

# Implementierung einer interruptgesteuerten Benutzerschnittstelle auf einem Low-Power-Mikrocontroller

## Bachelor-Thesis

zur Erlangung des akademischen Grades

Bachelor of Science (B. Sc.) im Studienfach Angewandte Informatik



**Hochschule Konstanz**

Technik, Wirtschaft und Gestaltung

vorgelegt von: Julian Rapp

Matrikelnummer: 304875

Erstgutachter: Prof. Dr. Irenäus Schoppa

Zweitgutachter: Prof. Dr. Heiko von Drachenfels

eingereicht in: Konstanz, am 30. Juni 2025

## Zitat

*So I listen to the radio and all the songs we used to know*

*So I listen to the radio remember where we used to go*

(The Corrs – Radio)

# **Abstract**

Die vorliegende wissenschaftliche Arbeit behandelt die Planung und Entwicklung einer interruptgesteuerten Benutzerschnittstelle zur Überwachung und Manipulation von Registern und Speicherzellen. Die Entwicklung soll auf Basis eines Low-Power-Microcontrollers der MSP430 Familie von Texas Instruments erfolgen.

Es erfolgt eine umfangreiche Planungs- / und Entwicklungsphase, die in Anlehnung an den Rational Unified Process dokumentiert wird. Auf Basis dieser Planung wird die Software dann als austauschbares Modul implementiert. Der Vorgang der praktischen Umsetzung wird schriftlich dokumentiert. Zum Abschluss der Arbeit wird die Erweiterung im Live-Betrieb des Praktikums von Microprozessorsysteme zum Einsatz kommen.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>I</b>
<b>Abkürzungsverzeichnis und Glossar</b>	<b>II</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Das Ziel dieser Arbeit . . . . .	1
1.2. Die Umgebung, in der die Arbeit entstand . . . . .	1
1.3. Der Aufbau dieser Arbeit . . . . .	2
1.4. Viele Informationen, wenige Quellen . . . . .	2
<b>2. Fazit und kritische Bewertung</b>	<b>3</b>
2.1. Das Ergebnis . . . . .	3
2.2. Die Bewertung der Frameworks Extbase und Fluid . . . . .	4
2.3. Ein Ausblick . . . . .	5
<b>Literaturverzeichnis</b>	<b>VII</b>
<b>Abbildungsverzeichnis</b>	<b>VIII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>Verzeichnis der Listings</b>	<b>X</b>
<b>Eidesstattliche Versicherung</b>	<b>XI</b>
<b>A. Anhang</b>	<b>i</b>
A.1. Beiliegende CD . . . . .	i
A.1.1. Inhaltsverzeichnis der CD . . . . .	i

## Abkürzungsverzeichnis und Glossar

FRAM .....	Ferroelectric RAM, ein nichtflüchtiger Speicher mit schneller Zugriffsgeschwindigkeit
interruptgesteuerten	Ein Mechanismus zur ereignisbasierten Unterbrechung des normalen Programmablaufs
Low-Power-Mikrocontroller	Ein Mikrocontroller, der für energieeffiziente Anwendungen optimiert ist. Siehe auch Fußzeile auf Seite <a href="#">1</a> .
Modul .....	Eine funktionale Einheit innerhalb eines größeren Systems, die separat entwickelt und gewartet werden kann
RAM .....	Random Access Memory, ein flüchtiger Speicher
Registern .....	Speicherzellen, die flüchtig sind und ihre Inhalte beim Ausschalten verlieren
RUP .....	Rational Unified Process. Siehe auch Fußzeile auf Seite ??.
TEST .....	Dummy
trial & error .....	Heuristische Methode, bei der durch Versuch und Irrtum eine Lösung gefunden wird.

# 1. Einleitung

## 1.1. Das Ziel dieser Arbeit

Diese Bachelor-Thesis befasst sich mit der Entwicklung einer *interruptgesteuerten*<sup>1</sup> Benutzerschnittstelle auf einem *Low-Power-Mikrocontroller*<sup>2</sup>, zur Überwachung und Manipulation von *Registern*<sup>3</sup> und Speicherzellen in *RAM*<sup>4</sup> und *FRAM*<sup>5</sup>.

Im Zuge des Arbeitsauftrags wird ein unabhängiges *Modul*<sup>6</sup> entwickelt, welches nach Wunsch aktiviert oder deaktiviert wird.

## 1.2. Die Umgebung, in der die Arbeit entstand

Die Entwicklung der Software geschah in Absprache mit Herr Prof. Dr. Irenäus Schoppa, welcher ein zusätzliches Tool für Studenten im Praktikum von Microprozessorsysteme benötigt.

Als Entwicklungsbasis kam der in *MIPS*<sup>7</sup> herangezogene MSP430FR5729 von Texas Instruments zum Einsatz, welcher bereits ein ausgereifter und etablierter *LP-MCU*<sup>8</sup> ist. Viele grundlegenden Technologien, die diesem Prozessor zugrunde liegen, werden in dieser Arbeit wissentlich nicht behandelt, um den Rahmen nicht zu sprengen und sich auf das wesentliche zu konzentrieren.

---

<sup>1</sup>Ein Mechanismus zur ereignisbasierten Unterbrechung des normalen Programmablaufs

<sup>2</sup>Typischerweise eingesetzt in batteriebetriebenen Embedded-Systemen

<sup>3</sup>Speicherzellen, die flüchtig sind und ihre Inhalte beim Ausschalten verlieren

<sup>4</sup>Random Access Memory, ein flüchtiger Speicher

<sup>5</sup>Ferroelectric RAM, ein nichtflüchtiger Speicher mit schneller Zugriffsgeschwindigkeit

<sup>6</sup>Eine funktionale Einheit innerhalb eines größeren Systems, die separat entwickelt und gewartet werden kann

<sup>7</sup>Microprozessorsysteme

<sup>8</sup>Low-Power-Mikrocontroller

## 1.3. Der Aufbau dieser Arbeit

**Aktueller Wissensstand:** Der aktuelle Wissensstand beschreibt, auf welchem Wissensniveau sich der Autor im Moment der Aufnahme der Arbeit befand.

**Methoden und Herangehensweisen:** Im Kapitel »Methoden und Herangehensweisen« werden die zur Planung verwendeten Methoden erläutert. Die grundlegenden Eigenschaften und der Aufbau des Softwareentwicklungsprozesses werden erklärt. Zudem werden die benötigten Dokumentationen spezifiziert.

**Die Konzeptionierung des Moduls:** Dieses Kapitel umfasst die Dokumentation der gesamten Planungsphase des Observer-Moduls. Hier wird eine Übersicht über die bereits vorhandene Lösung geschaffen und anschließend die zur Planung erforderlichen Dokumente angefertigt.

**Die Entwicklung des Observers:** Dieses Kapitel enthält die Dokumentation der tatsächlichen Programmierung der Software. Hier werden die Voraussetzungen zur Implementation geklärt und der Verlauf der Entwicklung anhand von Beispielen schrittweise abgearbeitet.

**Fazit und kritische Bewertung:** Im Fazit werden die gemachten Erfahrungen und die Ergebnisse der Planung und Entwicklung abschließend zusammengefasst und kritisch bewertet. Zusätzlich wird ein kleiner Ausblick auf Erweiterungsmöglichkeiten und mögliche Optimierungsschritte unternommen.

## 1.4. Viele Informationen, wenige Quellen . . .

Grundsätzlich war es einfach geeignete Quellen zu den Themen rund um die Technologien zu finden, da – wie bereits erwähnt – die Entwicklungsplattform und der Microcontroller weitestgehend etabliert sind. Allerdings können alle wichtigen Informationen aus erster Hand, von dem Hersteller entnommen werden, weshalb andere Quellen unnötig erscheinen.

## 2. Fazit und kritische Bewertung

### 2.1. Das Ergebnis

Der Relaunch des gesamten Webauftritts von HIT RADIO FFH erfolgte am 24. Juli 2011, mit dem auch der neue Webradio-Player zum ersten Produktiveinsatz gelangte. In Abbildung 2.1 ist die Webradio-Player-Extension im Live-Einsatz auf [webradio.ffh.de](http://webradio.ffh.de) zu sehen.

Abb. 2.1.: Der FFH Webradio-Player im Live-Einsatz

Die grundsätzliche Entscheidung, bei der Extension-Entwicklung auf die Frameworks Extbase und Fluid zu setzen, hat sich als absolut richtig erwiesen. Trotz der vielen Kinderkrankheiten und der unvollständigen Implementation vieler Funktionen ist die anwendungsdomänen-getriebene Herangehensweise von Extbase bzw. FLOW3 ein deutlicher Schritt in eine einfachere Richtung der Entwicklung.

Der Webradio-Player läuft seit dem Relaunch nahezu fehlerfrei und benötigte danach nur kleine Anpassungen wegen Schnittstellenänderungen des Stream-Providers Nacamar.



## 2.2. Die Bewertung der Frameworks Extbase und Fluid

Das bei der Radio/Tele FFH GmbH im Zuge des Relaunches zum Einsatz kommende Content Management System TYPO3 bildete die Basis der Extension-Entwicklung während dieser Arbeit. Aufgrund der angepriesenen Zukunftssicherheit wurde zudem entschieden, alle Extensions mit denen am Markt neu eingeführten Frameworks Extbase und Fluid zu entwickeln. Sowohl Extbase als MVC und Domain-Driven Design basiertes PHP-Framework, als auch Fluid als XML basierte Templating-Engine vereinen das gemeinsame Prinzip »Convention over Configuration«. Dieses Prinzip stellt viele Entwickler vor die große Herausforderung, ihre bisherigen Gewohnheiten beim Entwickeln von Software zu überdenken.

Die vielen zunächst übertrieben erscheinenden Konventionen beim Programmieren mit Extbase und Fluid haben schlussendlich trotz vieler Zweifel ermöglicht, dass drei Entwickler parallel an verschiedenen Extensions arbeiten konnten, auch wenn sie nicht am ursprünglichen Planungsprozess beteiligt waren. Einzig durch die strengen Konventionen in der Programmierung gelingt es, sich ohne große Probleme in andere Extensions einzulesen. So war es beispielsweise möglich, bestimmte Quellcode-Fragmente schnell in andere Extensions zu portieren, ohne viel am bereits vorhandenen Code ändern zu müssen.

Die Templating-Engine Fluid übernimmt diese hervorragenden Eigenschaften bei der Umsetzung in der View-Ebene der Extensions. Als XML-basierte Templating-Engine ist sie flexibel einsetzbar, erzeugt validen XHTML-Code und ist durch ihre Viewhelper theoretisch unendlich erweiterbar. Die bereits implementierten Viewhelper zeugen von der Macht, die von dieser Engine ausgeht. Aus diesem Grund ist es schade, dass bisher nur Basisfunktionen in Viewhelpers vorliegen. Geht es darum, exotischere Logiken in Viewhelpers zu verwenden, so ist man leider darauf angewiesen, diese selbst zu programmieren. Hierdurch geht ein nicht unerheblicher Teil an Entwicklungszeit verloren, weil die Rahmenbedingungen für die Umsetzung TYPO3 interner Funktionen leider meistens schlecht sind. Es fehlt an Dokumentationen der TYPO3 Funktionalitäten, aber auch an Möglichkeiten, diese in Viewhelpers zu realisieren. Aus diesem Grund ist man auch hier stark auf die Community im Internet angewiesen. Den Hauptanteil an Informationen bezieht man von anderen Ent-

## 2. Fazit und kritische Bewertung

---

wicklern, die sich die Lösungen meist umständlich durch *trial & error*<sup>9</sup> erarbeitet haben.

Theoretisch lässt sich Fluid mit jedem verfügbaren PHP-Framework verwenden. So kann eine simple Webseite auch komplett autark ohne Content Management System aufgebaut werden.

Zusammenfassend bleibt zu sagen, dass sowohl Extbase als auch Fluid ihren Weg in die Welt von TYPO3 gefunden haben. Die beiden alternativen Frameworks geben einen Vorgeschmack auf die Entwicklung mit TYPO3 v5 und FLOW3 und helfen Entwicklern schon heute sich auf die Portierung ihrer Extensions vorzubereiten. Für den Autor als Neueinsteiger in der Extension-Entwicklung boten die beiden Werkzeuge viele neue und intuitive Wege, an die Entwicklung von Software heranzugehen. Nicht zuletzt ist hervorzuheben, dass die neuartigen Ansätze der Entwicklung, wie etwa das Domain-Driven Design und Convention over Configuration, viel Erleichterung in den Alltag eines Software-Entwicklers bringen können, sofern dieser sich darauf einlässt.

### 2.3. Ein Ausblick

Es hat sich gezeigt, dass die Entwicklung der Frameworks Extbase und Fluid ständig voranschreitet, weshalb es in unregelmäßigen Abständen sinnvoll ist, die Webradio-Player-Extension zu aktualisieren und Neuerungen direkt zu implementieren. Dadurch ist eine spätere Umsetzung in FLOW3 für TYPO3 Version 5 einfacher zu realisieren.

Durch die grundverschiedenen Ansätze beider Varianten ist schnell erkennbar, dass mit der Entwicklung Schritt gehalten werden muss, damit die eigenen Extensions zukunftssicher bleiben. Aus diesem Grund werden alle bei der Radio/Tele FFH GmbH entwickelten Extensions – so auch der Webradio-Player – ständig auf dem neusten Entwicklungsstand gehalten.

Einer Portierung auf FLOW3 stünde somit nichts im Wege.

---

<sup>9</sup>Heuristische Methode, bei der durch Versuch und Irrtum eine Lösung gefunden wird.

## Literaturverzeichnis

### Webopedia a

WEBOPEDIA: *extension*. <http://www.webopedia.com/TERM/E/extension.html>, Abruf: 04. Juli 2011 09:21:48. Webseite

### Webopedia b

WEBOPEDIA: *RUP*. <http://www.webopedia.com/TERM/R/RUP.html>, Abruf: 30. Juni 2011 10:25:10. Webseite

## Abbildungsverzeichnis

2.1. FFH Webradio-Player im Live-Einsatz . . . . .	3
--	---

# Tabellenverzeichnis

# Verzeichnis der Listings

## Eidesstattliche Versicherung

Ich, Julian Rapp, Matrikel-Nr. 304875, versichere hiermit, dass ich meine Bachelor-Thesis mit dem Thema

*Implementierung einer interruptgesteuerten Benutzerschnittstelle auf einem Low-Power-Mikrocontroller*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Bachelor-Thesis zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in dreifacher Ausfertigung und gebunden im Prüfungsamt der Hochschule für Technik, Wirtschaft und Gestaltung abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Konstanz, den 27. April 2025

---

JULIAN RAPP

## **A. Anhang**

### **A.1. Beiliegende CD**

#### **A.1.1. Inhaltsverzeichnis der CD**

1. Die gesamte Bachelorarbeit als PDF-Datei
2. Alle verwendeten Online-Quellen als PDF-Ausdruck
3. Sonstige Quelltexte
4. Alle in der Bachelorarbeit verarbeiteten Diagramme
5. Alle in der Bachelorarbeit verarbeiteten Grafiken
6. Alle in der Bachelorarbeit verarbeiteten Screenshots