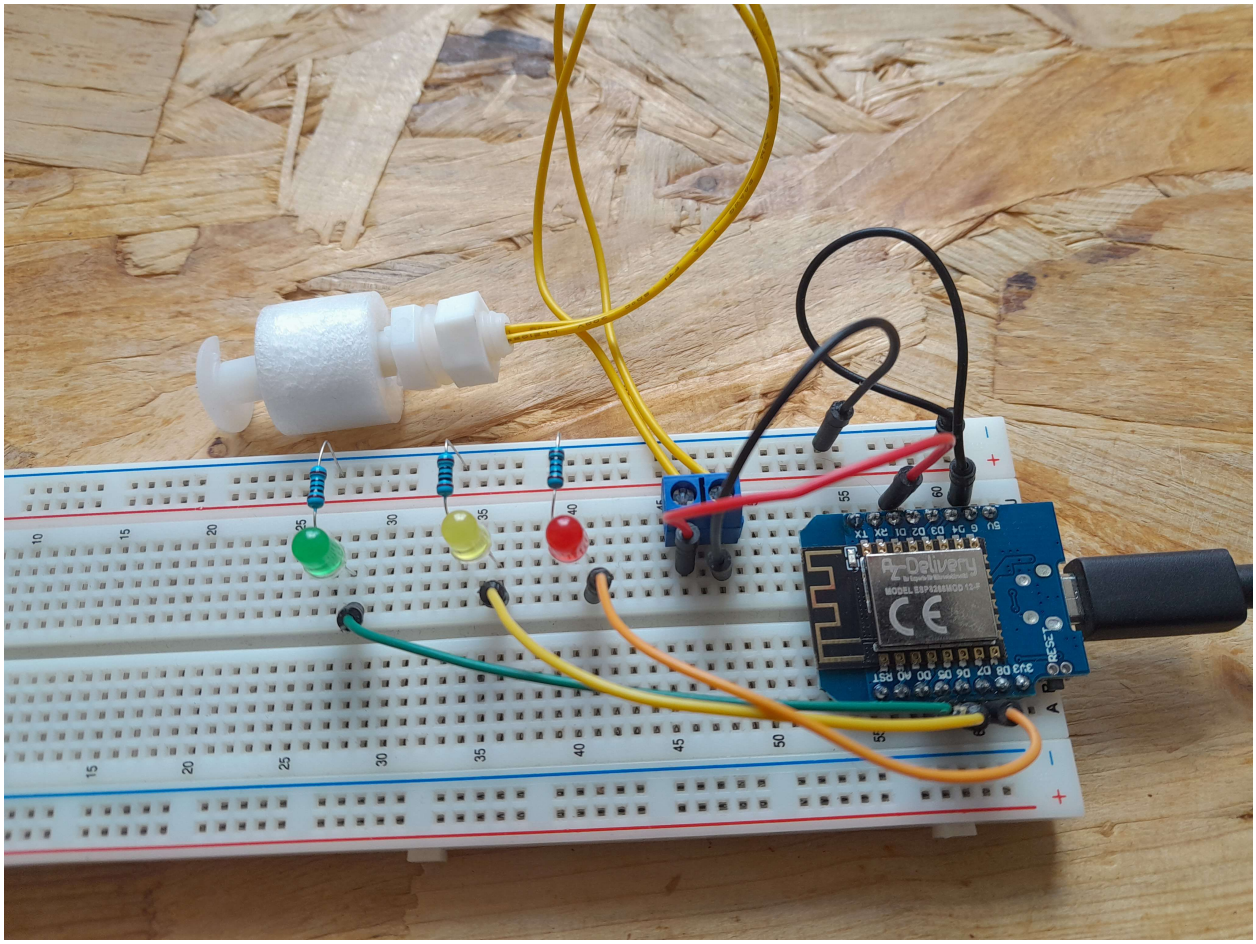
KARREE49

DELPHIN
Projekte gGmbH

GEMEINSAMerLEBEN | GEMEINSAM Stadt entwickeln

Grundkurs Elektronik
Teil 3 – Logiken und Programmierung



Die notwendige Fingerfertigkeit, um die Bauteile einzulöten, haben wir uns im Teil 1 dieses Kurses erworben. Das Hintergrundwissen zu den Bauteilen hat uns Teil 2 vermittelt. Nun kommen wir zu dem spannendsten Teil, einem Grundkurs im Programmieren.

Inhaltsverzeichnis:

3.0. Zielgruppe und Lerninhalte

Dieser Teil des Kurses bietet einen Einblick auf den Übergang zwischen Elektronik und frühzeitlicher Computertechnik.
Am praktischen Beispiel erfahren wir etwas über die Arbeitsweise von Speichern und Mikrocontrollern, indem wir eine einfache Schaltung für die Kontrolle einer Wasserstandshöhe aufbauen und programmieren.
Der Kurs bietet sowohl für Einzelne als auch für Gruppen, Schulen, ausbildende Betriebe oder Bildungsträger, Vereine, Einrichtungen für Menschen mit Behinderungen etc. eine gute Basis für die Berufs – oder Interessenorientierung und kann zur Lehrinhaltsvermittlung genutzt werden.
Aufgrund der oftmals geringen Budgets für Arbeitsmaterial wurden die benötigten Bauteile bewusst so gewählt, dass die Ausgaben erschwinglich (unter 10 €) bleiben und trotzdem eine möglichst große Menge an Wissen in den Unterrichtsstunden vermittelt oder im Selbst-Lern-Verfahren erlernt werden kann.

3.1. Materialliste:

- 1 Experimentierbrett (Breadboard)
- Board ESP32
- 3 Vorwiderstände 100 Ω
- 3 LEDs 5mm (rot, grün, gelb)
- 1 Wasserstandsensor
- 1 Kontaktklemme
- 6 Drahtverbindungen

3.2 Transistoren als binärer Informationsspeicher

Wie wir im Teil 2 erfahren konnten, kann man mit Hilfe von Transistoren mittels einer Basisspannung festlegen, ob der Transistor im Hauptkreis leitend ist oder nicht. So dienen uns die Transistoren als elektronische Schalter.
Ein Schalter, auch z. B. ein Lichtschalter, kann in gewisser Weise auch als Speicher betrachtet werden. Er speichert die Information EIN / AUS. Das ist der kleinstmögliche Wert einer digitalen Information. Man sagt auch **„Bool“** dazu oder dass dieser Wert **binär** ist. Daher kommt auch die Einheit „Bit“ (binary digit).
Ein Lichtschalter hat also genau 1 Bit, entweder Ein oder Aus.

In der Informationstechnik gibt es weitere Ausdrucksweisen, wie zum Beispiel:
- EIN / AUS
- TRUE / FALSE
- HIGH / LOW
- YES / NO
- TOP / DOWN

3.3. Boolesche Algebra, Speicher

Mit Transistoren lassen sich logische Schaltungen aufbauen. (zum Beispiel: UND-Schaltungen, ODER-Schaltungen, „wenn – dann – sonst"-Schaltungen usw.)
Grundlegende Boolesche Operatoren finden wir unter der sogenannten booleschen Algebra. Dazu gibt es in Büchern sowie im Internet zahlreiche Erklärungen. Wer sich für Elektronik interessiert, kommt an der booleschen Algebra nicht vorbei. Es gibt zahlreiche elektronische Bauteile, die Logikschaltungen / Logikgatter beinhalten. In jedem Fall lohnt es sich, einen Blick in die folgenden Grundoperatoren zu werfen:
AND, OR, NAND, NOR, XAND, XOR
Durch die Möglichkeit binäre Informationen zu verarbeiten, lassen sich individuelle „Schaltungen" aufbauen, nutzen und wieder löschen – und das im Bruchteil einer Sekunde.

Wenn wir große Datenmengen speichern wollen, brauchen wir also nichts anderes als viele Transistoren. Deshalb wird seit Langem erforscht, wie man immer mehr Transistoren auf eine immer kleinere Fläche bekommt.
Hat man in Anfangszeiten beispielsweise 10.000 Transistoren in mehrere große Schränke packen und verdrahten müssen, passen seit ein paar Jahren ca. **750 Milliarden Transistoren auf eine Fläche in der Größe eines Daumennagels.**

Auch werden unsere Transistoren immer schneller. Meistens haben sie eine Angabe in MHz, was nichts anderes bedeutet als anzugeben, wie oft man das Bauteil pro Sekunde ein- und ausschalten kann. Beispiel: 200 MHz bedeuten 200 Millionen Schaltvorgänge pro Sekunde. Aufgrund der Verbesserung der Transistoren und deren möglichen Schaltgeschwindigkeiten entstehen immer neue Möglichkeiten.

Wenn wir eine große Menge von Lichtern, wie sie z. B. bei einem Riesenrad verbaut sind, ansteuern wollen, brauchen wir dazu hohe Schaltgeschwindigkeiten. Wir steuern dazu jede Lampe der Reihe nach an und setzen das Bit entsprechend auf 0 oder 1.
Um diese Geschwindigkeiten zu erreichen braucht man einen Takt. Der Taktgeber ist ebenfalls ein elektronisches Bauteil, welches je nach angegebener Taktfrequenz seinen Ausgang auf HIGH oder LOW schaltet. Ein sehr verbreiteter Taktgeber ist der so genannte „Quarzoszillator".

Mithilfe dieser Technik ist der Weg in die Computertechnik geebnet. Wir haben einen Takt und verschiedene logische Bauteile um einen Speicher zu befüllen.

Weitere Elektronische Speicher - „ROM“

In der Elektronik gibt es viele verschiedene Speicher. Einer der frühzeitlichen Speicher ist der „ROM“. Hier ein Auszug der verschiedenen Varianten:

Kurzbezeichnung	
ROM	„Read-Only Memory“ – nur lesbar, fest verdrahtet
PROM	„Programmable Read-Only Memory“ – 1x durch Benutzer beschreibbar, nicht festverdrahtet
EPROM	„Erasable Programmable Read-Only Memory“ – kann elektronisch beschrieben und mit einem UV-Licht wieder gelöscht werden
EEPROM	„Electrically Erasable Programmable Read-Only Memory“ – durch Benutzer elektronisch beschreibbar und löschbar

3.4. Kleiner Exkurs zu Computern

Wie ihr bestimmt schon mal gehört habt, kennen die meisten unserer Computer nur 0 und 1. Das bedeutet, dass zumindest die Architektur der Elektronikebene binär ist. Der Strom kann entweder fließen oder nicht. Oder der Transistor kann EIN oder AUS geschaltet sein. Prinzipiell wären aber auch andere Architekturen möglich. Im Speicherbereich gibt es zum Beispiel auch jede Menge nicht digitale Varianten.

Wir gehen nochmal kurz zurück zu unseren Schaltern.
Wie wir bereits festgestellt haben, hat ein Lichtschalter genau 1Bit. Wir können den Lichtschalter entweder auf EIN (1) oder AUS (0) schalten.

Wir stellen das mal in einer so genannten Wahrheitstabelle dar:

Kombination	Schalter 1
1	0 (AUS)
2	1 (EIN)

Wie man sieht, bietet uns 1 Bit genau 2 Möglichkeiten.

Stellen wir uns jetzt 3 Lichtschalter neben einander vor. An jedem Lichtschalter ist eine Lampe angeschlossen. Wie viele Schaltmöglichkeiten (Kombinationen) ergeben sich dann?

Kombination	Schalter 1	Schalter 2	Schalter 3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Mit 3 Schaltern sind es also schon 8 mögliche Szenarien. (3 Bit = 8 Möglichkeiten).
Durch das Hinzufügen eines Schalters verdoppeln sich jeweils die Kombinationsmöglichkeiten.

Man rechnet deshalb **2 HOCH Anzahl der Schalter**

Beispiel bei 2 Schaltern:
2^3 = 8 mögliche Szenarien

Wie viel Möglichkeiten wären das bei 8 Schaltern?
2^8 = 256 mögliche Szenarien

Und bei 16 Schaltern ?
2^{16} = 65.536 mögliche Szenarien

In der Anfangszeit der Computerentwicklung hatte man sich überlegt, mit wie vielen Bits man auskommt, wenn man damit Zahlen und Buchstaben ausdrücken will. Damals hatte man sich auf 8 Bit fest gelegt und daraus die Einheit Byte gebildet. 1 Byte besteht als aus 8 Bits. Damit jeder Computer die gleiche „Sprache“ spricht, hatte man eine Tabelle erfunden und hinter den möglichen Bitkombinationen die jeweilige Zahl, den Buchstabe oder das Zeichen festgelegt.

Diese Tabelle heißt ASCII-Code-Tabelle. (American Standard Code for Information Interchange)

Hier ein kleiner Auszug:

Binärwert (8 Bit / 1 Byte)	Bedeutung in der ASCII-Tabelle
00000001	1
00000010	2
00000011	3
01100001	a
01000101	E

Wenn wir also irgendwo ein E speichern und dafür die ASCII Tabelle nutzen, wird irgendwo in unserem „Speicher“ die Information 01000101 festgehalten.

Auch schon der einzelne Druck der Taste „E“ erzeugt mit Hilfe eines Controllers (das ist ein elektronisches Bauteil mit einer integrierten Schaltung in der Tastatur) diesen Binärcode.
Über das Kabel wird diese Information dann an unseren Computer übertragen.

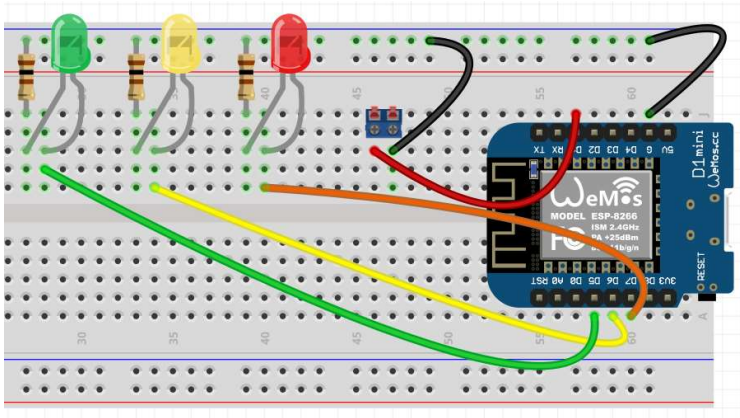
Um Binärcode kürzer darstellen zu können, werden die Daten als Hexadezimale Zahlen dargestellt. Da Hexadezimale Zahlen die Basis 16 haben, kann man die binären Daten schnell und vereinfacht darstellen. So kann man statt beispielsweise das Wort „hallo“ (ASCII-Code) entweder binär: „01001000 01100001 01101100 01101100 01101111“
oder hexadezimal: „48 61 6c 6c 6f“ darstellen.

Wir sehen also, dass es vereinfachte Darstellungsformen gibt. Auch die Sprachen in denen programmiert wird, sind vereinfachte, für den Menschen lesbare Formen. Dahinter stecken nach zahlreichen Ebenen der „Übersetzung“ am Ende wieder 0 und 1.

3.5. Experimentierboard mit Mikrocontroller, LEDs und Sensor bestücken

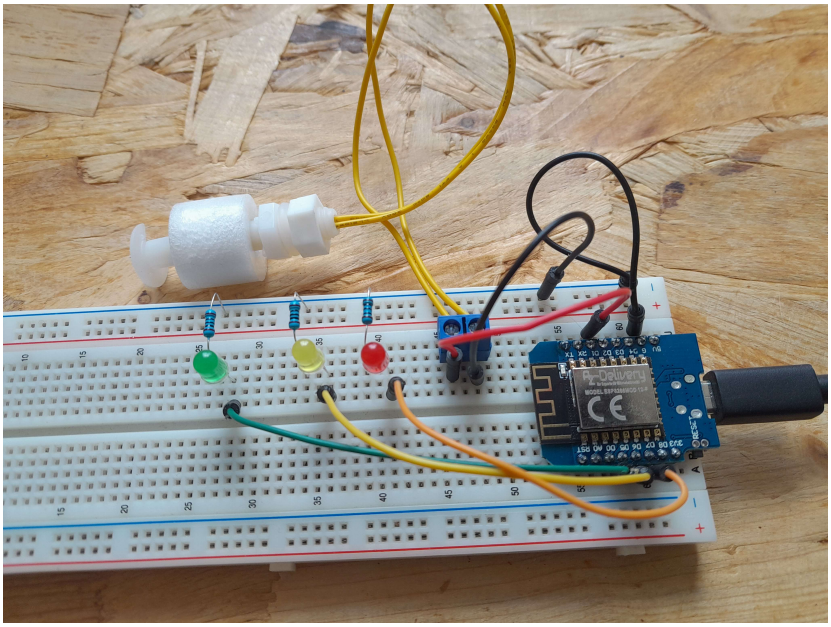
Jetzt, nachdem wir uns ein gewisses Grundverständnis für die Elektronik und Computerarbeitsweise angeeignet haben, wenden wir uns wieder etwas Praktischem zu. Wir löten einen Mikrocontroller (das ist ein kleiner Computer mit einigen integrierten Bauteilen) auf eine Platine und steuern mehrere LEDs an. Zusätzlich installieren wir einen Sensor.
Es gibt zahlreiche fertige Boards mit Mikrocontroller. Wir haben uns hier für das Board ESP32 der Firma Espressif Systems entschieden, weil dieser sehr verbreitet ist und es dafür zahlreiche Anwendungsmöglichkeiten gibt. So hat das Board beispielsweise WLAN, Bluetooth und jede Menge weitere Features.

Schauen wir uns das Board mal an. Es hat jede Menge Kontaktpins. Wir benötigen nur ein paar davon um die LEDs zum leuchten zu bringen und mit dem Sensor kommunizieren zu können. Da das Board glücklicherweise über einen USB-Anschluss verfügt, brauchen wir uns nicht um eine Stromversorgung zu kümmern und können das Board stattdessen über den USB-Anschluss unseres PCs mit Strom versorgen.
Wir stecken unsere LED’s, Vorwiderstände und unseren Wasserstandsensor nach dem folgenden Schaltbild auf das Steckbrett:



Diese Abbildung haben wir mit der Software „fritzing“ erstellt. Zu finden unter „https://fritzing.org“

An die blaue Klemme schließen wir einfach unseren Wassersensor an:



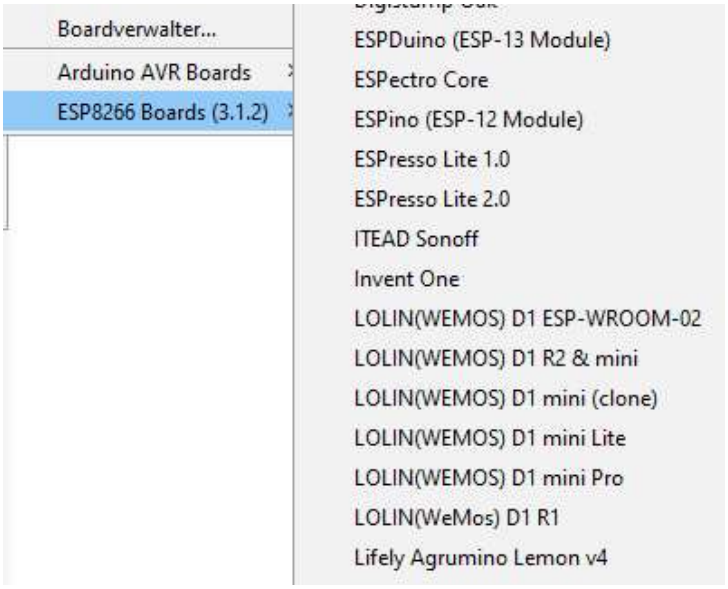
3.6. Programmierbeispiel

Um den Code zu erstellen und auf das Board zu übertragen, laden wir uns das Programm „Arduino IDE“ auf unseren PC herunter und installieren es:
https://docs.arduino.cc/software/ide-v1/tutorials/Windows

Die Arduino Software arbeitet mit zahlreichen Boards zusammen. Damit die Software unser Board kennt, müssen wir ihr die entsprechenden Informationen für das Board bereitstellen. Wir navigieren dazu in das folgende Menü:
Arduino IDE→ Datei → Voreinstellungen:

Dort kopieren wir in die Zeile „Zusätzliche Boardverwalter -URLs:“ den folgenden Link:
https://arduino.esp8266.com/stable/package_esp8266com_index.json
und klicken anschließend auf OK.
Jetzt navigieren wir zu: Werkzeuge→ Board → Boardverwalter und suchen nach „lolin“. Wir installieren das neueste Paket:

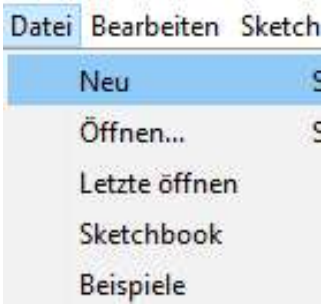
Nun steht unser Board „LOLIN (WEMOS) D1 R2 & mini“ zur Verfügung:



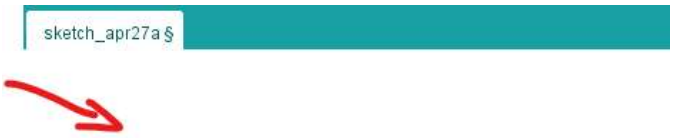
Jetzt legen wir nur noch den Port fest, über welcher unser Board angeschlossen ist. Dazu navigieren wir auf Werkzeuge → Port. Wenn wir nicht wissen welcher Port das ist, probieren wir ihn einfach aus.

Unter Windows kann man auch den Port des Gerätes über den Gerätemanager heraus finden.

Jetzt gehen wir auf Datei → Neu



Wir löschen den vom Programm generierten Code. Das neue Fenster muss leer sein. Danach markieren wir den nachfolgenden Code auf der Folgeseite (komplett) und fügen ihn in das neue Fenster ein.



//Code Beginn:

```
//Die Pinbezeichnungen auf dem Board sind nicht gleich
bezeichnet
//mit den Nummern zum Ansteuern. Hier die Übersicht:
//D1 = 5,  D2 = 4, D3 = 0, D4 = 2, D0 = 16,   D5 = 14, D6 = 12
//D7 = 13    D8 = 15

//Grundeinstellungen
void setup() {
 // Starten der Kommunikation mit unserem seriellen Monitor mit
 //einer Rrate von  115200 Baud
 //(Baud ist die Einheit für die Symbolrate - Symole je
Sekunde):
  Serial.begin(115200);

  //Festlegen, dass unsere PINs Ausgänge sind:
  pinMode(14, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  //Bei Pin 5 mit PULLUP:
  pinMode(5, INPUT_PULLUP);
}

// Hauptschleife, die sich immer wiederholt
void loop() {
   //Variable „Wasser" des Types Integer festlegen

  int wasser=digitalRead(5);    //und den Pin 5 auslesen

  // WENN keine Strom am PIN 5 fließt:
  if (wasser == LOW){
        Serial.println("Wasserstand OK");
        //Wir schalten die grüne LED ein
```

```
        digitalWrite(14, HIGH);
  }
  // WENN Strom am PIN 5 fließt:
  if (wasser == HIGH){
        Serial.println("Wasserstand KRITISCH");
        //Grüne LED AUS
        digitalWrite(14, LOW);
        //GELBE LED EIN
        digitalWrite(12, HIGH);
        delay(500);  // warte 500 Millisekunden
        digitalWrite(12, LOW);

        //und nochmal mit der roten LED
        digitalWrite(13, HIGH);
        delay(500);  // warte 500 Millisekunden
        digitalWrite(13, LOW);
  }
}
//Code Ende
```

Wir klicken auf das Symbol „Hochladen“. Dadurch wird der Code compiliert. (der im Programm eingebaute Compiler übersetzt unseren Programmiercode in Maschinencode (Binärcode 0 und 1).
Wenn wir den richtigen COM-Port ausgewählt haben, wird unser Code auf das Board übertragen.

Wir klicken auf „Werkzeuge“ → Serieller Monitor. Ein Kommunikationsfenster öffnet sich. Somit können wir sehen, was das Board gerade entsprechend unserem Code meldet.

In diesem Beispiel wird permanent der Wasserhöhensensor abgefragt, was wir über den seriellen Monitor beobachten können. Die grüne LED leuchtet permanent. Sobald jedoch das Wasser steigt und den Schwimmer des Sensors nach oben treibt, blinken die gelbe und die rote LED abwechselnd.

Jetzt können wir mit dem Code etwas herumspielen, weitere LEDs hinzufügen die LEDs schneller oder langsamer blinken lassen usw… Je mehr wir experimentieren, desto mehr werden wir mit dem Code vertraut.

3.7. Jede Menge Sensoren und jede Menge Möglichkeiten

In diesem Teil des Kurses haben wir grundlegende Dinge kennen gelernt, die uns helfen ein gewisses Verständnis für die Zusammenhänge zu bekommen.
Es gibt jede Menge weitere interessante Bauteile, die wir teilweise für wenig Geld erhalten können. Wir können beobachten, dass in vielen, teilweise sehr teuren Fertigsysteme mitunter die gleichen Sensoren verbaut sind, die man im Internet für ein paar Euros kaufen kann.

Haben wir ein schönes System entwickelt, können wir nun auch die Bauteile fest auf einer Platine verlöten. Dank der in Teil 1 erlernten Fähigkeiten sollte das für uns kein Problem mehr darstellen. Auch können wir uns Gehäuse dazu kaufen oder bauen und so unser Bauteil schützen. Sollten uns die PINs auf dem Board ausgehen, weil wir beispielsweise 10 LEDs anschließen wollen, gibt es dafür sogenannte „Schieberegister“. Wenn man sich mit deren Funktion vertraut macht, lernt man wieder etwas mehr hinzu.

Übrigens kann man dieses oder andere Boards internetfähig machen. Dazu aktiviert man auf diesen Boards mittels Code das integrierte WLAN-Modul, installiert einen kleinen Webserver darauf und stellt beispielsweise eine Verbindung mit dem eigenen Smartphone her. So könnte man beispielsweise von seinem Urlaubsort aus verschiedene Daten hin und her senden, seine Beleuchtungen einschalten, Pflanzen gießen, die eigene Katze oder den Hund orten und noch vieles mehr. Der Fantasie ist hierbei keine Grenzen gesetzt.

Nun gutes Gelingen und viel Erfolg und Freude mit dem neu erlernten Wissen.

3.8. Andere Bauteile für weitere Visionen

Hier noch ein paar weitere schöne Bauteile, die man für eigene Projekte nutzen kann.

Sensoren / INPUT- Geräte:

- Temperaturfühler für Flüssigkeiten oder Luft
- Gassensoren, Luftdruck und Luftfeuchte
- Lichtsensoren, Helligkeitssensoren, Lichtschranken, Infrarotsensoren, Bewegungsmelder
- Drehzahlmesser
- Mikrofone, Kameras
- Erschütterungssensoren, Neigungssensoren, Klopfsensoren, Hindernissensoren
- Magnetsensoren
- Herzschlagsensoren
- PH-Wert Sensoren, Sauerstoffsensoren
- Nässesensoren

Ausgabeeinheiten:

- Displays, Touchscreens
- Relaiskarten zum Schalten von großen Leistungen
- Sirenen, Lautsprecher

Sonstige:

- Funk wie WLAN, LoRa oder Bluetooth, GPS
- Ultraschall Abstandsmesser

Im KARREE49 werden die meisten der oben genannten Sensoren bei der Steuerung unserer Aquaponik-Anlage eingesetzt. Oftmals haben wir diese aus Kostengründen selbst zusammengebaut. Ihr seht also, die Anwendung der in diesem Kurs vermittelten Inhalte hat für uns eine große Bedeutung. Gerne könnt ihr euch mit uns in Verbindung setzen, wenn ihr euch am Aufbau einer einfachen Steuerung beteiligen wollt.