JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll
State-Machines

Systemtechnik Labor
5BHITT 2015/16

Stefan Geyer

Version 0.1

Note:

Betreuer: Prof. Weiser

Begonnen am 25. September 2015

Beendet am 2. Oktober 2015

1 Inhaltsverzeichnis

2 Einführung

2.1 Ziele

Umsetzung der State Machines in Form einer Ampel auf einem STM32F3

2.2 Voraussetzungen

Die notwendigen Informationen für die Laborübung können im Skriptum von Elicia White „Making Embedded Systems" im Kapitel 5 nachgelesen werden. Dort werden die unterschiedlichen State-Machines ausführlich beschrieben. Bevor die Aufgabe begonnen wird, sollten jedoch die Vor- und Nachteile dieser definiert werden.

State-Centric State Machine

In einem Switch-Case Konstrukt wird der derzeitige Status der Ampel abgefragt (Rot, Gelb, …). Wenn das zugehörige Event stimmt, nimmt die Ampel den nächsten Status an.

Vorteile

- Relativ einfach zu Verstehen und umzusetzen

Nachteile

- Komplett statisch
- Jeder Zustand muss über die anderen Bescheid wissen
- Switch Konstrukt wird sehr verschachtelt wenn viele Zustände vorhanden sind

State-Centric State Machine with hidden Transitions

Funktioniert vom Prinzip her wie die State-Centic Version, verzichtet allerding komplett auf die Abfrage der Events. (z.B. Wenn der Status Rot ist wird er auf Rot-Gelb geändert)

Vorteile

- Vereinfachte Version der State-Centric Version
- Bessere Verkapselung
- Geringere Abhängigkeiten

Nachteile

- Komplett statisch

Event-Centric State Machine

Eine gespiegelte Version der State Centric Version. Das Switch Statement handelt nach den Events ab. In den einzelnen Case Statements wird allerdings noch nach dem Status der Ampel abgeprüft.

Vorteile

- Geeignet wenn viele Events auftreten müssen, damit sich der Satus ändern kann

Nachteile

- Komplett Statisch
- Je mehr Events, desto ineffizienter Arbeitet das Programm

State-Pattern State Machine

Dies ist eine objektorientierte Umsetzung einer State Machine. Es wird eine Klasse erstellt welche verschiedene Methoden besitzt die jeweils auf ein Event reagieren:

- Enter – wird aufgerufen, wenn man dem Status beigetreten ist.
- Exit – wird aufgerufen, wenn man den Status verlässt.
- EventGo – verarbeitet den Start eines bestimmten Events.
- EventStop – verarbeitet das Ende eines bestimmten Events.
- Housekeeping – geeignet für Zustände, die periodisch überprüft werden (z.B. Timeout)

Vorteile

- objektorientiert
- gekoppelt
- dynamischer
- eigene Funktion um Event abzuhandeln

Nachteile

- komplexeste Version
- Implementierung in C nicht möglich, da C keine objektorientierte Sprache ist

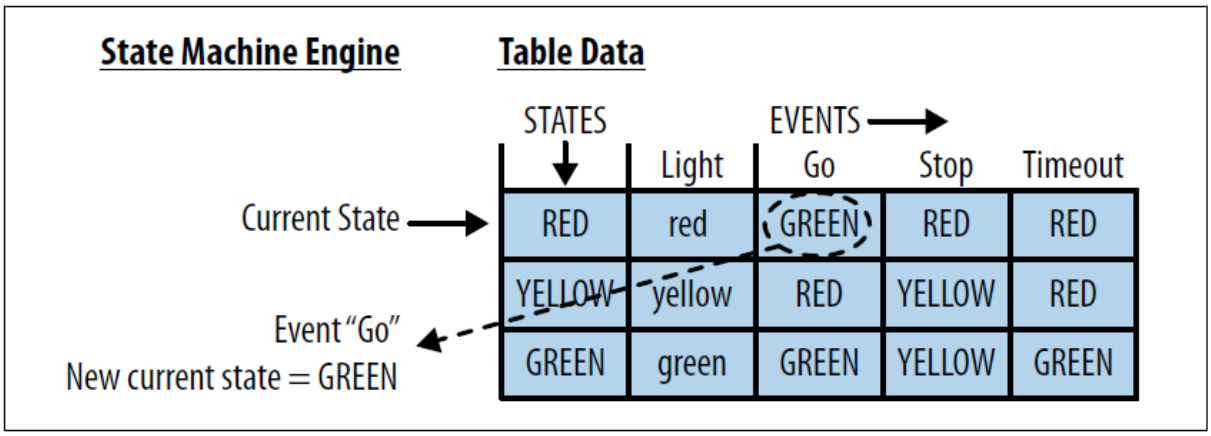
Beispiel

```
class Context {
	class State Red, Yellow, Green;
	class State Current;

constructor:
	Current = Red;
	Current.Enter();

destructor:
	Current.Exit();

Go:
	if (Current.Go() indicates a state change)
		NextState();

Stop:
	if (Current.Stop() indicates a state change)
		NextState();

Housekeeping:
	if (Current.Housekeeping() indicates a state change)
		NextState();

NextState:
	Current.Exit();
	if (Current is Red) Current = Green;
	if (Current is Yellow) Current = Red;
	if (Current is Green) Current = Yellow;
	Current.Enter();
}
```

[2]

Table-Driven State Machine

Hierbei wird eine Tabelle verwendet um die verschiedenen Status darzustellen.



[2]

Vorteile

- Besser lesbar

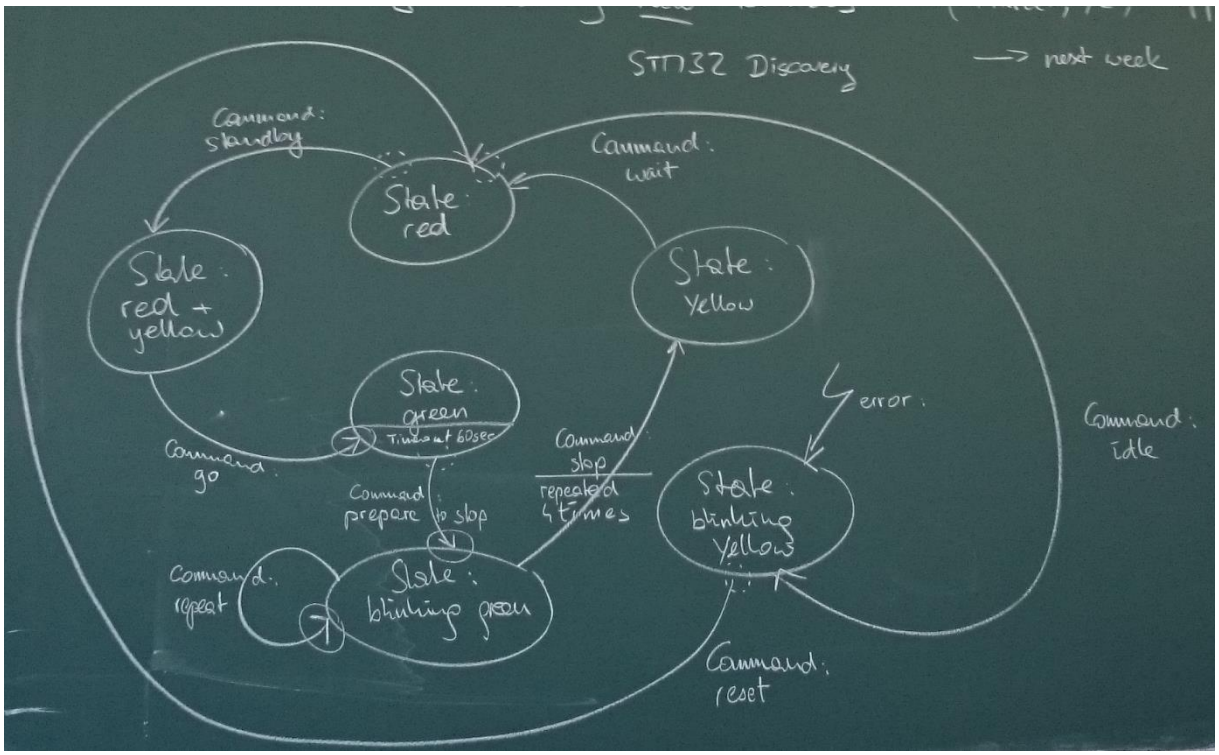
Nachteile

- Tabelle schlechter lesbar, Fehler passieren leichter

2.3 Aufgabenstellung

Implement a component based C-Program to show the difference of the 5 types of state machines presented in the book of Mrs. Elicia White "Making Embedded Systems" with traffic light system we discussed in the lesson. To test your implementation you can use simple output functions (e.g. fprintf), but be prepared to implement it also on hardware (GPIO with Leds, Timers, etc.).

Don't forget to document the differences (advantages/disadvantages) in your protocol.



3 Ergebnisse

3.1 Vorbereitung

Um die Applikationen zu testen und zu auf den STM zu flashen wird mit eclipse gearbeitet.
Die Website http://www.openstm32.org bietet ein Tutorial zur korrekten Konfiguration der IDE.
Damit mit der HAL-Library gearbeitet werden kann, muss eine eclipse Erweiterung über „Install new software" von der Website http://www.ac6-tools.com/Eclipse-updates/org.openstm32.system-workbench.site geladen werden.

Damit der STM unter Windows erkannt wird und für das spätere flashen bereitsteht, muss ein Treiber von der Website http://www.st.com/web/catalog/tools/FM147/SC1887/PF260218 geladen und installiert werden

3.2 Projekterstellung

Beim Wizard müssen die Einstellungen der neu installierten Software benutzt und die STM Einstellungen angepasst werden (Model, …). Dafür wird die Toolchain Ac6 STM32 MCU GCC verwendet.
Nun kann die spezielle Debug-Configuration (AC6) verwendet werden um das Projekt zu debuggen.
In manchen Fällen kann es notwendig sein, dass der STM an einem USB2.0 Port angeschlossen ist, und eclipse als Administrator ausgeführt wird.

3.3 Erstellen eines Header Files

Ein Header-File wurde erstellt. Dieses beinhält Enumerations für die Events und für den Status der Ampel, eine Struktur in welcher das derzeitige Event, der derzeitige Status, einen Counter zum Übergeben der Blink-Anzahl gespeichert werden und den Prototyp der State-Machine Methode.

3.4 Erstellen der LED Funktionen

Für jeden Zustand (Rot, Gelb, Gelb-Blink, …) wird eine Eigene LED Methode erstellt die später von der State Machine aufgerufen werden kann.

3.5 Umsetzen der State Machines

Die State Machines werden jeweils in einem eigenen File implementiert welches das Header File included.

3.6 Aufrufen in der Main Methode

Nun wird die State Machine Methode in der Main aufgerufen und kann auf den STM geflasht werden.

4 Quellen

[1] OpenSTM32 Community Site - http://www.openstm32.org/
zuletzt abgerufen am 25.09.2015 (online)

[2] Elicia White: Making Embedded Systems –
http://it-ebooks.info/book/549/
zuletzt abgerufen am 02.10.2015 (online)

[3] STMicroelectronics: ST-LINK/V2-1 USB Driver für Windows -
http://www.st.com/web/catalog/tools/FM147/SC1887/PF260218
zuletzt abgerufen am 25.09.2015 (online)