

Embedded Betriebssystem

für ARM Cortex-A8

Nicolaj Höss, Marko Petrović, Kevin Wallis



Fachhochschule Vorarlberg
Master Informatik

S1: Softwarelösungen für ressourcenbeschränkte Systeme

14. Mai 2015

1. Was ist YAML
2. Aufbau und Syntax von YAML
3. Design des Kommunikationsprotokolls
4. Bewertung

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung

Referenzen

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung

Referenzen

1. Was ist YAML

2. Aufbau und Syntax von YAML

3. Design des Kommunikationsprotokolls

4. Bewertung

2 / 30

2
30

- ▶ „YAML Ain't Markup Language“
- ▶ vereinfachte Auszeichnungssprache
mensch- und maschinenlesbar
Gliederung und Formatierung von Text und Daten
- ▶ zur Datenserialisierung
- ▶ ursprünglich an XML angelehnt
- ▶ Datenstrukturen von Perl, Python und C
- ▶ entwickelt von Clark Evans (2001)

Die Designziele von YAML sind: [6]

- ▶ leicht von Menschen lesbar
- ▶ Portabilität zwischen Programmiersprachen
- ▶ Übereinstimmung mit natürlichen Datenstrukturen
- ▶ konsistentes Schema für generische Tools
- ▶ aussagekräftig und erweiterbar
- ▶ leicht zu Implementieren und zu Verwenden

C	C++
JavaScript	Java
Perl	PHP
Python	ActionScript
Ruby	.NET Plattform

Und viele weitere mehr.

Inhalt

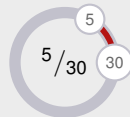
Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung

Referenzen



1. Was ist YAML

2. Aufbau und Syntax von YAML

2.1 Besonderheiten

2.2 Semantik

2.3 Andere Features

2.4 Anwendungsbeispiel (Ruby)

2.5 Vergleich mit JSON

3. Design des Kommunikationsprotokolls

4. Bewertung

Inhalt

Was ist YAML

**Aufbau und Syntax
von YAML**

Besonderheiten

Semantik

Andere Features

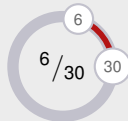
Anwendungsbeispiel
(Ruby)

Vergleich mit JSON

Design des
Kommunikations-
protokolls

Bewertung

Referenzen



Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Besonderheiten

Semantik

Andere Features

Anwendungsbeispiel
(Ruby)

Vergleich mit JSON

Design des
Kommunikations-
protokolls

Bewertung

Referenzen

Jede Datenstruktur darstellbar durch:

- Scalars (Einzelwerte)
 - Sequences (Arrays)
 - Mappings (Assoziativen Listen)
-
- ▶ Leerzeichen zum Kennzeichnen von Strukturen
 - ▶ Tabs sind nicht erlaubt

Unterstützte Typen:

- Strings
- Numerische Werte
- Boolesche Ausdrücke
- Spezielles Format für Datum

```
1 #Strings
2 'String'
3 "String"
4 #Numbers
5 12      #int
6 014     #octal
7 0xC     #hex
8 13.4    #float
9 1.2e+34 #exponential
10 .inf    #infinity
11 #Nulls
12 null
13 ~
14 #Boolean
15 true
16 false
17 #Dates
18 2002-12-14
```

Quellcode 1: Einzelwerte

Quelle: [2]

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Besonderheiten

Semantik

Andere Features

Anwendungsbeispiel (Ruby)

Vergleich mit JSON

Design des Kommunikationsprotokolls

Bewertung

Referenzen

Erzeugen von Array:

- ▶ Dash+Space
- ▶ oder im Flowstyle

```
1 a_sequence:  
2   - PHP  
3   - Perl  
4   - Python  
5 #or Flowstyle:  
6 a_sequence:[PHP, Perl, Python]
```

Quellcode 2: Arrays

Quelle: [2]

Assoziative Listen (Mappings)

```
1 a_map:
2   PHP: 5.2
3   MySQL: 5.1
4   Apache: 2.2.20
5 #or Flowstyle:
6 a_map: {PHP: 5.2, MySQL: 5.1,
7   Apache: 2.2.20}
```

Quellcode 3: Assoziative

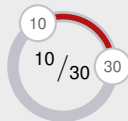
Quelle: [2]

- ▶ bestehend aus Key und Value
- ▶ Key: Value
- ▶ auch im Flowstyle

```
1 #Mix Mappings und Sequences:
2 'Chapter1': [Introduction, Event Types]
3 "Symfony1": {PHP: 5.0, Propel: 1.2}
```

Quellcode 4: Mischung

Quelle: [2]



```
1 #Anchors:
2 anchored_content: &anchor_name
3 other_anchor: *anchor_name
4
5 #Declare Types:
6 !!str 0.5
7
8 #Set type:
9 set:
10   ? item1
11   ? item2
12   ? item3
13 #entspricht
14 set:
15   item1: null
16   item2: null
17   item3: null
```

Quellcode 5: Features

Quelle: [3]

- ▶ Anchor: ein Value für zwei Keys (Inhalt duplizieren)
- ▶ mit !! wird ein Typ explizit angegeben (tags)
- ▶ !!binary: base64-encoded representation of a binary blob
- ▶ Set entspricht Map mit Null Values

```
1 #require 'yaml'
2 # Set.new([1,2]).to_yaml
3 --- !ruby/object:Set
4 hash:
5   1: true
6   2: true
```

Quellcode 6: Features
Quelle: [4]

- ▶ Kommentar: Zeile in Ruby
- ▶ Der Rest zeigt das Ergebnis in YAML
- ▶ —: Tag der von Ruby erzeugt wurde
- ▶ YAML erlaubt das Hinzufügen von Metadaten

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Besonderheiten

Semantik

Andere Features

Anwendungsbeispiel (Ruby)

Vergleich mit JSON

Design des Kommunikationsprotokolls

Bewertung

Referenzen

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Besonderheiten
Semantik
Andere Features
Anwendungsbeispiel (Ruby)

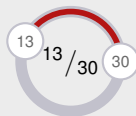
Vergleich mit JSON

Design des Kommunikationsprotokolls

Bewertung

Referenzen

- JSON ist Untermenge von YAML
- JSON Dokument ist gleichzeitig YAML 1.2 Dokument (V1.1 nicht!)
- JSON Designziele: Einfachheit, Universalität
- YAML Designziele: Lesbarkeit durch Mensch, Unterstützt serielle Übertragung von Datenstrukturen
- Was YAML erlaubt und JSON nicht: [7]
 - Kommentare
 - Tags für Metadaten
 - Anchors
 - Dokument Start- und Endmarkierungen



Vergleich mit JSON (Performance)

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Besonderheiten
Semantik
Andere Features
Anwendungsbeispiel (Ruby)

Vergleich mit JSON

Design des Kommunikationsprotokolls

Bewertung

Referenzen

Method	Simple	Complex
JSON.generate	0.1550s	0.5830s
JSON.pretty_generate	0.1470s	0.6060s
YAML.dump	2.4531s	3.4732s

Execution times in seconds for each method on both data sets.

Abb. 1: Serialization (dump) performanceQuelle: [8]

Method	Simple	Complex
JSON.parse	0.0440s	0.0790s
YAML.load	0.2750s	0.3360s

Execution times in seconds for each method on both data sets.

Abb. 2: Deserialization (load) performanceQuelle: [8]

1. Was ist YAML

2. Aufbau und Syntax von YAML

3. Design des Kommunikationsprotokolls

3.1 Message Typen

3.2 Designkriterien des Protokolls

3.3 Aufbau Protokoll

3.4 Messages

3.5 Beschreibung und Abhängigkeiten

4. Bewertung

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen
Designkriterien des
Protokolls
Aufbau Protokoll
Messages
Beschreibung und
Abhängigkeiten

Bewertung

Referenzen



- ▶ Start
- ▶ Stop
- ▶ Set Default Values
- ▶ Get Default Values
- ▶ Report -Default Values
- ▶ Report -Status
- ▶ Request -Status
- ▶ Report -Error

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen
Designkriterien des
Protokolls
Aufbau Protokoll
Messages
Beschreibung und
Abhängigkeiten

Bewertung

Referenzen

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen

**Designkriterien des
Protokolls**

Aufbau Protokoll

Messages

Beschreibung und
Abhängigkeiten

Bewertung

Referenzen

Designziele (nach Priorität):

- Lesbarkeit für den Menschen
- Defaultparameter sind vorhanden
- Reihenfolge der Parameter beliebig
- Wenig Overhead (Geschwindigkeit)

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen
Designkriterien des
Protokolls

Aufbau Protokoll

Messages
Beschreibung und
Abhängigkeiten

Bewertung

Referenzen

Tabelle 1: Protokollaufbau

"MessageType"	:	{	Parametername	:	Wert	,	usw	}
---------------	---	---	---------------	---	------	---	-----	---

YAML Ergebnis:

MessageType:

Parametername1: Wert1

Parametername2: Wert2

Parametername3: Wert3

Start Message

```
1 Start: {  
2 Delay: 0,  
3 FinalSpeed: 100,  
4 RiseTime: 2,  
5 Direction: Right,  
6 AutoStatus: False }
```

Quellcode 7: Start Message

Command or Message Type	Command option or Data	Default	Description
Start			
	Delay (in sec)	0	Delay start by x seconds
	Final speed (in %)	100	Final speed/amplitude as percentage value of maximum speed/amplitude (e.g. max. speed = max. amplitude = 30 s ⁻¹)
	Rise time (in sec)	2	Raise time in seconds = Time to reach final speed in seconds = rise time (t _r)
	Direction	Right	Direction: right (clockwise) or left (counter clockwise) rotation
	Auto Status	False	If this option is set, the addressed controller unit will send a report message automatically after the delay time (if there is one) and after reaching the final speed.

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Design des Kommunikationsprotokolls

Message Typen
Designkriterien des Protokolls
Aufbau Protokoll

Messages

Beschreibung und Abhängigkeiten

Bewertung

Referenzen

Stop Message

```
1 Stop: {  
2 Delay: 0,  
3 FinalSpeed: 0,  
4 FallTime: 2,  
5 AutoStatus: False,  
6 EmergencyStop: False }
```

Quellcode 8: Stop Message

Stop			
Delay (in sec)	0	Delay stop by x seconds	
Final speed (in %)	0	Final speed/amplitude as percentage value of maximum speed/amplitude (e.g. max. speed = max. amplitude = 30 s^{-1})	
Fall time (in sec)	2	Raise time in seconds = Time to reach final speed in seconds = rise time (t_r)	
Auto Status	False	If this option is set, the addressed controller unit will send a report message automatically after the delay time (if there is one) and after reaching the final speed.	
Emergency Stop	False	If this option is set to true, all other options will be ignored and the motor will stop immediately	

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Design des Kommunikationsprotokolls

Message Typen
Designkriterien des Protokolls
Aufbau Protokoll

Messages
Beschreibung und Abhängigkeiten

Bewertung

Referenzen

Set und Get Default Values Messages

```
1 SetDefaultValues: {  
2   Delay: 0,  
3   FinalSpeed: 100,  
4   FallTime: 2,  
5   RampTime: 3,  
6   Direction: Right,  
7   FinalSpeedStart: 100,  
8   FinalSpeedStop: 0 }
```

Quellcode 9: Set Default Values Message

```
1 GetDefaultValues: {}
```

Quellcode 10: Get Default Values Message

Set Default Values	
Delay	
Rise Time	
Fall Time	
Ramp Time	Sets rise time and fall time to the same value
Direction	
Final Speed – Start	
Final Speed – Stop	

Get Default Values	Requests the addressed controller unit to send its default values report
--------------------	--

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Design des Kommunikationsprotokolls

Message Typen
Designkriterien des Protokolls
Aufbau Protokoll

Messages

Beschreibung und Abhängigkeiten

Bewertung

Referenzen

Report Default Values Message

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen
Designkriterien des
Protokolls
Aufbau Protokoll

Messages

Beschreibung und
Abhängigkeiten

Bewertung

Referenzen

```
1 ReportDefaultValues: {  
2   Delay: 0,  
3   RiseTime: 2,  
4   FallTime: 2,  
5   Direction: Right,  
6   FinalSpeedStart: 100,  
7   FinalSpeedStop: 0 }
```

Quellcode 11: Report Default Values
Message

Report -- Default Values	Message which contains the current default values of the reporting controller unit
Delay	
Rise Time	
Fall Time	
Direction	
Final Speed – Start	
Final Speed – Stop	

22 / 30

30

22

Report und Request Messages

1 `ReportStatus: { Status: }`

Quellcode 12: Report Status Message

1 `RequestStatus: {}`

Quellcode 13: Request Status Message

1 `ReportError: {}`

Quellcode 14: Report Error Message

Report – Status	Reports current status as answer to a Request – Status message (see below)
Status	Status is one of (ready/bereit, wait/warten, release/freigabe)

Request – Status	Requests the status message from the addressed controller unit (anfrage)
------------------	--

Report – Error	Report on certain error conditions
----------------	------------------------------------

Inhalt

Was ist YAML

Aufbau und Syntax von YAML

Design des Kommunikationsprotokolls

Message Typen
Designkriterien des Protokolls
Aufbau Protokoll

Messages

Beschreibung und Abhängigkeiten

Bewertung

Referenzen

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

**Design des
Kommunikations-
protokolls**

Message Typen
Designkriterien des
Protokolls
Aufbau Protokoll
Messages

**Beschreibung und
Abhängigkeiten**

Bewertung

Referenzen

Parametername hätte auch weggelassen werden können.
(Verwendung von Array statt Assoziativen Listen)

Wir haben uns dagegen entschieden weil:

- Lesbarkeit verschlechtert wird
- Reihenfolge der Parameter nicht mehr beliebig wäre

Nachteile dieser Entscheidung:

- Overhead (Geschwindigkeit)

1. Was ist YAML
2. Aufbau und Syntax von YAML
3. Design des Kommunikationsprotokolls
4. Bewertung
 - 4.1 Vorteile
 - 4.2 Nachteile

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

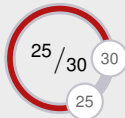
Design des
Kommunikations-
protokolls

Bewertung

Vorteile

Nachteile

Referenzen



YAML:

- ++ Leicht vom Mensch Schreiben, Lesen (vgl. XML)
- ++ Weiterverarbeitung der Daten in anderen Sprachen
 - + Hinzufügen von Tags erlaubt
 - + Sprache ist leicht zu Verwenden

Entwickeltes Protokoll:

- ++ Leicht vom Mensch Schreiben, Lesen (vgl. XML)
 - + Defaultparameter sind vorhanden (nicht alle Senden)
 - + Reihenfolge der Parameter ist beliebig

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung
Vorteile

Nachteile

Referenzen

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung

Vorteile

Nachteile

Referenzen

YAML:

- – Geschwindigkeit (vgl. JSON)
- – Verbreitung und Bekanntheit
 - Wird immer noch weiterentwickelt (Ausgereiftheit)

Entwickeltes Protokoll:

- Vermeidbarer Overhead (Siehe Abhängigkeiten)
- Sender und Empfänger brauchen Befehlsliste

- [1] Wikipedia Community
YAML.
<http://en.wikipedia.org/wiki/YAML>, 2015.
- [2] Symfony
The YAML Format.
http://symfony.com/doc/current/components/yaml/yaml_format.html, 2015.
- [3] Adam Brenecki
Learn X in Y minutes. Where X=yaml.
<http://learnxinyminutes.com/docs/yaml/>, 2015.
- [4] Xavier Shay
Robot Has No Heart. YAML Tutorial.
<http://rnhh.net/2011/01/31/yaml-tutorial>, 2015.

Inhalt

Was ist YAML

Aufbau und Syntax
von YAML

Design des
Kommunikations-
protokolls

Bewertung

Referenzen

- [5] yaml-cpp
HowToEmitYAML.
<https://code.google.com/p/yaml-cpp/wiki/HowToEmitYAML>,
2012.
- [6] Clark Evans et al.
YAML Ain't Markup Language Version 1.2.
<http://yaml.org/spec/1.2/spec.html>, 2009.
- [7] CowTalk
What me like YAML?
http://www.cowtowncoder.com/blog/archives/2012/04/entry_473.html, 2009.

- [8] Malin Eriksson und Viktor Hallberg
*Comparison between JSON and YAML for data
serialization.*
[http://www.csc.kth.se/utbildning/kth/kurser/
DD143X/dkand11/Group2Mads/victor.hallberg.malin.
eriksson.report.pdf](http://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand11/Group2Mads/victor.hallberg.malin.eriksson.report.pdf), 2011.



The End

Danke für Ihre Aufmerksamkeit.