

## IoT SW Framework

SW Framework für IoT Devices für verschiedene Mesh-Netzwerke

<b>Programm / Ausschreibung</b>	IWI, IWI, Basisprogramm Ausschreibung 2023	<b>Status</b>	abgeschlossen
<b>Projektstart</b>	03.07.2023	<b>Projektende</b>	02.07.2024
<b>Zeitraum</b>	2023 - 2024	<b>Projektlaufzeit</b>	13 Monate
<b>Keywords</b>			

### Projektbeschreibung

Beim Projekt "SW Framework für IoT Devices für verschiedene Mesh-Netzwerke" geht es darum, eine modulare bibliotheksbasierte SW-Basis für ereignisgesteuerte IoT Anwendungen zu entwickeln, die unterschiedliche Hardware integrieren kann, verschiedene Netzwerkkommunikation wie Bluetooth Mesh, Matter/Thread und proprietäre Mesh-Netzwerke unterstützt und weiters für eine IoT Multiprozessor Struktur geeignet ist.

Nachdem die zu unterstützenden Mesh-Netzwerke unterschiedlich konzipiert sind, muss eine Architektur entwickelt werden, mit welcher die Netzwerkfunktionalität einheitlich bedient werden kann.

Dies erfordert die Konzeptionierung der Funktionen und Dienste nach eingehender Analyse der Systeme, gefolgt von der Umsetzungsphase des SW Frameworks. Dies umfasst die Unterstützung aller Kommunikationsschichten, Sicherheitsfunktionen, Gerätemanagement und andere relevante Aspekte. Darüber hinaus sollen Anpassungen und Optimierungen am Framework vorgenommen werden um Leistung, Skalierbarkeit und Sicherheit zu erhöhen - dies kann die Implementierung von effizienten Algorithmen, Datenstrukturen und Fehlerbehandlungsmechanismen umfassen.

Die erfolgreiche Integration unterschiedlicher Mesh-Netzwerke in ein Software Framework soll es Entwicklern ermöglichen, IoT Geräte für unterschiedliche Eco-Systeme effizienter zu entwickeln, die Wiederverwendbarkeit von SW Funktionalitäten zu gewährleisten und dadurch die Interoperabilität und Benutzerfreundlichkeit von IoT Lösungen zu verbessern.

### Endberichtkurzfassung

Kurzfassung der Projektergebnisse:

Dieses Projekt zielte auf die Entwicklung eines hochmodernen Software-Frameworks für Smart-Home-Anwendungen ab. Im Fokus standen Modularität, Flexibilität und Zukunftssicherheit als Kernprinzipien der Entwicklung. Das Framework integriert fortschrittliche Kommunikationsprotokolle wie Matter/Thread und Bluetooth Mesh 1.1, um eine vielseitige und robuste Plattform für die Entwicklung verschiedener Smart-Home-Geräte zu schaffen.

Der Entwicklungsprozess umfasste mehrere kritische Phasen: von der detaillierten Anforderungsanalyse über die sorgfältige Architekturkonzeption bis hin zur präzisen Implementierung und gründlichen Testung. Besondere Aufmerksamkeit wurde der Entwicklung einer leistungsfähigen Job Engine und den von ihr unterstützten Jobs (verschiedene Anwendungsszenarien: Schalten, Dimmen, Szenenauswahl,...) gewidmet, was eine flexible Anpassung an verschiedene Anwendungsszenarien ermöglicht. Die Integration verschiedener Kommunikationsprotokolle und die Anpassungsfähigkeit an unterschiedliche Hardware-Konfigurationen stellten weitere Schwerpunkte dar.

Das Projekt wurde in zehn Arbeitspakete unterteilt, die jeweils spezifische Aspekte der Entwicklung abdeckten. Von der initialen Konzeptphase bis zur finalen Qualifizierung und Testung wurden innovative Lösungen erarbeitet und implementiert, die den hohen Anforderungen an Qualität, Leistung und Zukunftssicherheit gerecht werden.

#### Arbeitspaket-spezifische Ergebnisse

##### Ergebnisse Arbeitspaket 1: Anforderungsanalyse und Konzeption

Dieses grundlegende Arbeitspaket bildete das Fundament für das gesamte Projekt. Es umfasste eine tiefgreifende Analyse des Marktes, der Kundenanforderungen und der technologischen Möglichkeiten.

Erstellung eines detaillierten Anforderungskatalogs mit präzise definierten funktionalen und nicht-funktionalen Anforderungen

Entwicklung eines Architekturkonzepts mit Fokus auf Flexibilität, Skalierbarkeit und Modularität als Kernprinzipien

Konzeption eines geschichteten SW Frameworks zur Gewährleistung der Anforderungen

Erfolgreiche Durchführung eines Proof-of-Concept (PoC) in einem Test-Netzwerk unter Einbeziehung von Drittkomponenten, um die Machbarkeit des Konzepts zu validieren

##### Ergebnisse Arbeitspaket 2: Konzepterstellung für SW Architektur

Aufbauend auf den Erkenntnissen aus Arbeitspaket 1 konzentrierte sich dieses Paket auf die detaillierte Ausarbeitung der Softwarearchitektur. Hier wurde besonderer Wert auf die Schaffung einer modularen und flexiblen Struktur gelegt, die sowohl aktuelle Anforderungen erfüllt als auch zukünftige Erweiterungen ermöglicht. Die Einhaltung bewährter Softwareentwicklungsprinzipien stand dabei im Vordergrund.

Konzeption einer modularen Kernarchitektur, die einfache Erweiterbarkeit und Anpassbarkeit an verschiedene Anwendungsszenarien ermöglicht

Entwicklung eines konfigurierbaren OPAL-Layers (Operating System Abstraction Layer) mit durchdachter SW-Schnittstelle zur Abstraktion hardwarenaher SW-Funktionen, wahlweise für Zephyr oder CubeMX

Konzeption der LOLA-Schicht (Low Level Layer) zur Bereitstellung elementarer Basisfunktionen wie Logging, Tracing, Monitoring und Time Management für die darüberliegenden Schichten  
Entwurf einer konfigurierbaren MOLA-Schicht (Modul Layer), bestehend aus einer Auswahl von Bibliotheksmodulen. Diese Schicht bildet die oberste Ebene des Frameworks, wobei die Gesamtheit der Modul-APIs das flexible Framework-API darstellt  
Strikte Einhaltung von SOLID-Prinzipien und Clean Code Praktiken bei der Core Framework Konzeptionierung, um eine hochqualitative, gut strukturierte Architektur zu gewährleisten, die einfach zu warten und zu erweitern ist

### Ergebnisse Arbeitspaket 3: Umsetzung SW Framework

Nach der sorgfältigen Konzeptionsphase folgte in diesem Arbeitspaket die praktische Umsetzung des Software-Frameworks. Hier ging es darum, die theoretischen Konzepte in funktionsfähigen Code zu überführen und dabei die hohen Qualitätsstandards und Architekturprinzipien beizubehalten.

Implementierung der in AP2 konzipierten Schichten:

Umsetzung der OPAL-Schicht für Zephyr, um eine effiziente Abstraktion der hardwarenahen Funktionen zu gewährleisten  
Parallele Implementierung der OPAL-Schicht für CubeMX, um Flexibilität in der Wahl des Betriebssystems zu bieten  
Realisierung der universellen Schichten LOLA und MOLA, die die Kernfunktionalität des Frameworks bilden und die Basis für die darauf aufbauenden Anwendungen schaffen

### Ergebnisse Arbeitspaket 4: Entwicklung Job Engine

Die Job Engine ist die Basis für konfigurierbare Funktionalität der Endgeräte, die in Form von verschiedenen Jobs vom Benutzer gewählt werden kann.

Entwicklung einer leistungsfähigen Job Engine, die auf dem Bluccino Framework aufsetzt und auf die MOLA APIs zugreift  
Implementierung der Job Engine und der darauf aufbauenden Jobs in C++, während das zugrundeliegende Bluccino Framework in C realisiert wurde. Diese Entscheidung ermöglicht die Nutzung fortschrittlicher objektorientierter Konzepte in der Job-Implementierung

Nutzung polymorpher Schnittstellen durch virtuelle Methoden der C++-Klassen zur Gewährleistung hoher Flexibilität verschiedener Jobs, vergleichbar mit Apps auf einem Smartphone

Integration der SOLID-Prinzipien und Clean Code-Praktiken in die Job Engine-Entwicklung, um eine flexible und einfache Erweiterbarkeit sicherzustellen

## Ergebnisse Arbeitspaket 5: Entwicklung Core Module zur Matter/Thread Kommunikation

Dieses Arbeitspaket konzentrierte sich auf die Implementierung der notwendigen Funktionen zur Unterstützung dieses aufstrebenden Smart-Home-Standards.

Implementierung der Kernfunktionalität basierend auf den bereitgestellten Nordic-Bibliotheken, um eine solide Grundlage für die Matter/Thread-Kommunikation zu schaffen

Integration der entwickelten Funktionalität in eine Interfaceschicht, die vollständig kompatibel zum Bluccino Framework ist, um eine nahtlose Einbindung zu gewährleisten

Spezifische Entwicklung für Nordic-Prozessoren, die Matter-Netzwerkereignisse effizient über die Job Engine an den ausgewählten Job weiterleitet

Implementierung bidirektionaler Kommunikation, die es dem ausgewählten Job ermöglicht, entweder mit dem Matter-Netzwerk zu kommunizieren oder den 230V-Netzcontroller zu bedienen

## Ergebnisse Arbeitspaket 6: Entwicklung Core Module zur Bluetooth Mesh1.1 Kommunikation

Parallel zur Matter/Thread-Integration wurde in diesem Arbeitspaket die Unterstützung für Bluetooth Mesh 1.1 entwickelt. Dies erweitert die Konnektivitätsoptionen des Frameworks und ermöglicht die Integration in bestehende Bluetooth-basierte Smart-Home-Ökosysteme.

Implementierung der Kernfunktionalität basierend auf den bereitgestellten Nordic-Bibliotheken für eine robuste Bluetooth Mesh 1.1-Unterstützung

Integration in eine Interfaceschicht, die vollständig kompatibel zum Bluccino Framework ist, um eine konsistente Architektur über verschiedene Kommunikationsprotokolle hinweg zu gewährleisten

Spezifische Entwicklung für Nordic-Prozessoren zur effizienten Weiterleitung von Bluetooth Mesh-Netzwerkereignissen über die Job Engine an den ausgewählten Job

Implementation einer Option zur Einbindung der Bluetooth Mesh-Firmware von Silvair durch Konfiguration eines UART-Proxymoduls, um zusätzliche Flexibilität und Kompatibilität zu bieten

## Ergebnisse Arbeitspaket 7: Entwicklung des Core Modules zur flexiblen Anpassung an unterschiedliche Hardware

Dieses Arbeitspaket zielte darauf ab, ein hochflexibles Core Modul zu entwickeln, das sich an verschiedene Hardware-Konfigurationen anpassen lässt. Dies ist entscheidend für die breite

Anwendbarkeit des Frameworks in verschiedenen Smart-Home-Geräten.

Entwicklung eines Core Modules mit folgenden Sub-Modulen:

**SYS:** Implementierung der Steuerung von drei Interrupt-getriebenen Hauptprozessen (Zero Cross, ADC, PLL), die die Basis für präzise Energiesteuerung und -messung bilden

**METER:** Realisierung der Analog/Digital-Wandlung (ADC) und Verarbeitung der ADC-Rohdaten zu skalierten und gefilterten Nutzdaten, einschließlich RMS-Wert-Berechnung für akkurate Energiemessungen

**OCU (Output Control Unit):** Integration von Kalman-Filterung der Zero Cross-Zeitstempel als Basis für die PLL-Regelschleife, Implementierung von PWM-Funktionen (Pulsweitenmodulation) zur Realisierung von Phasenanschnitt- oder Phasenabschnittsteuerung für verschiedene Lasttypen, Kommunikation mit der Kurzschlussabschaltheardware  
Entwicklung von GUARD-Funktionalitäten zum Schutz des Gerätes vor Überstrom, Exzess-Verlusten, Übertemperatur, Kurzschluss und Asynchronität der PLL-Regelschleife

**GPIO:** Implementierung von Funktionen zum Lesen und Entprellen der 230V-Eingänge sowie zur Ansteuerung diverser Ausgänge (MOSFET-Steuerung, Kurzschlusslogik, LEDs)

**NVM:** Entwicklung eines Moduls zum Speichern und Abrufen persistenter Informationen (z.B. Job ID, Betriebsmodus)

**UART:** Implementierung einer seriellen Schnittstelle zur Kommunikation zwischen den beiden Prozessoren

Ergebnisse Arbeitspaket 8: Definition und Implementierung automatisierter Tests

Um die Zuverlässigkeit und Leistungsfähigkeit des entwickelten Systems sicherzustellen, wurde in diesem Arbeitspaket ein umfangreiches Testframework definiert und implementiert.

Dies umfasste sowohl grundlegende Funktionstests als auch komplexe Szenarien zur Überprüfung der Systemstabilität unter verschiedenen Bedingungen.

Definition detaillierter Anforderungen für automatisierte Tests und Erstellung spezifischer Testfälle

Implementierung und Durchführung folgender elementarer Tests:

Überprüfung der  $\mu$ s-Clocktime mit Fokus auf minimale Code-Instruktionen und Re-entrante Prozesse

Durchführung von Timing-Analysen für kritische Prozesse: Zero Cross, ADC und PLL

Genauigkeitsanalyse des ADC-Prozesses zur Sicherstellung präziser Messungen

Jitter-Analyse des Zero Cross-Prozesses für stabile Energiesteuerung

Effizienz- und Robustheitstests des implementierten Kalman-Filters

Spezielle Tests zur Provokation hoher Einschaltströme

Tests zur Störung der PLL-Schleife durch fehlerhafte Zero Cross-Trigger

Zuverlässigkeitstests der Watchdog-Funktion zur Gewährleistung der Systemstabilität

Analyse der Varianz des PLL-Regelfehlers

Test des PLL Settling-Verhaltens für optimale Energiesteuerung

Ergebnisse Arbeitspaket 9: Implementierung der OTA Funktionalität

Die Implementierung von Over-the-Air (OTA) Update-Funktionalitäten war ein kritischer Aspekt, um die langfristige Wartbarkeit und Aktualisierbarkeit der entwickelten Geräte sicherzustellen. Dieses Arbeitspaket konzentrierte sich auf die Umsetzung dieser Funktionalität für beide unterstützten Kommunikationsprotokolle.

Bluetooth:

Implementierung der elementaren OTAU-Funktionalität basierend auf der Nordic Zephyr Bibliothek, um eine zuverlässige und sichere Update-Möglichkeit zu gewährleisten

Integration der Möglichkeit, Firmware-Updates mittels der Nordic nRF Connect App durchzuführen, was eine benutzerfreundliche Update-Lösung bietet

Matter/Thread:

Implementierung der von Matter definierten OTA-Funktionalität basierend auf der Nordic Matter Bibliothek, um Kompatibilität mit dem Matter-Ökosystem sicherzustellen

Sicherstellung, dass die OTA-Funktionalität den Sicherheits- und Zuverlässigkeitsstandards des Matter-Protokolls entspricht

Ergebnisse Arbeitspaket 10: Qualifizierung und Test

Das abschließende Arbeitspaket diente zur Sicherstellung der Qualität, Zuverlässigkeit und Praxistauglichkeit des entwickelten Frameworks und der darauf basierenden Anwendungen. Es umfasste eine umfangreiche Reihe von Tests und Qualifizierungsmaßnahmen, die sowohl im Labor als auch unter realen Bedingungen durchgeführt wurden.

Durchführung von Unit Tests im Rahmen der Zephyr-Funktionalität, um die Korrektheit und Robustheit der grundlegenden

Systemkomponenten sicherzustellen

Separate Tests der einzelnen Schichten (OPAL, LOLA, MOLA) zur Verifizierung ihrer Funktionalität und der korrekten Interaktion zwischen den Schichten

Umfassende Modultests in Zusammenarbeit mit den darunterliegenden Schichten auf der Ziel-Hardware, um die korrekte Integration und das Zusammenspiel aller Komponenten zu gewährleisten

Durchführung von Qualifikationstests über eine Serie von 50 Fertigungsmustern, um die Konsistenz und Zuverlässigkeit innerhalb eines Fertigungsloses zu ermitteln und Fertigungsvariationen quantitativ zu identifizieren

Verbundtests in Testnetzwerken mit Drittgeräten, um die Interoperabilität und korrekte Funktion in heterogenen Smart-Home-Umgebungen sicherzustellen

Durchführung von einmonatigen Langzeit-Feldtests in privaten Heimnetzwerken, um die Stabilität, Zuverlässigkeit und Benutzerfreundlichkeit unter realen Bedingungen zu evaluieren und potenzielle Probleme oder Verbesserungsmöglichkeiten zu identifizieren

## Zusammenfassung

Das Projekt zur Entwicklung eines innovativen Software-Frameworks für Smart-Home-Anwendungen wurde erfolgreich abgeschlossen und hat in allen zehn Arbeitspaketen die gesetzten Ziele erreicht. Von der initialen Anforderungsanalyse bis zur finalen Qualifizierungs- und Testphase wurden durchgängig innovative Lösungen erarbeitet und implementiert, die den hohen Anforderungen an Qualität, Leistung und Zukunftssicherheit gerecht werden.

Zentrale Erfolge des Projekts umfassen:

1. Entwicklung einer hochmodularen, flexiblen und zukunftssicheren Softwarearchitektur:

Die geschaffene Architektur, bestehend aus den Schichten OPAL, LOLA und MOLA, bietet eine modulare Grundlage für die Entwicklung verschiedener Smart-Home-Geräte. Die strikte Einhaltung von SOLID-Prinzipien und Clean Code Praktiken gewährleistet eine hohe Codequalität und erleichtert zukünftige Wartungs- und Erweiterungsarbeiten.

2. Erfolgreiche Integration zweier Kommunikationsprotokolle mit offenem Standard:

Die Implementierung von Matter/Thread und Bluetooth Mesh 1.1 ermöglicht eine breite Kompatibilität mit aktuellen und zukünftigen Smart-Home-Ökosystemen. Dies stellt sicher, dass auf dem Framework basierende Geräte in verschiedensten Umgebungen einsetzbar sind und langfristig relevant bleiben.

3. Entwicklung einer leistungsfähigen Job Engine:

Die in C++ implementierte Job Engine bietet eine flexible Plattform für die Entwicklung verschiedener Anwendungen. Durch die Nutzung polymorpher Schnittstellen wird eine hohe Anpassungsfähigkeit an unterschiedliche Anwendungsszenarien erreicht.

4. Schaffung eines flexiblen Core Modules:

Das entwickelte Core Modul mit seinen verschiedenen Sub-Modulen (SYS, METER, OCU, GPIO, NVM, UART) ermöglicht eine präzise Anpassung an unterschiedliche Hardware-Konfigurationen und Anwendungsfälle. Dies bildet die Grundlage für ein breites Spektrum an Smart-Home-Geräten.

#### 5. Implementierung robuster Over-the-Air (OTA) Update-Funktionalitäten:

Die Möglichkeit, Firmware-Updates über Bluetooth und Matter/Thread durchzuführen, gewährleistet die langfristige Wartbarkeit und Verbesserungsfähigkeit der entwickelten Geräte im Feld.

#### 6. Durchführung umfangreicher Tests und Qualifizierungsmaßnahmen:

Die Kombination aus Unit Tests, Modultests, Qualifikationstests und Langzeit-Feldtests stellt sicher, dass das entwickelte Framework und die darauf basierenden Anwendungen höchsten Qualitäts- und Zuverlässigkeitsstandards entsprechen. Die Durchführung von Tests in realen Umgebungen und mit einer Vielzahl von Fertigungsmustern gewährleistet die Praxistauglichkeit und Konsistenz der Lösung.

Das entwickelte Framework bietet eine zukunftssichere, flexible und leistungsfähige Plattform für die Entwicklung innovativer Smart-Home-Geräte. Es zeichnet sich durch seine

Anpassungsfähigkeit, Erweiterbarkeit und Kompatibilität mit offenen Kommunikationsstandards aus. Die umfangreichen Tests und Feldversuche bestätigen die Praxistauglichkeit und Zuverlässigkeit der entwickelten Lösung.

Abschließend lässt sich festhalten, dass das Projekt für Bluenetics sehr wichtig ist, das die weiteren Arbeiten an Softwarelösungen für weitere Smart Home Geräte wesentlich effizienter und daher kostengünstiger durchführbar sind.

### **Projektpartner**

- Bluenetics GmbH