

Assessment

Vorname / Name: *(via GitHub)*

Punkte: ____ / 60, Note: ____

Klasse: 5ibb1

Frei lassen für Korrektur.

Hilfsmittel:

- Aufgaben werden allein, am eigenen Computer gelöst.
- Alle Unterlagen (Slides, Bücher) sind erlaubt, im Sinn von *open book*.

Nicht erlaubt:

- Jegliche Kommunikation mit anderen Personen.

Bewertung:

- Offene Fragen: Bewertet wird die Korrektheit, Vollständigkeit und Kürze Ihrer Antwort.
- Programme: Bewertet wird die inhaltliche Umsetzung, aber nicht ob es kompiliert.

Fragen zur Prüfung:

- Während der Prüfung werden vom Dozenten keine Fragen zur Prüfung beantwortet.
- Ist etwas unklar, machen Sie eine Annahme und notieren Sie diese bei Ihrer Lösung.

Bearbeiten der Aufgaben:

- Lesen Sie die Aufgabenstellungen in dieser PDF Datei.
- Bearbeiten Sie die bestehenden *TXT* Dateien im Repo.

Abgabe via GitHub:

- Committen Sie alle Änderungen an bestehenden *TXT* Dateien mit

```
$ git commit -m "update" *.*
```
- Übermitteln Sie alle lokalen Commits an GitHub mit

```
$ git push
```
- Es zählt der Stand *auf GitHub* beim letzten Commit vor dem / am Ende der Prüfung.

o: name.txt

Wie lautet Ihr Vorname und Name?

Verbindlich

Antworten Sie in der Datei *name.txt*.

1: humialert.txt

Ändern Sie das in der Datei *humialert.txt* gegebene Arduino Programm eines Feuchtigkeits-alarms so, dass die hier gezeigte State Machine korrekt implementiert wird. Punkte: ___ / 8

```

b:  user button
h:  humidity %
bl: blue LED
rl: red LED
t0: time

+----+                                     +----+
| S0 |---pressed(b)|threshold=h+10%;rl=on----->| S1 |
+----+                                     +----+
    ^                                         |
    |                                         |
    |                                         |h>threshold|rl=off;bl=on
millis()-t0>1000|rl=off                      |
    |                                         v
+----+                                     +----+
| S3 |<---pressed(b)|rl=on;bl=off;t0=millis()---| S2 |
+----+                                     +----+

```

Fügen Sie Ihren Code bzw. Ihre Änderungen in die Datei *humialert.txt* ein.

2: projects.txt

Welche IoT Architektur und Hardware passt am besten zu welchem Projekt? Punkte: ___ / 13

In der Datei *projects.txt* finden Sie Architektur-Referenzmodelle für je eines dieser Projekte:

- *Wilderer-Alarm*, welcher bei Jagd in einem Waldstück im Hauptquartier Alarm schlägt.
- *Ski-Logger*, der jede Skiabfahrt aufzeichnet und die Anzahl Abfahrten auf dem Ski angibt.
- *Futterspender*, der von einer Katze selber ausgelöst, oder per App gesteuert werden kann.

Ihre Hardware ist: 1 *Accelerometer*, 1 *BLE Modul*, 1 *Buzzer*, 1 *LED Strip*, 1 *LoRaWAN Modul*, 1 *Mikrofon*, 1 *Motor*, 1 *PIR Sensor*, 1 *Wi-Fi Modul*, plus ein weiteres Funkmodul Ihrer Wahl.

Füllen Sie alle mit "..." markierten Stellen in der Datei *projects.txt* mit den obigen Begriffen.

3: senddata.txt

Ändern Sie das in der Datei *senddata.txt* gegebene Programm so ab, dass es statt ThingSpeak das hier mittels Curl gezeigte API verwendet, ohne eine zusätzliche Library. Punkte: ___ / 8

```
$ curl -vX POST -d 'value=42' \
"https://io.adafruit.com/api/v2/IO_USER/feeds/FEED/data?x-aio-key=IO_KEY"
```

Fügen Sie Ihren Code bzw. Ihre Änderungen in die Datei *senddata.txt* ein.

4: quakehouse.txt

Gegeben die folgenden MQTT Topics eines Hauses mit Erdbebenalarm: Punkte: ___ / 13

```
house
  /quake "none|alert"
  /rooms
    /ROOM_ID
      /switch "on|off"
      /lights
        /LIGHT_ID "on|off"
```

Ergänzen Sie das Node.js Programm in der Datei *quakehouse.txt* um folgende Funktionalität:

- Beim Betätigen eines Schalters werden alle Lichter im selben Raum ein- bzw. ausgeschaltet.
- Bei einem Erdbeben (quake) Alarm werden alle Lichter im ganzen Haus einschaltet.

Node.js Beispiele sind hier: <https://github.com/tamberg/fhnw-iot/tree/master/07/Nodejs>

Nutzen Sie Wildcards der Form "a/+/"c", um mehrere Topics (oder hier IDs) zu matchen.

Sowie die `parts = topic.split("/")` Funktion, um einen String in ein Array zu zerlegen.

Und `topic.startsWith(prefix)`, um in Node.js den Anfang eines Strings zu prüfen.

Fügen Sie Ihren Code bzw. Ihre Ergänzungen in die Datei *quakehouse.txt* ein.

5: blescale.txt

Gegeben die BLE Spezifikationen [WSS V1.0.0.pdf](#) und [16-bit UUID Numbers Document.pdf](#), suchen Sie die wesentlichen Angaben zum GATT API einer BLE Personenwaage, und tragen Sie diese in die Datei *blescale.txt* ein. Geben Sie zudem einen Hinweis darauf, wo (Dokument, Seite, Abschnitt) steht, wie man die Bits einer Messung interpretieren würde. Punkte: ___ / 8

Fügen Sie Ihre Antworten in die Datei *blescale.txt* ein.

6: loralock.txt

Welche drei Gründe sprechen dagegen, ein aus dem Internet ferngesteuertes Türschloss mit LoRaWAN zu bauen, und welche drei Vorteile hätte ein solches Produkt? Punkte: ___ / 6

Fügen Sie Ihre Antworten in die Datei loralock.txt ein.

7: voiceframes.txt

Welche zwei Use Cases werden möglich, wenn eine Sprachsteuerung wie Alexa in eine Brille mit "open-ear" Audio eingebaut wird, was könnten zwei Probleme damit sein? Punkte: ___ / 4

Fügen Sie Ihre Antworten in die Datei voiceframes.txt ein.