

# JUGEND FORSCHT 2021

INTERNET  
OF STUPID  
THINGS



Von Niklas Krieger und Mirko Weih

aus der Ludwig-Geißler-Schule in Hanau

in Betreuung durch

Herr Dr. Martin Löffler und Herr Fabian Bott

## KURZFASSUNG

**Projektname:** Internet Of Stupid Things

| <b>Teilnehmer: Name (Alter)</b> | <b>Ort</b> | <b>Schule/Institution/Firma</b> |
|---------------------------------|------------|---------------------------------|
| Niklas Krieger (17)             | Bad Orb    | Ludwig-Geissler-Schule Hanau    |
| Mirko Weih (17)                 | Hainburg   | Ludwig-Geissler-Schule Hanau    |

**Projektbetreuung:** Martin Löffler, Fabian Bott

**Erarbeitungsort:** Ludwig-Geißler-Schule Hanau

---

Mit unserem Projekt wollen wir herausfinden, was man in öffentlichen Gebäuden alles digitalisieren kann, ohne neue Gerätschaften mit entsprechenden Schnittstellen anschaffen zu müssen. Wegen der aktuellen Covid-19-Pandemie haben wir uns dabei vor allem auf die Hygiene in Räumen spezialisiert. Hierbei geht es darum, den Füllstand von Papier- oder Seifen- bzw. Desinfektionsmittelpender an ein elektronisches System weiterzugeben, damit einfacher zu erkennen ist, wann und wo diese nachgefüllt werden müssen. Dazu verwenden wir Mikrocomputer mit verschiedenen Sensoren. Im weiteren Verlauf unserer Entwicklung wollen wir außerdem auch noch weitere analoge Zustände wie offene Fenster oder Statuslampen in Räumen einbinden. Darüber hinaus versuchen wir unser Produkt bzw. unsere Produkte so zu verkleinern, dass sie überall verwendbar und vielseitig einsetzbar sind und deren Akkubetrieb so effizient wie möglich zu gestalten, damit nur selten ein Akkuwechsel bzw. eine Akkuladung notwendig ist.

## INHALTSVERZEICHNIS

|   |    |
|---|----|
| <b>EINLEITUNG</b> .....   | 3  |
| Möglichkeiten, IoT-Systeme aufzubauen .....                                 | 3  |
| Herstellerspezifische IoT-Systeme .....                                     | 3  |
| Herstellerübergreifende IoT-Systeme .....                                   | 4  |
| IoT-Sicherheit .....  | 4  |
| Praktische IoT-Anwendung .....  | 5  |
| <b>FORSCHUNG</b> .....  | 6  |
| Hardware .....  | 6  |
| Software .....  | 7  |
| Sensorsuche .....   | 8  |
| Überleitung .....   | 9  |
| Stromversorgung & Gehäuse .....   | 9  |
| System der TellMe's .....   | 10 |
| <b>WIE GEHT ES WEITER?</b> .....  | 11 |
| <b>Quellenverzeichnis</b> .....   | 13 |
| <br>  |    |
| Abbildung 1: Anwendungsgebiete von IoT-Systemen im Smart-Home .....         | 3  |
| Abbildung 2: AHA-Regeln .....   | 5  |
| Abbildung 3: Handelsüblicher Papiertuchspender .....                        | 5  |
| Abbildung 4: Mikrocomputer .....  | 6  |
| Abbildung 5: WLAN-Access-Point aus dem Schulgebäude .....                   | 7  |
| Abbildung 6: Arduino IDE .....  | 8  |
| Abbildung 7: Hardware-Versuchsaufbau .....                                  | 8  |
| Abbildung 8: Versuchsaufbau mit Infrarot-Abstandssensor .....               | 8  |
| Abbildung 9: Serieller Monitor der Arduino IDE .....                        | 9  |
| Abbildung 10: Stromversorgung des ESPs mit Ultraschall-Abstandssensor ..... | 10 |
| Abbildung 11: Anwendungsentwurf .....                                       | 12 |

## EINLEITUNG

IoT und vernetzte Systeme sind unsere Zukunft. IoT steht als Abkürzung für „Internet of Things“, was übersetzt so viel wie „Internet der Dinge“ heißt. Unter solch einem Netzwerk versteht man die Vernetzung von Gegenständen aus unterschiedlichen Bereichen.

Beispiele hierfür sind: Eine Heizung, die mit Hilfe eines intelligenten Thermostates automatisch die Raumtemperatur anpasst, oder ein intelligenter Briefkasten, der eine Statusmeldung auf das Mobiltelefon sendet, sobald er vom Briefträger geöffnet wurde.

Häuser und Wohnungen, deren Geräte genau durch solch ein System vernetzt sind, nennt man neudeutsch auch „Smart Home“. Bei der Vernetzung von Haustechnik (z.B. Heizung oder Lichtquellen) und Haushaltsgeräten (bspw. Kühlschrank oder Waschmaschine) sowie von Komponenten der Unterhaltungselektronik strebt man eine Erhöhung der Wohn- und Lebensqualität, der Sicherheit und einer effizienten Energienutzung an.



Abbildung 1: Anwendungsgebiete von IoT-Systemen im Smart-Home

### Möglichkeiten, IoT-Systeme aufzubauen

Heute gibt es eine Vielzahl von Herstellern, die IoT-Systeme anbieten. Diese Systeme dienen unter anderem zum Schalten von Lampen, Rollläden und Heizungen oder zum Protokollieren und Visualisieren von Ereignissen wie der Temperatur, der Feuchtigkeit und der Stromverbrauch.

Hersteller sind zum Beispiel: AVM, Bosch, eQ3, Phillips, Rademacher oder Siemens.

Die meisten oben genannten Systeme funktionieren ohne Kabel und lassen sich einfach in bestehende Hausinstallationen integrieren. Des Weiteren gibt es auch leitungsgebundene Systeme (z.B. KNX oder Homematic IP Wired), die allerdings bereits bei der Hausplanung berücksichtigt werden sollten.

### Herstellerspezifische IoT-Systeme

Jeder Hersteller nutzt für seine Systeme eigene Protokolle, welche oftmals nicht herstellerübergreifend unterstützt werden. Diese Protokolle bestimmen die Regeln für den Austausch von Daten zwischen den Komponenten der IoT-Lösung und laufen über verschiedenste Schnittstellen. Einige Hersteller setzen bspw. auf das ZigBee-Light-Link-Protokoll. Es belastet das WiFi-Netz nicht mit Datendurchsatz und hat einen geringeren Stromverbrauch als eine WiFi-Anbindung. Im Vergleich zu Kurzstreckenfunktechnologien wie Bluetooth oder ANT ist der Stromverbrauch noch recht hoch. Ein ZigBee-Netzwerk nutzt das 2,4-GHz-Frequenzband. Der in Deutschland weitverbreitete Hersteller eQ3 nutzt für seine

Systeme 868,3 MHz als Funk-Frequenz. AVM verwendet für seine Smart-Home-Lösung den DECT-Standard.

Der Vorteil einer homogenen Herstellerlösung ist, dass die Integration der herstellerspezifischen Aktoren und Sensoren sowie die Programmierung von Steuerungen recht einfach umzusetzen ist. Einige dieser Lösungen nutzen Cloud-Server, auf denen die Programme ablaufen. Hierbei ist jedoch zu beachten, dass es sich um seriöse Betreiber handeln sollte, die mit den Daten vertrauenswürdig und sicher umgehen.

### Herstellerübergreifende IoT-Systeme

Um die Systeme verschiedenster Hersteller kombinieren zu können, gibt es Heimautomatisierungssysteme, die die verschiedenen Protokolle unterstützen und somit herstellerübergreifend die Steuerung von Aktoren und Sensoren ermöglichen. Hierzu gehört beispielsweise FHEM (Akronym für „Freundliche Hausautomation und Energie-Messung“). Dabei handelt es sich um ein Programm, welches zum Beispiel auf einem Raspberry Pi läuft und häufig auftretende Aufgaben herstellerübergreifend automatisiert.

#### Von FHEM unterstützte Protokolle

- Amazon Alexa, Apple Siri, Google Home
- eQ3 spezifisch: HomeMatic (IP), MAX!, FS20, EM1000, FHT80b, HMS, S300, ESA2000
- Philips HUE, FRITZ!Dect, KNX, ZWave, EnOcean, Intertechno, HomeEasy, Zigbee
- 1Wire, Firmata, webio, panStamp, LIRC, JeeLink, RFXCOM/RFXTRX, Wireless M-Bus

### IoT-Sicherheit

So toll und komfortabel eine Hausautomation auch ist, birgt diese leider auch Sicherheitslücken, welche Hackern den Zugriff auf das Smart-Home-System ermöglicht.

Das Bundesamt für Sicherheit in der Informationstechnologie (BSI) warnt vor Sicherheitslücken in vielen vernetzten Geräte wie IP-Kameras, smarten Türschlössern oder smarten Stromzählern. Auch Umgebungssensoren, zum Beispiel zur Erfassung von Temperatur und Luftfeuchtigkeit, sowie Lichtsysteme, Netzwerkdrucker und vernetzte Audiosysteme sind gefährdet.

Angreifer können über Sicherheitslücken in die Netzwerke eindringen, Daten stehlen und Computersysteme außer Gefecht setzen oder in ihrem Sinne fernsteuern. Bei einem smarten Türöffner beispielsweise bedeutet das: Der Angreifer kann sich Zugang zu der damit gesicherten Wohnung verschaffen.

Dies ist eine große Herausforderung für Hersteller und Nutzer, welche es zu bewältigen gilt.

## Praktische IoT-Anwendung

Als Anlass für dieses große Thema haben wir die aktuelle Covid-19-Pandemie genommen.

Weltweit kämpft die Bevölkerung gegen das neuartige Coronavirus an. Um die Pandemie eindämmen zu können, sind vor allem die AHA-Regeln (Abbildung 2) für uns alle von großer Bedeutung. Ein gründliches Waschen und Desinfizieren der Hände sind nötig. Damit dies sichergestellt ist, muss überall eine optimale Versorgung mit Seife, Papiertüchern und Desinfektionsmittel gewährleistet sein. Da das leider nicht immer der Fall ist, wollen wir daran forschen, wie eine solche Versorgung und auch anderweitig die Hygiene in Räumen sichergestellt werden kann. Mit Hilfe eines Mikrocomputers versuchen wir, vorhandenes Equipment in öffentlichen Gebäuden zu digitalisieren, sodass keine neuen Gerätschaften mit entsprechenden Schnittstellen angeschafft werden müssen.



Abbildung 32: AHA-Regeln

Ohne alle Räume ständig überprüfen zu müssen, soll die Versorgung gesichert werden, was auch eine große Zeitersparnis darstellt. Hausmeister werden vom System automatisch benachrichtigt, ohne dass sie angerufen oder aufgesucht werden müssen. Die Meldung kann somit nicht verloren gehen und die Arbeitskraft wird effizienter.

Da wir uns jedoch zuerst nur einem realisierbaren Ziel widmen wollten, entschieden wir uns dafür, das Problem mit den fehlenden Papiertüchern beim Abtrocknen zu lösen und Papierspender (Abbildung 3) zu digitalisieren. Ein richtiges Abtrocknen der Hände ist nebenbei genauso wichtig wie das Waschen: Durch nasse Hände können Keime leichter übertragen werden, wodurch die Wahrscheinlichkeit, die Umgebung zu kontaminieren viel höher ist. Laut der Bundeszentrale für gesundheitliche Aufklärung (BZgA) können sich Mikroorganismen in einer feuchten Umgebung besser halten und vermehren. Aus diesem Grund rät die BZgA, die Hände direkt nach dem Waschen gründlich abzutrocknen. Darüber hinaus streift man beim Trocknen sämtliche Keime ab, die noch an den Händen haften.



Abbildung 3: Handelsüblicher Papiertuchspender

Wir besuchen auf der Ludwig-Geißler-Schule im beruflichen Gymnasium die Bereiche Informatik und Elektrotechnik, weshalb dieses Projekt genau zu unseren Interessen passt. Dadurch können wir unsere bis jetzt erworbenen Fähigkeiten verwenden und uns in unserer Fachrichtung weiterbilden.



## FORSCHUNG

Zu Beginn bestand unsere Vorgehensweise darin, verschiedene Ideen und Fragestellungen zu sammeln. Bis wir endlich unser Thema gefunden hatten, dauerte es einige Zeit. Schließlich hat uns aber die Idee gut gefallen, etwas im Bereich „Internet Of Things“ beziehungsweise „Internet Of Stupid Things“ zu entwickeln. Zunächst überlegten wir, welche Materialien wir benötigen, und informierten uns über diese.

### Hardware

Unser Gerät sollte möglichst klein und kompakt, damit es zum Beispiel gut in einem Papierspender platzierbar ist, günstig, damit man viele Einheiten davon einsetzen kann, und energieeffizient sein, damit der Hausmeister nicht jeden Tag die Batterien wechseln muss. Also entschieden wir uns, einen Mikrocomputer bzw. Mikrocontroller (Abbildung 4) zu verwenden. Wir haben kein Produkt eines gängigen Smart-Home-Lösungen-Herstellers gefunden, was diese Kriterien erfüllt, zumal wir auch eine lokale Lösung bevorzugt haben, die nicht jedes Ereignis beispielsweise auf einem Server in China sichert.



Abbildung 4: Mikrocomputer

Darüber hinaus merkten wir schnell, dass unsere Geräte nur eine simple Aufgabe haben, wofür wir nicht ein vollständiges IoT-System eines Herstellers benötigen. Deshalb kam für uns nur ein Einplatinencomputer in Frage. Wir entschieden uns schließlich für ein besonders kompaktes Modell des Herstellers Espressif namens „ESP 8266 Plus“, welches im Vergleich zu den Konkurrenten der Raspberry Pi Foundation oder des Arduino-Projekts am besten unsere Erwartungen erfüllt. Von Haus aus ist eine WLAN-Schnittstelle vorhanden, die Kosten sind, besonders bei vielen Einheiten, sehr gering, der Stromverbrauch kann durch geschickte Programmierung auf bis zu 25µA minimiert werden (**Quelle 1**) und als Entwicklungsumgebung kommt auch die Arduino IDE in Frage, mit der wir aus dem Unterricht schon Vorerfahrung sammeln durften.

Es bestand nun die Möglichkeit, unsere Geräte und unser System über kabelgebundene oder kabellose Schnittstellen kommunizieren zu lassen.

Obwohl wir in unserer Schule in nahezu jedem Raum einen Ethernet-Anschluss haben, mit dem wir unseren Mikrocomputer ins Netzwerk bringen könnten, schlossen wir eine derartige Verkabelung in jedem Raum aus, da sie zu zeitintensiv gewesen wäre und man die vorhandene Infrastruktur insoweit verändern hätte müssen, dass man Löcher, zum Beispiel am Papierspender, für die Kabel hätte bohren müssen. Durch dieselben Gründe und der Tatsache, dass man in der ganzen Schule hätte Kabel verlegen müssen, um die Geräte in den Räumen zu verbinden, haben wir auch die kostengünstige, serielle Schnittstelle RS232 (Recommended Standard 232) und das zu umfangreiche Bussystem USB (Universal Serial Bus) ausgeschlossen.

Zusätzlich hätte mit den erforderlichen Schnittstellen-Steckern ein Mikrocomputer ausgesucht werden müssen, der deutlich an Kompaktheit verloren hätte.

Als kabellose Schnittstelle haben wir uns dann gegen ZigBee entschieden, weil wir keine zusätzliche Hardware wie einen ZigBee-Router kaufen wollten, und nahmen die höhere Energieaufnahme von WLAN in Kauf, da wir dafür schon alle benötigten Geräte in der Schule vorfinden können.

Dieses Privileg haben die Schüler nun schon seit beinahe fünf Jahren auf der Ludwig-Geißler-Schule. Die IT-Abteilung hat ein äußerst performantes WLAN-Netzwerk aufgebaut, welches den Schülern im Schüler-WLAN sogar in der Sporthalle eine kostenfreie Internetanbindung zur Verfügung stellt. Die Schule setzt hierbei auf ca. 50 WLAN-Access-Points der Firma Ubiquiti, also auf „Zugriffspunkte“, die im Gegensatz zu WLAN-Repeatern (Verstärkern) ihr eigenes WLAN aufbauen, somit nicht nur das vom Router verstärkt und dabei mit ebendiesem per LAN verbunden sind.



Abbildung 5: WLAN-Access-Point aus dem Schulgebäude

Bluetooth haben wir darüber hinaus als Schnittstelle ausgeschlossen, da wir wollten, dass man die Statusmeldungen von allen Geräten zu jeder Zeit mit unserem System einsehen kann und nicht nur von denen in der direkten Umgebung. Da das Schulnetzwerk auch von außen erreichbar ist, kann der Hausmeister nebenbei, egal wo er auch ist, sich die Füllstände anzeigen lassen.

Für den Betrieb unseres Einplatinencomputers verwenden wir ein Breadboard (Steckbrett), verschiedene Arten von Jumper-Kabeln und Sensoren, um unterschiedliche Versuchsaufbauten erstellen zu können (siehe Abbildung 6ff).

## Software

Bevor wir mit dem Programmieren anfangen konnten, mussten wir uns erst auch noch einmal über unsere Entwicklungsumgebung informieren. Die einfachste und effektivste Lösung, den Mikrocomputer anzusteuern, fanden wir in der Software vom Arduino-Projekt namens „Arduino-IDE“, in der wir mit der ESP8266 Erweiterung unsere Hardware mit C++ programmieren können. Anschließend versuchten wir, unseren ESP mit unserem neu gelernten Wissen anzusteuern. Nachdem dieser erste Versuch erfolgreich war und eine kleine LED auf der Platine unseres Microcomputers leuchtete, setzten wir diesen auf unser Breadboard und schlossen per Jumperkabel als ersten Test mit erweiterter Hardware ein Knopf-Modul (KY-004) und ein Zwei-Farben-LED-Modul (KY-011) an. Diese waren unter anderem Inhalt im „Sensorkit X40“ des Herstellers Joy-IT, welches uns, genauso wie die davor beschriebene Hardware von unseren betreuenden Lehrern ausgehändigt wurde. Dann programmierten wir den ESP so, dass die LED leuchtet, wenn der Taster gedrückt wird. Als Erweiterung ließen wir die LED dauerhaft rot leuchten, bis sie bei einem Knopfdruck zu grün



wechselt und anschließend beim Loslassen des Knopfes wieder auf die vorherige Farbe wechselt.



```

Knopf
int Led_Rot = D0;
int Led_Gruen = D7;
int Knopf = D6;

void setup ()
{
  pinMode (Led_Rot, OUTPUT);
  pinMode (Led_Gruen, OUTPUT);
  pinMode (Knopf, INPUT);
  digitalWrite (Knopf, HIGH);
}

void loop ()
{
  if (digitalRead (Knopf) == HIGH)
  {
    digitalWrite (Led_Rot, LOW);
    digitalWrite (Led_Gruen, HIGH);
  }
}
    
```

Abbildung 6: Arduino IDE

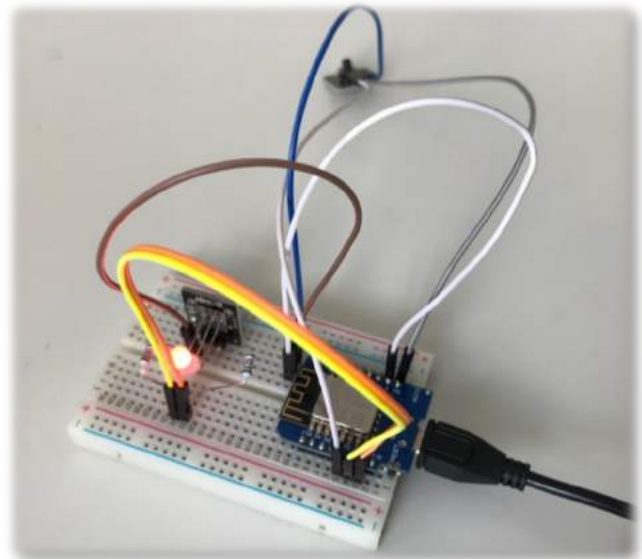


Abbildung 7: Hardware-Versuchsaufbau

## Sensorsuche

Nachdem diese ersten Versuche erfolgreich waren, überlegten wir, welcher Sensor am besten für unsere Anwendung, die Füllstandmessung eines Papiertuchspenders, geeignet ist.

Wir tauschten deshalb zunächst den Taster mit einem Infrarot-Abstandssensor (KY-032) aus und erhofften uns mit diesem im Inneren des Tuchspenders, also im Dunkeln wegen des nicht geöffneten Zustandes, zuverlässig einen gewissen Abstand messen zu können.

Es stellte sich bereits nach kurzer Zeit heraus, dass der Sensor ab einer bestimmten Entfernung anschlägt und sich auch auf verschiedene Entfernungen einstellen lässt. Dies war zwar schon mal ein Schritt in die richtige Richtung – da man aber die Entfernung, bei der der Sensor anschlägt nur mechanisch und nicht per Programmierung einstellen kann und wir zudem nicht die genaue Entfernung abfragen konnten, gingen wir erneut auf die Suche. Wir behielten ihn aber im Hinterkopf, da wir im fortgeschrittenen Verlauf unserer Entwicklung gerne noch weitere Zustände in öffentlichen Räumen wie zum Beispiel ein offenes Fenster digitalisieren wollen, wofür dieser sehr gut geeignet wäre.

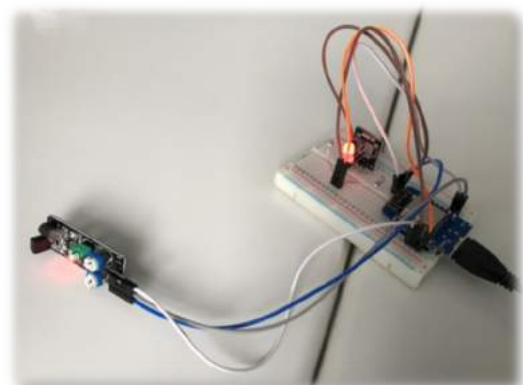


Abbildung 8: Versuchsaufbau mit Infrarot-Abstandssensor

Wir schauten also erneut auf der Webseite mit Informationen zu unserem Sensorkit (**Quelle 2**), wo die ungefähr 40 Sensoren alle aufgelistet sind und deren Funktion beschrieben werden. Es stach uns schließlich der Ultraschall-Abstandssensor (KY-050) ins Auge, der perfekt für unser Anwendungsgebiet ist. Der einzige Nachteil gegenüber dem vorherigen Sensor ist, dass er etwas teurer, aber mit einem Mengenrabatt trotzdem noch sehr erschwinglich ist. Die

Vorteile wiederum überwiegen deutlich. Der HC-SR04 (Herstellerbezeichnung unseres Sensors) gibt in einem seriellen Monitor (Abbildung 8) nach einer internen Rechnung die genaue Entfernung zu einem bestimmten Gegenstand in cm aus, was möglich macht, dass wir dem Hausmeister am Ende einen fast genauen Füllstand in Prozent angeben

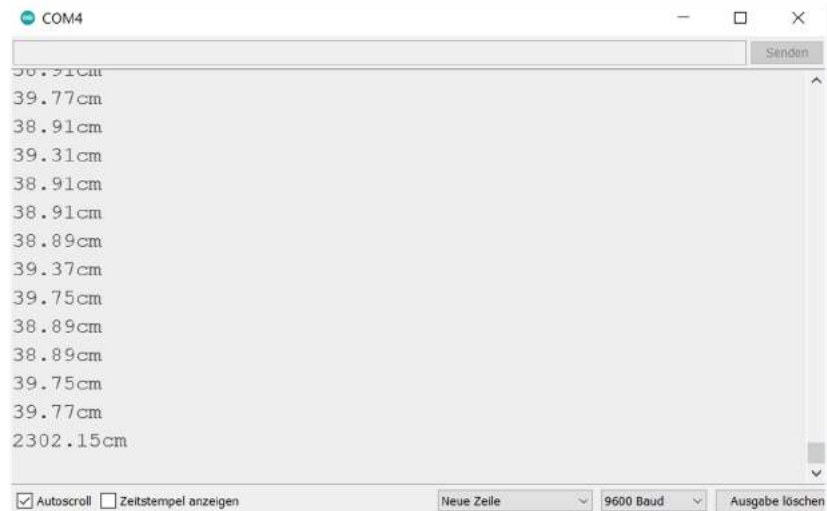


Abbildung 9: Serieller Monitor der Arduino IDE

können. Die Messwerte sind meistens relativ genau und ähnlich (bei derselben Entfernung), doch manchmal schleichen sich fehlerhafte Messdaten ein, wie man gut in der Abbildung erkennen kann. Deshalb entschieden wir uns dafür, in unserem System später einige Messungen durchzuführen und dann den Median (Mittelwert) und nicht den Durchschnittswert zu übermitteln, da ein solch großer Wert diesen stark verfälschen würde.

## Überleitung

Die Sensorsuche war somit erfolgreich beendet und wir konnten uns nun zwei weiteren wichtigen Aspekten widmen: Der Stromversorgung und dem System hinter den Mikrocomputern.

Wegen der vor den Winterferien wieder akuten Pandemielage hatten wir von diesem Zeitpunkt an nur noch getrennt Unterricht und konnten uns natürlich auch nicht privat treffen in unserer Freizeit. Deshalb führten wir unsere gemeinsame Projektarbeit über unsere Schulplattform iServ und andere Hilfsmittel wie Discord für Sprach- und Videokonferenzen fort und teilten unsere Weiterarbeit aber auf die eben beschriebenen Gebiete auf: Niklas Krieger übernahm als Elektrotechniker die Aufgabe der Stromversorgung und wollte auch ein erstes Gehäuse für unser inzwischen „TellMe“ getauftes Produkt anfertigen. Mirko Weih als Informatiker übernahm die Entwicklung des Systems hinter unserem Produkt. Gemeinsames Überlegen und Planen erfolgte natürlich aber weiterhin.

## Stromversorgung & Gehäuse

Da wir auch schon ein Kabel für unsere Netzwerkverbindung ausgeschlossen hatten, waren wir uns natürlich einig, dass wir auch hier auf eine kabellose Lösung in Form eines Akkumulators (Akku), einer wieder aufladbaren Batterie, setzen.

Zunächst führte Niklas erstmal zuhause einen Versuch durch, in dem er eine alte Batteriehalterung mit einer Metallsäge zurechtgeschnitten und anschließend ein altes USB-Kabel, welches er zuvor zerschnitten und isoliert hatte, angelötet hat. Dieses Konstrukt

verband er anschließend mit unserem ESP, was funktioniert hat, denn die LEDs am Mikrocomputer waren eingeschaltet und somit die Stromversorgung erfolgreich sichergestellt. Zudem führte er Messungen zum Stromverbrauch durch, die schnell zeigten, dass der Einplatinencomputer, besonders mit einer effizienten Programmierung, wirklich nicht viel Strom braucht, was dem Hausmeister ein häufiges Wechseln ersparen sollte.

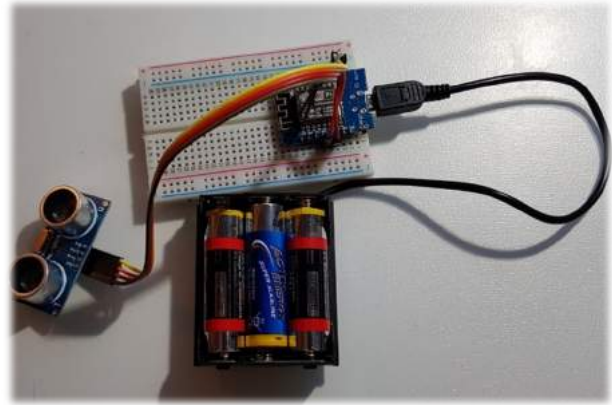


Abbildung 10: Stromversorgung des ESPs mit Ultraschall-Abstandssensor

Wir überlegten trotzdem intensiv, wie wir die Aufladung des Akkus für diesen vereinfachen können. Unser Ergebnis waren zwei Varianten: Entweder der Hausmeister lädt den Akku vor Ort oder er tauscht den Akku aus. Ersteres ist für ihn ein wenig umständlich, da er dann einmal vor Ort ein Kabel legen muss und auch noch einer bestimmten Zeit ein zweites Mal im Raum vorbeischaun muss, um seine Verkabelung wieder abzubauen. Das ist ein wenig umständlich und könnte zudem von Schülern, die bspw. das Kabel ziehen, erschwert werden.

Deshalb neigten wir eher zur zweiten Variante. Diese wollten wir zudem noch erweitern, indem man unser ganzes Produkt austauschen soll. Dadurch können wir einen fest verbauten Akkumulator verwenden, der zwar bei einem Defekt gewechselt werden kann, aber grundsätzlich nicht mehr extrem einfach erreichbar sein muss, weshalb wir unser Produkt in ein kompaktes Gehäuse stecken wollen, wo nach außen hin nur der Sensor und ein USB-Ladeanschluss sichtbar sind. Dadurch können wir eine Halterung in jedem Papiertuchspender anbringen, durch die der TellMe immer in der besten Position hängt.

Generell sollen die Geräte oben im Spender befestigt werden, da dort genug Platz und eine zuverlässige Messung möglich ist.

Ein Problem hatten wir bis jetzt noch nicht beseitigt: Der Hausmeister muss nach dem Laden des Akkus trotzdem nochmal zurück in den besagten Raum kommen, da unser Produkt natürlich im System immer mit einem Raum verknüpft sein muss, da der Hausmeister sonst nicht weiß, wo der leere Papierspender ist.

Genau da kommt unser erdachtes System für die Geräte ins Spiel...

### [System der TellMe's](#)

Für die Aufladung haben wir uns überlegt, dass jeder TellMe mit einem QR-Code oder einer Nummer versehen ist. Wenn der Hausmeister nun ein Gerät auswechseln muss, da er auf der TellMe-Statusanzeige gewarnt wurde, dass unser Produkt in einem bestimmten Raum einen niedrigen Akkustand hat, nimmt er den leeren TellMe aus der Halterung und steckt einfach einen neuen, geladenen hinein. Dann scannt er den QR-Code oder gibt die Nummer ein und teilt in der TellMe-Plattform mit, dass das neue, aufgeladene Gerät nun in dem aktuellen Raum ist. Anschließend kann er das leere laden und muss es nicht wieder im selben Raum

verwenden. Darüber hinaus gibt es dann auch keine Zeiträume, in denen ein Papiertuchspender nicht mit einem TellMe bestückt ist.

Als nächstes stand dann die Überlegung, wie der Hausmeister benachrichtigt wird, auf der Tagesordnung. Dafür haben wir uns die oben schon einmal erwähnte TellMe-Plattform ausgedacht. Zunächst soll es erstmal eine Webseite sein, die lokal im Netzwerk gehostet ist. Auf ihr sieht man einen Raumplan des öffentlichen Gebäudes und dann die mit diesem Raum verknüpften Papiertuchspender. Wenn man diese anklickt, bekommt man nähere Informationen zum Papier- und Akkufüllstand. Zusätzlich kann man auch die Nummer des aktuell mit dem Spender verknüpften TellMe's einsehen.

Die genauen Informationen werden von der Systemplattform ca. einmal die Stunde aktualisiert, indem der TellMe kurz eine WLAN-Verbindung zum Netzwerk herstellt, was er nicht durchgehend tut, um Energie zu sparen, und dann alle wichtigen Informationen übermittelt. In der Nacht oder am Wochenende soll er das natürlich nicht so häufig vollführen.

Das System überwacht dann alle übermittelten Daten und informiert denn Hausmeister, sobald Handlungsbedarf eintritt.

Im weiteren Verlauf der Entwicklung soll diese Plattform vielleicht auch eine App erhalten, was wir aber grundsätzlich erstmal vernachlässigt haben, weil wir diese für mehrere Betriebssysteme, also hauptsächlich Android und iOS, entwickeln müssten, was besonders bei letzterem einige Schwierigkeiten bereiten wird, da man Apps nicht einfach so in Apples AppStore veröffentlichen kann.

## WIE GEHT ES WEITER?

Bis zum Wettbewerb werden wir natürlich unser Produkt finalisieren und unser System dahinter programmiertechnisch umsetzen.

Was wir schon einige Male haben anklingen lassen und was man sicher auch schnell bemerken wird, ist, dass man unser System auch auf andere Mikrocomputer mit anderen Sensoren und anderen Einsatzzwecken anwenden kann.

Konkret schweben uns folgende Projekte für die Zukunft vor:

Wir könnten einen TellMe konstruieren, der oben an den Fenstern befestigt wird und diese dann ins System einbindet, damit offene Fenster, bspw. über das Wochenende, der Vergangenheit angehören.

Ein weiteres Produkt wäre ein TellMe, der Statuslampen in Räumen ins System einbindet. Zum Beispiel, wenn in den Fachräumen für Biologie, Physik oder Chemie noch Strom auf den Steckdosen der Schüler oder die Gaszufuhr für die Bunsenbrenner noch eingeschaltet ist, was beides meistens mit einer leuchtenden Lampe im Raum signalisiert wird.

Darüber hinaus wäre in den aktuellen Zeiten natürlich auch ein TellMe sehr sinnvoll, der den Füllstand von Seifen- und Desinfektionsmittelspendern abfragt. Dies ist aber im Vergleich mit

der Digitalisierung des Füllstands von Papiertuchspendern um einiges schwerer, da man unser Gerät nicht einfach in den Innenraum verbauen kann, da dieser zu klein ist und dort natürlich auch nicht die besten Bedingungen für Elektronik herrschen. Dementsprechend müsste man den Füllstand von außen durch die Gehäusewand des Spenders hindurch messen. Eine äußere und somit sichtbare Anbringung erhöht zudem das Risiko eines Diebstahls oder einer Beschädigung von außen.

In Abbildung 9 sieht man einen Anwendungsentwurf, der verdeutlicht, dass durch die Lage des Sensors entschieden werden kann, wie früh der TellMe dem Hausmeister einen niedrigen Füllstand mitteilt. An der ersten Position erfolgt eine frühe und an der zweiten Position eine späte Mitteilung.



Abbildung 11: Anwendungsentwurf

## Quellenverzeichnis

- <https://www.infektionsschutz.de/haendewaschen/>
- <https://www.freundin.de/gesundheit-haende-richtig-abtrocknen-so-gehts>
- <https://www.telekom.de/smart-produkte/iot>
- [https://fhem.de/fhem\\_DE.html](https://fhem.de/fhem_DE.html)
- <https://www.elektronikpraxis.vogel.de/von-ant-bis-zigbee-die-wichtigsten-funkstandards-im-smart-home-a-796103/>
- <https://www.computerwoche.de/a/sicherheitsluecken-bedrohen-millionen-iot-geraete,3550302>
- <https://www.giga.de/extra/android-spezials/specials/apn-access-point-name-was-ist-das/>
- <https://www.mikrocontroller.net/articles/ESP8266>
- [https://praxistipps.chip.de/programmieren-fuer-den-arduino-ein-anfaenger-tutorial\\_102897](https://praxistipps.chip.de/programmieren-fuer-den-arduino-ein-anfaenger-tutorial_102897)
- [https://sensorkit.en.joy-it.net/index.php?title=Main\\_Page](https://sensorkit.en.joy-it.net/index.php?title=Main_Page)
- <https://www.heinzinger.de/glossar/akkumulator/>
- [https://de.freepik.com/vektoren-premium/smart-home-iot-internet-der-dinge-steuern-komfort-und-sicherheit-isometrische-infografik-poster-abstrakt\\_12089696.htm](https://de.freepik.com/vektoren-premium/smart-home-iot-internet-der-dinge-steuern-komfort-und-sicherheit-isometrische-infografik-poster-abstrakt_12089696.htm)
- [https://www.bundesgesundheitsministerium.de/fileadmin/Dateien/5\\_Publikationen/Gesundheit/Flyer\\_Poster\\_etc/Corona/AHA-Formel\\_Blau-Rot\\_barr\\_V2.pdf](https://www.bundesgesundheitsministerium.de/fileadmin/Dateien/5_Publikationen/Gesundheit/Flyer_Poster_etc/Corona/AHA-Formel_Blau-Rot_barr_V2.pdf)
- [https://static.praxisdienst.com/out/pictures/generated/product/1/800\\_800\\_90/marplast\\_haandtuchspender\\_134940\\_1.jpg](https://static.praxisdienst.com/out/pictures/generated/product/1/800_800_90/marplast_haandtuchspender_134940_1.jpg)
- [https://media.nbb-cdn.de/images/products/250000/257732/Ubiquiti\\_UAP-AC-PRO\\_Front.jpg?size=2800](https://media.nbb-cdn.de/images/products/250000/257732/Ubiquiti_UAP-AC-PRO_Front.jpg?size=2800)