

Inhaltsverzeichnis

1	Einleitung	1
1.1	Industriepartner2	1
1.2	Motivation	1
1.3	BeagleBone Black	1
1.4	Aufgabenstellung	1
2	Einleitung	3
2.1	Vorwort	3
2.2	EEROS	3
2.3	Klarstellung der Benennungen	3
2.4	Aufgabenstellung	3
3	EEROS aktueller Stand	5
3.1	EEROS generell	5
3.2	Aktuelle Implementierung des Sequenzers	5
3.3	Fallbeispiel Parallelsara	5
4	Anforderungen an den Sequencer	6
4.1	seciton	6
5	Aufbau des Sequencers	7
5.1	Caller Stack	7
6	Test des Sequencer	8
6.1	section	8
7	Fazit	9
7.1	section	9

1 Einleitung

1.1 Vorwort

1.2 EEROS

EEROS (Easy Elegant Reliable ...TODO) ist ein open source Roboter Framework, welches an der NTB entwickelt wurde und auch immer noch in Entwicklung ist. Das Ziel von EEROS ist, möglichst einfach, zuverlässig und einfach in der Bedienung zu sein. Da das Framework besonders auch in industriellen Robotern zum Einsatz kommen soll, ist besonders auch die Zuverlässigkeit der Software ein wichtiger Punkt. Für die Software wird die objektorientierte Programmiersprache C++ verwendet.

EEROS kann in vier Hauptbereiche unterteilt werden.

- 1.) Die HAL (Hardware Abstraction Layer) welche als Schnittstelle zur Hardware dient.
- 2.) Das CS (Control System). Im CS wird die Regelung des Roboters aufgebaut. In diesem System wird aber nicht nur die Regelung gerechnet, sondern auch Aufgaben wie die Berechnung der Vorwärts- und inversen Kinematik werden hier erledigt.
- 3.) Der Sequencer steuert den Ablauf des Roboters. Hier werden nicht nur Wegpunkte aufgelistet, sondern auch das allgemeine Verhalten definiert.
- 4.) Im SS (Safety System) werden sicherheitsrelevante Parameter überwacht. Das SS arbeitet unabhängig vom CS und vom Sequencer. Es löst einen Notaus aus, wenn der Roboter ausserhalb der zulässigen Parameter operiert. Ein möglicher Grund für einen Notstopp wäre zum Beispiel, wenn sich der Roboterarm in einem Sicherheitsbereich zu schnell bewegt.

1.3 Klarstellung der Benennungen

Mit den meisten Programmiersprachen werden in englisch codiert. Auch die offizielle Online-Dokumentation von EEROS, und die Benennung von Komponenten und Konzepten, ist in Englisch. In diesem Dokument wird an vielen Stellen bewusst darauf verzichtet, englische Bezeichnungen auf Deutsch zu übersetzen. Dies kann zu Deutsch - Englischen Mischwörter führen. Solche Mischwörter sind zwar nicht elegant, können aber besser für die Verständlichkeit sein und werden deshalb bewusst verwendet. Auch einige Eigennamen, wie z.B. *Sequencer* anstelle von *Sequencer* werden in diesem Dokument nicht auf Deutsch übersetzt.

In dieser Arbeit wird oft von drei verschiedenen Arten von Entwicklern gesprochen. Es wird zwischen EEROS-, Steuerungs-, und Applikations-Entwickler unterschieden.

Der EEROS-Entwickler hat vertiefte Kenntnisse der Programmiersprache C++ und vom EEROS Framework. Seine Hauptaufgabe ist die Weiterentwicklung des Frameworks, welches vom Steuerungs-Entwickler verwendet wird.

Der Steuerungs-Entwickler hat ebenfalls gute C++ Kenntnisse und nutzt das Framework, um eine Steuerung für einen Roboter zu entwickeln. Dafür muss er seine Software speziell auf den Roboter anpassen. Er bereitet auch erste Sequenzen für den Applikations-Entwickler vor. Oft wird die Entwicklung der Steuerung und der Applikation von der selben Person übernommen.

Um den Ablauf des Roboters anzupassen, kann der Applikations-Entwickler bestehende Sequenzen einfach anpassen. Dazu werden nur grundlegende Programmierkenntnisse benötigt. Mit etwas erweiterten Kenntnissen kann er auch neue Sequenzen erstellen.

1.4 Aufgabenstellung

In der aktuellen Version von EEROS existiert bereits eine erste Version von einem Sequencer. Dieser ist in seiner Funktionalität und Übersichtlichkeit aber stark eingeschränkt. Oft musste auf Tricks zurück gegriffen werden, damit bestimmte Aufgaben mit dem bestehenden Sequencer gelöst werden konnten.

Um eine bestehende Sequenz anpassen zu können, auch wenn der Ablauf nur geringfügig geändert werden soll, ist schon viel Fachwissen notwendig.

In dieser Vertiefungsarbeit sollen diese Probleme gelöst werden. Es soll ein neuer Sequencer entwickelt werden, der flexibel für verschiedenste Arten von Robotern eingesetzt werden kann. Die Sequenzen, welche den Ablauf des Roboters beschreiben, sollen dabei möglichst einfach und übersichtlich aufgebaut sein. Dank dem einfachen Aufbau soll es auch für einen Applikationsentwickler möglich sein, Sequenzen anzupassen und zu erstellen, auch wenn dieser Entwickler keine vertiefte Kenntnisse von C++ besitzt.

2 EEROS aktueller Stand

2.1 EEROS generell

2.2 Aktuelle Implementierung des Sequenzers

2.3 Fallbeispiel Parallels cara

3 Anforderungen an den Sequencer

3.1 seciton

4 Aufbau des Sequencers

4.1 Caller Stack

Das unterste (älteste) Element ist die ID-Nummer der Hauptsequenz. Als nächstes folgt

5 Test des Sequencer

5.1 section

6 Fazit

6.1 section

Anhang