

## Automatische Bewässerung einer Gemüsegärtnerei um Füllstandsmessung erweitern

Fach	IT-Technik
Lernfeld	LF 7: Cyber-physische Systeme ergänzen
Querverweise zu weiteren Lernfeldern des Lehrplans	<p>LF 2: Arbeitsplätze nach Kundenwunsch ausstatten</p> <p>LF 3: Clients in Netzwerke einbinden</p> <p>LF 8: Daten systemübergreifend bereitstellen</p>
Zeitrahmen	10 Unterrichtsstunden (5 Doppelstunden)
Benötigtes Material	<p>SBC's (alternativ <math>\mu</math>C oder Kleinsteuerung) mit - falls erforderlich - Erweiterungskarten für die A/D-Wandlung, Laptops als Programmiergeräte für die SBC's und als Clients für das eingesetzte IoT-Kommunikationsprotokoll, Entwicklungsumgebung (Editor, Compiler, Debugger etc.) für gängige Skript- bzw. Programmiersprache, Füllstandssensoren, Optional: Wassergefäße zur Füllstandssimulation oder im Idealfall Hardware-Modell der Regenwasserbewässerung, Internetzugang sowie kabelgebundene oder kabellose Netzwerkanbindung, Moderatorenkoffer, Posterpapier oder Flipcharts etc., Optional: Zugang zu Online-Nachrichtendienst bzw. Webmail-Anbieter o.ä.</p>

### Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- ... erfassen und visualisieren das Zusammenwirken der Komponenten cyber-physischer Systeme. (Medienkompetenz)
- ... setzen Vorgehensmodelle und -methoden sowie Entwicklungsumgebungen und -bibliotheken ein. (Anwendungs-Know-how)
- ... erstellen Testkonzepte, führen Tests durch und dokumentieren diese. (Anwendungs-Know-how)

- ... erstellen Programme, passen diese an und konfigurieren Signal- und Datenübertragungseinrichtungen. (Informatische Grundkenntnisse)

## Aufgabe

### 1. Orientieren und Informieren:

#### *2 Unterrichtsstunden*

Der Inhaber einer Gemüsegärtnerei bewässert Jungpflanzen im Anzuchtgewächshaus zeitgesteuert aus einer Regenwasserzisterne. Im fingierten Kundengespräch erläutert die Lehrkraft den Schülerinnen und Schülern als Auftragnehmern die bisher vorhandene, mit Hilfe eines SBC's (hier: Raspberry Pi) realisierte Zeitsteuerung, die in einer gängigen Programmier- bzw. Skriptsprache (hier: Python) implementiert wurde. Zusätzlich stellt die Lehrkraft die umzusetzende Erweiterung der Bewässerungssteuerung dar, die zukünftig den Füllstand der Regenwasserzisterne stetig messen soll. Die Messdaten sollen über ein geläufiges IoT-Kommunikationsprotokoll (hier: MQTT) an einen Client weitervermittelt werden, um sie dort grafisch aufbereiten zu können (LF 8). Im gleichen Benutzerdialog soll das Gärtnerpersonal mit Farbsymbolen auf unkritischen, niedrigen bzw. zu niedrigen (kritischen) Füllstand der Zisterne hingewiesen werden. *Optional: Um für die Anlagenbediener zusätzlichen Komfort zu schaffen, soll ein kritischer Wasserstand mit Hilfe einer Push-Benachrichtigung an ein mobiles Endgerät gemeldet werden.* Die Schülerinnen und Schüler protokollieren die im Klassenplenum vorgetragenen Anforderungen in Einzelarbeit informell.

Auf Grundlage ihrer Protokolle informieren sich die Schülerinnen und Schüler anschließend in Partnerarbeit mittels Internetrecherche, bei der sie durch Recherchehinweise von der Lehrkraft gecoacht werden. Dabei gilt es vor allem, offizielle Webseiten (z.B. [python.org](http://python.org), [mosquitto.org](http://mosquitto.org) etc.) und informative Foren zur für die Implementierung der Steuerungserweiterung einzusetzenden Hardware (SBC, Sensorik, Erweiterungskarten etc.) und Software (Betriebssystem, Skriptsprache, IoT-Kommunikationsprotokoll etc.) zu eruieren und beispielsweise als Favoriten im Browser verfügbar zu machen, die während der Planungs- und Durchführungsphasen als Informationsquellen genutzt werden können.

## 2. Planen:

### *2 Unterrichtsstunden*

Anhand ihrer Protokolle und der recherchierten Informationen entwickeln die Schülerinnen und Schüler in der Planungsphase in Partnerarbeit ein Technologieschema (Abschnitt "Schülerlösungen"), in dem sie sowohl die bereits bestehende Bewässerungssteuerung als auch die vorgesehenen Erweiterungen unter Verwendung von Stift und Papier übersichtlich darstellen. Zusätzlich extrahieren sie aus ihrem Technologieschema eine Liste der zu beschaffenden bzw. zu installierenden Hardware bzw. Softwarepakete. **Optional:** *Technologieschema und Beschaffungsliste können auf Kosten effektiver Unterrichtszeit auch digital erstellt werden. Der Lerneffekt ist dabei aber - bezogen auf LF 7 - verhältnismäßig gering!* Die Lehrkraft sollte in der Planungsphase speziell darauf achten, dass die für die Durchführungsphase wichtigen Aspekte - gerade Hard- und Software betreffend - in den Technologieschemata und Beschaffungslisten der Schülerinnen und Schüler Berücksichtigung finden.

Als Ergänzung zum Technologieschema skizzieren die Schülerinnen und Schüler die Anbindung des - von der Lehrkraft vorausgewählten - Füllstandssensors (hier: [sainsmart.com](http://sainsmart.com)) an die bestehende Steuerung. Abhängig von der Sensorauswahl (z.B. Abstandsmessung über Ultraschall, füllstandsabhängige Widerstandsmessung, Füllstandssimulation über Potentiometer etc.) sind in der Skizze (Abschnitt "Schülerlösungen") nötigenfalls Erweiterungsboards (hier: MCP3008, [microchip.com](http://microchip.com)) für die A/D-Wandlung bzw. Bussysteme für die Signalübertragung zu berücksichtigen.

**Optional:** *Für die Realisierung der Push-Benachrichtigung wegen eines kritischen Wasserstands der Regenwasserzisterne müssen in der Planung entsprechende technische Möglichkeiten Berücksichtigung finden. Unter Umständen sind dabei auch Konten bei Messenger-Diensten (z.B. Twitter) einzurichten, die über eine einfach einzusetzende API verfügen.*

## 3. Durchführen:

### *4 Unterrichtsstunden*

Die Durchführungsphase des hier beschriebenen illustrierenden Beispiels besteht aus zwei - bei Umsetzung der Push-Benachrichtigung drei - Arbeitsschritten, die die

Schülerinnen und Schüler in Partnerarbeit in den bereits bestehenden Schülertandems ausführen. In einem ersten Schritt realisieren sie die hard- und softwaremäßige Anbindung des von der Lehrkraft zur Verfügung gestellten Füllstandssensors an den bisher verwendeten SBC. Wie bereits erwähnt, muss dabei nötigenfalls ein Erweiterungsboard für die A/D-Wandlung mit eingebunden werden (in diesem illustrierenden Bsp.: MCP3008 am SPI des Raspberry Pi, s.o.). Für den Unterrichtseinsatz sollten die erforderlichen Materialien über von den Schülerinnen und Schülern einfach zu handhabende Steckverbindungen, einadrige Leitungen und z.B. Breadboards miteinander verbunden werden können. Zusätzlich zur Sensoranbindung muss das bereits vorhandene Steuerungsskript (Abschnitt "Schülerlösungen") für die Verarbeitung der Sensormessdaten angepasst und erweitert werden. Die Sensordaten werden im hier dargestellten Python-Skript über SPI eingelesen und dann weiterverarbeitet. Dafür ist der Import der entsprechenden Programmierbibliotheken erforderlich. Für die grafische Darstellung des Füllstandsverlaufs auf beispielsweise einem MQTT-Client müssen die Sensordaten normiert werden. Außerdem ist ein Datum festzulegen, das die Meldung über unkritische, niedrige bzw. kritische Füllstände der Regenwasserzisterne über das IoT-Kommunikationsprotokoll (hier: MQTT) ermöglicht.

Im zweiten Schritt installieren die Schülertandems die für die IoT-Kommunikation nötigen Softwarepakete und Bibliotheken auf dem SBC. In diesem illustrierenden Beispiel kommen dafür der MQTT-Broker Mosquitto (mosquitto.org), auf den MQTT-Clients als Publisher oder Subscriber (dann auch mittels API, LF 8) zugreifen können, sowie eine als MQTT-Client verwendbare Python-Bibliothek (mosquitto.org, Paho MQTT) zum Einsatz. Über die von Paho MQTT zur Verfügung gestellten Schnittstellen-Funktionen können die Messdaten zum Zisternenfüllstand aus dem Python-Skript heraus an den (möglichst auf demselben Gadget laufenden) MQTT-Broker weitergeleitet werden, von wo aus sie auch von Remote-Clients für die Weiterverarbeitung abgerufen werden können (LF 8). Für den MQTT-Broker müssen die Schülerinnen und Schüler in Partnerarbeit außerdem die entsprechenden Konfigurationen und Betriebssystemanpassungen sowie im Steuerungsskript (Abschnitt "Schülerlösungen") die notwendigen Erweiterungen vornehmen. Um sich zu informieren, nutzen die Schülertandems die in der Orientierungs- und Informationsphase gefundenen Quellen wie z.B. Websites zu MQTT und/oder entsprechende Foren.

**Optional:** Die Schülertandems schließen die Durchführungsphase ab, indem sie für das Gärtnerpersonal als ergänzende Komfortfunktion eine Push-Benachrichtigung über

*einen Messengerdienst oder per E-Mail-Benachrichtigung realisieren. In der hier dargestellten illustrierenden Lernsituation nutzen sie dafür eine Python-Bibliothek, die den Zugriff auf SMTP-Funktionen aus ihrem Steuerungsskript heraus ermöglicht. Bei zu niedrigem, also kritischem Füllstand der Regenwasserzisterne wird eine E-Mail (von einem Web-E-Mail-Account) an einen (anderen) Web-E-Mail-Account versendet, um das Bedienpersonal zu benachrichtigen. Die Schülerinnen und Schüler sind hier im Sinne der User-Experience gefordert, die Anzahl der versendeten E-Mails zu begrenzen und den Mailversand auf sinnvolle Art und Weise mit dem Füllstandsverlauf der Regenwasserzisterne zu verknüpfen.*

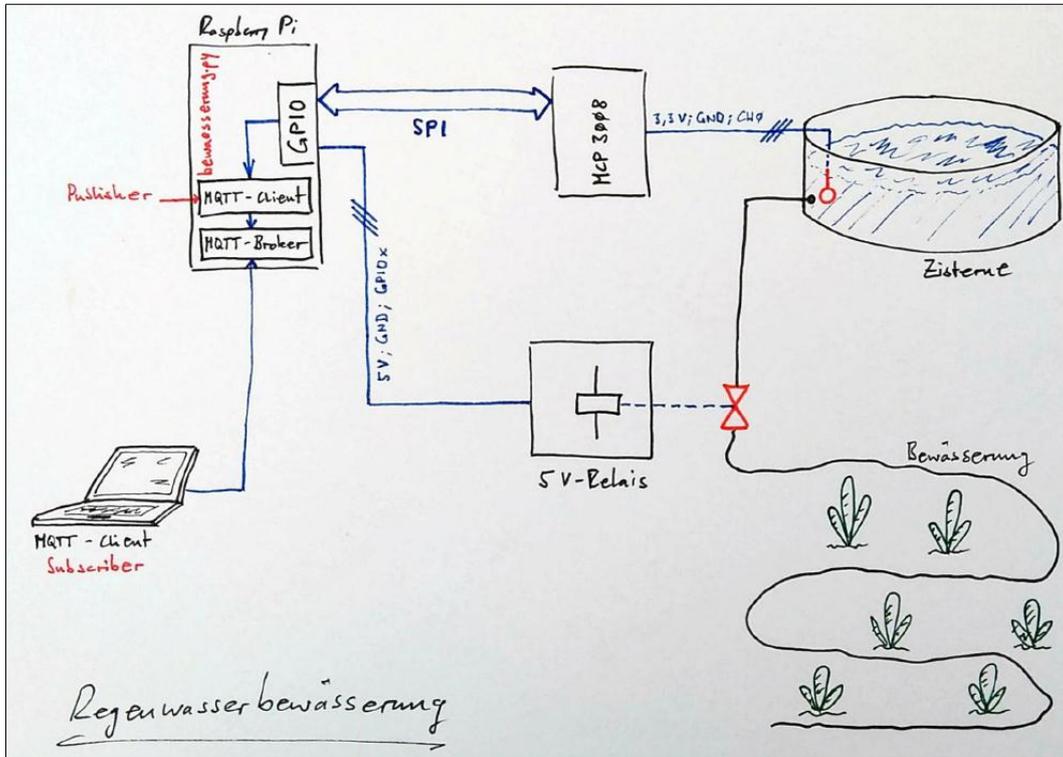
#### **4. Kontrollieren und Bewerten:**

*2 Unterrichtsstunden*

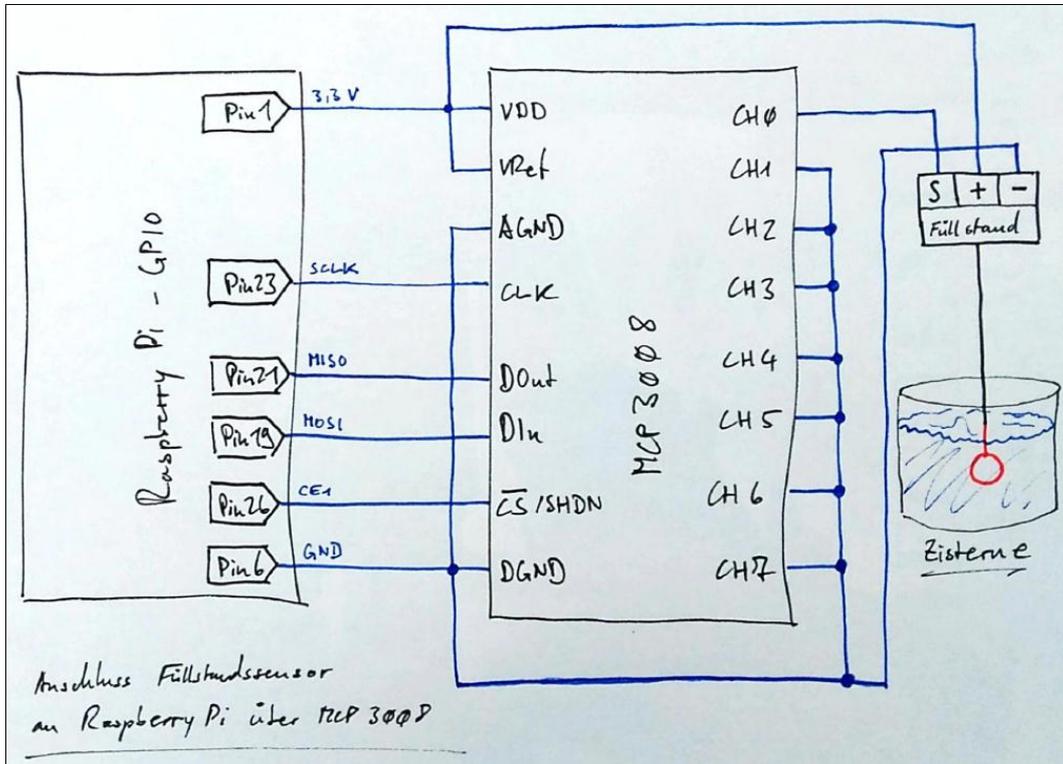
Den Abschluss dieses illustrierenden Beispiels bildet ein Entwicklertest anhand eines Testprotokolls, den die bereits bestehenden Schülertandems durchführen. Die Testprotokolle sollten von den Schülerinnen und Schülern schon während Planung und Durchführung entworfen und weiterentwickelt werden. Es bietet sich an, in der Phase „Kontrollieren und Bewerten“ Testprotokolle und in den Planungs- und Durchführungsphasen erstellte Schülerartefakte zwischen jeweils zwei Tandems auszutauschen und die Tests gegenseitig durchführen zu lassen. Dabei bietet sich den Schülerinnen und Schülern die Möglichkeit, auftretende Fehler und Ungereimtheiten im wechselseitigen Austausch zu diskutieren. Auf dieser Grundlage können Lösungsvorschläge zur Beseitigung von Fehlern in Gruppenarbeit formuliert werden, die bei verfügbarer Unterrichtszeit in die jeweiligen Schülerlösungen eingearbeitet werden können und damit die gesamte Lernsituation abrunden.

## Beispiele für Produkte und Lösungen der Schülerinnen und Schüler

Planen: Technologieschema



Planen: Anschluss-Skizze Füllstandssensor



Durchführen: Steuerungsskript bewaesserung.py

```

1  #!/usr/bin/python3
2  #bewaesserung.py
3
4  #Import der benötigten PyPI-Packages
5  import time
6  import RPi.GPIO as GPIO
7  import spidev
8  import paho.mqtt.publish as publish
9  import smtplib, sys
10 from email.mime.text import MIMEText
11 from email.header import Header
12
13 #Initialisierung
14 emailSent = False
15 waterLevel = 0
16
17 spi = spidev.SpiDev()
18 spi.open(0,1)
19 spi.max_speed_hz = 5000
20
21 frm = 'gemuese-bauer@gmail.com'
22 to = 'info@gemuese-bauer.de'
23 subj = 'WARNMELDUNG Zisterne!!!'
24 msg = 'Wasserstand KRITISCH!'
25 gmail_user = 'gemuesebauer'
26 gmail_pwd = 'Wir_bewaessern_nachhaltig'
27
28 #Bei Programmabbruch Grundzustand des Systems herstellen
29 def clean_up():
30     finish = input('Automatische Bewaesserung mit RETURN beenden! ')
31     GPIO.cleanup()
32     return
33
34 #Bewässerung starten (True, Pegel LOW) bzw. stoppen (False, Pegel HIGH)
35 def automatic_watering(switchOn):
36     if switchOn==True:
37         GPIO.output(7, GPIO.LOW)
38     else:
39         #Warnmeldungen unterdrücken
40         GPIO.setwarnings(False)
41         GPIO.output(7, GPIO.HIGH)
42     return
43
44 #Wasserstandsmessung durchführen
45 def get_water_level():
46     value = spi.xfer([1,128,0])
47     if 0 <= value[1] <= 3:
48         #Wasserstand in cm (Wasserglas-Simulation! Sturm im ...)
49         level = ((value[1] * 256) + value[2]) * 0.01
50         return level
51     else:
52         return 0
53
54 #Bei kritischem Wasserstand Warnmeldung senden
55 def send_email():
56     mime = MIMEText(msg, 'plain', 'utf-8')
57     mime['From'] = frm
58     mime['To'] = to
59     mime['Subject'] = Header(subj, 'utf-8')
60
61     smtpserver = smtplib.SMTP("smtp.gmail.com",587)
62     smtpserver.starttls()
63     smtpserver.login(gmail_user, gmail_pwd)
64     smtpserver.sendmail(frm, [to], mime.as_string())
65     smtpserver.quit()
66
67     return
68
69
70 #Main
71 try:
72

```

```

73     #Pin- anstelle der GPIO-Nummern verwenden
74     GPIO.setmode(GPIO.BOARD)
75     GPIO.setup(7, GPIO.OUT)
76
77     while True:
78
79         #Füllstandsmessung 6x pro halbe Stunde durchführen
80         #und Messwert auf MQTT-Broker publizieren
81         for cntMeasurement in range(1,7):
82             waterLevel = get_water_level()
83             publish.single("bewaesserung/wasserstand", payload=str(waterLevel),
84                             qos=0, retain=False, hostname="localhost",
85                             port=1883, client_id="", keepalive=60, will=None, auth=None,
86                             tls=None, protocol=mqtt.MQTTv31)
87
88             #Warnmeldung bei kritischem Wasserstand absetzen
89             if(waterLevel<1):
90                 if emailSent==False:
91                     send_email()
92                     emailSent = True
93                 else:
94                     emailSent = False
95
96             if(cntMeasurement<6): time.sleep(300)
97
98             #Bei unkritischem Wasserstand 1x pro halbe Stunde automatisch bewässern
99             if(waterLevel>=1):
100                 automatic_watering(True)
101                 time.sleep(300)
102                 automatic_watering(False)
103             else:
104                 time.sleep(300)
105
106     #Bei Strg+C abbrechen und Grundzustand herstellen!
107     except KeyboardInterrupt:
108         clean_up()
109
110

```

## Hinweis zum Unterricht

Fachliche Informationen, die zur Umsetzung des hier vorgestellten illustrierenden Beispiels benötigt werden, können zum Großteil dem vorgegebenen und von den Schülerinnen und Schülern zu erweiternden sowie kommentierten Python-Steuerungsskript sowie Technologieschema und Anschluss-Skizze für das Erweiterungsboard entnommen werden. Was Ergänzungen, Änderungen und Erweiterungen angeht, sind in dieser Unterrichtssituation Fantasie und Kreativität keine Grenzen gesetzt, gerade dann, wenn man auch Gemüsegeärtnereien oder die sogenannte Home Automation (auch: Smart Home, Gardening 4.0, Smart Gardening etc.) in die Unterrichtsentwicklung mit einbezieht.

## Querverweise zu anderen Fächern / Fachrichtungen

Werden bereits die Lernfelder 2 und 3 der 10. Jahrgangsstufe so geplant, dass SBC's zur Vermittlung grundlegender Hardwarekenntnisse (*LF 2*) eingesetzt bzw. als Clients in ein bestehendes Netzwerk eingebunden werden (*LF 3*), können die entsprechenden Vorkenntnisse der Schülerinnen und Schüler die Arbeit in *LF 7* im Sinne eines wirkungsvollen Einsatzes von Unterrichtszeit deutlich erleichtern. Gleiches gilt für rudimentäre Kenntnisse zur Linux-Kommandozeile und zu unter Linux verwendeten Benutzer- und Dateisystemstrukturen: „7 flies in one swat!“

## Quellen- und Literaturangaben

- Microchip Technology Inc., 2355 West Chandler Blvd., Chandler, Arizona, USA , <https://www.microchip.com/wwwproducts/en/en010530> (Zugriff 13-06-2020. 15:07 MEZ)
- Vastmind LLC, 251 Little Falls Drive, Wilmington, New Castle, DE 19808 USA , <https://www.sainsmart.com/products/water-sensor-with-free-cables> (Zugriff 02-06-2020. 16:15 MEZ)
- Python Software Foundation, 9450 SW Gemini Dr., ECM# 90772, Beaverton, OR 97008, USA, <https://www.python.org> (Zugriff 14-06-2020. 14:30 MEZ)
- Eclipse Foundation, Inc., 2934 Baseline Road, Suite 202, Ottawa, Ontario, Canada, <http://mosquitto.org> (Zugriff 14-06-2020. 15:23 MEZ)
- HiveMQ GmbH, Ergoldingerstr. 2a, 84030 Landshut, Bavaria, Germany <https://www.hivemq.com/blog/mqtt-client-library-paho-python> (Zugriff 15-07-2020. 15:50 MEZ)
- Kofler, Kühnast, Scherbeck (2020): Raspberry Pi, Das umfassende Handbuch, 6. Auflage, Bonn Verlag: Rheinwerk Verlag GmbH, 2020