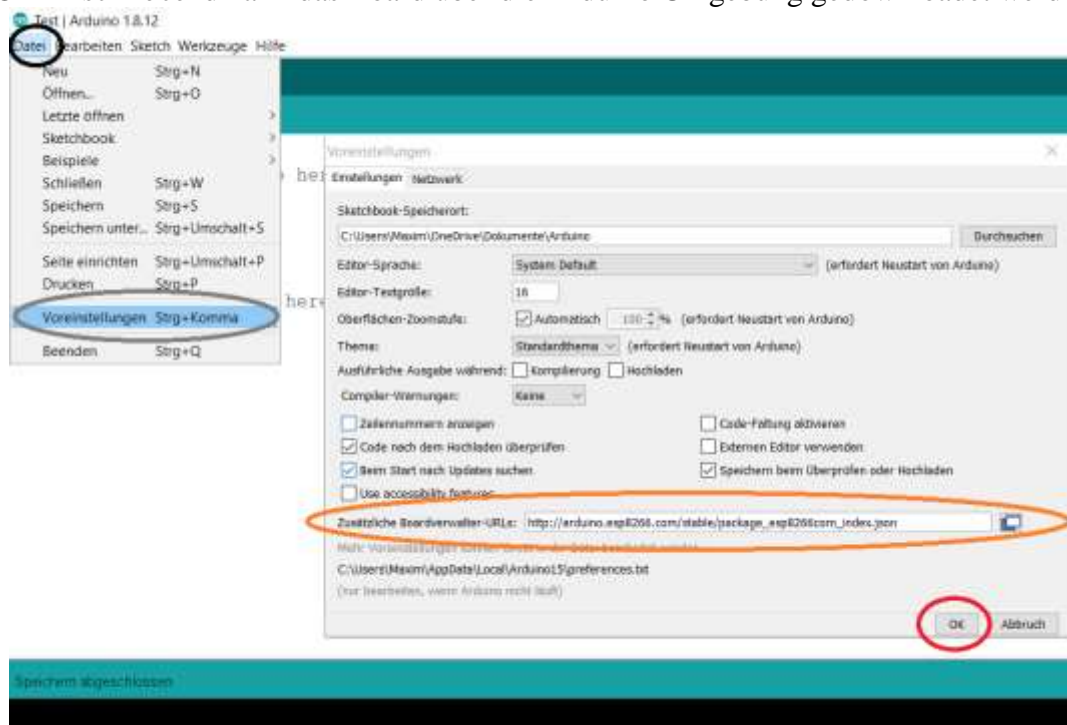


Abbildung 5.2: Konfiguration: URL

Teil 2 - Konfiguration zum Hinzufügen vom Board (Download)

- 2.1 Hierfür muss im Reiter der Punkt Werkzeuge geöffnet werden, um dann den Punkt Board zu selektieren.
- 2.2 Anschließend muss der Punkt Boardverwalter ausgewählt werden.
- 2.3 In der Suche muss der ESP8266 eingegeben
- 2.4 und die Datei - esp8266 by ESP8266 Community installiert werden.

Wie in Abbildung 6.3 dargestellt.

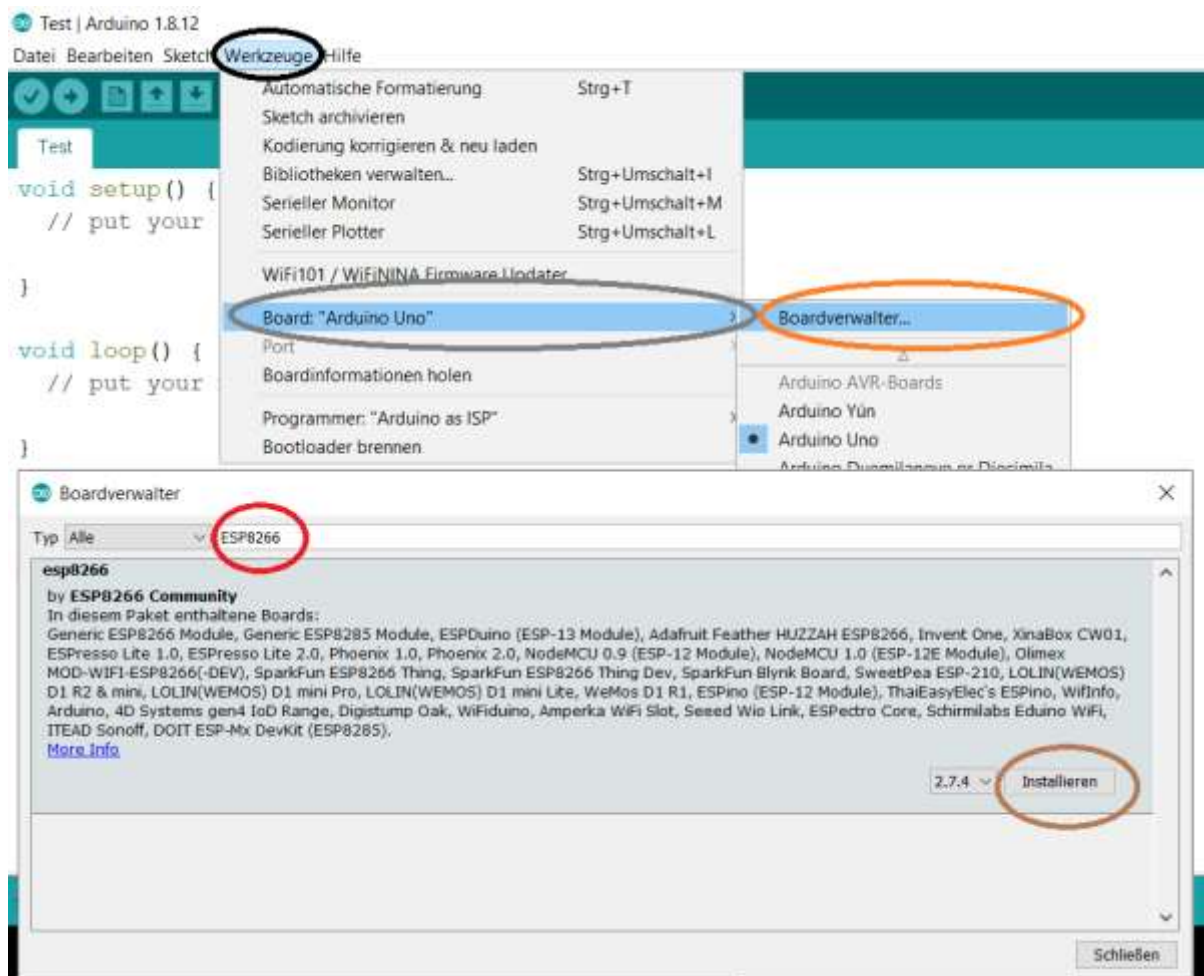


Abbildung 5.3: Konfiguration: Installieren vom Board

Teil 3 - Hineinladen von einem Programm - als Beispiel

- 3.1 Für das hineinladen von einem Programm, muss der ESP8266 mit dem Computer verbunden werden.
- 3.2 Als Test soll das folgende Programm dienen:

```
void setup()
{

  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }
}

void loop()
{
  Serial.println("Guten Tag, geschafft!!");
}
```



Abbildung 5.4: Programm: Austausch vom Programmcode

3.3 Anschließend muss unter Werkzeuge – das Board ESP8266 ausgewählt werden, um der Software mitzuteilen, welches Board du programmieren möchtest. (Abb 6.5)

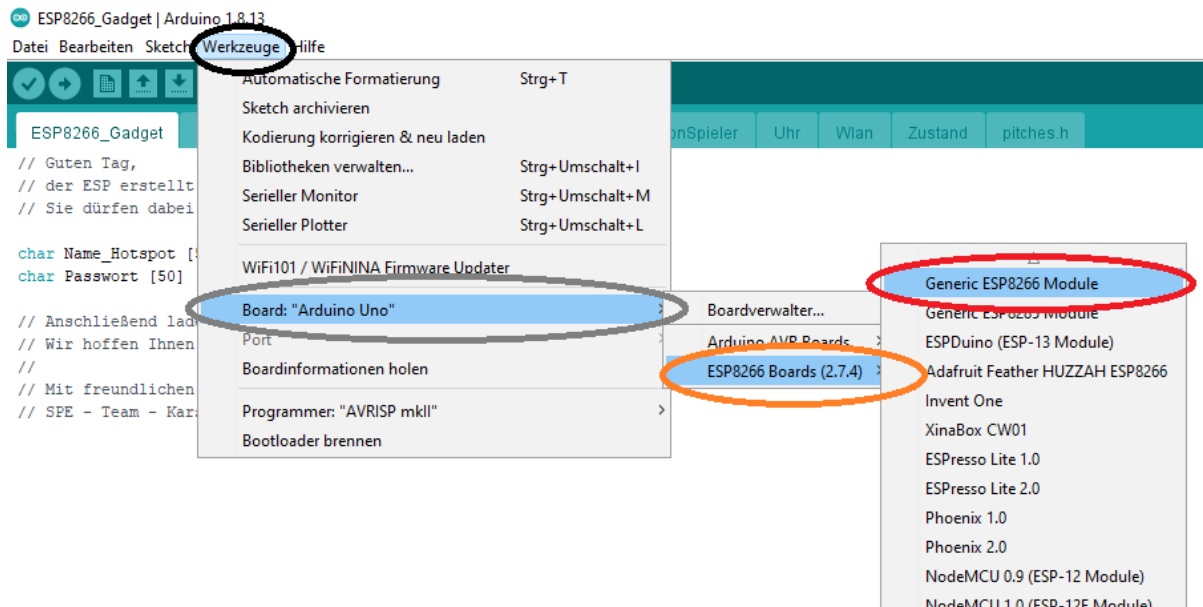


Abbildung 5.5: Programm: Einstellen vom verwendeten Board

3.4 Wähle unter dem Punkt Werkzeuge – den richtigen Port aus. (Vermutlich wird nur einer angezeigt) (Abb. 6.6)

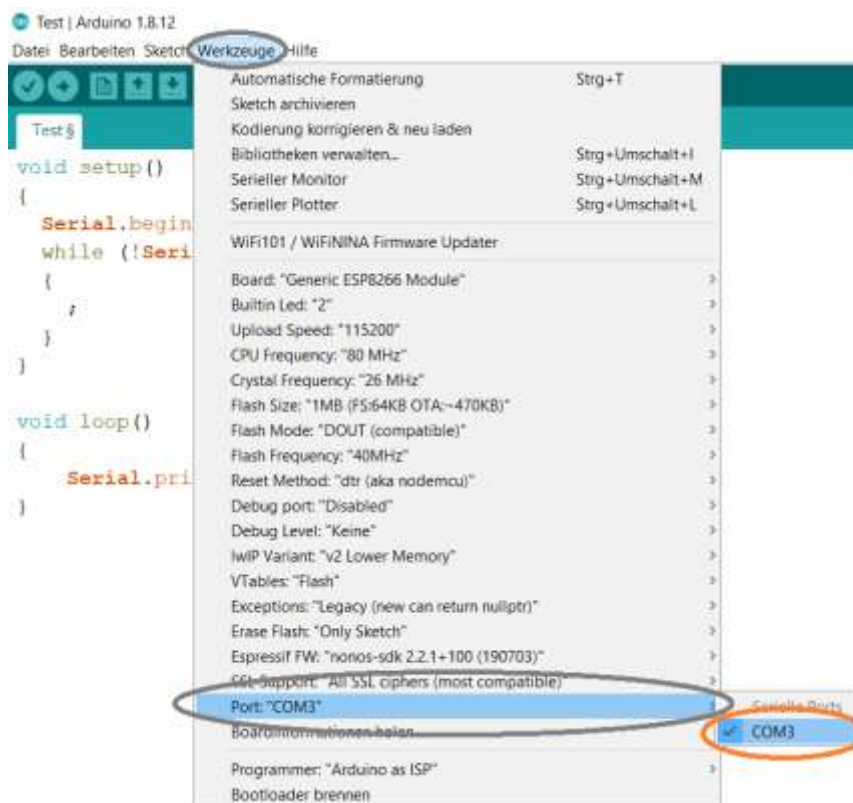


Abbildung 5.6: Konfiguration: Einstellen vom verwendeten Port

- 3.4 Danach auf den Pfeil klicken - welcher nach rechts zeigt. (Abb. 6.7)
- 3.5 Du wirst aufgefordert die Datei zu speichern – wenn du es nicht schon getan hast.



Abbildung 5.7: Programm: Programmierung - Beispiel

- 3.6 Wähle unter Werkzeuge - den Seriellen Monitor aus, um die Kommunikation mit dem ESP8266 zu testen. (Abb 6.9)

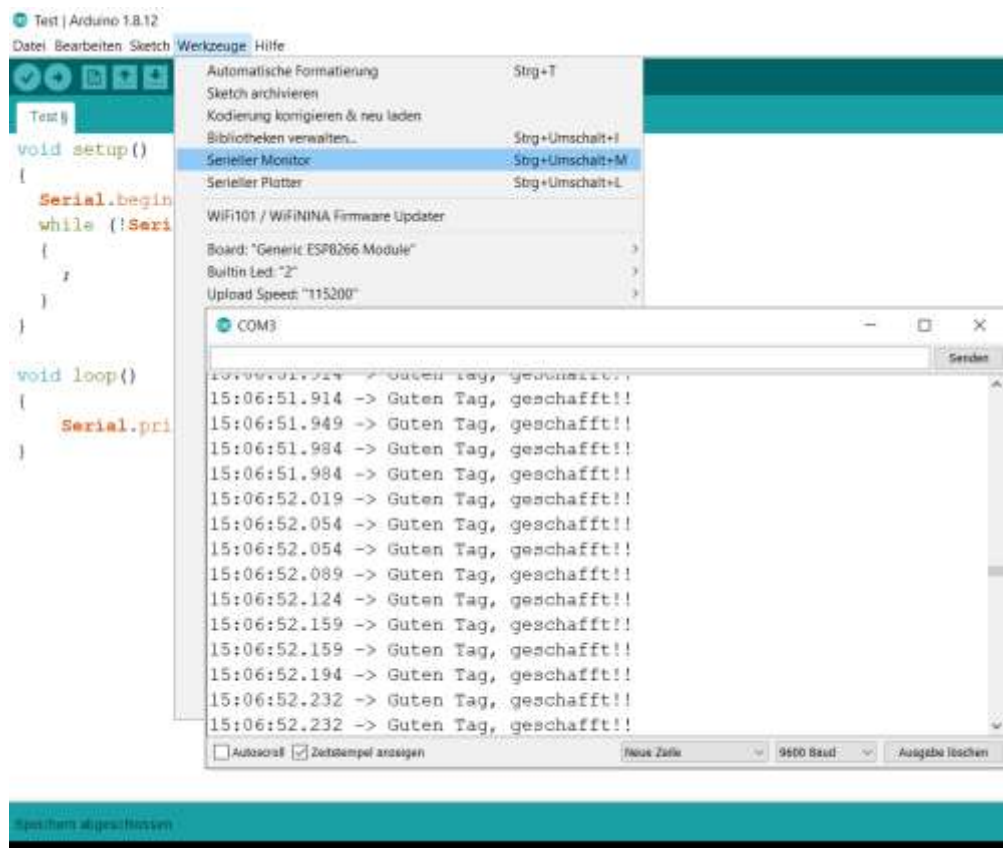


Abbildung 5.8: Kommunikation: Serial

Teil 4 – Entpacken der Datei und verschieben der Bibliotheken (libraries)

- 4.1 Datei aus GitHub Herunterladen (...)
- 4.2 Datei unter - Dokumente - im Ordner Arduino entpacken. (Abb. 6.9)

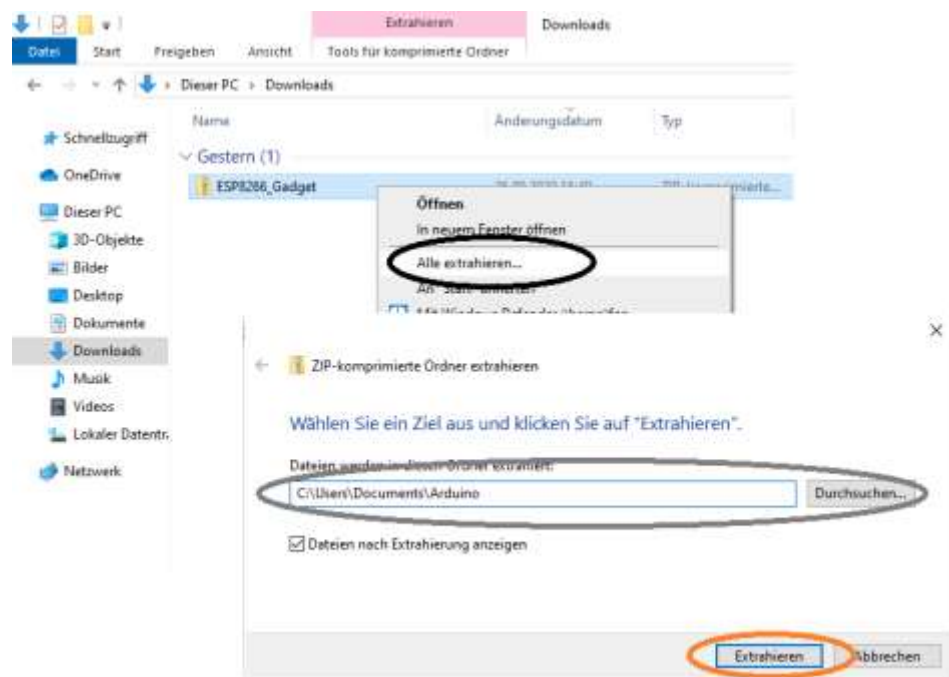


Abbildung 5.9: Programm: Entpacken der Datei

- 4.3 Bibliotheken (libraries) aus dem Projekt ESP8266_Gadget ausschneiden und im Ordner Dokumente unter Arduino (libraries) ablegen. (Abb. 6.10)

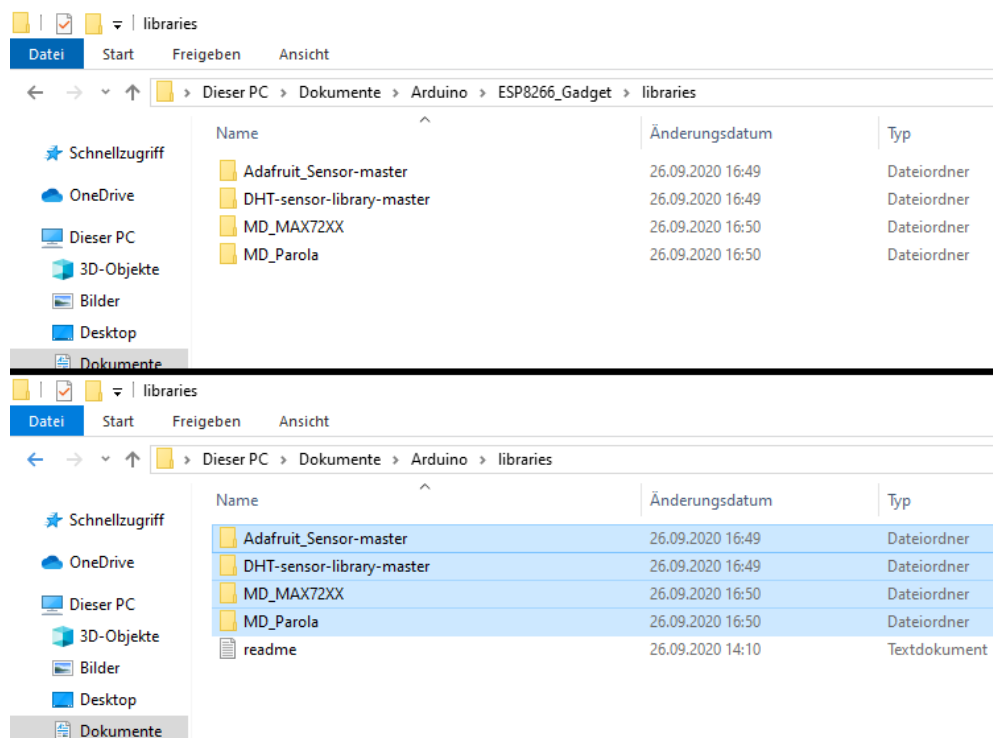


Abbildung 5.10: Entpacken: Hinzufügen der Bibliotheken

Teil 5 – Programmierung der Datei

5.1 Datei öffnen - ESP8266_Gadget. (Abb. 6.11)

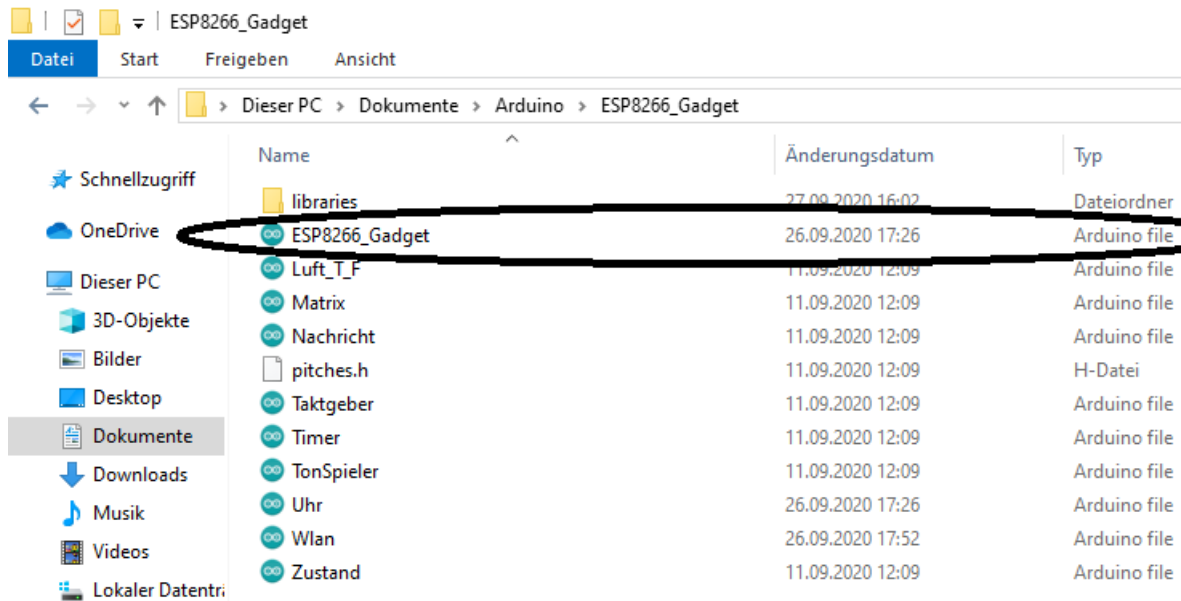


Abbildung 5.11: Programmierung: Öffnen der Datei

5.2 Übertragen auf den ESP8266. (Mit dem Pfeil nach rechts) (Abb. 6.12)

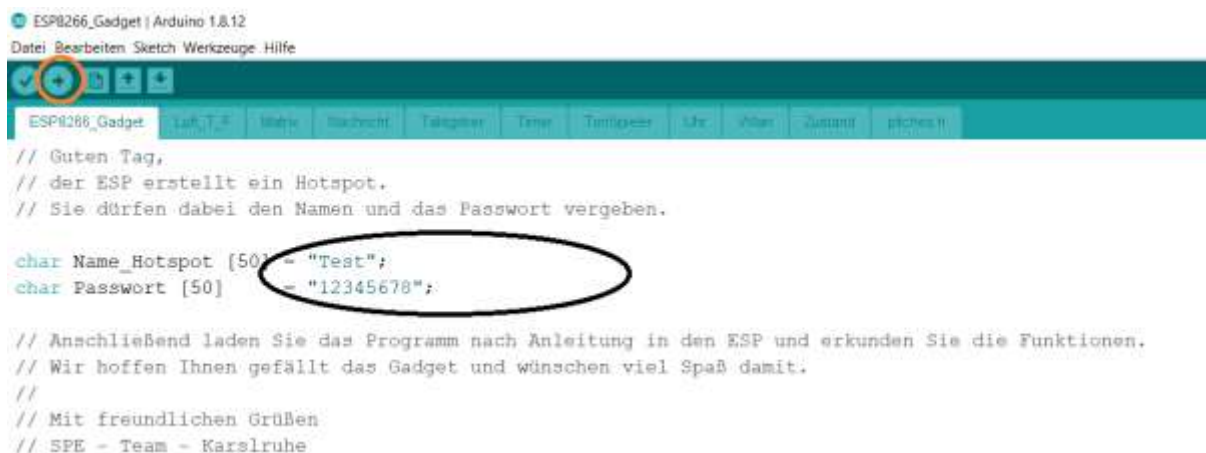


Abbildung 5.12: Programmierung: Übertragung auf das Gadget

6 Inbetriebnahme

Der erste Schritt besteht darin, für die Spannungsversorgung des Gadgets zu sorgen. Es sollte eine Melodie ertönen, die den Start signalisiert. Anschließend wird auf der Led-Matrix die aktuelle Temperatur dargestellt. Den aktuellen Modus kannst du mit dem Taster unten rechts auswählen. Der Wechsel zwischen den Funktionen findet in Chronologischer Reihenfolge statt. Ab der Funktion sieben startet es wieder mit der ersten Funktion. In der Abbildung 4.1 ist dies schematisch dargestellt.

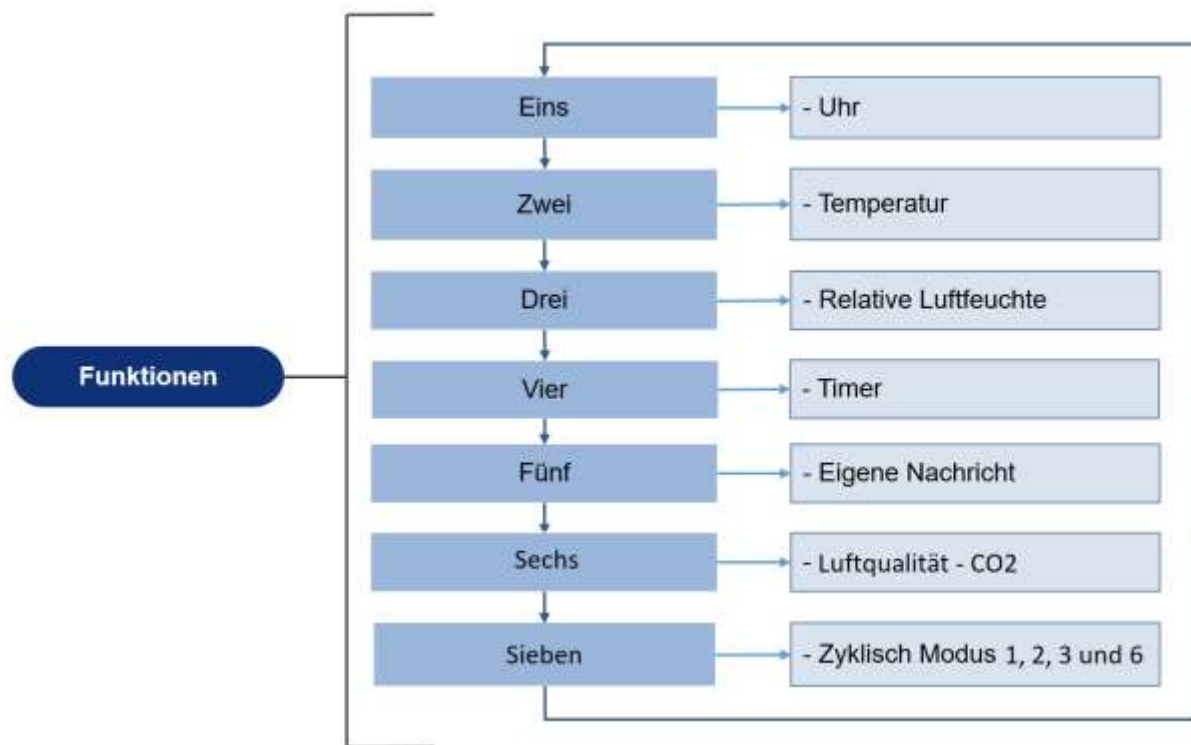


Abbildung 6.1: Schematische Darstellung: Wechsel zwischen den Funktionen

Auf dem ESP8266 befindet sich eine Internetseite. Für den Aufruf der Internetseite muss man sich mit dem Wlan-Netzwerk des ESP8266 verbinden. Jeder verwendete ESP8266 hat einen eigenen „Wlan-Namen“. Den Namen und das Passwort kannst du selbst eintragen.

Wenn die Verbindung erfolgreich hergestellt wurde, kann man auf die Internetseite bzw. den Server zugreifen. Hierzu muss ein Browser (z.B. Safari, Firefox) geöffnet werden und die IP-Adresse, 192.168.4.1, des ESP 8266 eingegeben werden.

Dadurch sollte sich die Internetseite, wie in Abbildung 4.2 zu sehen, öffnen.

Gadget - MultiMonG

Modus:

Timer:

Nachrichte:

Stunden:

Minuten:

Sekunden:

Messwert - Luft:

SIEMENS
Ingenuity for life

Anleitung:
Mit dieser Internetseite kann das Gadget - MultiMonG gesteuert werden.

Modus:
- Zahl 1: Uhr
- Zahl 2: Temperatur
- Zahl 3: Luftfeuchte
- Zahl 4: Timer
- Zahl 5: Laufnachricht
- Zahl 6: CO2 Messwert in ppm
- Zahl 7: Wechselt zwischen Uhr, Temperatur und Luftfeuchte

Nachricht:
Hier kann eine beliebige Nachricht eingetragen werden, welche dann im Modus 5 angezeigt wird

Timer:
Hier kann die Timerzeit in Minuten eingetragen werden

Stunden, Minuten, Sekunden:
Hier kann die Uhrzeit gestellt werden

Messwert - Luft
Trage hier den folgenden Messwert ein, wenn du dich an der frischen Luft befindest: 75

Schaltfläche Submit:
Mit dieser kann Ihre Eingabe bestätigt und an den ESP8266 uebertragen werden

Abbildung 6.2: Internetseite zur Konfiguration des Gadgets

Auf der Internetseite sind sieben Textfelder zu erkennen. Die Funktionen dieser Textfelder sind in der Tabelle 5.1 aufgelistet.

Nr.	Name	Beschreibung
1	Modus	Auswahl - Eintragung der Zahlen 1 bis 7, um zwischen den Modis zu wechseln.
2	Timer	Setzen - Eingabe der Minuten, die der Timer herunterzählt.
3	Nachricht	Spruch - Eingabe einer Nachricht, die im Modus fünf durchläuft.
4	Stunden	Uhrzeit - Stellen der Stunden
5	Minuten	Uhrzeit - Stellen der Minuten
6	Sekunden	Uhrzeit - Stellen der Sekunden
7	Messwert - Luft	Eintragung des Messwertes an der frischen Luft

Tabelle 6.1: Internetseite: Überblick über die Funktionen

7 Inbetriebnahme des Sensors für die Luftqualität

Für die Benutzung sind einige Schritte zu beachten. Hierzu gehört zum einen die empfohlene Inbetriebnahme, sowie Interpretation der Messdaten.

Der Luftsensord besitzt vier Anschlüsse, welche in Tabelle 6.1 dargestellt sind. Für unser Gadget ist der vierte Anschluss der entscheidende, denn diesen schließen wir an unseren Eingang an, welcher in der Lage ist, eine Spannung einzulesen bzw. zu messen. In Abhängigkeit der Spannung resultiert eine gewisse Luftqualität, welche wir ermitteln, um diese dann zu interpretieren.

Nr.	Name
1	Versorgung (Vcc)
2	GND (Bezugspotential)
3	Rückmeldung
4	Ausgabe einer Spannung – In Abhängigkeit der Luftqualität

Tabelle 7.1: Inbetriebnahme des Luftsensors – Anschlüsse

Im Folgenden wird darauf eingegangen, wie der MQ135 in Betrieb zu nehmen ist und welche Eigenschaften zu konfigurieren sind. Damit im nächsten Schritt die Interpretation der Messwerte erfolgen kann.

Schritt 1

Nachdem Sie das Gadget eingeschaltet haben und das Programm überspielt haben, ist es notwendig, dass der Luftsensord eine gewisse Zeit bekommt. Dies bedeutet so viel, dass Sie das Gadget 24 Stunden betreiben müssen, damit die Messung vom Sensor geeignete Werte ausgibt.

Schritt 2

Im Anschluss gehen Sie mit Ihrem Gadget an die frische Luft. Wir wissen, dass die Luft draußen ca. 350 ppm bis 450 ppm hat [5]. Dieser Wert ist von dem Ort abhängig an dem du dich befindest, also in der Stadt oder in einem ländlichen Gebiet [5]. Aus diesem Grund gehen wir davon aus, dass die Außenluft 400 ppm hat.

Wenn du draußen mit deinem Multi-MonG stehst, dann nimmst du jetzt einen Wert auf und kannst dem Gadget diesen Wert über die Internetseite mitteilen. Wie in Abbildung 7.1 dargestellt. **Tipp:** du kannst den Messwert auch mit deinem Laptop aufnehmen, indem du mit deinem Laptop und dem Gadget an die frische Luft gehst und den seriellen Monitor nutzt. **Tipp:** Gebe dabei den niedrigsten Wert ein, wenn du die Messwerte 60, 61, 62, 61 usw. siehst, dann entscheide dich für den niedrigeren (60). **Achtung,** denn die eingetragene Zahl kann sich das Gadget nicht merken. Deshalb gibt es die Möglichkeit die Zahl im Programm einzutragen, damit sich das Gadget die Zahl merken kann. In Abbildung 7.2 ist die Stelle eingetragen, an der du die Zahl eintragen kannst, um dann das Gadget Multi-MonG erneut zu programmieren. Den Code findest du fast ganz oben.

Anleitung:

Mit dieser Internetseite kann das Gadget - MultiMonG gesteuert werden.

Modus:

- Zahl 1: Uhr
- Zahl 2: Temperatur
- Zahl 3: Luftfeuchte
- Zahl 4: Timer
- Zahl 5: Laufnachricht
- Zahl 6: CO2 Messwert in ppm
- Zahl 7: Wechselt zwischen Uhr, Temperatur und Luftfeuchte

Nachricht:

Hier kann eine beliebige Nachricht eingetragen werden, welche dann im Modus 5 angezeigt wird

Timer:

Hier kann die Timerzeit in Minuten eingetragen werden

Stunden, Minuten, Sekunden:

Hier kann die Uhrzeit gestellt werden

Messwert - Luft

Trage hier den folgenden Messwert ein, wenn du dich an der frischen Luft befindest: **75**

Schaltflaeche Submit:

Mit dieser kann Ihre Eingabe bestaetigt und an den ESP8266 uebertragen werden

Abbildung 7.1: Internetseite zur Konfiguration des Gadgets



Abbildung 7.2: Eintragen vom Messwert an der frischen Luft

Schritt 3

Die Interpretation der Messwert sollte mit Bedacht erfolgen, denn der Sensor misst mehr als den CO₂ Gehalt in der Luft, liefert aber nur einen Wert. Der Sensor reagiert somit auf folgende Gase; CO₂, Alkohol, Benzol, NO_x, NH₃.

Betrachten wir die Ein- und Ausatemluft des Menschen, dann zeigt sich, dass diese aus den folgenden Bestandteilen besteht [5];

Einatemluft - 78 % Stickstoff und 21 % Sauerstoff

Ausatemluft -78 % Stickstoff, ca. 17 % Sauerstoff und ca. 4% Kohlenstoffdioxid

Diese Eigenschaft machen wir uns zu Nutze, um eine Abschätzung darüber zu treffen, wie hoch der CO₂ Gehalt in der Luft ist, um festzustellen, ob es sinnvoll wäre zu Lüften. Dabei behalten wir die Eigenschaften des Sensors im Hinterkopf, um die Messwerte mit einem geschärften Blick zu betrachten.

Worauf ist zu achte?

Aufgrund der verschiedenen Gase, auf die der Sensor reagiert, muss darauf geachtet werden, in welchem Raum er denn betrieben wird. Hierbei unterscheiden wir solche Räume, wie Büroräume, also solche, welche geringe Gase wie z.B. von Parfümen aufweisen und solchen Räumen, wie wir sie zu Hause haben. Dies liegt darin, dass wir eine Abschätzung über den CO₂ Gehalt treffen möchten.

Aus diesem Grund gebe bitte an, um welchen Raum es sich denn handelt. In Abbildung 7.3 ist die Stelle vom Code dargestellt, an der du entweder eine 1 für Büroräume, eine 2 für private Räume oder eine 3 für Räume mit verschiedenen Düften eingibst.

Den Code findest du fast ganz oben.

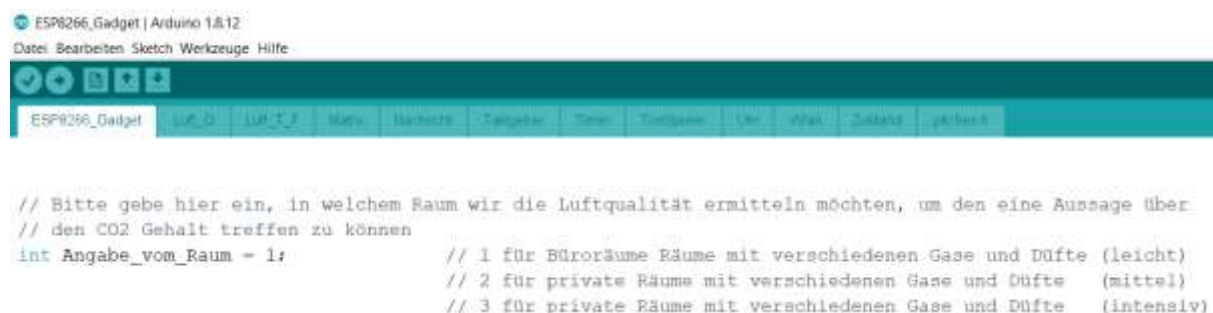


Abbildung 7.3: Angabe vom Raum

8 Schlusswort

Damit mit einem Projekt gestartet werden kann, benötigt es mehrere Schritte. Es beginnt an dem Ort, an dem Gedanken und Ideen kommuniziert und umgesetzt werden. So auch für Ihr Gadget für zu Hause. Denn für dieses haben wir, das SPE Team (Siemens Professional Education) in Karlsruhe, uns zusammengesetzt und zusammengearbeitet. Damit wir dich erfolgreich erreichen und ansprechen können.

Der erste Schritt bestand darin, Ideen zu sammeln, und zu einem sinnvollen Gerät zusammenzusetzen. Somit haben wir ein Brainstorming, und eine Prüfung auf deren Umsetzung durchgeführt. Des Weiteren mussten Platine und das Holzbrett, geplant, sowie erstellt werden. Im ersten Schritt war es wichtig, ein Prototyp zu erstellen und die endgültige Funktionsweise zu testen. Nach Abschluss dieser Phase ging es an die Bestellung der Materialien und der Bauteile.

Zu erwähnen ist, dass wir uns dagegen entschieden haben, das Gadget in eine Schachtel zu packen. Damit wollen wir zum einen, dein Interesse wecken, und zum anderen, dich zum Basteln auffordern. Denn auch in Zukunft wollen wir, über die Github Seite Updates hochladen, mit der du dein Gadget auf dem aktuellen Stand halten kannst.

Die Fähigkeiten Ideen und Gedanken umzusetzen lernt man nicht über Nacht. Aus diesem Grund würden wir uns freuen, dich in den nächsten Jahren bei uns begrüßen zu dürfen. Bis dahin wünschen wir Ihnen eine gute Zeit und viel Erfolg.

#Technik #ist #nicht #langweilig

Ihr Siemens Professional Education Team

#Technik
#ist
#mehr
#als
#eine
#Kiste
#mit
#irgendeiner
#Funktion

#Wirfreuenunsaufdich
#Mintfluencer

Abbildungsverzeichnis

Abbildung 2.1: Darstellung des ESP8266	3
Abbildung 3.1: Aufbau: Bauteile	4
Abbildung 3.2: Gadget: Schaltplan.....	5
Abbildung 3.3: Aufbau: Steckverbinder	5
Abbildung 3.4: Aufbau: Vorderseite.....	6
Abbildung 3.5: Aufbau: Rückseite	6
Abbildung 3.6: Aufbau: DHT11 Sensor	6
Abbildung 3.7: Aufbau: Verbindung der LED-Matrix mit der Platine	7
Abbildung 3.8: Aufbau: Gadget.....	7
Abbildung 0.1: Programmierung - Programmaufbau	9
Abbildung 0.2: Programmierung - Uhrzeit Flussdiagramm	10
Abbildung 4.1: Anleitung: Folge von Schritten.....	11
Abbildung 4.2: Konfiguration: URL.....	11
Abbildung 4.3: Konfiguration: Installieren vom Board.....	12
Abbildung 4.4: Programm: Austausch vom Programmcode	13
Abbildung 4.5: Programm: Einstellen vom verwendeten Board	14
Abbildung 4.6: Konfiguration: Einstellen vom verwendeten Port	14
Abbildung 4.7: Programm: Programmierung - Beispiel.....	15
Abbildung 4.8: Kommunikation: Serial.....	15
Abbildung 4.9: Programm: Entpacken der Datei.....	16
Abbildung 4.10: Entpacken: Hinzufügen der Bibliotheken.....	16
Abbildung 4.11: Programmierung: Öffnen der Datei	17
Abbildung 4.12: Programmierung: Übertragung auf das Gadget.....	17
Abbildung 5.1: Schematische Darstellung: Wechsel zwischen den Funktionen	18
Abbildung 5.2: Internetseite zur Konfiguration des Gadgets	19
Abbildung 6.1: Internetseite zur Konfiguration des Gadgets	21
Abbildung 6.2: Eintragen vom Messwert an der frischen Luft	21
Abbildung 6.3: Angabe vom Raum	22

Tabellenverzeichnis

Tabelle 2.1: Gadget: Funktionen im Überblick	2
Tabelle 3.1: Gadget: Übersicht über die Bauteile	4
Tabelle 0.1: Programmierung - Variablen für die Uhrzeit.....	9
Tabelle 0.2: Programmierung - Variablen für die Uhrzeit.....	9
Tabelle 0.3: Programmierung - Symbole	10
Tabelle 5.1: Internetseite: Überblick über die Funktionen	19
Tabelle 6.1: Inbetriebnahme des Luftsensors – Anschlüsse	20

9 Quellenverzeichnis

- [1] http://www.i6.in.tum.de/pub/Main/TeachingWs2014ProseminarMicrocontrollerEmbedded/Was_ist_ein_Microcontroller.pdf, zuletzt abgerufen am 10.02.2021
- [2] <https://www.brickobotik.de/was-ist-ein-mikrocontroller/>, zuletzt abgerufen am 10.02.2021
- [3] <https://www.questionpro.de/likert-skala-beispiel-umfrage/>, zuletzt abgerufen 10.02.2021
- [4] [CO2 - Bedeutung für die Raumluftqualität \(cik-solutions.com\)](https://www.cik-solutions.com/co2-bedeutung-fuer-die-raumluftqualitaet/), zuletzt abgerufen 10.02.2021
- [5] https://de.wikipedia.org/wiki/Atem#Zusammensetzung_der_Atemluft, zuletzt abgerufen 10.02.2021

10 Änderungsliste – Version 1.1

Die Änderungen sind in der folgenden Tabelle aufgelistet.

Datum	Änderung
22.01.2021	Überarbeitung für das Gadget mit MQ135 - Luftqualitätssensor