

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
4	Inbetriebnahme.....	18
5	Programmierung.....	8
6	Anleitung – Programm in das Gadget hineinladen	11
7	Schlusswort	20
	Abbildungsverzeichnis.....	21
	Tabellenverzeichnis	22
8	Quellenverzeichnis	23
9	Änderungsliste – Version 1.0.....	24

1 Vorwort

Dieses Dokument ist in mehreren Kapiteln gegliedert und soll Ihnen das Gadget (Multi-Monitoring Gadge) 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 dargestellt. 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 ihres Gadgets vorgestellt wird.

2 Einführung

Das Gadget ist eine Station mit insgesamt sechs 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	Zyklisches wechseln (5 Sekunden), zwischen Funktion eins, zwei und drei

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 bewerkstelligende 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.

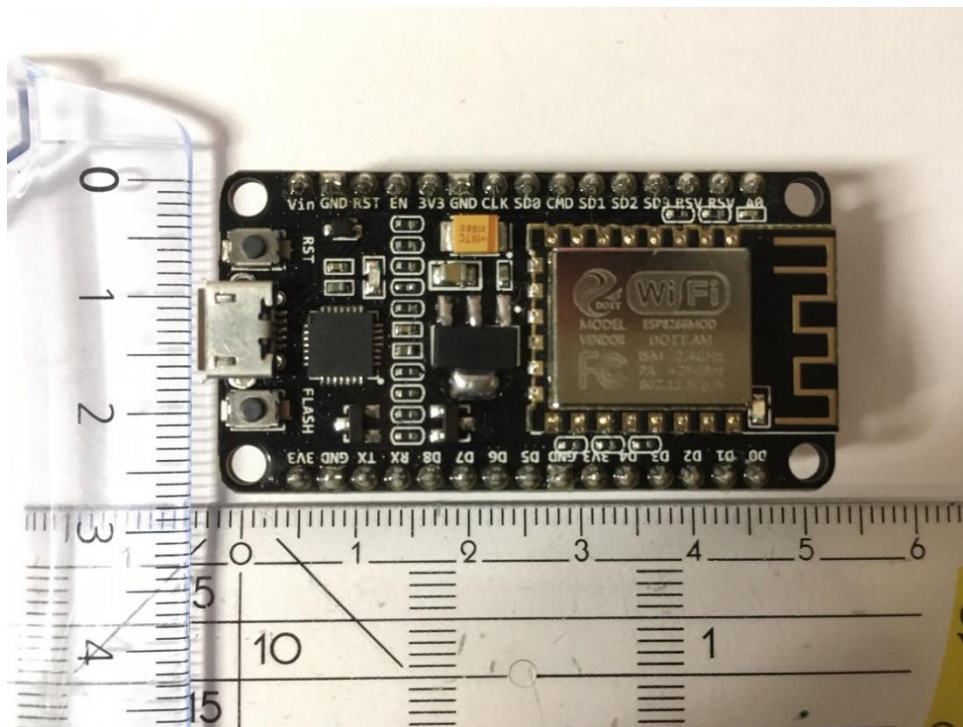


Abbildung 2.1: Darstellung des ESP8266

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]. Der Vollständigkeit, soll erwähnt werden, dass es Computer oder Laptops mit passiver Kühlung gibt.

3 Aufbau

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

Nr.	Bauteil	Nr.	
1	Platine	5	ESP8266
2	Holzbrett	6	DHT11 (Sensor)
3	Taster	7	Konstruktionsteilen
4	Lautsprecher	8	Verbinder

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.

Oben Links ist die Platine abgebildet, rechts davon die Steckleisten und rechts oben, die Verbindungssteckleiste zwischen LED-Matrix und Verbindungssteckleiste.

In der Mitte - links ist der Lautsprecher abgebildet, rechts davon die Taster, danach folgen die Abstandshalter zwischen Platine und LED-Matrix und ganz rechts befindet sich der Temperatur und Luftsensor, DHT11. Ganz unten, wie eindeutig zu erkennen ist, befindet sich die LED-Matrix.

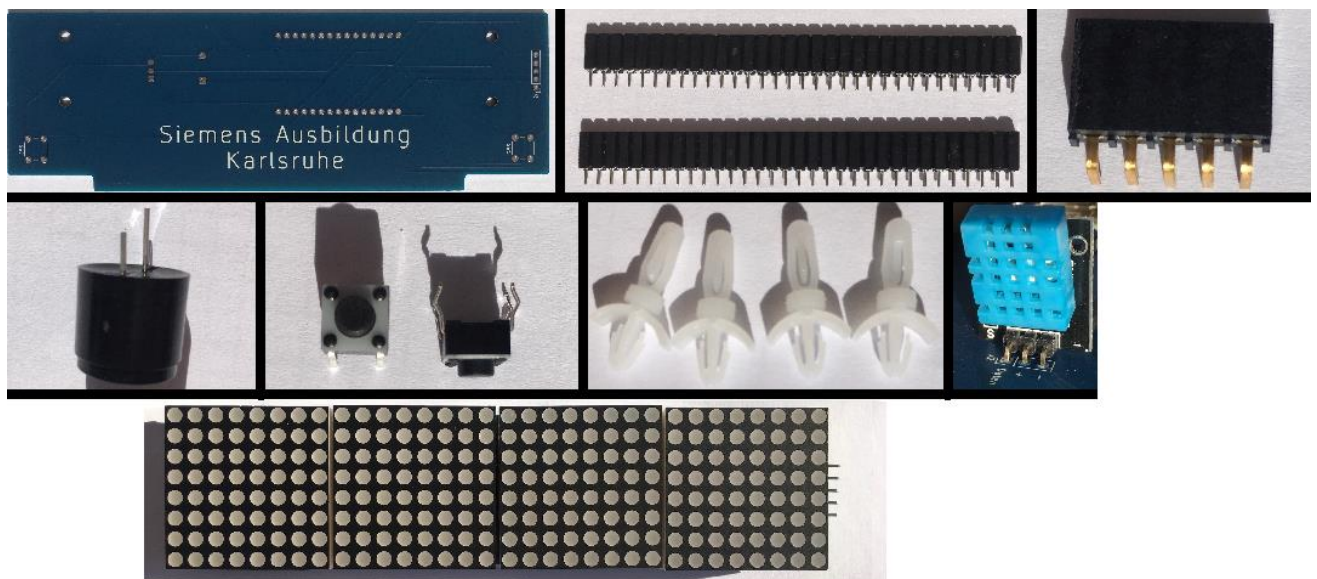


Abbildung 3.1: Aufbau: Bauteile

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 und verlötet werden. Hierbei ist es wichtig, auf die Polarität der Bauteile zu achten, und auf eine angemessene Verlötung wert zu legen, damit jede Verbindung korrekt durchkontaktiert bzw. verbunden ist.

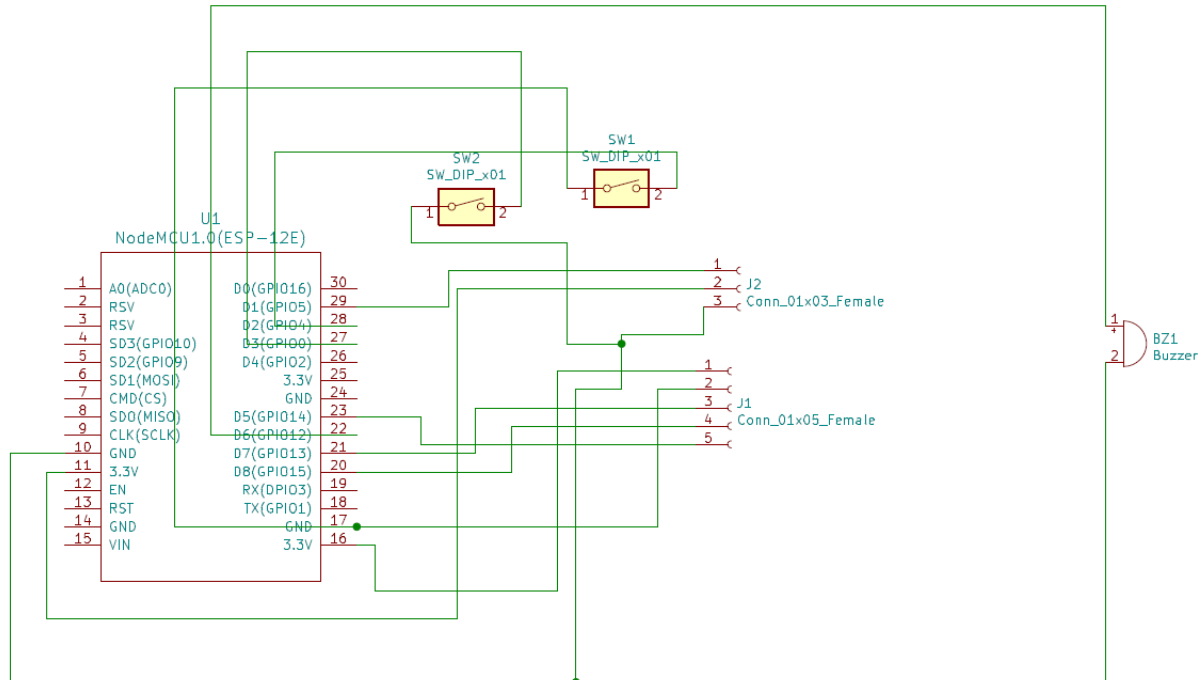


Abbildung 3.2: Gadget: Schaltplan

Als erstes wollen wir die Steckverbinder verlöten. Diese müssen zu Beginn gekürzt werden, und anschließend verlötet werden. Dabei muss darauf geachtet werden, dass diese relativ gerade verlötet werden, damit die Bauteile, und insbesondere der ESP8266 hineingesteckt 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 ebenfalls verlötet werden muss, auch hier muss auf die Polarität geachtet werden. Aus diesem Grund ist dieser in Abbildung 3.6 einzeln 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

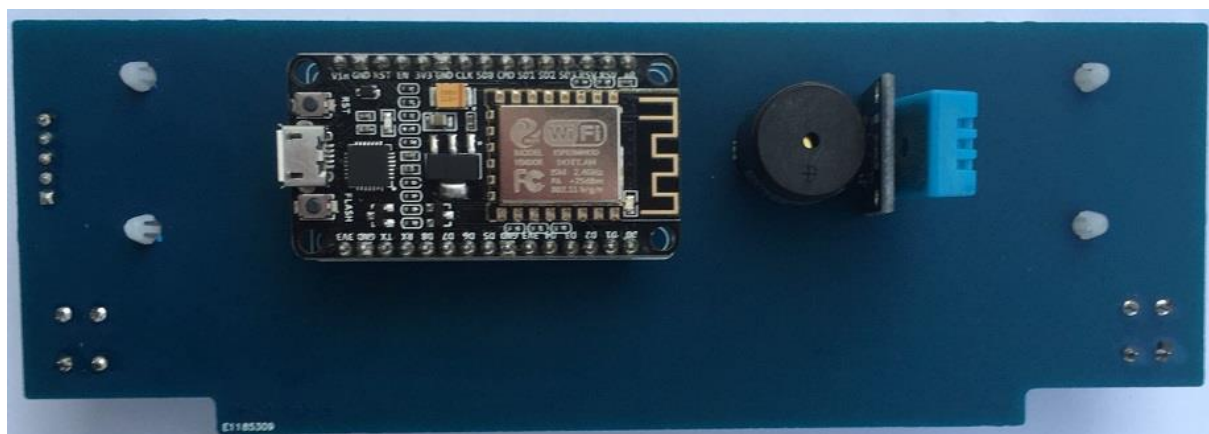


Abbildung 3.5: Aufbau: Hintere Seite

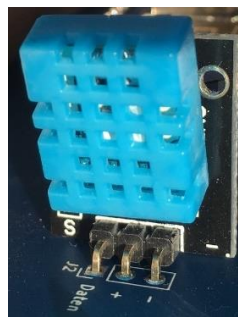


Abbildung 3.6: Aufbau: DHT11 Sensor

damit die Led-Matrix mit der Verbindungssteckleiste eine Verbindung, über die Steckleiste, mit der Platine hat.

Dies ist in Abbildung 3.7 dargestellt. 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 kann, mit der Zuführung der Spannung, das Gadget getestet werden und mit der Erklärung in Kapitel 4 gesteuert werden.



Abbildung 3.7: 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, dass vorhin vorgestellt wurde, vergessen. Somit würde man daraus schließen, dass es sieben Funktionen und das Hauptprogramm gebe. 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 dazu.

Unter dem Strich bedeutet dies, dass es 10 Funktionen und ein Hauptprogramm gibt. Schematisch dargestellt, sieht dies wie in Abbildung 5.1 aus.

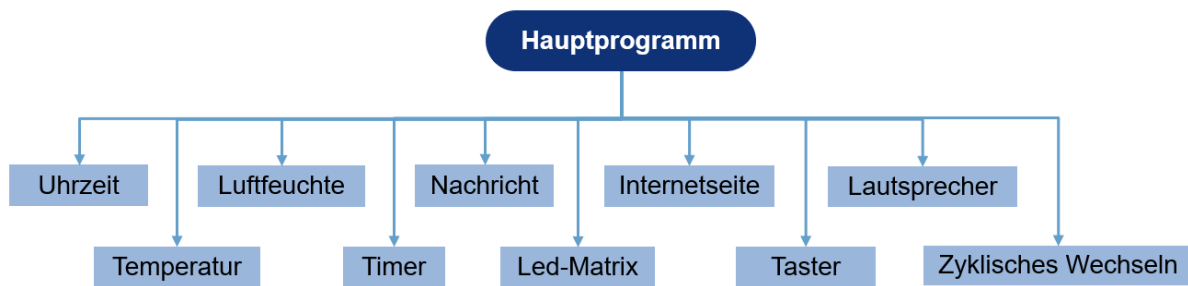


Abbildung 4.1: Internetseite zur Konfiguration des Gadgets

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: 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 nötig.

Nr.	Name der Variable
1	Sekunden_stellen
2	Minuten_stellen
3	Stunden_stellen

Tabelle 4.2: Variablen für die Uhrzeit

Diese Variablen sind dafür, 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 es der Fall sein sollte, dass die Variable `test` kleiner als die Zahl 10 ist, dann wird die Variable `test` um 1 hochgezählt.

```
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: Variablen für die Uhrzeit

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

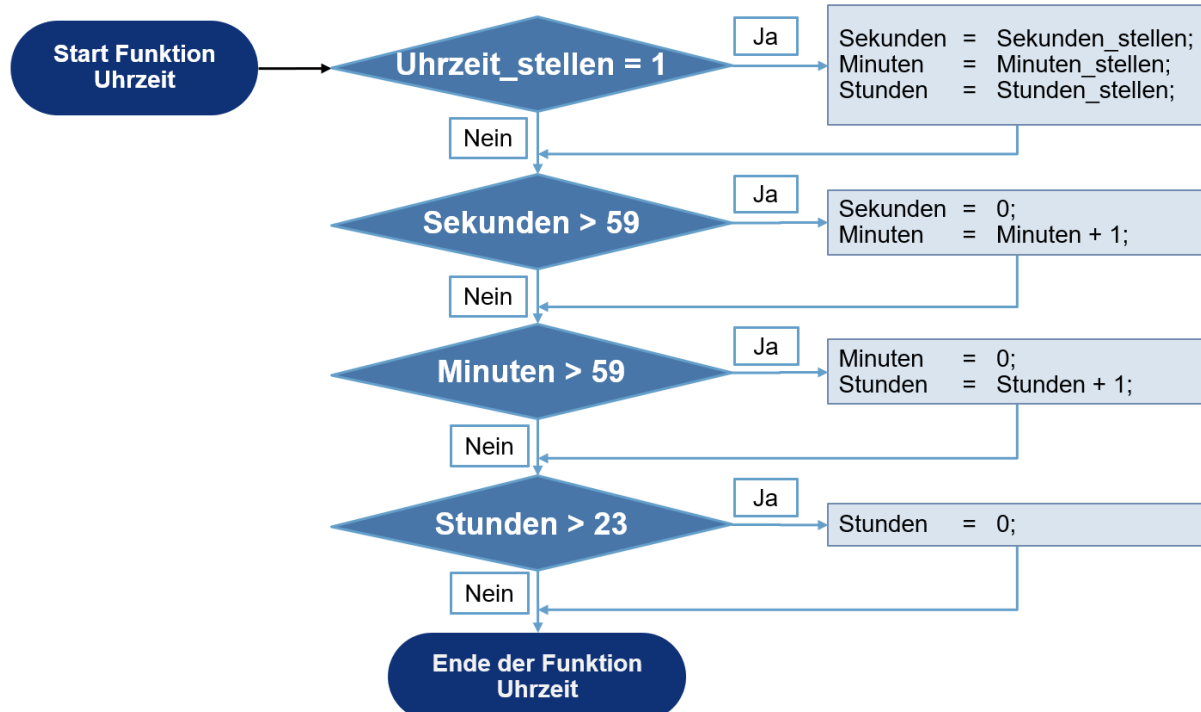


Abbildung 4.2: Funktion: Uhrzeit Flussdiagramm

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.

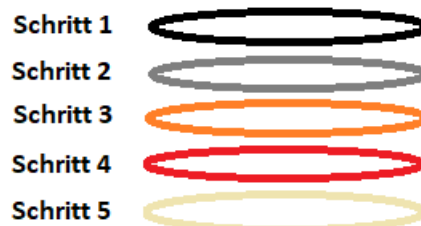


Abbildung 5.1: Anleitung: Folge von Schritten

Teil 1 - Konfiguration zum Hinzufügen vom Board (URL)

- 1.1 Der erste Schritt besteht darin, die Arduino Software zu starten.
- 1.2 Anschließend muss auf Datei (oben links) und dann auf Voreinstellungen.
- 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.

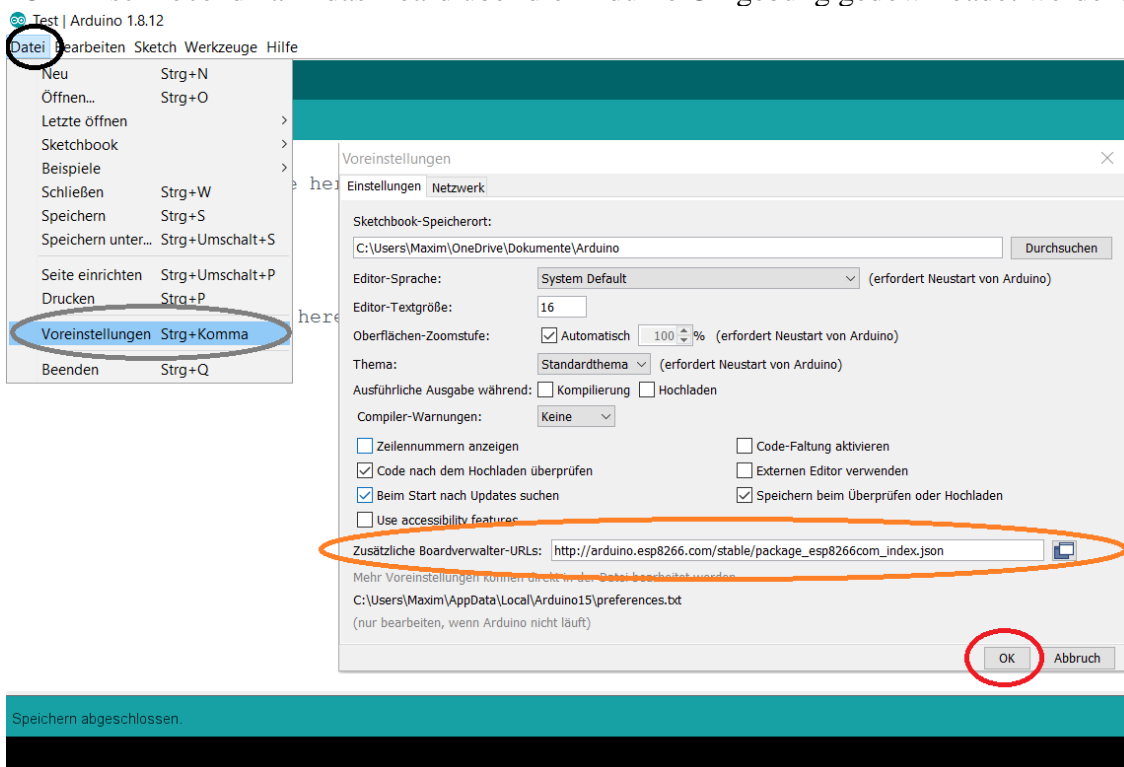


Abbildung 5.2: Konfiguration: URL

Teil 2 - Konfiguration zum Hinzufügen vom Board (Download)

- 2.1 Hierfür muss auf Werkzeuge und dann auf den Punkt Board gegangen werden.
INFO - Es werden die Standard Boards angezeigt.
- 2.2 Anschließend muss der Punkt Boardverwalter ausgewählt werden.
- 2.3 In der suche muss dann ESP8266 eingegeben werden
- 2.4 und die Datei - esp8266 by ESP8266 Community installiert werden.

Wie in Abbildung 6.3 dargestellt.

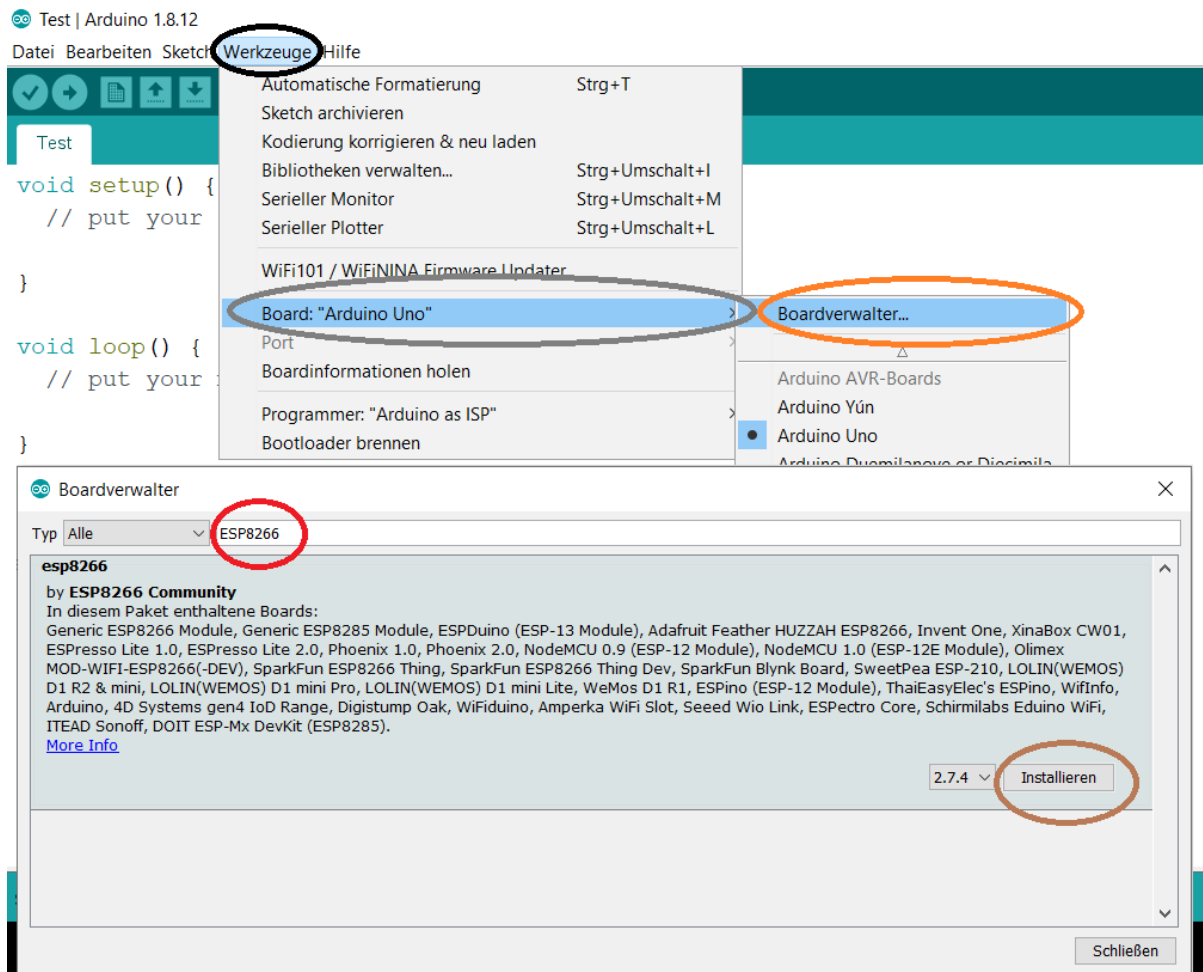


Abbildung 5.3: Konfiguration: Installieren vom Board

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)

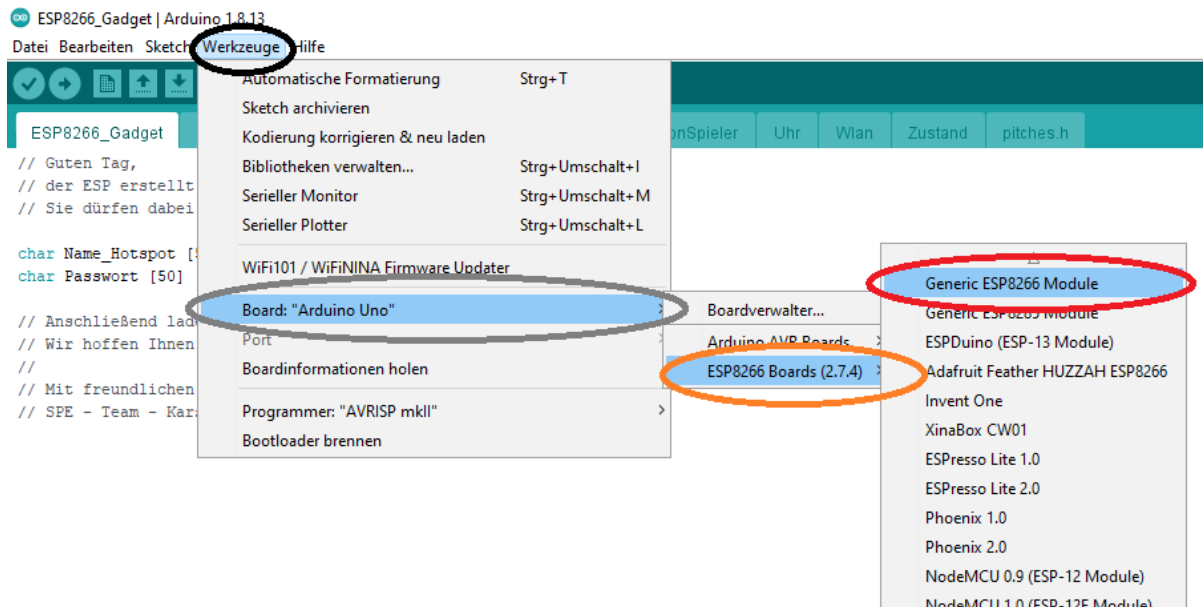


Abbildung 5.5: Programm: Einstellen vom verwendeten Board

3.4 Wählen unter dem Punkt Werkzeuge - Port und dann den richtigen Port aus. (Vermutlich wird nur einer angezeigt) (Abb. 6.6)

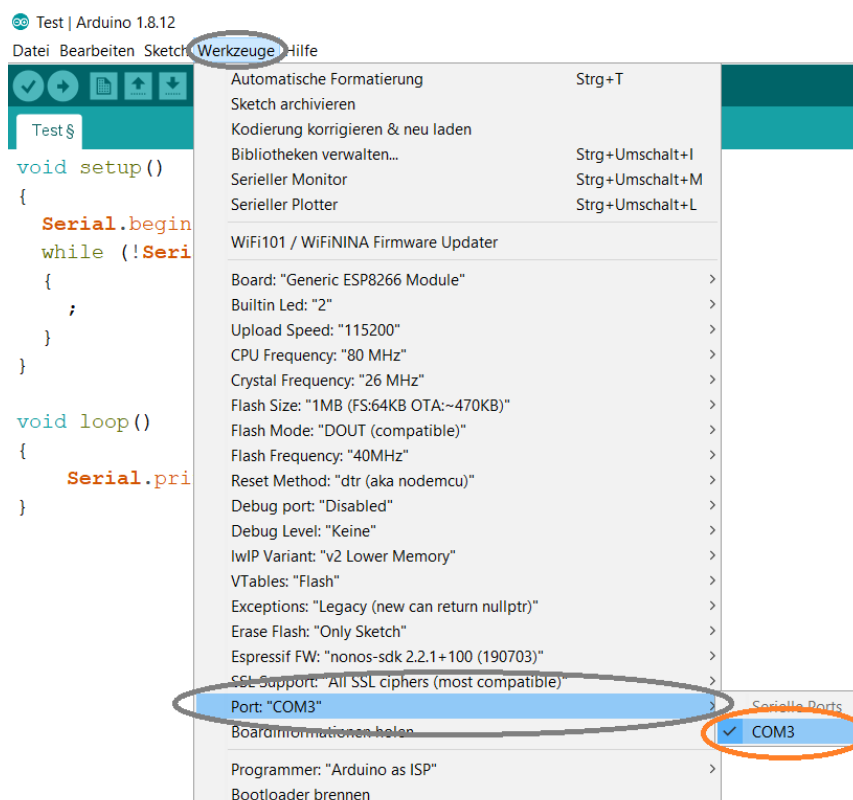


Abbildung 5.6: Konfiguration: Einstellen vom verwendeten Port

- 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)

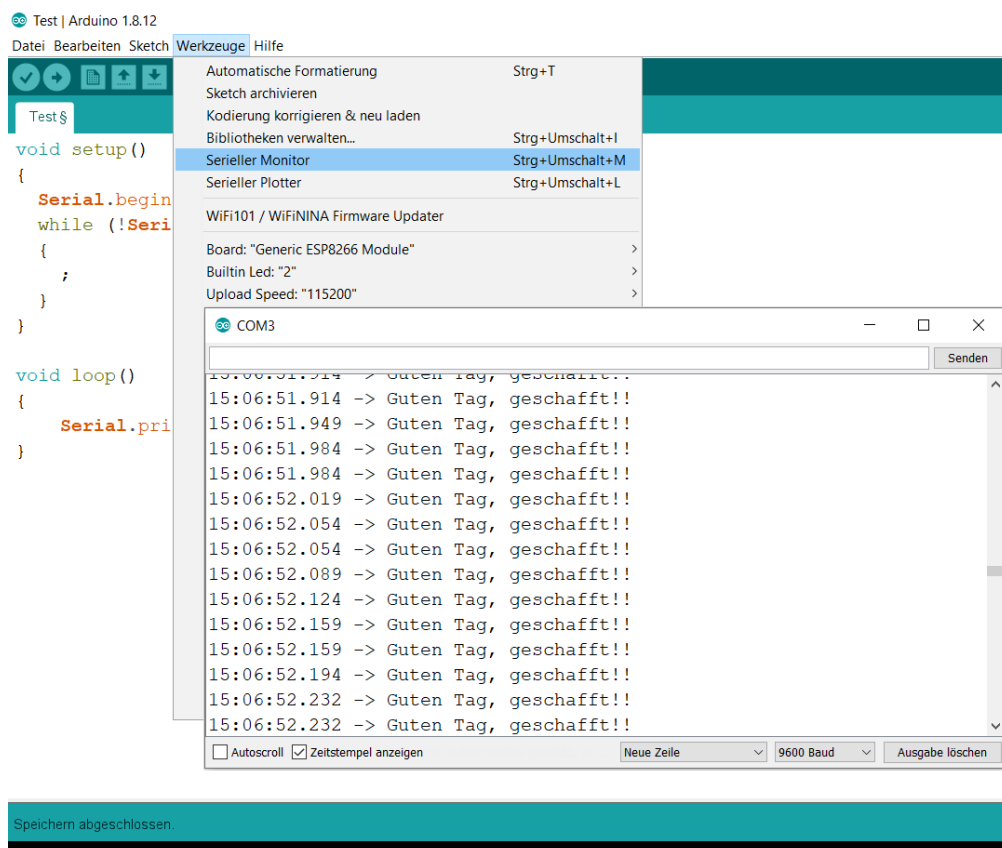


Abbildung 5.8: Kommunikation: Serial

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)

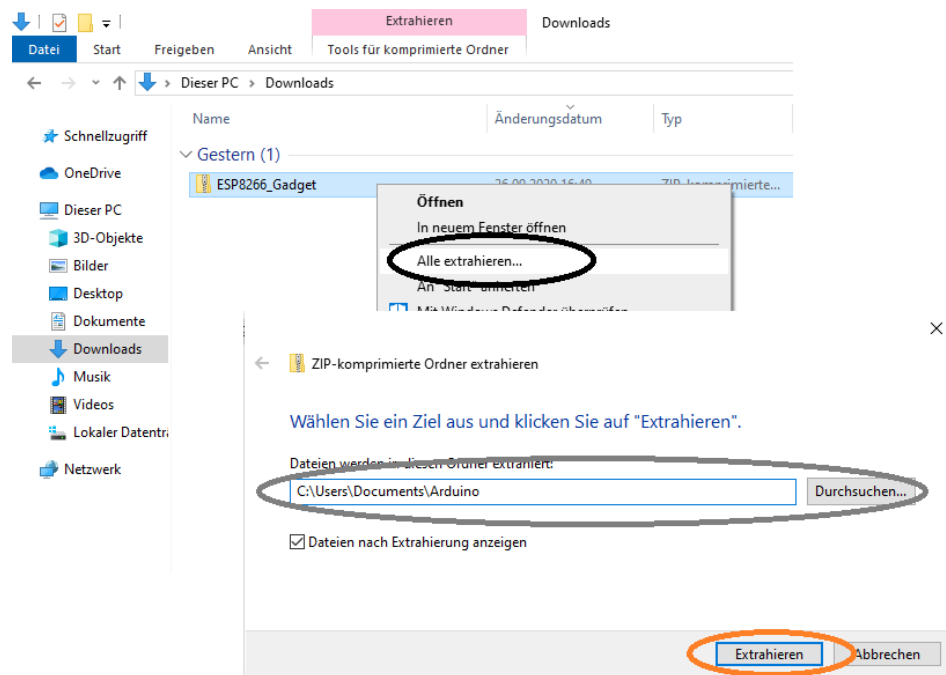


Abbildung 5.9: Programm: Entpacken der Datei

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

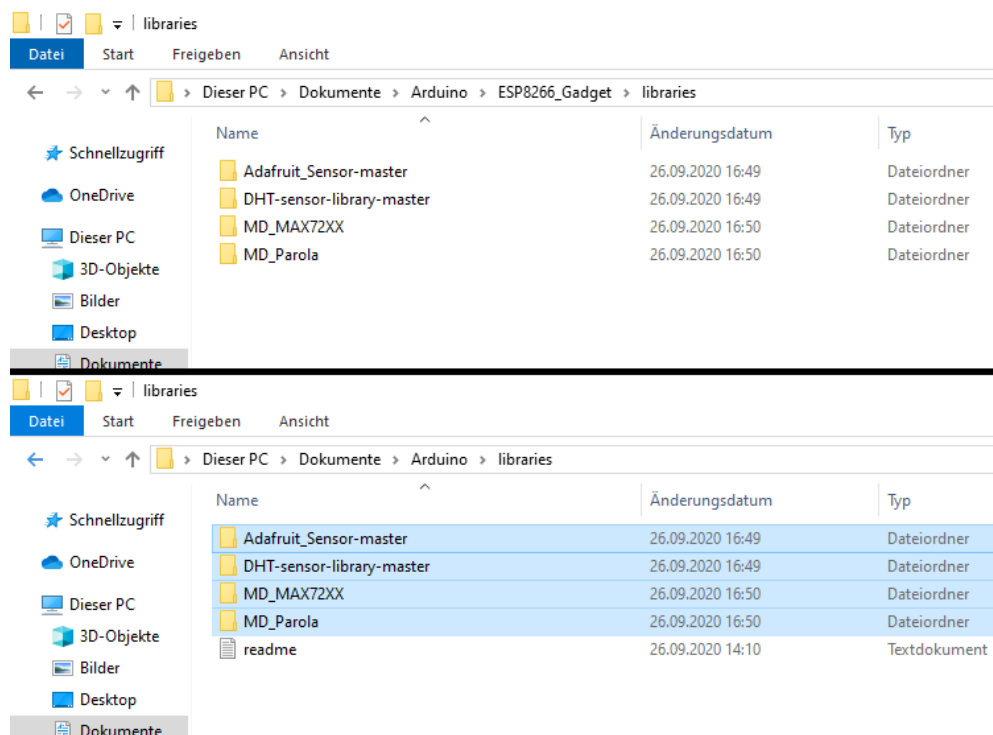


Abbildung 5.10: Entpacken: Hinzufügen der Bibliotheken

Teil 5 – Programmierung der Datei

5.1 Datei öffnen - ESP8266_Gadget. (Abb. 6.11)

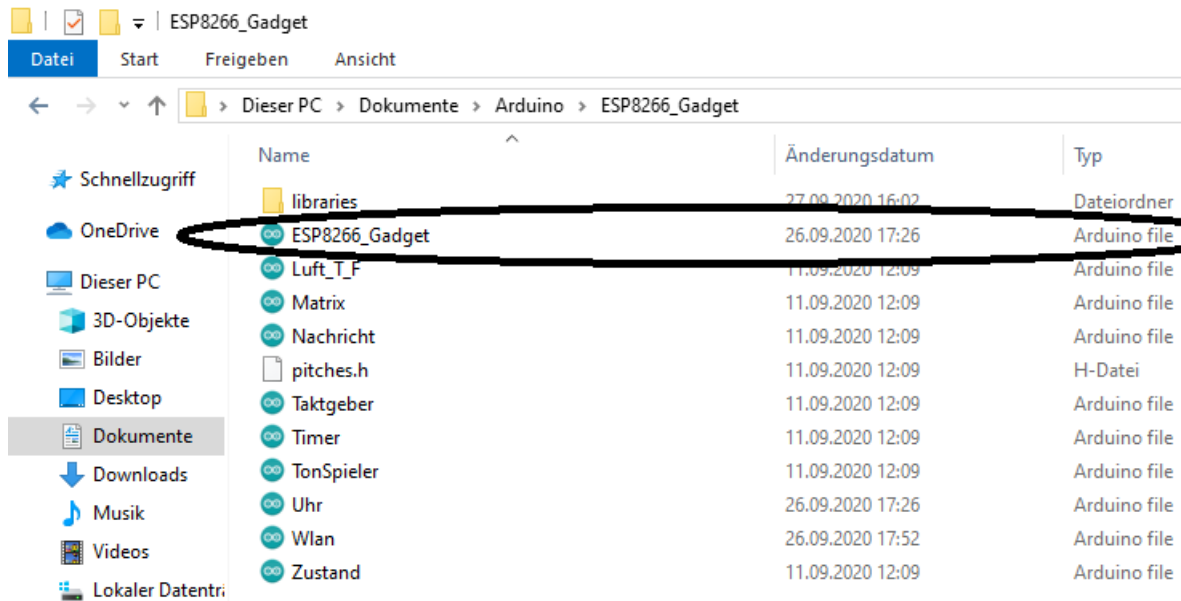


Abbildung 5.11: Programmierung: Öffnen der Datei

5.2 Übertragen auf den ESP8266. (Wie schonmal mit dem Pfeil nach rechts) (Abb. 6.12)

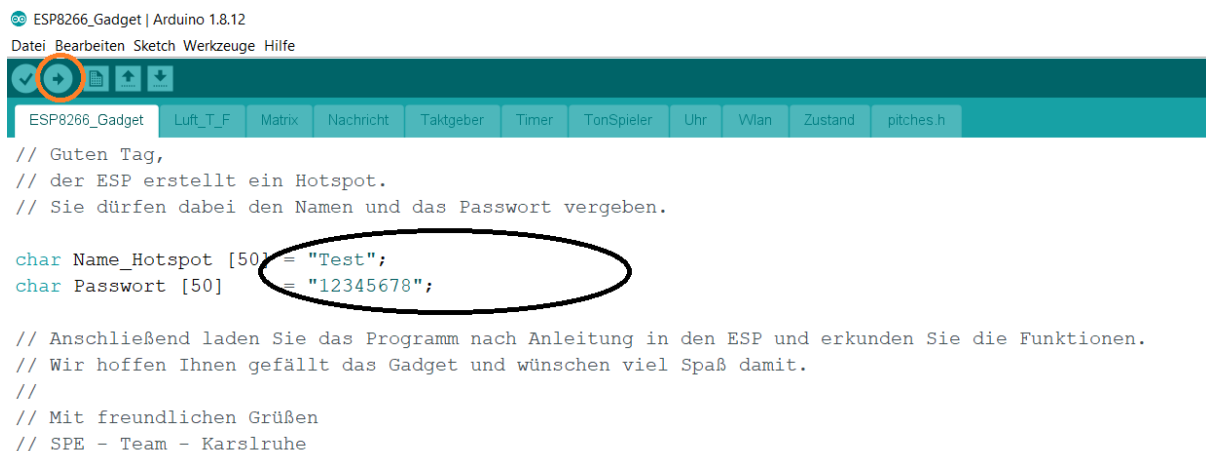


Abbildung 5.12: Programmierung: Übertragung auf das Gadget

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 den Tastern unten rechts und links auswählen. Der Wechsel zwischen den Funktionen findet in Chronologischer Reihenfolge statt. Ab der Funktion sechs, startet es wieder mit der ersten Funktion. In der Abbildung 4.1 ist dies Schematisch dargestellt.

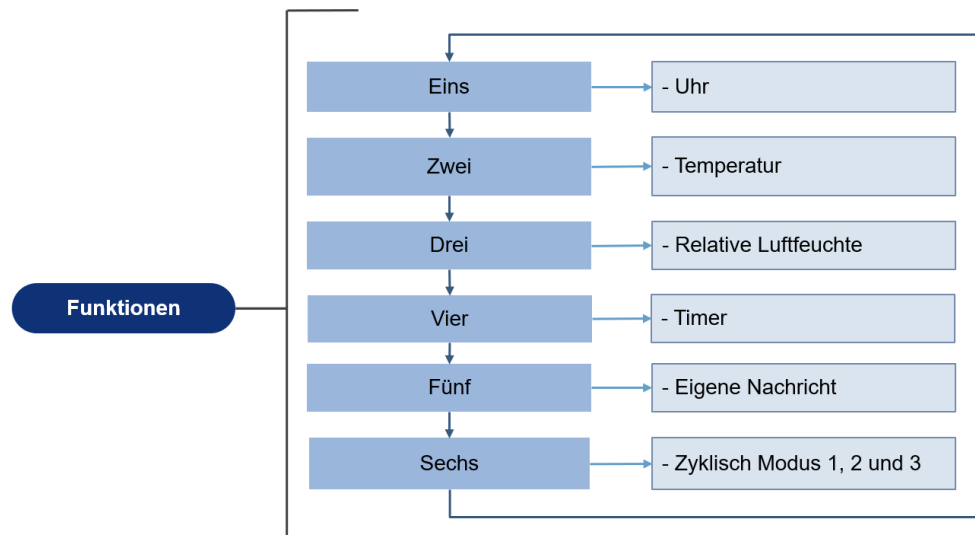


Abbildung 6.1: Schematische Darstellung: Wechsel zwischen den Funktionen

Auf dem ESP8266 befindet sich eine Internetseite, um Zugriff auf diese zu erhalten, 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 - ESP8266 Station

Modus

Timer

Nachricht

Stunden

Minuten

Sekunden

SIEMENS
Ingenuity for life

Anleitung:
Mit dieser Internetseite, kann die ESP8266 Station gesteuert werden.

Modus:
- Zahl 1: Uhr
- Zahl 2: Temperatur
- Zahl 3: Luftfeuchte
- Zahl 4: Timer
- Zahl 5: Laufnachricht
- Zahl 6: Wechselt zwischen Uhr, Temperatur und Luftfeuchte

Timer:
Hier kann die Timerzeit in Minuten eingetragen werden

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

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

Abbildung 6.2: Internetseite zur Konfiguration des Gadgets

Auf der Internetseite sind sechs Textfelder zu erkennen. Die Funktionen dieser Textfelder sind in der Tabelle 4.1 aufgelistet.

Nr.	Name	Beschreibung
1	Modus	Auswahl - Eintragung der Zahlen 1 bis 6, 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

Tabelle 6.1: Internetseite: Überblick über die Funktionen

7 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 können.

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: Hintere Seite	6
Abbildung 3.6: Aufbau: DHT11 Sensor	6
Abbildung 3.7: Aufbau: Gadget	7
Abbildung 4.1: Schematische Darstellung: Wechsel zwischen den Funktionen	18
Abbildung 4.2: Internetseite zur Konfiguration des Gadgets	19
Abbildung 5.1: Internetseite zur Konfiguration des Gadgets	9
Abbildung 5.2: Funktion: Uhrzeit Flussdiagramm	10
Abbildung 6.1: Anleitung: Folge von Schritten	11
Abbildung 6.2: Konfiguration: URL	11
Abbildung 6.3: Konfiguration: Installieren vom Board	12
Abbildung 6.4: Programm: Austausch vom Programmcode	13
Abbildung 6.5: Programm: Einstellen vom verwendeten Board	14
Abbildung 6.6: Konfiguration: Einstellen vom verwendeten Port	14
Abbildung 6.7: Programm: Programmierung - Beispiel	15
Abbildung 6.8: Kommunikation: Serial	15
Abbildung 6.9: Programm: Entpacken der Datei	16
Abbildung 6.10: Entpacken: Hinzufügen der Bibliotheken	16
Abbildung 6.11: Programmierung: Öffnen der Datei	17
Abbildung 6.12: Programmierung: Übertragung auf das Gadget	17

Tabellenverzeichnis

Tabelle 2.1: Gadget: Funktionen im Überblick	2
Tabelle 3.1: Gadget: Übersicht über die Bauteile	4
Tabelle 4.1: Internetseite: Überblick über die Funktionen	19
Tabelle 5.1: Variablen für die Uhrzeit	9
Tabelle 5.2: Variablen für die Uhrzeit	9
Tabelle 5.3: Variablen für die Uhrzeit	10

8 Quellenverzeichnis

- [1] http://www.i6.in.tum.de/pub/Main/TeachingWs2014ProseminarMicrocontrollerEmbedded/Was_ist_ein_Microcontroller.pdf, zuletzt abgerufen am 14.10.2020
- [2] <https://www.brickobotik.de/was-ist-ein-mikrocontroller/>, zuletzt abgerufen am 14.10.2020
- [3] <https://www.questionpro.de/likert-skala-beispiel-umfrage/>, zuletzt abgerufen am 14.10.2020

9 Änderungsliste – Version 1.0