

TODO

Wecker (Wecker.ino):

- `getCurrentTime()` : int
 - Gibt die aktuelle Uhrzeit zurück. Holt sich diese über WLAN. Methode wird in `setup()` aufgerufen. Danach wird die Zeit über `millis()` weiterberechnet.
- `updateTime()`
 - Aktualisiert die Uhrzeit, welche als Attribut gespeichert wird. Dazu kann `millis()` und die Startuhrzeit, welche in `setup()` geladen wird, genutzt werden.
 - Ruft `alarm()` auf, wenn die Uhrzeit mit der aktualisierten Weckzeit übereinstimmt.
- `alarm()`
 - Startet den Alarm und setzt ein globales Attribut „alarmAktiv“ auf true. Der Alarm wird so lange abgespielt, bis „alarmAktiv“ false ist. Dazu muss ein Buzzer angesteuert werden.
- `onMqttMessage(int messageSize)`
 - Ist schon vorhanden. Muss so bearbeitet werden, dass auch auf das Topic mit dem Alarm-Stopp-Signal geprüft wird. Wenn ein Signal erhalten wurde, wird das globale Attribut „alarmAktiv“ auf false gesetzt.

Display (Wecker.ino):

- `displayTime(int hour, int minute)`
 - Die aktuelle Zeit wird auf dem Display angezeigt. Dazu wird die Adafruit-Library genutzt und Inputs und Outputs in `setup()` gesetzt.

Mobiler-Aus-Knopf (neu zu erstellende Arduino-Projektdatei):

- `sendSignal(String signal)`
 - Es wird ein Signal an den Wecker gesendet. Dazu müssen Broker und Topic mit dem Wecker übereinstimmen.
- Das Signal wird gesendet, wenn der Knopf gedrückt wurde. Diese Bedingung wird in der `loop()` geprüft. Außerdem muss der digitale Pin des Knopfes definiert- und als Input in `setup()` gesetzt werden.