**Simulation der Tür mit Türklingel**

Für die Simulation der Tür mit Türklingel haben wir insgesamt zwei LEDs und einen Taster verwendet. Eine LED zeigt den Zustand der Tür, die zweite LED simuliert den Klingelton und der Taster simuliert den Klingelschalter.

Die Komponenten müssen in der Items-Datei eingebunden werden:

Switch raspiLED { gpio="pin:17" } //LED simuliert Tuer

Switch LED2 {gpio="pin:27"} //LED simuliert Klingelton

Contact taster "Taster"{ gpio="pin:22"}

Die GPIO-Belegung ist von der Raspberry Pi-Version abhängig. In unserem Fall, dem Raspberry Pi 3, waren es die Pins 17, 22 und 27.

Um die Komponenten auf diese Weise einbinden zu können, mussten die entsprechenden GPIO-Bindings im Verzeichnis Addons installiert werden.

**Simulation der Tür**

Für die Simulation der Tür haben wir zwei Rules geschrieben. Wenn die LED leuchtet, ist die Tür offen und wenn die LED aus ist, ist die Tür geschlossen:

rule tuerOeffnen

when

Item tuerSwitch received command ON

then

sendCommand(raspiLED, ON) // Tueroeffnung wird simuliert

end

rule tuerSchliessen

when

Item tuerSwitch received command OFF

then

sendCommand(raspiLED, OFF) // Tuerschliessung wird simuliert

end

**Simulation des Klingeltons**

Wenn der Taster betätigt wird, simuliert ein zwei Sekunden langes Leuchten der LED den Klingelton. Die letzten drei Zeilen dieser Rule implementieren diesen Vorgang:

rule klingeln

when

Item taster changed to OPEN

then

sendBroadcastNotification("Ding Dong!") // Push-Notification an das Smartphone

sendCommand(camSwitch,ON) // rule „KameraAN“ wird aktiviert

sendCommand(LED2,ON) // Klingelton wird simuliert

Thread::sleep(2000)

sendCommand(LED2,OFF)

end

**Simulation des Klingelschalters**

Wie an der Rule „klingeln“ zu erkennen ist, dient das Status-Update des Tasters als Trigger für die dann folgenden Befehle. Wird der Taster betätigt wird eine Push-Notification an die Smartphone-App gesendet und die Kamera und der Klingelton werden aktiviert.

Um den Taster zu entprellen, haben wir in der Datei openhab.cfg den Debounce auf 10 ms gestellt:

gpio:debounce=10