

mbed und mbed Target: Toolkette zur Prototypenentwicklung am Beispiel einer künstlichen Hand

Olaf Hagendorf¹, Alexander Martens¹ und Olaf Simanski¹

¹Fakultät für Ingenieurwissenschaften, Hochschule Wismar, Wismar, Deutschland

Kontakt: olaf.hagendorf@hs-wismar.de

Einleitung

Die prototypische Entwicklung von mechanischen Konstruktionen in Lehre und Forschung hat sich in den letzten Jahren durch die Entwicklung neuer Technologien wesentlich geändert. Bis vor wenigen Jahren waren im Hochschulumfeld xy-Fräsen zur 2D oder 2.5D Bearbeitung eine der wenigen Möglichkeiten für den mechanischen Bau von Prototypen. Komplexere Konstruktionen mussten in Auftrag gegeben werden. Die Umsetzung erfolgte hier z.B. durch CNC Maschinen oder durch 3D Druckverfahren wie Lasersintern. In der Regel waren diese Möglichkeiten sowohl zeit- als auch kostenintensiv. Durch die Entwicklung der Fused Deposition Modeling (FDM) Technologie haben sich die Möglichkeiten des Prototypenbaus wesentlich geändert. Mittels FDM 3D Druckern können in Forschungs- aber auch in studentischen Projekten mit herkömmlichen 3D CAD Systemen Konstruktionen entworfen und auf 3D Druckern unmittelbar gedruckt werden.

Aber ein zweiter, wesentlicher Aspekt des Prototypenbaus ist die Softwareentwicklung. Durch die Vorstellung des Arduino-Projektes 2005, eine Kombination aus einer 8-Bit Mikrocontroller-Plattform, einer Softwareentwicklungsumgebung und einer Softwarebibliothek mit typischen Mikrocontroller-Basisfunktionen können mit geringem Aufwand und nur grundlegendem Basiswissen über Softwareentwicklung Steuerungen für Prototypen entwickelt werden.

Einschränkend bei dieser Plattform sind die mangelnde Flexibilität bei der Auswahl geeigneter Mikrocontroller und die Ressourcenknappheit der meisten Arduino Mikrocontroller-Plattformen. Komplexe Regelungsalgorithmen sind dadurch nicht oder nur eingeschränkt realisierbar. Mangels hardwareunterstützter Fließkommaberechnung sind diese unter Umständen nicht schnell genug.

mbed Plattform

Die mbed Plattform wurde 2009 als Prototyping- und Lernplattform für Cortex-M Mikrocontroller vorgestellt. Eine im Mikrocontrollerumfeld einmalige Besonderheit ist die nicht notwendige lokale Entwicklungsumgebung, da die Softwareentwicklung webbasiert mit einer Online Toolchain erfolgt. Das Programmierinterface ist objektorientiert und dem Arduino Interface sehr ähnlich [1].

Auf der Embedded World 2014 wurden von ST Micro (STM) verschiedene Mikrocontrollerboards in der neuen Nucleo Produktfamilie mit mbed Kompatibilität vorgestellt [2]. Ebenfalls 2014 wurde auf der ARM TechCon 2014 von ARM CTO Mike Muller die Weiterentwicklung

von mbed, das mbed OS, als Leuchtturmprojekt von ARM für das Internet of Things vorgestellt [3]. Die bis dahin als mbed bezeichnete Plattform ist Kernbestandteil des mbed OS.

Die Bibliotheken (Hardware Abstraction Layer (HAL), Device Driver u.a.), die in der Online Toolchain verwendet werden, sind seit 2013 als Offline Toolchain bei der Open Source Plattform Github verfügbar [4].

Aktuell werden von der mbed Offline Toolchain 178 Hardwareplattformen (mbed Online 106) von 12 Mikrocontrollerherstellern unterstützt [1] [4]. Dabei wird der gesamte Bereich der ARM Cortex Kerne von M0 bis M7, mit single precision, double precision und ohne FPU, Gehäusen von DIP28 bis zu BGA216, Speichergrößen von 4k Flash /1K RAM bis zu 8MB Flash / 10MB RAM und z.T. mit integrierten Funkschnittstellen (BLE, SubGHz) abgedeckt. Auch wenn von mbed nur Cortex ARM Mikrocontroller unterstützt werden, ist die angebotene Bandbreite an Leistungsklassen wesentlich größer als durch das Arduino Projekt. Dieses unterstützt einige Derivate aus den Atmel ATmega und Atmel Cortex-M Familien, Intel x86 Prozessoren und Mini-Linux Systeme [5] [6].

Um die Einfachheit der Programmierung unter mbed zu demonstrieren, zeigt Abb. 1 wie ein Analogwert eingelesen, eine LED als Comparator-Anzeige mit Umschaltswelle 50% V_{inmax} verwendet und der Winkel eines Servos in Abhängigkeit des Analogwertes gesteuert wird:

```
#include "mbed.h"
#include "Servo.h"

AnalogIn ain(A0);
DigitalOut led(LED1);
Servo myServo(D12);

int main() {
    while (1){
        led = (ain > 0.5f) ? 1 : 0;
        myServo = ain;
        wait(0.2);
    }
}
```

Abb. 1: Beispiel für ein mbed Programm

Diese objektorientierte Programmierung erlaubt eine effektive prototypische Problemlösung, ohne sich umfangreich in die Mikrocontroller-Interna einarbeiten zu müssen. Ein Nachteil dieser einfachen Programmierbarkeit ist ein deutlich höherer Ressourcenverbrauch, der sich durch die zusätzlichen Abstraktionsebenen ergibt (siehe Abb. 2).

Matlab/Simulink ist als Engineering-Plattform weit verbreitet. Es gibt sowohl von Mathworks [7] als auch von Drittanbietern Erweiterungen für den Simulink und Embedded Coder [8], um für ausgewählte Plattformen, darunter auch einige Mikrocontroller, C/C++ Code zu erzeugen.

Damit ist eine Entwicklung auf sehr hohem Abstraktionsniveau möglich. Nachteil der verfügbaren Erweiterungen sind die Einschränkung auf bestimmte Mikrocontroller und/oder der Closed Source Charakter, der eigene Erweiterungen verhindert oder erschwert.

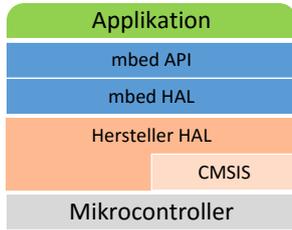


Abb. 2: Schichtenaufbau der mbed Plattform

mbed target

Das mbed target für Matlab/Simulink kombiniert die Vorteile der breiten Hardware Basis und der effektiven, objekt-orientierten Programmierschnittstelle von mbed mit dem hohen Abstraktionsniveau, das Simulink bietet. Abb. 3 zeigt ein zu Abb.1 funktionsäquivalentes Beispiel

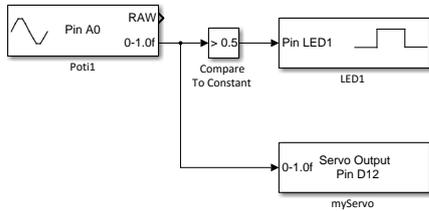


Abb. 3: Beispiel für ein Simulink Modell

Aus dem Modell bzw. aus Simulink heraus kann der Mikrocontroller direkt programmiert werden, d.h. zusätzlich zur Code Generierung durch den Embedded Coder werden automatisch Compiler, Linker und Flash Programm ausgeführt. Ein Umschalten auf einen anderen Mikrocontroller ist in der Simulink Modell Konfiguration mit wenigen Maus-Klicks möglich.

Das mbed target steht als Open Source Projekt auf GitHub zur Verfügung [9]. Es sind die wesentlichen mbed Objekte (GPIO, ADC, DAC, UART, SPI, I2C, ...) als Simulink Blöcke realisiert. Zusätzlich sind verschiedene Sensoren und Aktoren integriert. Aufgrund der einfachen mbed Programmierschnittstelle und anhand der vorhandenen im Source Code vorliegenden Blöcke können mit einem geringen Aufwand neue Funktionen integriert werden.

Anwendungsbeispiel: Prototyp einer künstlichen Hand

Im Folgenden wird die Toolkette zur Prototypenentwicklung am Beispiel einer künstlichen Hand demonstriert. Die Hand dient als Werkzeug an einem KUKA KR3 Roboter. Der 3D Entwurf wurde vor dem Druck simuliert, um die Beweglichkeit und den Arbeitsraum zu optimieren. Für den Druck wurden sowohl Laser Sinter als auch FDM Methoden verwendet, um die mechanische Stabilität zu testen. Abb. 4 zeigt CAD Teil-Zeichnungen von Fingergliedern, ein Modell zur Simulation und Optimierung des

Greifvorgangs, die komplette Hand als CAD Modell und ein Foto der fertigen Hand (im Uhrzeigersinn, links oben beginnend) und Abb. 5 das Simulink Modell.

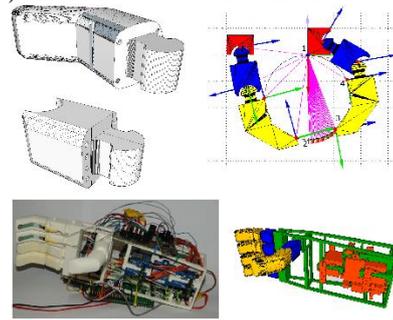


Abb. 4: künstliche Hand

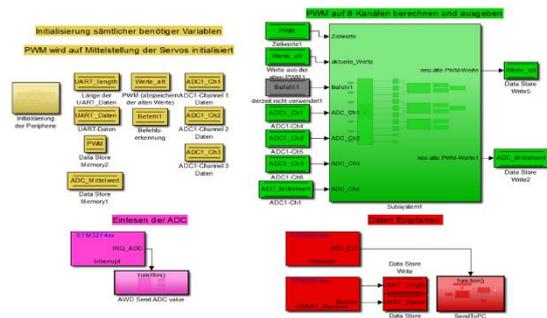


Abb. 5: Simulink Modell für die künstliche Hand

Zusammenfassung

Durch die Kombination von 3D Druckern und Fräsen mit 3D Simulation lassen sich mechanische Konstruktionen prototypisch aufbauen. Bauteile aus einem FDM 3D Drucker genügen einfachen Anforderungen an die mechanische Stabilität. Eine adäquate prototypische Softwareentwicklung ist mit dem vorgestellten mbed target für Matlab/Simulink möglich. Durch die Veröffentlichung als Open Source Projekt und die Verwendung von mbed als Basis sind Erweiterungen um eigene Simulink Blöcke realisierbar.

Literatur

- [1] mbed Projekt: <https://www.mbed.com/en/> (Zugriff 09.2016)
- [2] Pressenotiz: <https://developer.mbed.org/blog/entry/STM-microelectronics-mbed-enabled-Nucleo/> (Zugriff 09.2016)
- [3] Mike Muller, A Free mbed OS Accelerates Internet of Things, ARM TechCon 2014, Keynote
- [4] mbed-os GitHub Projekt: <https://github.com/ARM-mbed/mbed-os> (Zugriff 09.2016)
- [5] Arduino Project, <https://arduino.cc> (Zugriff 09.2016)
- [6] ARDUINO S.R.L., www.arduino.org (Zugriff 09.2016)
- [7] Simulink Targets: <http://de.mathworks.com/hardware-support/index.html?q=%20product:%22Embedded+Coder%22> (Zugriff 09.2016)
- [8] STM Simulink Target: <http://www.st.com/content/st.com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stm32-mat-target.html> (Zugriff 09.2016)
- [9] mbed target GitHub Projekt: https://github.com/ATM-HSW/mbed_target (Zugriff 09.2016)