The Drunken Clock

Motivation

Starke und durchzechte Nächte sind oft der Grund für das Verschlafen. Das Verschlafen von Vorlesungen ist oft ein großes Problem, da eine Abwesenheit auch bedeutet, dass der Unterrichtsstoff verpasst wird. Besonders in der Situation eines Studenten kann das schwerwiegende Nachteile haben. Unsere persönlichen Erfahrungen mit dem Verschlafen und unsere Frustration über einen Tiefschlaf, verursacht durch einen hohen Alkoholkonsum, haben uns auf die Idee gebracht, eine endgültige Lösung für dieses Problem zu finden.

Szenario

Serhat war gestern auf dem FKF. Da er um 08:50 zur nächsten Vorlesung muss, benötigt er einen verlässlichen Wecker. Serhat tendiert gerne länger zu schlafen und den Wecker auszuschalten.
The Drunken Clock erkennt, dass Serhat alkoholisiert ist und verändert dadurch den Weckton. Der erste Wecker befindet sich neben dem Bett von Serhat und kann auch da deaktiviert werden. Mit Hilfe des Displays kann Serhat die aktuelle Uhrzeit und weitere Daten einsehen. Nachdem Serhat den ersten Wecker deaktiviert hat, ertönt nach 5 Minuten der zweite Wecker. Dieser muss mittels RFID deaktiviert werden. So wird Serhat gezwungen, tatsächlich aufzustehen, um den zweiten Wecker abzuschalten.

Projektidee

Unser Projekt hat zwei Wecker. Beim ersten Wecker muss ein Alarm (Uhrzeit) eingestellt werden. Dies funktioniert über die Blynk App. Der erste Wecker hat einen Alkoholsensor, der testet, ob der Nutzer alkoholisiert ist. Dieser misst den Gehalt von Alkohol-Gasen in der Luft. Falls er alkoholisiert ist, wird die Melodie bei beiden Weckern umgestellt. Wenn der erste Wecker klingelt, muss der Nutzer auf einen Button drücken, damit er aufhört zu läuten. Fünf Minuten nach dem Läuten vom ersten Wecker schaltet sich der zweite Wecker ein. Dieser kann nur ausgeschaltet werden, indem man den Chip oder die Karte über den Scanner hält. Die Identifikation findet durch die Seriennummer des Chips statt.

Einsatzgebiet

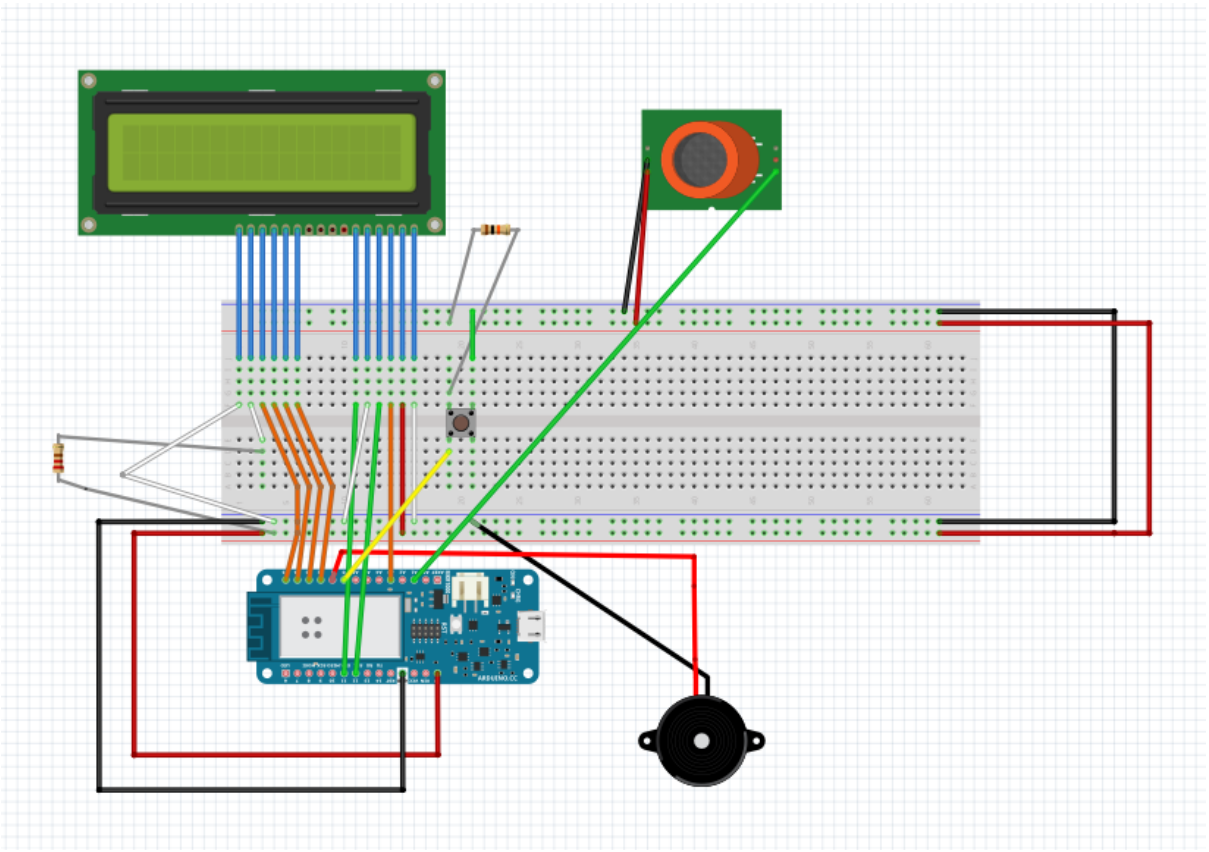
The Drunken Clock kann generell allen Interessenten zur Verfügung gestellt werden. Besonders für Studenten, mit einem Alkoholproblem, wäre unsere Drunken Clock ein gutes Werkzeug, um es rechtzeitig aus dem Bett zu schaffen und nie wieder Vorlesungen zu verpassen. Aber auch für Arbeitende, die gerne tief ins Glas schauen, ist The Drunken Clock ein Berufsretter.

Komponenten

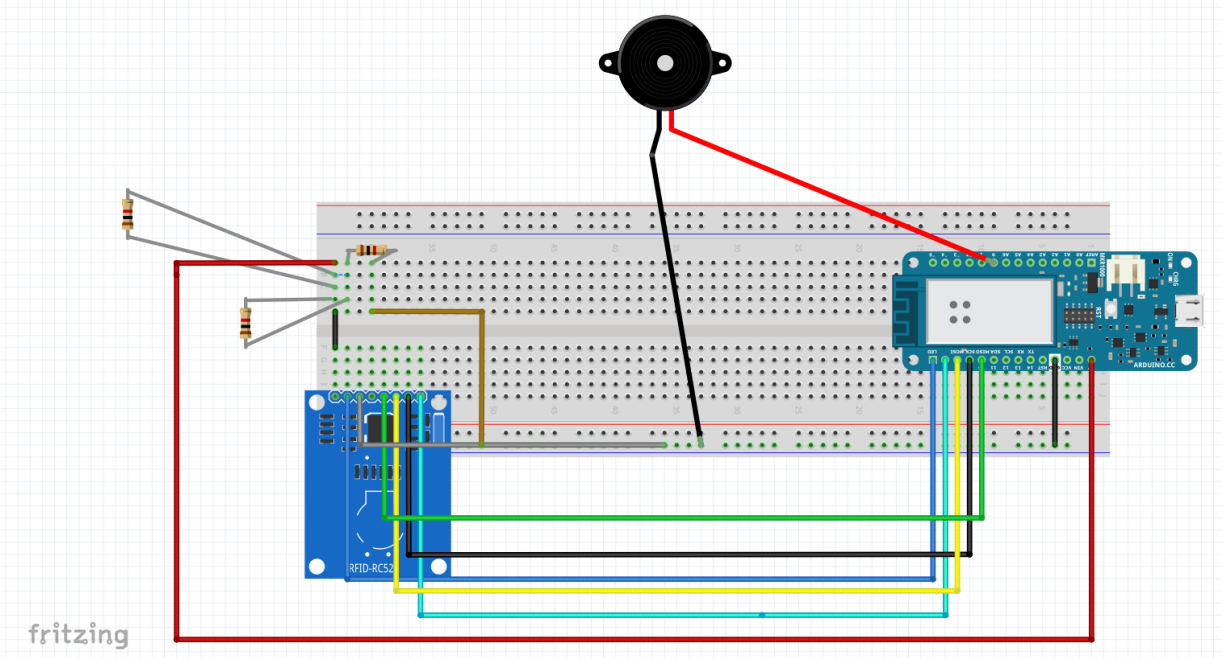
- Allgemein
    - Buzzer
        - jeweils ein Buzzer pro Wecker
        - wenn der Wecker aktiv ist, läutet dieser zur gewählten Uhrzeit
- Main (Arduino MKR1000)
    - Alcohol Sensor (MQ 3)
        - https://www.mikroe.com/alcohol-click
        - misst den Alkoholgehalt in der Luft
        - Veränderung des Weckmusters, wenn dieser Wert über einem bestimmten Schwellwert ist
        - Funktionsweise
            - 5V und Ground
            - Analoger Output
            - Schwellwert war bei ca. 850
    - LCD
        - Uhrzeit
        - Alarmzeit
        - Ist der Alarm aktiviert?
        - Ist der Benutzer betrunken?
    - Button
        - deaktiviert den ersten Wecker durch Drücken
        - 10 kΩ Widerstand vor den Button
- Second (Arduino MKR1000)
    - RFID-Modul (RC522)
        - https://www.az-delivery.de/products/rfid-set
        - deaktiviert den zweiten Wecker
        - überprüft, ob der Benutzer tatsächlich aufgestanden ist
            - nur mit dem richtigen RFID-Chip
        - Identifikation findet durch die Seriennummer des Chips statt.
        - <10 cm Leseabstand
        - Funktionsweise
            - 3.3 V, Ground und RST
            - SDA, SCK, MOSI, MISO
        - Spannung mit 3 x 1 kΩ Widerständen von 5 V auf 3.3 V reduziert

Schaltplan

Erster Wecker:



Zweiter Wecker:



Beschreibung der Software

Blynk

Blynk wird als Konfigurations-App verwendet, bei der man die gewünschte Uhrzeit einstellen bzw. den Wecker ein- und ausschalten kann.

Blynk Channels

- V0 - Status von User
    - 0 = Nicht betrunken
    - 1 = Betrunken
- V1 - Stellen der Alarm Zeit
    - Zeit vom aktuellen Tag in Sekunden ab 00:00 Uhr
- V2 - Alarm Aktiv
    - 0 = Alarm ist deaktiviert
    - 1 = Alarm ist aktiviert

Real Time Clock

- RTC Blynk (WidgetRTC.h)
- setup()
    - sekundlichen Timer initialisieren
        - ruft die changeTime-Methode auf
- BLYNK_CONNECTED
    - RTC synchronisieren

Song Player - basierend auf EasyBuzzer library

- setSong() - Song setzen (unterschiedliche Files und Noten Arrays)
- start() - Song starten
- stop() - Song stoppen (noTone auf Buzzer setzen)
- update() - wird in loop() aufgerufen
    - Check ob der Song gestoppt wurde -> nicht abspielen
    - aktuellen Index von Noten Array abspielen
    - Index erhöhen (nächste Note)

Erster Wecker

- **setup()**
    - Blynk Setup (Verbindung mit WIFI und Cloud)
    - Button als Input setzen
    - LCD Setup
        - Helligkeit von LCD auf Maximum

            - Nummer der Zeilen und Spalten festlegen
- **loop()**
    - Alkoholwert einlesen (analogRead)
    - falls der Wecker aktiviert ist und der Alkoholwert über den Schwellwert liegt
        - Benutzer ist alkoholisiert → anderes Weckmuster
        - auf LCD schreiben
    - falls der Wecker gerade läutet
        - Button einlesen (digitalRead)
        - wenn Button == LOW → stop
        - sonst update
- **changeTime()**
    - falls der Wecker aktiviert ist und die aktuelle Zeit gleich der Alarmzeit ist
        - Wecker starten
        - Weckton abhängig von Alkohol-Status
            - Drunk: Cantina Band
            - Sober: Star Wars Theme

Zweiter Wecker

- **setup()**
    - Initialisierung von SPI Bus and RFID Reader
    - Blynk Setup (Verbindung mit WIFI und Cloud)
- **loop()**
    - Check ob der Alarm gerade läutet
    - Falls ja,
        - RFID einlesen und abfragen ob es der Tag für den User ist
        - Alkoholwert zurücksetzen
        - Alarmton stoppen
    - Falls nein,
        - loop fortsetzen
- **changeTimer()**
    - falls der Wecker aktiviert ist und die aktuelle Zeit gleich der Alarmzeit ist
        - Wecker starten
        - Weckton abhängig von Alkohol-Status
            - Drunk: Cantina Band
            - Sober: Star Wars Theme

Wie haben wir die Anforderungen erfüllt?

1) **2 Boards**: Wir haben ein Board, das als Hauptwecker dient. Das zweite Board übernimmt das Scannen des RFID-Chips und dient als zweiter Wecker.
2) Als Sensoren haben wir einen Alkoholsensor (MQ 3) und ein RFID-Modul (RC522) verwendet, welche nicht Teil des IOT Bundles sind.
3) **Sensoren:** Alkoholsensor, Button, RFID-Modul
   Aktuatoren: Buzzer, LCD-Display
4) Blynk zum Einstellen der Alarm Uhrzeit.

5) Wir haben die Blynk Cloud für die Kommunikation zwischen den Arduinos benutzt, um die aktuelle Uhrzeit zu ermitteln.