

Projekt: I2EA_SmaBui

Autor: Leard, David

Datum: 09.07.2025

Hochschule Bremen

Fakultät Elektrotechnik und Informatik

Veranstaltung: Informatik 2 – SoSe 2025

Inhaltsverzeichnis

1	Einleitung	2
2	Projektbeschreibung	2
3	Funktionsweise	2

1 Einleitung

Die folgende Projektdokumentation befasst sich im Rahmen der abschließenden Prüfungsleistung der Veranstaltung Informatik 2 mit dem Projekt „I2EA_SmaBui“.

2 Projektbeschreibung

Bei dem Projekt handelt es sich um ein „Smart Building“, in welchem als zentrale Recheneinheit ein Espressif ESP32-WROOM-32-Mikrocontroller verbaut ist. Bei diesem Modell eines intelligenten Gebäudes sind mehrere verschiedene Sensoren verbaut. Enthaltene Sensoren sind unter anderem ein Regensensor, ein Feuchtigkeits- und Temperatursensor, ein Gassensor, ein Bewegungssensor, ein RFID-Leser, zwei verschiedene Arten von LED-Modulen, ein LC-Display, ein passiver Summer, ein 180 Grad Servo und zwei Taster. Ein zweiter Servo ist verbaut, wurde aus zeitlichen Gründen nicht im Code implementiert. Mit der Hilfe von Visual Studio Code und PlatformIO wurde der zu implementierende Programmcode geschrieben. Dieser Code soll die verbauten Sensoren nach unseren Vorgaben zum Laufen bringen, wobei dieser erweiterbar und verständlich dokumentiert sein soll.

3 Funktionsweise

Zu Beginn verbindet sich das „Smart Building“ mit dem WLAN, gibt die IP-Adresse an und synchronisiert die interne Real Time Clock mit der aktuellen Uhrzeit. Daraufhin misst es die Umweltdaten durch die Sensoren und übermittelt sie alle fünf Minuten per MQTT-Protokoll an die Plattform ThingSpeak, welche die Daten speichert und visualisiert. Zusätzlich löst ein Alarm aus, wenn der Gassensor aktiviert wird. Mit der Hilfe des richtigen RFID-Chips ist es möglich die Tür zu öffnen. Dabei leuchtet die RGB-LED grün auf. Sollte man einen anderen RFID-Chip benutzen, öffnet die Tür nicht und die RGB-LED leuchtet rot. Die Buttons am Smart Building ermöglichen das Navigieren durch ein Menü zum Anzeigen der Werte auf dem LC-Display. Zusätzlich stellt das Smart Building eine webbasierte Bedienoberfläche bereit, die per WebSocket mit dem Gerät kommuniziert. Nach erfolgreicher Anmeldung mit Benutzernamen und Passwort erhalten autorisierte Benutzer Zugriff auf diese Oberfläche. Sie zeigt den aktuellen Zustand aller Sensoren an und erlaubt die Steuerung des Türmechanismus und der LED. Wenn der Bewegungssensor ausgelöst wird, geht die LED auf dem Dach an. Sie geht wieder aus, wenn der Bewegungssensor 10 Sekunden lang nicht ausgelöst wird. Für die Umsetzung dessen an unserem Projekt haben wir das State Pattern verwendet. Das State Pattern hilft dabei, ein Objekt je nach Zustand unterschiedlich reagieren zu lassen, ohne dass man überall if- oder switch-Anweisungen braucht. Jeder Zustand bekommt eine eigene Klasse und das Objekt, welches sich unterschiedlich verhalten soll, gibt das Verhalten an die Klasse seines aktuellen Zustands. Bei unserem Code gibt es zwei Zustände: IdleState (warten auf Bewegung) und ActiveState (LED ist an, Timer läuft). Die Hauptklasse MotionSensorAutomation merkt sich, in welchem Zustand sie gerade ist, und gibt die Steuerung an diesen Zustand weiter. Das bedeutet: Sie entscheidet nicht selbst, was bei Bewegung oder nach Ablauf der Zeit passieren soll – stattdessen übernimmt der aktuelle Zustand diese Aufgabe. So wird z.B. im IdleState bei Bewegung die LED eingeschaltet, der Timer gestartet und in den ActiveState gewechselt. Im ActiveState sorgt jede neue Bewegung dafür,

dass der Timer zurückgesetzt wird. Erst wenn 10 Sekunden lang keine Bewegung erkannt wird, geht die LED aus und das System wechselt zurück in den IdleState.

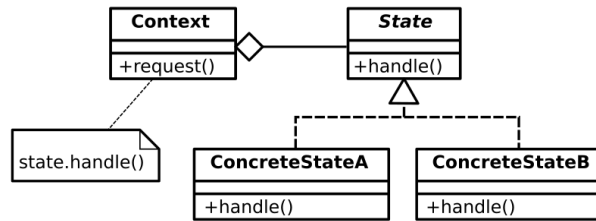


Abbildung 1: State Design Pattern UML Class Diagram

Literatur

- [1] Wikipedia. State pattern. Online verfügbar unter: https://en.wikipedia.org/wiki/State_pattern#/media/File:State_Design_Pattern_UML_Class_Diagram.svg, abgerufen am 9. Juli 2025.
- [2] Wikipedia. *State Pattern*. Online verfügbar unter: https://en.wikipedia.org/wiki/State_pattern#cite_note-GOF-1, abgerufen am 9. Juli 2025.

Erklärung zur individuellen Leistungsbeteiligung

Da es sich bei diesem Projekt um eine Gruppenarbeit handelt, wird gemäß Prüfungsordnung folgendes zur individuellen Beteiligung der Gruppenmitglieder am Projekt „I2EA__SmaBui“ erklärt:

Name des Gruppenmitglieds	Beteiligung in %
David Kiseljov	50 %
Leard Mucolli	50 %

Wir bestätigen hiermit, dass die oben angegebene Aufteilung unserer Einschätzung der individuellen Arbeitsleistung entspricht.

Uns ist bewusst, dass der betreuende Lehrende im Falle abweichender Wahrnehmung eine andere Bewertungsgrundlage heranziehen kann.



David Kiseljov
Unterschrift



Leard Mucolli
Unterschrift