

Groß, Hannes

## Die App "Blynk". Automatisieren einer Gewächshausbewässerung

*technik-education (tedu). Fachzeitschrift für Unterrichtspraxis und Unterrichtsforschung im allgemeinbildenden Technikunterricht 1 (2021) 2, S. 40-43*



Quellenangabe/ Reference:

Groß, Hannes: Die App "Blynk". Automatisieren einer Gewächshausbewässerung - In: *technik-education (tedu). Fachzeitschrift für Unterrichtspraxis und Unterrichtsforschung im allgemeinbildenden Technikunterricht 1 (2021) 2, S. 40-43* - URN: urn:nbn:de:0111-pedocs-243012 - DOI: 10.25656/01:24301

<https://nbn-resolving.org/urn:nbn:de:0111-pedocs-243012>

<https://doi.org/10.25656/01:24301>

in Kooperation mit / in cooperation with:



<https://tec-edu.net/tedu>

### Nutzungsbedingungen

Dieses Dokument steht unter folgender Creative Commons-Lizenz: <http://creativecommons.org/licenses/by-nc/4.0/deed.de> - Sie dürfen das Werk bzw. den Inhalt vervielfältigen, verbreiten und öffentlich zugänglich machen sowie Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen, solange Sie den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen und das Werk bzw. den Inhalt nicht für kommerzielle Zwecke verwenden.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

### Terms of use

This document is published under following Creative Commons-License: <http://creativecommons.org/licenses/by-nc/4.0/deed.en> - You may copy, distribute and render this document accessible, make adaptations of this work or its contents accessible to the public as long as you attribute the work in the manner specified by the author or licensor. You are not allowed to make commercial use of the work, provided that the work or its contents are not used for commercial purposes.

By using this particular document, you accept the above-stated conditions of use.



### Kontakt / Contact:

peDOCS  
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation  
Informationszentrum (IZ) Bildung  
E-Mail: [pedocs@dipf.de](mailto:pedocs@dipf.de)  
Internet: [www.pedocs.de](http://www.pedocs.de)

Mitglied der

  
Leibniz-Gemeinschaft

# technik – education

1. Jahrgang

Fachzeitschrift für Unterrichtspraxis und Unterrichtsforschung  
im allgemeinbildenden Technikunterricht

2|2021



[www.tec-edu.net](http://www.tec-edu.net)

**tedu**

# Fachzeitschrift für Unterrichtspraxis und Unterrichtsforschung im allgemeinbildenden Technikunterricht

<https://tec-edu.net/tedu>

## HERAUSGEBER

Dr. Hannes Helmut Nepper  
Armin Ruch, OStR  
Prof. Dr. Lars Windelband

## REDAKTION

Dr. Dierk Suhr

## Mail

herausgeber@tec-edu.net

## Anschrift

Pädagogische Hochschule Schw. Gmünd  
Institut für Bildung, Beruf und Technik  
Abteilung Technik  
Oberbettringer Straße 200  
73525 Schwäbisch Gmünd  
[www.tec-edu.net](http://www.tec-edu.net)

## AUTOR\*INNEN IN DIESEM HEFT

Samuel Benz  
Sindi Dressnandt  
Stefan Ginthum  
Hannes Groß  
Till Lohse  
Jörg Ostertag  
Sarah Ryser

Namentlich gekennzeichnete Beiträge  
geben nicht unbedingt die Meinung der  
Herausgeber wieder.  
Titelfoto: Armin Ruch

**ISSN: 2748-2022**

## Inhalt

**Grußwort der Herausgeber** ..... 2

*Unterrichtsforschung*

S. Ryser

**Bildung für nachhaltige Entwicklung im techni-  
schen und textilen Gestalten** ..... 3

*Unterrichtsforschung*

J. Ostertag

**Comicvignetten als fallbasierte Methode zur  
Sensibilisierung von Lehrkräften** ..... 12

*Diskussionsbeitrag*

tedu

**Technik im Bereich Sonderpädagogik** ..... 21

*Diskussionsbeitrag*

S. Dressnandt & S. Ginthum

**Learning by Making**..... 23

*Unterrichtspraxis*

T. Lohse

**Alternativen zum gekauften Bausatz**..... 26

*Unterrichtspraxis*

S. Benz

**Ein Trinkwasserspender für das Klassenzimmer** ..... 32

*Unterrichtspraxis*

H. Groß

**Die App Blynk**..... 40

# Die App „Blynk“ Automatisieren einer Gewächshausbewässerung

Hannes Groß

## SCHLAGWORTE

Blynk-App  
Smart Home  
Messen-Steuern-Regeln  
Smartphone im Unterricht

## ABSTRACT

In der heutigen Zeit rückt ein nachhaltiger Lebensstil, die Nähe zum Grünen und eine gesunde Ernährung immer mehr in den Fokus unsere Gesellschaft. „Home-Gardening“ ist im Trend und viele Kräuter-, Obst- und Gemüsepflanzen werden auf dem heimischen Balkon oder im Garten angepflanzt. Auch Schülerinnen und Schüler beschäftigt dieses Thema und sie können die nachfolgende Projektidee in ihren Alltag integrieren. Im vorliegenden Beitrag soll die Nutzung der Blynk-App am Beispiel einer Gewächshausbewässerung verdeutlicht werden. Für die Umsetzung in einem mehrperspektivischen Technikunterricht bietet sich dabei die Fertigung eines Gewächshauses im Fachwerkstil an. Dies fordert nicht nur das Pflanzenwachstum, sondern soll außerdem die alte traditionelle Handwerkstechnik (Fachwerkbau) mit neuen Innovationen (automatisierte Bewässerung) verknüpfen, um so einen kulturhistorischen Kontext zu schaffen.

## Projektvorstellung

Der Grundaufbau des Projekts stellt ein Gewächshaus aus Holz im Fachwerkstil dar. Teil des Gewächshauses sind neben einem Wassertank auch eine Pumpe, welche die Pflanzen automatisiert bewässert (vgl. Abb. 1). Die Bewässerung der Pflanze(n) wird mit Hilfe eines Mikrocontrollers gesteuert. Dieser ist über WLAN mit einer App auf dem Smartphone verbunden. Durch einen Feuchtigkeitssensor wird die Bodenfeuchtigkeit der Pflanze gemessen. Wird der vorher eingestellter Wert unterschritten, erhält der Benutzer eine Push-Benachrichtigung auf sein Smartphone und kann anschließend mit der App die Pumpe einschalten und somit die Pflanze(n) bewässern.



Abbildung 1: Ein automatisiertes Miniaturgewächshaus

## Fertigungs- und Konstruktionsprozess

Die fachpraktische Umsetzung setzt sich aus dem Konstruktion- und Fertigungsprozesse des Fachwerk-Gewächshauses zusammen. Aufgrund der offenen projektbezogenen Aufgabe, erfordern diese Prozesse von den Schülerinnen

und Schüler einen hohen Grad an Eigenständigkeit und Problemlösefähigkeit. Die Schüler\*innen probieren, konstruieren, experimentieren und fertigen, wenn möglich, in Eigenarbeit ihr Fachwerkhaus.

Die Schüler\*innen können hier ihre bereits gelernten Fähigkeiten und Fertigkeiten mit CAD Programmen bzw. Zeichenbrett sowie den Werkzeugen der Holzbearbeitung wie der Japansäge oder dem Stecheisen unter Beweis stellen. Ebenfalls sind Kenntnisse in der Bautechnik nötig, um die Holzverbindungen im Fachwerkstil korrekt umzusetzen (vgl. Abb. 2).

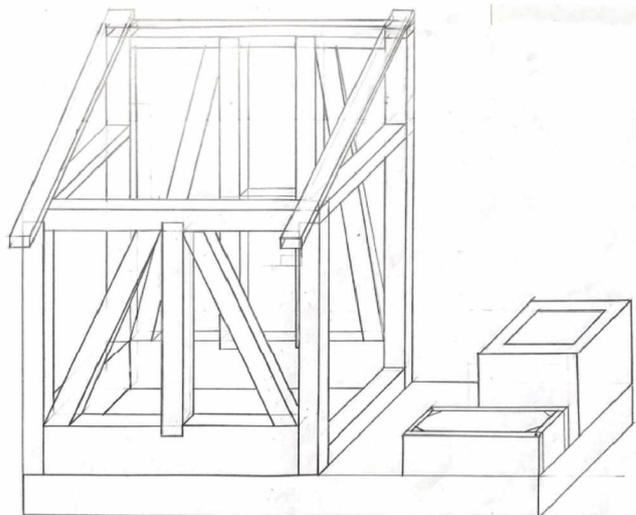


Abbildung 2: Beispiel einer Projektskizze des Fachwerks

## Programmierung

Der komplexeste Teil des Projekts stellt die Programmierung des Mikrocontrollers dar. Auch hier sollten bereits Grundkenntnisse und Grundfertigkeiten im Umgang mit Mikrocontrollern vorhanden sein. Zur Ansteuerung des Relais und der damit verbundenen Wasserpumpe sowie zum

Auslesen des Feuchtigkeitssensors wird ein WLAN-fähiger Mikrocontroller benötigt. Hier wurde ein ESP32 verwendet. Zur Steuerung des ESP32 wurde die App „Blynk“ verwendet, welche es ermöglicht, eine eigene Benutzeroberfläche aus sogenannten Widgets wie Buttons, Drehregler oder Graphen zu erstellen, um damit einen Mikrocontroller zu steuern oder dessen Eingangswerte (wie hier einen Feuchtigkeitswert) anzeigen zu lassen. Die App ist kostenlos für Android und Mac iOS in den jeweiligen Stores herunterladbar.

### Anleitung zur Steuerung eines Mikrocontrollers mit der Blynk-App

Die folgenden Schritte erklären, wie das Gewächshaus mit dem Smartphone gesteuert werden kann. Zur Umsetzung wird ein Smartphone mit installierter Blynk-App (für Android oder mac iOS), ein ESP32- Mikrocontroller, WLAN-Zugang sowie ein Verbindungskabel benötigt.

#### Registrierung

Nach der Registrierung in der Blynk-App wird die Zugangsberechtigung (Auth-Token) an die angegebene E-Mail-Adresse gesendet. Dieses Token wird später benötigt, damit sich der ESP32 mit der Blynk-App verbindet.

#### Projekt anlegen

Durch Klicken auf „New Project“ wird ein neues Projekt angelegt. Zusätzlich muss die Hardware und Verbindungsart ausgewählt werden. Als Hardware wird der ESP32 ausgewählt und als Verbindungsart Wi-Fi. Nach einem Klick auf Create gelangen wir auf eine Arbeitsoberfläche, auf der wir unsere Visualisierung konfigurieren können.

#### Bauteile auswählen

Ein Klick auf den Arbeitsbereich öffnet die Widget-Box, in der sich die verschiedenen zur Verfügung stehenden grafischen Objekte befinden. Die kostenfreie Version bietet eine limitierte Anzahl gleichzeitig nutzbarer Objekte. Jedes Bauteil benötigt einen bestimmten Energiebetrag. Die kostenlose Version stellt dafür 2000 Energieeinheiten zur Verfügung. Die Bauteile können per Drag and Drop beliebig im Arbeitsbereich angepasst werden. Zur Realisierung der Automatisierten Bewässerung benötigen wir drei Widgets. Alle Widgets können per Drag and Drop auf der Oberfläche platziert

werden. Benennung, Farbe, Schriftgröße, Einheit etc. lassen sich beliebig anpassen.

#### Bauteil Konfigurierung

Klickt man auf die jeweiligen Bauteile, können diese konfiguriert und benannt werden. Das „Notification-Widget“ muss nicht konfiguriert werden.

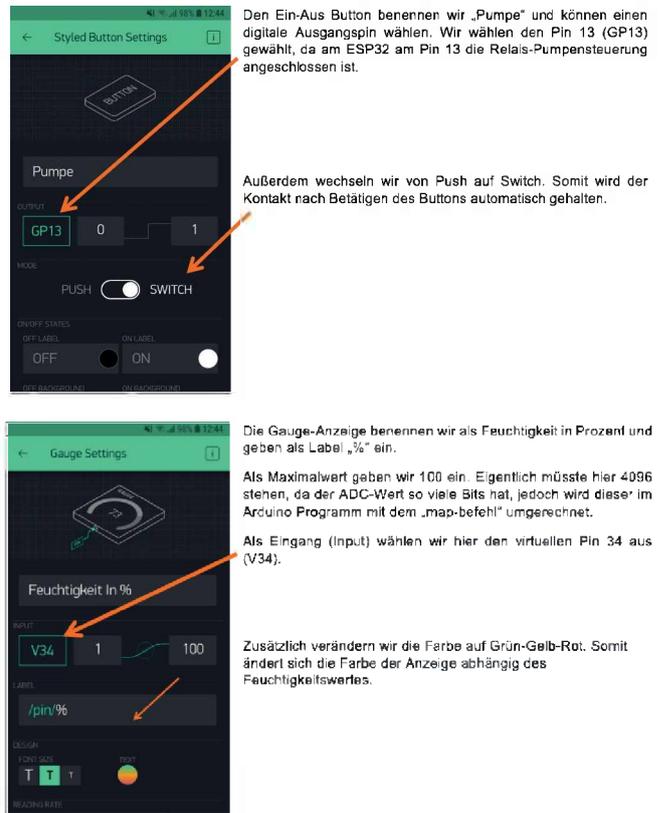


Abbildung 4: Einstellungen bei der Konfigurierung

#### Verbindung mit Arduino IDE

In der Arduino IDE muss im Boardverwalter das Board ausgewählt werden („ESP32 Dev Module“). Dafür müssen die benötigten Boardinformationen heruntergeladen und installiert werden. Dazu fügt man in der Arduino IDE in den Voreinstellungen (Datei → Voreinstellungen) in der Zeile „Zusätzliche Boardverwalter-URLs:“ den Eintrag [https://raw.githubusercontent.com/esp8266/Arduino/master/packages/package\\_index.json](https://raw.githubusercontent.com/esp8266/Arduino/master/packages/package_index.json)

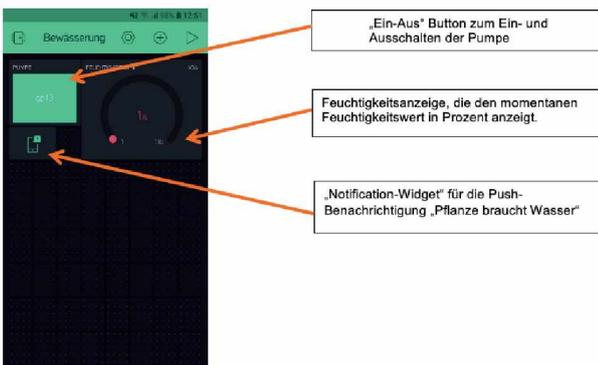


Abbildung 3: Die Oberfläche der Blynk-App

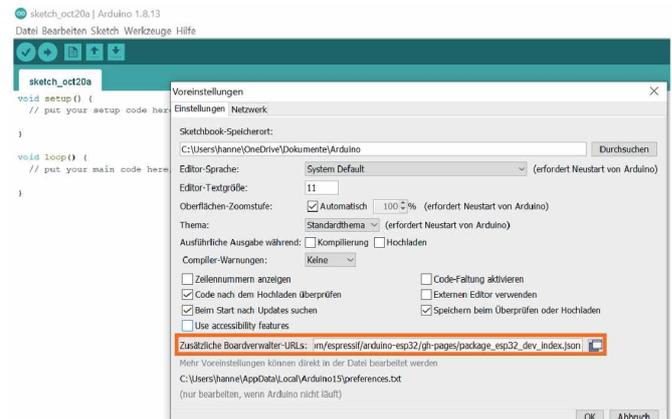


Abbildung 5: Herstellen der Verbindung zum Arduino

githubusercontent.com/esp8266/arduino-esp32/gh-pages/package\_esp32\_dev\_index.json ein und klickt „OK“.



Abbildung 6: Herunterladen der Boardinformationen

Danach öffnet man unter Werkzeuge → Board → Boardverwalter den Boardverwalter und kann im Suchfeld esp32 eingeben und die dazu benötigten Boardinformationen herunterladen.

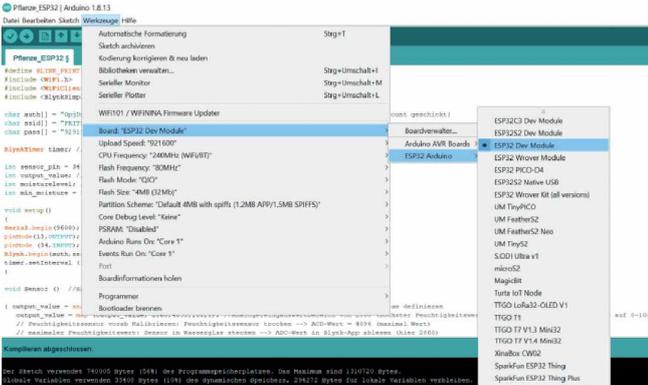


Abbildung 7: Auswahl des Boards

Anschließend lässt sich das „ESP32 Dev Module – Board“ auswählen.



Abbildung 8: Herunterladen der Blynk-Bibliothek

Neben der Boardinformationen für den Mikrocontroller muss zusätzlich die Blynk-Bibliothek heruntergeladen werden. Diese lässt sich unter Werkzeuge → Bibliotheken verwalten mit dem Suchbegriff „Blynk“ suchen und installieren.

Zur Verwendung der Blynk-App müssen im Programmcode folgende Zeilen enthalten sein. Die Kommentare in der IDE erklären die Funktion der jeweiligen Zeilen.

```

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkClient.h>
#include <BlynkSimpleEsp32.h> //ESP32 mit Blynk App steuern

char auth[] = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"; //Blynk-Token (wird von Blynk-App an E-Mail-Account geschickt)
char ssid[] = "XXXXXXXXXX"; // Wifi-Name
char pass[] = "XXXXXXXXXX"; // Wifi-Passwort
    
```

Abbildung 9: Anpassen des Codes zur Verwendung der App

**Programmierung:**

Im Folgenden wird nun das Programm erläutert. Im Setup werden wie gewohnt die Ein- und Ausgänge deklariert. Mit „Blynk.begin(auth,ssid,pass);“ wird die Blynk App gestartet. Der Timerinterval legt fest, wie oft der ESP32 den am Feuchtigkeitssensor gemessener Wert an die App sendet (1000L = jede Sekunde).

Eine „if-Abfrage“ legt fest, wann die Push-Benachrichti-

```

BlynkTimer timer; // Funktion sendet Betriebszeit von ESP32 jede Sekunde an Virtual Pin (V34)

int sensor_pin = 34; //Pin-Nummer 34 als Analogeingang Feuchtigkeitssensor definieren
int output_value; // als Ausgabe definieren
int moisturelevel; // als Ausgabe definieren
int min_moisture = 30; //Mindest Feuchtigkeitsswert kann
    
```

Abbildung 10: Deklarieren der Variablen

gung „Pflanze braucht Wasser“ erfolgen soll. Der Mindestfeuchtigkeitsswert (hier 30) wird am Anfang des Programms deklariert und kann beliebig angepasst werden. Der Hauptloop besteht nur aus 2 Zeilen.

```

void setup()
{
  Serial.begin(9600);
  pinMode(13,OUTPUT); //Definition Ausgangspin Pumpe
  pinMode(34,INPUT); // Definition Eingangspin Sensor
  Blynk.begin(auth,ssid,pass); //Starte Blynk App
  timer.setInterval(1000L,Sensor); // Timer setzen
}

void Sensor () //Hauptfunktion zum Lesen von Sensoren und zum Drucken auf blynk
{
  output_value = analogRead(sensor_pin); // Analogsignal von sensor_pin lesen und als output_value definieren
  output_value = map(output_value, 2860, 4095, 100, 0);
  //Analogereingangsbereich von 2860 (Höchster Feuchtigkeitsswert)-4096 (Trockener Sensor)Bits auf 0-100%
  // Feuchtigkeitssensor vorab Kalibrieren: Feuchtigkeitssensor trocken -> ADC-Wert = 4096 (maximal Wert)
  // maximaler Feuchtigkeitsswert: Sensor in Wasserglas stecken -> ADC-Wert in Blynk-App ablesen (hier 2860)
  Blynk.virtualWrite(V34, output_value); //Feuchtigkeitsswert an virtuellen Pin 34 senden.

  if(output_value < min_moisture) //Wenn moisturelevel kleiner als der eingestellte Mindestwert ist
  // (hier 30, kann aber beliebig verändert werden)
  {
    Blynk.notify("Pflanze braucht Wasser"); // Push-Benachrichtigung durch die Blynk-App
  }
}

void loop()
{
  Blynk.run();
  timer.run();
}
    
```

Abbildung 11: Setup für das Programm

Abbildung 12: Der Hauptlook des Programms ist sehr kurz

Das Programm kann nun von der Arduino IDE auf den ESP32 geladen werden.

**Anwendung:**

Nun öffnen wir die Blynk-App und drücken Play (rechte obere Ecke), um die App zu starten. Daraufhin erhält man eine Nachricht, dass die Hardware online ist.

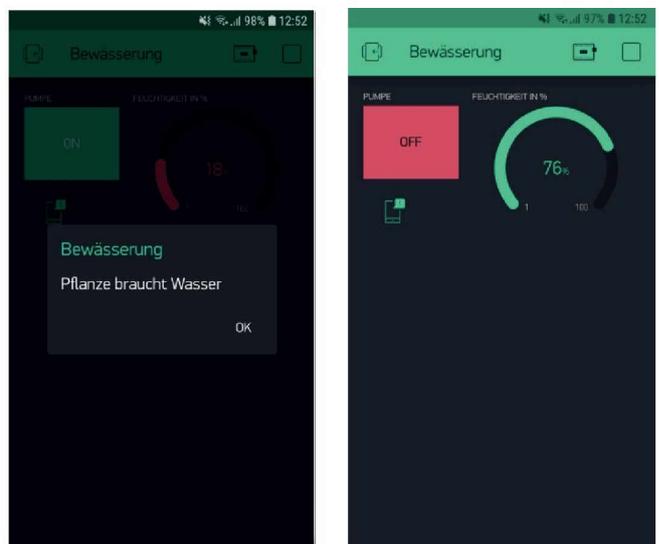


Abbildung 13: Die Oberfläche der laufenden Blynk-App

## Funktionsablauf

Der Ablauf der Einstellungen über die App ist sehr anschaulich und intuitiv bedienbar.

1. Feuchtigkeit unter 30 % → Push Benachrichtigung „Pflanze braucht Wasser“
2. Pumpe wird eingeschaltet (Ein-Aus-Button = Rot) → Pflanze wird bewässert, Feuchtigkeitswert steigt an, Push-Benachrichtigung setzt aus.

## Mögliche Herausforderungen in der Umsetzung im Unterricht

Um die Schülerinnen und Schüler vorab mit den Grundfunktionen und dem Umgang der „Blynk-App“ vertraut zu machen, empfiehlt es sich vorab, einfach Grundschaltungen wie das Ein- und Ausschalten von LEDs mit der App auszuprobieren. Leider gibt es keine Möglichkeit, grafische Programme wie „Open Roberta Lab“ zu verwenden, wodurch die Programmierung mithilfe der Arduino IDE realisiert werden muss. Das Programm setzt sich aus Elementen der Programmiersprachen C, C+ und Java zusammen und könnte sich als Einstieg in das Thema Programmieren ohne Vorkenntnisse schwierig gestalten. Eine Möglichkeit zur vereinfachten Umsetzung wäre es, das Programm selbst grafisch darzustellen. Auch ein Puzzle aus Programmschnipseln (Die Schüler\*innen ordnen nur die richtige Reihenfolge an.) könnten sich hier als hilfreich darstellen.

## Erweiterungsmöglichkeiten

Um aufgrund der Komplexität des Projekts die Schüler\*innen nicht zu überfordern, wurde die Bauteilauswahl überschaubar gestaltet. Neben einem Feuchtigkeitssensor und einer Pumpe sind keine weitere Eingangs- bzw. Ausgangsbaueteile vorhanden, um eine erhöhte Komplexität zu vermeiden. Es gibt jedoch zahlreiche Möglichkeiten, das Projekt zu verändern. Je nach Niveau und Motivation der Schülerinnen und Schüler kann der Aufbau erweitert bzw. vereinfacht werden. Als Erweiterung des Projekts könnte beispielsweise die Steuerung um LEDs, einen Temperatursensor oder einen optischen Sensor zur Überwachung des Füllstandes im Tank erweitert werden. Als vereinfachte Variante kann die Steuerung auch mit einem Arduino Uno bzw. Nano (beide nicht WLAN-fähig) realisiert werden. Hier würde die Pumpe die Pflanze automatisch bewässern, sobald der Feuchtigkeitswert zu gering ist. Zur Erstellung des Programms kann hierfür dann auch die Online-Plattform mit graphischer Programmiersprache „Open-Roberta-Lab“ verwendet werden. Weitere Ergänzungsmöglichkeiten wären eine selbst entworfene und gefräste Platine sowie ein durch 3D-Druck gefertigter witterungsbeständiger Schaltkasten für das Steuerungselement.

## Fazit

Zusammenfassend stellt das „Gewächshaus mit Mikrocontroller-gesteuerter Pumpe“ im mehrperspektivischen Tech-

nikunterricht ein interessantes Projekt dar. Es lässt sich flexibel als klassisches Projekt, Abschlussarbeit oder als Grundidee eines Schul- oder Klassenprojekts umsetzen und bietet zudem eine Vielzahl an Erweiterungsmöglichkeiten.

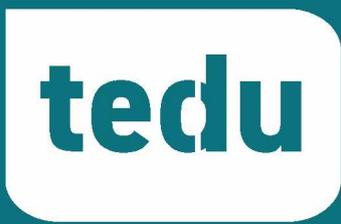
Die im Projekt erlernten Grundfertigkeiten und Kompetenzen beim Bau einer automatisierten Gewächshausbewässerung könnte viele Schülerinnen und Schüler in ähnliche Projekte im heimischen Garten oder auf dem eigenen Balkon selbst in die Tat umsetzen. Das Projekt setzt mit seiner Abgeschlossenheit und Einsetzbarkeit den Grundstein und lässt viele Möglichkeiten und Perspektiven auf einen digital vernetzten Garten bzw. eine digital vernetzte Wohnung offen.

## Autoreninformation

### Hannes Groß

studiert an der PH Schwäbisch Gmünd Lehramt für die Sekundarstufe 1 (B.A./M.Ed.) mit dem Fach Technik. In seiner Bachelorarbeit beschäftigt er sich momentan mit der Frage, wie das Internet der Dinge (IoT) im Technikunterricht thematisiert und für die Bildung nutzbar gemacht werden kann.





2|2021