



Projekt myFinder MK3

Inhalt

Vorbetrachtungen	3
Einführung.....	3
Aufgabenstellung	3
Anforderungen	4
Entwicklungsumgebung	4
Anwendungsfälle myFinder MK3	4
Blockdefinition myFinder MK3.....	4
Blockdefinition mySmartControl MK3.....	5
Blockdefinition myFinder	5
Anforderungen myFinder MK3	6
Entwurf	6
Vorgehensweise	6
Grobentwurf Systemverhalten.....	6
CRC Karten.....	7
Realisierung.....	8
Vorgehensweise	8
Pakete.....	8
Framework und Codegenerierung.....	8
Zustände	9
Detailliertes Systemverhalten.....	10
Quellcodebeispiel.....	10
Fertiges System.....	11
Anwendung	11
Einkaufsliste (Vorschlag).....	11

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Dokument erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden.

© Laser & Co. Solutions GmbH
Promenadenring 8
02708 Löbau
Deutschland

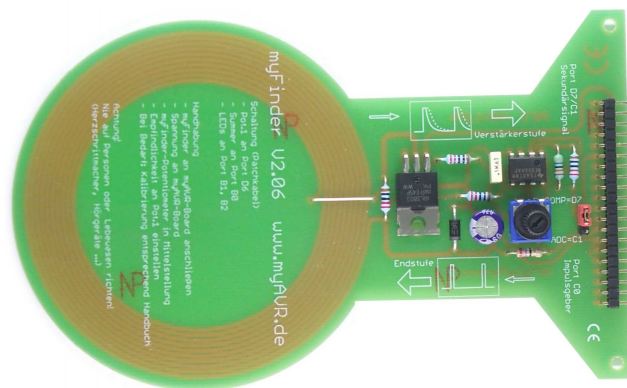
www.myAVR.de
support@myavr.de

Tel: ++49 (0) 358 470 222
Fax: ++49 (0) 358 470 233

Vorbetrachtungen

Einführung

Metalldetektoren sind in unserem Alltag weit verbreitete Sensorsysteme. Von der Parkplatzschanke über das Sicherheitssystem in einem Flughafen oder öffentlichem Gebäude bis zur Verkehrsüberwachungstechnik finden sich unzählige Anwendungsgebiete für Metalldetektoren. Genauso vielfältig wie die Anwendungsgebiete sind auch die Verfahren zur Erkennung von Metall. Das [Projektpaket myFinder](#) beschreibt eine mögliche Metalldetektorlösung ausgehend von den physikalischen Grundlagen über die Analogtechnik bis zur digitalen Signalverarbeitung in einem Mikrocontroller. Diese Projektbeschreibung soll Anregungen liefern, wie das [Projektpaket myFinder](#) weitergeführt werden kann. Hierbei sollen besonders die Möglichkeiten der Softwaretechnologie (Software-Engineering) und des Systementwurfes (Systems-Engineering) aufgezeigt werden.



Aufgabenstellung

Entwickeln Sie eine Mikrocontrollerlösung für einen Metalldetektor nach dem Puls-Induktions-Verfahren mit grafischer Visualisierung der erfassten Messwerte. Nutzen Sie dabei die Paradigmen, Methoden, Techniken und Werkzeuge der objektorientierten Systementwicklung. Die Metalldetektorlösung soll auf folgenden Systembausteinen basieren:

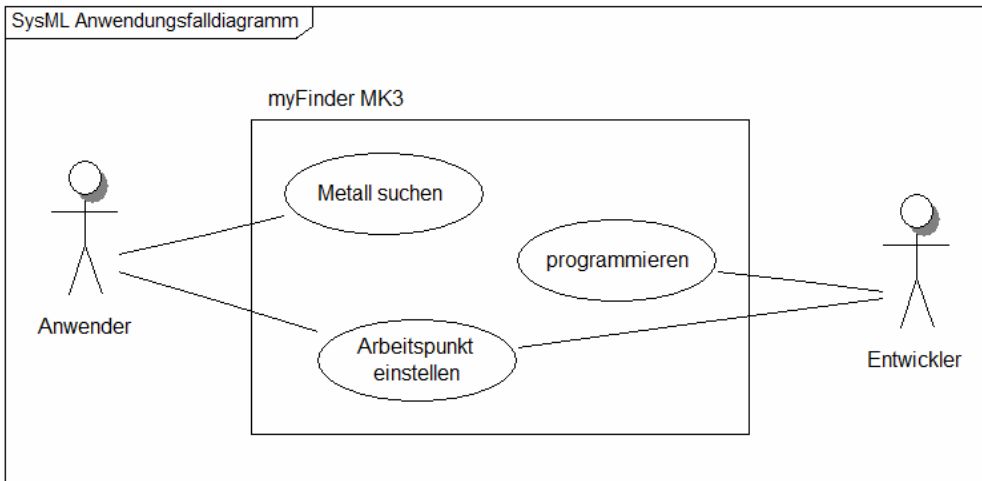
- mySmartControl MK3
- myAVR Stamp256 PLUS
- Metalldetektor myFinder

Anforderungen

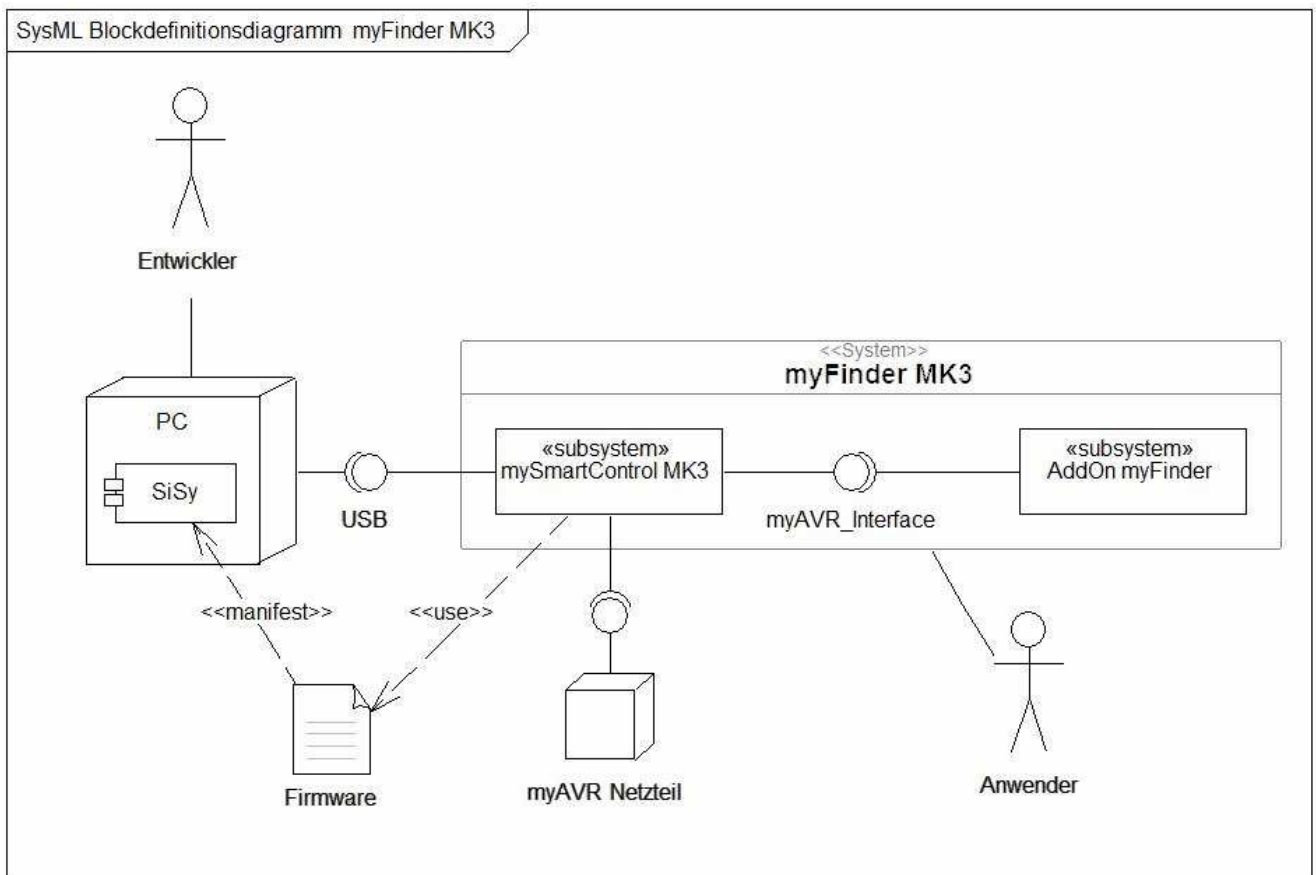
Entwicklungsumgebung

Nutzen Sie für die Entwicklung der Metalldetektorlösung die Darstellungstechniken der UML und der SysML! Als Modellierungswerkzeug empfiehlt sich SiSy mit folgenden Add-Ons: AVR, UML und SysML. Im Folgenden werden Beispiele für die Arbeitsergebnisse des Entwicklungsprozesses aufgezeigt. UML und SysML sind international verbindliche Standards für die Visualisierung von Systemen, die aus Hardware und Software bestehen.

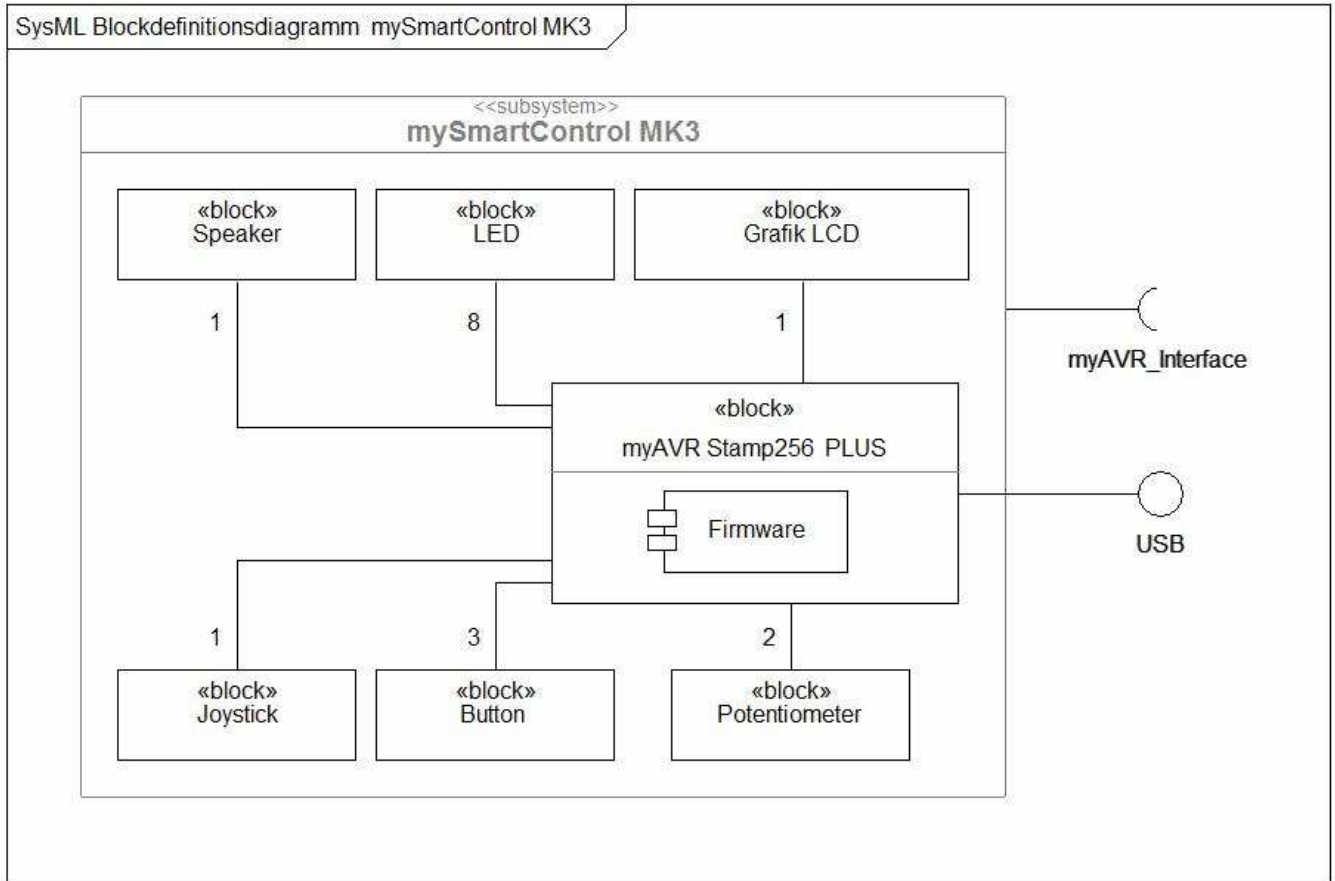
Anwendungsfälle myFinder MK3



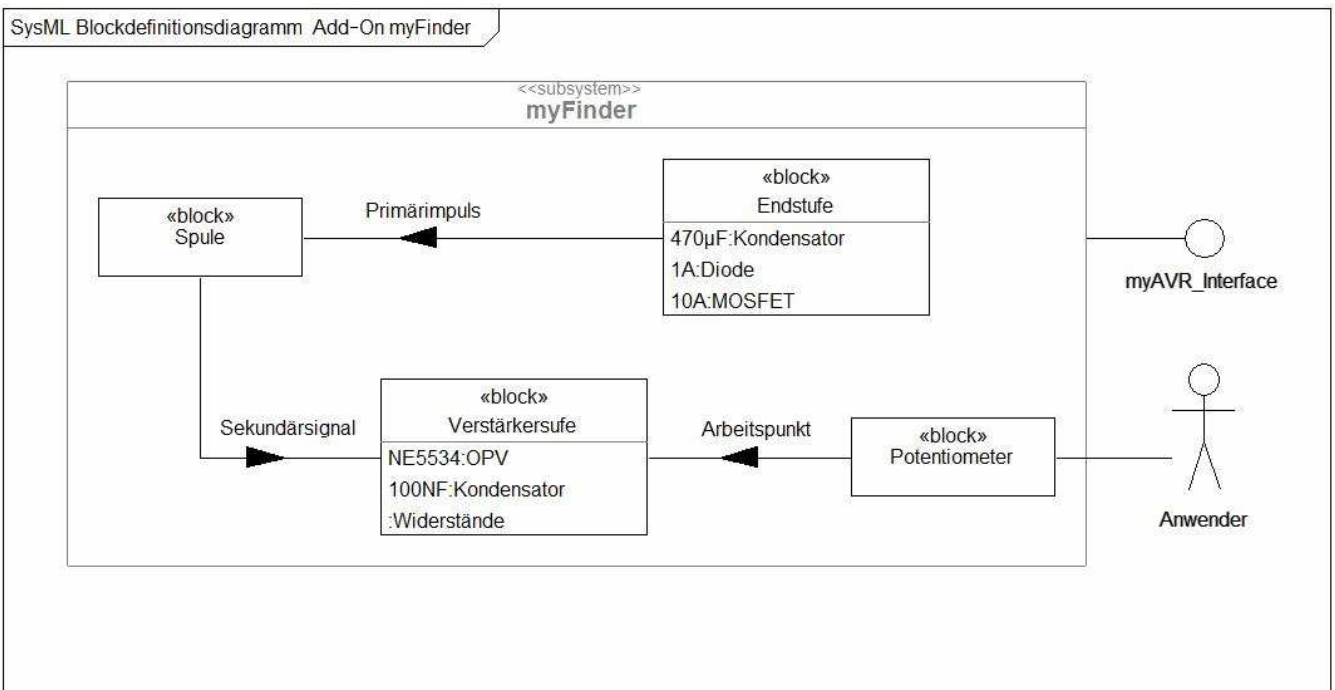
Blockdefinition myFinder MK3



Blockdefinition mySmartControl MK3



Blockdefinition myFinder



Anforderungen myFinder MK3

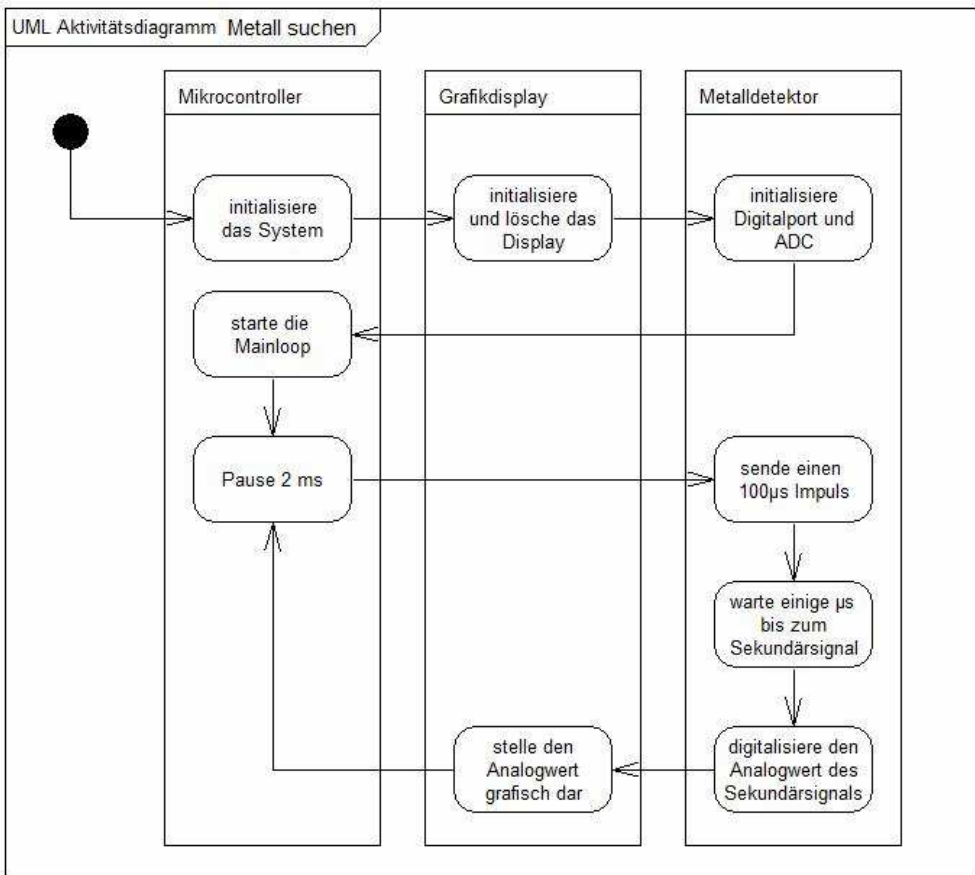
Anforderungen an den myFinder MK3		
ID	Anforderung	Beschreibung
1	Zielplattform	Plattform ist der mySmartControl MK3
2	Zielsystem	Zielsystem ist die myAVR Stamp256 PLUS
3	Portierbarkeit	portierbar auf myAVR Board MK3
4	Kompatibilität	kompatibel mit ATmega640
5	Benutzerfreundlichkeit	der Signalverlauf wird grafisch angezeigt
6	Qualität	...

Entwurf

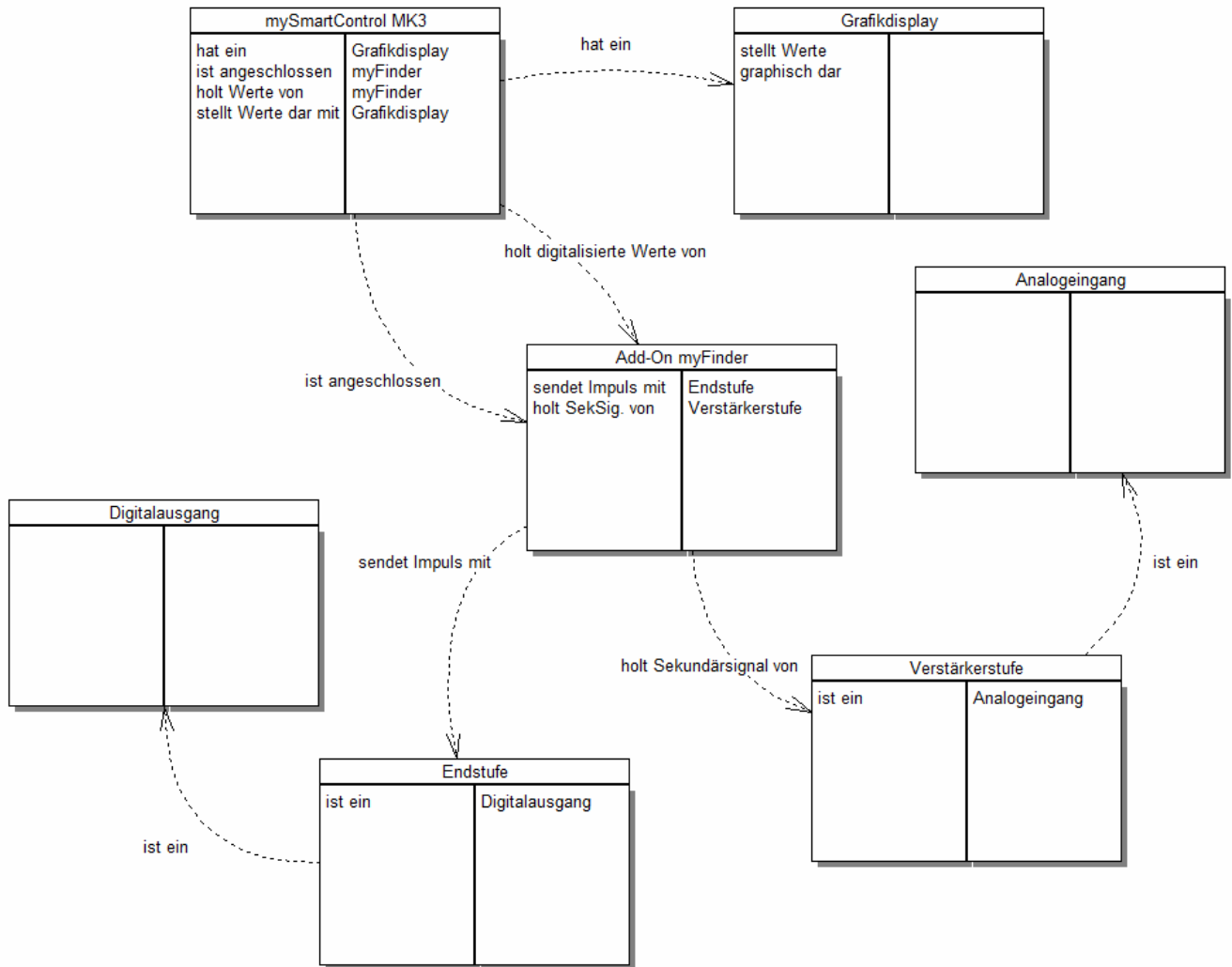
Vorgehensweise

Der Entwurf der Mikrocontrollerlösung basiert auf den Strukturmodellen des Systems und seiner Aufgabe (WAS soll das System leisten - Anwendungsfälle). Die Anwendungsfälle werden im ersten Entwurfsschritt als grobe Szenarien des geforderten Systemverhaltens (WIE soll das System die Leistung erbringen) modelliert. Ausgehend von der gegebenen Systemstruktur (Hardware), den Anwendungsfällen und dem geforderten Systemverhalten werden die benötigten Objekte und Klassen abgeleitet. Eine mögliche Methodik zum Auffinden der Klassen bietet die CRC-Kartentechnik.

Grobentwurf Systemverhalten



CRC Karten

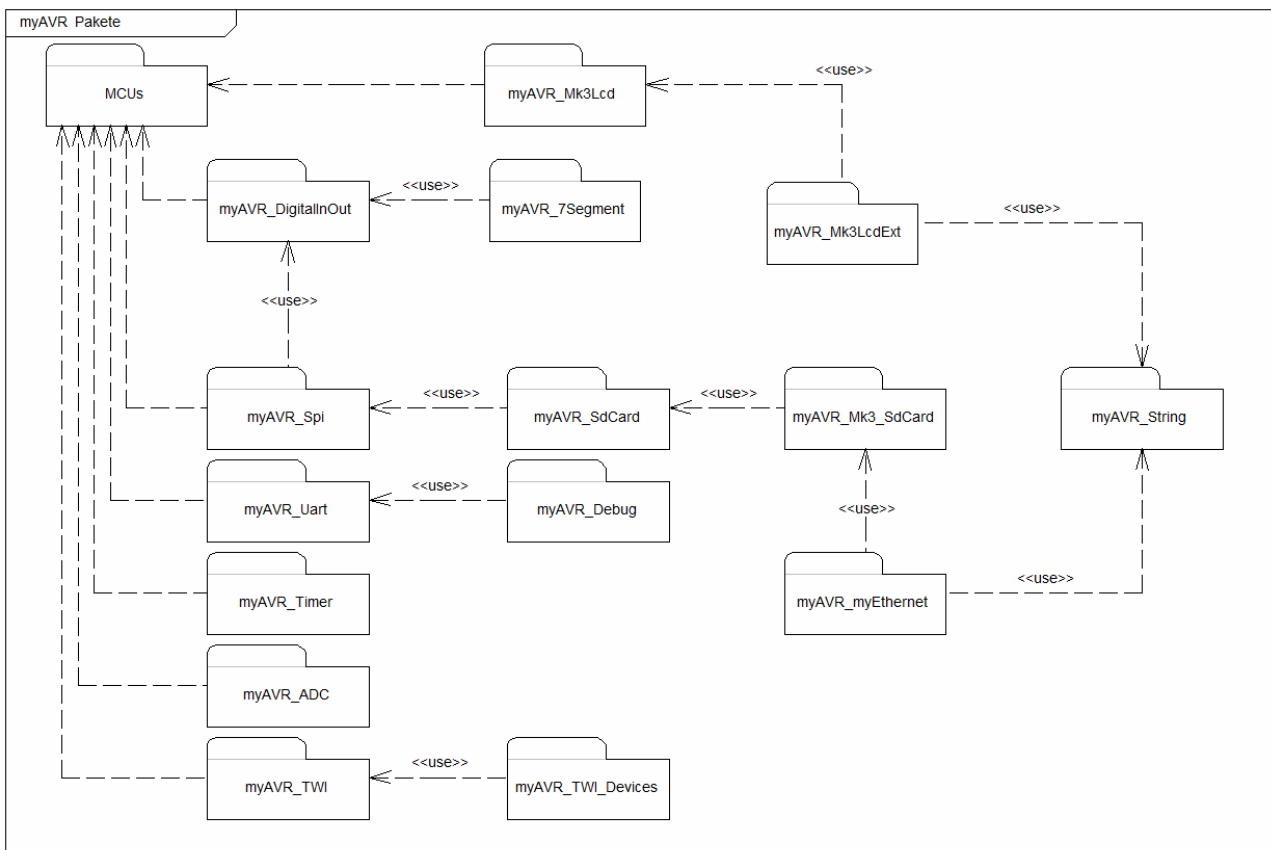


Realisierung

Vorgehensweise

Für den Feinentwurf und die Realisierung der Mikrocontroller-Firmware bietet es sich an, die Fähigkeiten der UML-Sprachmittel zur Systemgenerierung in einer objektorientierten Programmiersprache zu nutzen. Dabei fungiert vor allem das Klassendiagramm als Konstruktionszeichnung der Softwarekomponenten des Systems. Für das vorgegebene Mikrocontrollersystem von myAVR gibt es leistungsfähige C++ Klassenbibliotheken, auf die eine Firmwareentwicklung aufbauen kann (Aspekt der Wiederverwendung).

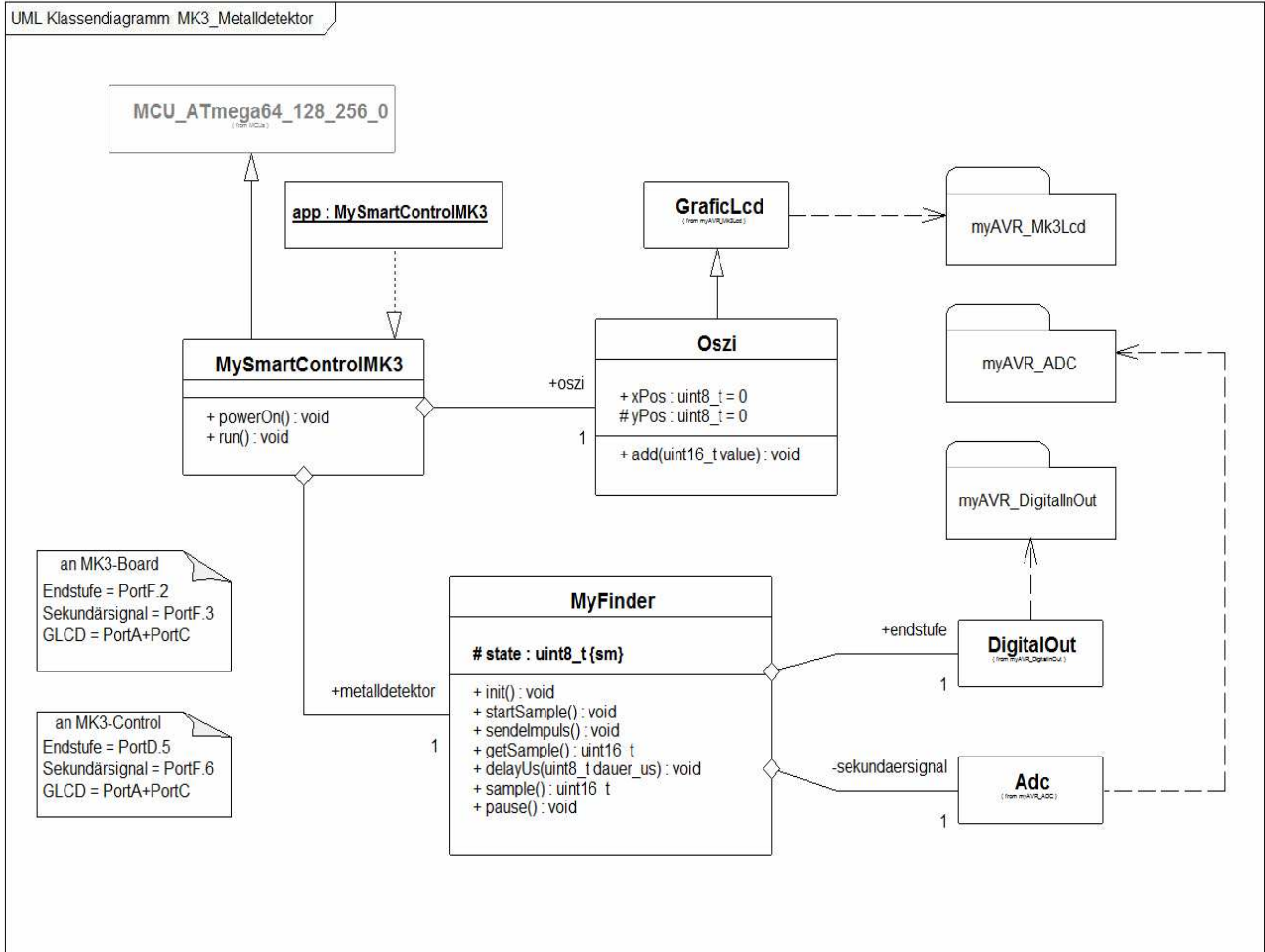
Pakete



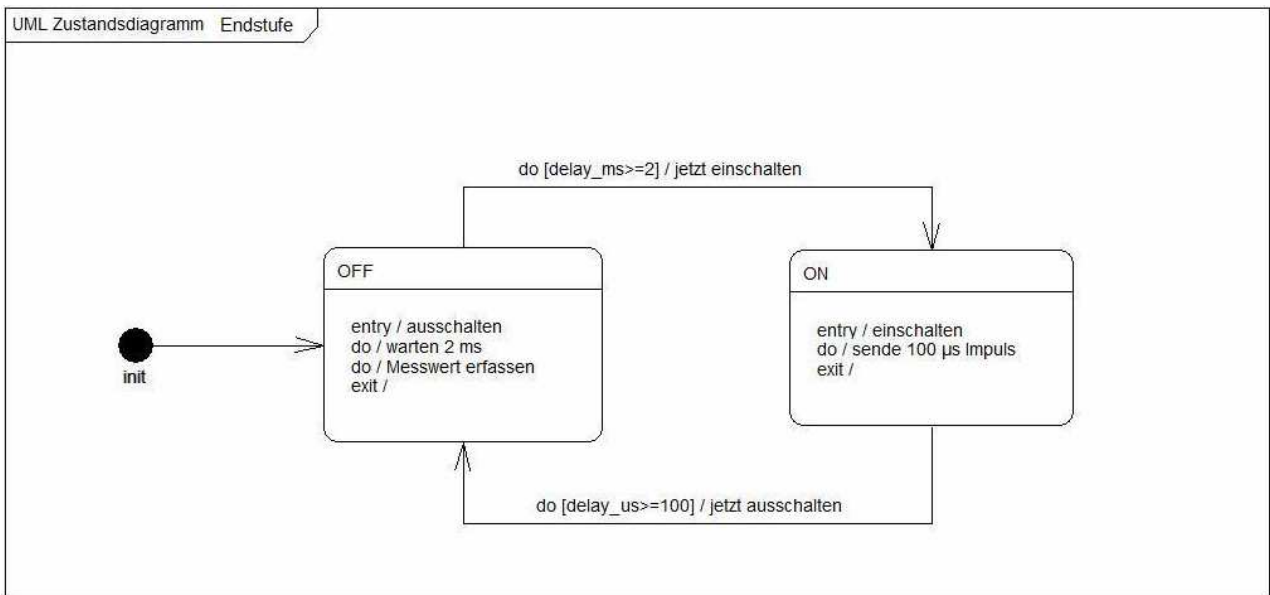
Framework und Codegenerierung

Das myAVR Framework bietet eine Reihe von Paketen mit leistungsfähigen Klassen und Templates, welche auf die vorgegebene Hardware optimal abgestimmt sind. Diese ermöglichen dem Entwickler weitestgehend von konkreten Registern und Hardwarebesonderheiten zu abstrahieren und das System bausteinartig im Klassendiagramm zu konstruieren. Aus dem Klassendiagramm wird der benötigte C++ Code generiert und kann über die integrierten myAVR-Tools (myAVR CodeWizard, myAVR ProgTool, myAVR ControlCenter) direkt auf den Mikrocontroller gebrannt und sofort getestet werden. Im folgenden werden Beispiellösungen mit wichtige Konstruktionstechniken der UML aufgezeigt.

Klassen

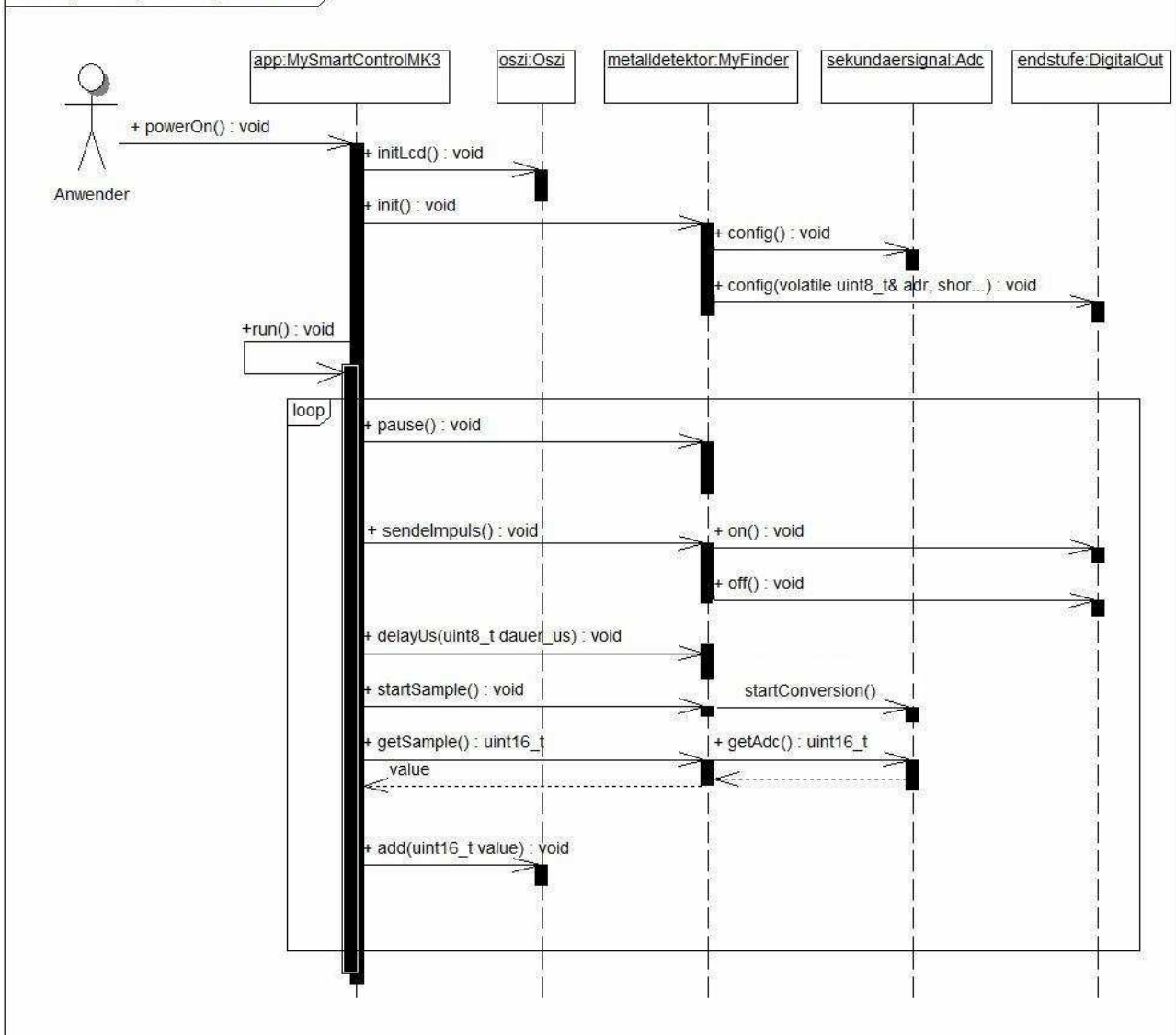


Zustände



Detailliertes Systemverhalten

UML Sequenzdiagramm myFinder MK3



Quellcodebeispiel

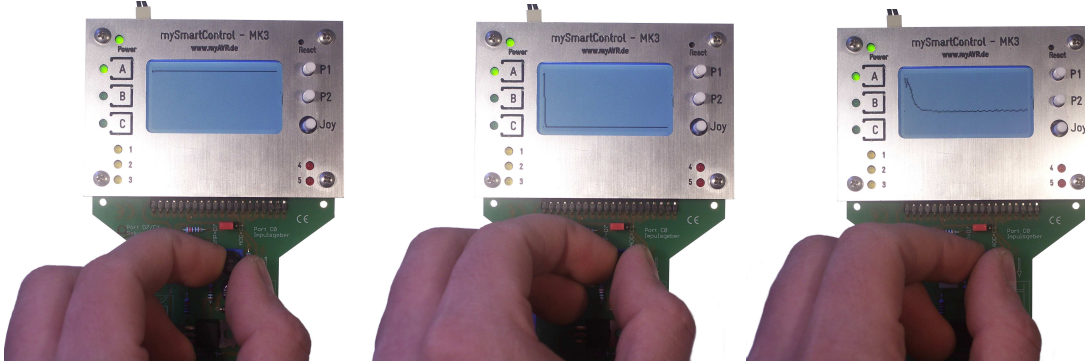
```

void MySmartControlMK3::run () {
0050 do {
0051     pause();
0052     metalldetektor.sendeImpuls();
0053     for (uint8_t n=0;n<128;n++)
0054     {
0055         metalldetektor.delayUs(1);
0056         if(n==oszi.xPos) metalldetektor.startSample();
0057     }
0058     oszi.add(metalldetektor.getSample());
0059 } while (true);
0060 }
  
```

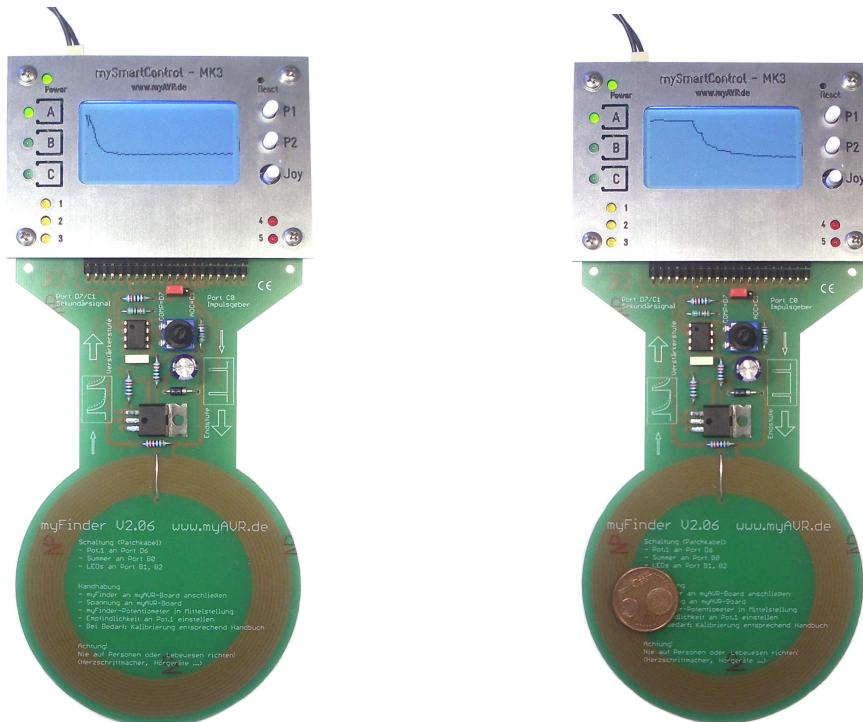
Fertiges System

Anwendung

- mySmartControl MK3 und myFinder verbinden
- Spannungsversorgung anschließen
- Firmware brennen
- Arbeitspunkt einstellen (vgl. Bilder: übersteuert, untersteuert, optimaler Arbeitspunkt)



- Metall suchen (vgl. Bilder: Signalverlauf ohne Metall in Spulennähe, Signalverlauf mit einem 5 Cent Stück in Spulennähe)



Einkaufsliste (Vorschlag)

- [Projektpaket myFinder](#)
- [mySmartControl MK3](#)
- [myAVR Stamp256 PLUS](#)
- [Mini-USB-Kabel](#)
- [myAVR Netzteil](#)
- [SiSy AVR Entwicklungsumgebung](#)