

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
4	Softwarearchitektur	3
5	Kritik und Verbesserungsvorschläge	4
6	Reflexion zur Zusammenarbeit und Fazit	5

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

Nach dem Einschalten stellt das Smart Building zunächst eine WLAN-Verbindung her und gibt seine IP-Adresse bekannt. Zudem wird die interne Echtzeituhr (Real Time Clock) automatisch mit der aktuellen Uhrzeit synchronisiert. Danach werden kontinuierlich verschiedene Umweltdaten mittels integrierter Sensoren erfasst und in einem regelmäßigen Intervall von fünf Minuten an die Online-Plattform ThingSpeak übermittelt. Diese Übertragung erfolgt über das MQTT-Protokoll, welches eine effiziente Kommunikation zwischen Gerät und Cloud ermöglicht. Auf ThingSpeak werden die empfangenen Daten gespeichert und in übersichtlichen Diagrammen visualisiert. Ein spezieller Gassensor überwacht die Luftqualität kontinuierlich. Wird Gas erkannt, erfolgt unmittelbar die Auslösung eines Alarms, welcher sowohl lokal als auch über das System visuell angezeigt wird. Zur Zugangskontrolle nutzt das System RFID-Technologie: Bei Nutzung eines autorisierten RFID-Chips öffnet sich die Tür automatisch und die RGB-LED signalisiert mit grünem Licht den erfolgreichen Zugang. Bei Verwendung eines nicht autorisierten Chips bleibt die Tür geschlossen und die LED leuchtet rot auf. Die Bedienung vor Ort erfolgt über physische Buttons, die eine Navigation durch verschiedene Anzeigeoptionen auf dem integrierten LC-Display ermöglichen. Zusätzlich bietet das Smart Building eine webbasierte Steuerungsoberfläche an, die über WebSockets kommuniziert. Nach erfolgreicher Authentifizierung mit Benutzername und Passwort ermöglicht diese Oberfläche die Abfrage aktueller Sensorwerte sowie eine manuelle Steuerung des Türmechanismus und der LED. Wird vom Bewegungssensor Bewegung erkannt, aktiviert sich eine LED-Beleuchtung auf dem Dach des Gebäudes. Bleibt für 10 Sekunden Bewegung aus, deaktiviert sich die Beleuchtung automatisch.

4 Softwarearchitektur

Die Softwarearchitektur des Smart Building Systems folgt einem modularen und objektorientierten Design. Die Architektur basiert auf standardisierten Interfaces, konkreten Klassen und der Nutzung von Design Patterns. Ein zentrales Element der Architektur ist das Interface `ISmaBui_IO`. Dieses Interface definiert zwei grundlegende Methoden: `begin()` zur Initialisierung und `handle()` zur zyklischen Verarbeitung. Alle Hardware-Module implementieren dieses Interface, wodurch eine einheitliche Verwaltung und zyklische Abfrage innerhalb der Hauptschleife (`loop()`) möglich wird. Beispiele dafür sind:

- **SensorDHT:** Verantwortlich für die Erfassung und Bereitstellung von Temperatur- und Luftfeuchtigkeitswerten des DHT11-Sensors.
- **DisplayLCD:** Steuert die Anzeige von Messwerten, Systemmeldungen und Menüs auf dem 16x2 LCD-Display.
- **MotionSensorAutomation:** Implementiert die Bewegungserkennung mittels PIR-Sensor und steuert den LED-Streifen basierend auf erkannten Bewegungen.

Die Bewegungserkennung nutzt das State Pattern, um unterschiedliche Verhaltensweisen klar zu strukturieren. Dabei sind zwei konkrete Zustände definiert:

- **IdleState:** Wartet auf Bewegungserkennung. Bei Bewegung aktiviert dieser Zustand die LED-Beleuchtung und wechselt in den `ActiveState`.
- **ActiveState:** Aktiviert die LEDs dauerhaft und überwacht eine Timeout-Periode, nach deren Ablauf die LEDs ausgeschaltet und der Zustand zurück zu `IdleState` gewechselt wird.

Die zentrale Verwaltung der Zustände erfolgt in der Klasse `MotionSensorAutomation`, die alle Zustandsübergänge und zeitgesteuerten Abläufe koordiniert. Für die Zugangskontrolle mittels RFID wird der `MFR522_I2C` Leser eingesetzt. Eingelesene RFID-Karten werden mit einer hinterlegten Liste autorisierter IDs abgeglichen. Die Steuerung der Türöffnung erfolgt über eine eigenständige Zustandsmaschine mit folgenden Zuständen:

- `DOOR_CLOSED`
- `DOOR_OPENING`
- `DOOR_OPEN`
- `DOOR_CLOSING`

Diese Zustände werden zeitgesteuert verwaltet, um präzise Steuerungen des Servo-Motors zu gewährleisten. Die Kommunikation mit externen Systemen erfolgt über MQTT mittels der `PubSubClient`-Bibliothek. Sensorwerte wie Temperatur, Luftfeuchtigkeit und Regenwert werden regelmäßig alle fünf Minuten an ThingSpeak übermittelt. Das Webinterface ist eine weitere zentrale Komponente der Softwarearchitektur. Es nutzt `AsyncWebServer` und `WebSockets` für Echtzeitdaten und direkte Steuerungen aus dem Browser. HTTP Basic Authentication schützt den Zugang zur Weboberfläche, auf der Benutzer Echtzeitdaten visualisieren und Steuerbefehle (z.B. Tür öffnen/schließen, LED-Test) direkt an das Gerät senden können.

Die Hauptschleife (`loop()`) steuert die zyklische Verarbeitung aller Funktionen:

- Überwachung und Verarbeitung der Sensoren (Bewegung, Gas, Temperatur)
- Menü- und Button-Interaktionen
- Netzwerkkommunikation (MQTT und WebSockets)
- Verwaltung der RFID-basierten Zugangskontrolle
- Steuerung der Alarmfunktionen und Zustandsmaschinen

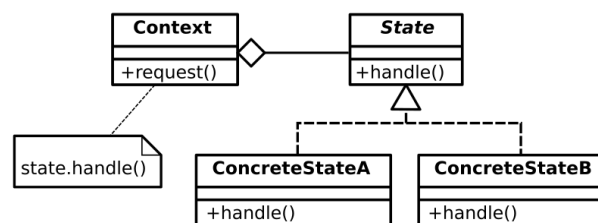


Abbildung 1: State Design Pattern UML Class Diagram

5 Kritik und Verbesserungsvorschläge

Mit der Zeit wurde der Code immer länger und schwerer zu überblicken. Neue Funktionen wie das Webinterface, die Sensoren oder die Kommunikation haben alles noch komplexer gemacht. Es wurde schwierig, bestimmte Stellen im Code schnell zu finden oder gezielt anzupassen. Auch das Testen war mühsam, weil viele Bereiche voneinander abhängig waren. Ein weiteres Problem war die fehlende Trennung der Aufgaben. Alles lag in einer Datei. Das machte es fast unmöglich, einzelne Teile unabhängig voneinander zu ändern oder zu erweitern. Für die Zusammenarbeit im Team war das ebenfalls unpraktisch. Man konnte schlecht parallel arbeiten, weil man sich oft gegenseitig im Weg stand. Besser wäre es gewesen, den Code gleich in mehrere Dateien aufzuteilen. Zum Beispiel eine für die Sensoren, eine für die Anzeige, eine für das Netzwerk und so weiter. Das hätte den Überblick verbessert und vieles einfacher gemacht. Auch für Außenstehende wäre der Code verständlicher gewesen. Für künftige Projekte haben wir daraus gelernt, dass eine klare Struktur von Anfang an wichtig ist. Sie hilft nicht nur beim Entwickeln, sondern macht das Projekt insgesamt robuster und angenehmer zu pflegen.

6 Reflexion zur Zusammenarbeit und Fazit

Die Zusammenarbeit im Rahmen des Projekts verlief insgesamt sehr positiv. Zu Beginn der Projektphase kam es aufgrund gesundheitlicher Einschränkungen zu einer ungleichmäßigen Verteilung der Aufgaben. In dieser Zeit wurde die Projektarbeit durch gegenseitige Unterstützung dennoch erfolgreich vorangetrieben. Es wurde gemeinsam an einem Rechner gearbeitet, sodass trotz der erschwerten Bedingungen frühzeitig Fortschritte erzielt werden konnten und wichtige Entscheidungen gemeinsam getroffen wurden. Im weiteren Verlauf wurde die Arbeit zunehmend aufgeteilt. Verschiedene Aufgabenbereiche wie Sensoranbindung, Webentwicklung oder Benutzeroberfläche konnten dadurch parallel bearbeitet werden. Regelmäßige Absprachen und gegenseitige Hilfestellung haben dafür gesorgt, dass das Projekt trotz der Aufgabenteilung als zusammenhängendes Gesamtsystem entwickelt wurde. An dieser Stelle möchten wir uns ausdrücklich bei Herrn Roloff für die gewährte Nachfrist bei der Abgabe bedanken. Die zusätzliche Zeit hat es uns ermöglicht, die Dokumentation ausführlicher und strukturierter zu verfassen. Dadurch konnten wir neben der technischen Umsetzung auch unsere Überlegungen und Erfahrungen ein bisschen umfassender darstellen. Insgesamt hat uns das Projekt große Freude bereitet. Der starke Praxisbezug war besonders motivierend, da viele Inhalte aus der Lehrveranstaltung direkt in einem funktionierenden Gesamtsystem zur Anwendung kamen. Die Kombination aus Softwareentwicklung, Hardwareansteuerung, Netzwerkanbindung und Benutzerinteraktion machte das Projekt für uns besonders abwechslungsreich. Neben dem fachlichen Wissen konnten auch unsere Fähigkeiten in der Planung, Zusammenarbeit und Problemlösung gestärkt werden. Für uns war dieses Projekt nicht nur eine Prüfungsleistung. Es war eine wertvolle und praxisnahe Lernerfahrung.

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