

git für die tägliche Arbeit

Andreas Bießmann

30. April 2016

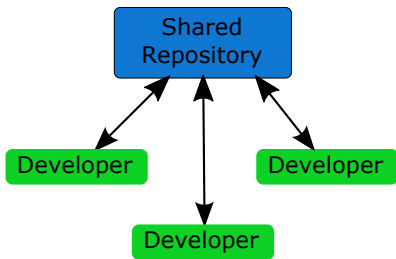
Inhalt

- 1 Einführung
 - Modelle der Zusammenarbeit
- 2 git: Grundlagen
 - Wichtige Kommandos
 - Globale Konfiguration
 - Mein U-Boot Repository
- 3 git: Erweiterte Funktionen
 - Rebase des Repositories
 - Arbeiten mit Patches
 - Reviewprozess
- 4 Patman
 - Was ist Patman
 - Wie funktioniert Patman

Über mich

- Embedded Linux Entwicklung im medizinischen Umfeld.
- Privat im U-Boot Umfeld aktiv.

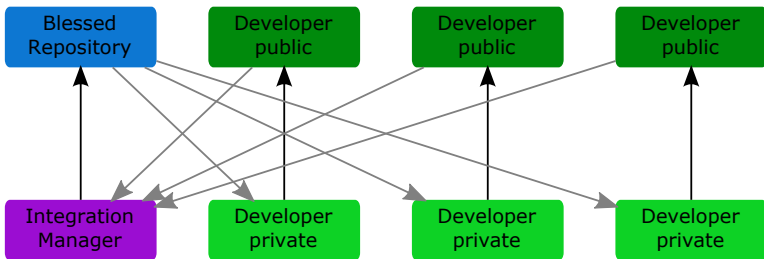
Zentralisierter Workflow



- ✓ Verbreitetes, bekanntes Vorgehen von SCMs¹ wie z.B. SVN.
- ✓ Schneller Einstieg, wenig Verwaltungs-Overhead.
- ✗ Jeder muss regelmäßig Integration durchführen.
- ✗ Skaliert nicht für große Umgebungen.
- ✗ Jeder benötigt Zugriff zum Server.

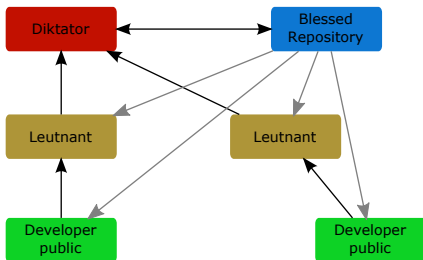
¹Source Control Managements

Integration Manager Workflow



- ✓ Überprüfungsmöglichkeit an zentraler Stelle.
- ✓ Zugriff auf Server kann beschränkt werden.
- ✗ Erhöhter Verwaltungsaufwand für Entwickler Repos.
- ✗ Erhöhter Kommunikationsaufwand, kann schon eine Mailingliste rechtfertigen.

Diktator und Leutnants Workflow



- ✓ Gute Steuerung der Entwicklung.
- ✓ Skaliert sehr gut bei großen Projekten (Linux Kernel).
- ✗ Maximaler Kommunikationsaufwand, erzwingt eine Mailingliste.

Inhalt

- 1 Einführung
 - Modelle der Zusammenarbeit
- 2 git: Grundlagen
 - Wichtige Kommandos
 - Globale Konfiguration
 - Mein U-Boot Repository
- 3 git: Erweiterte Funktionen
 - Rebase des Repositories
 - Arbeiten mit Patches
 - Reviewprozess
- 4 Patman
 - Was ist Patman
 - Wie funktioniert Patman

Übersicht

`git add` Hinzufügen von Dateien oder Änderungen zum 'staging' Bereich.

`git commit` Den 'staging' Bereich in einen Kommit überführen.

`git checkout` Zwischen lokalen Zweigen umschalten.

`git remote` Management von Remote Repositories.

`git fetch` Aktualisieren von Remote Repositories.

`git status` Abfragen des aktuellen Status.

`git merge` Zusammenbringen verschiedener Entwicklungsstränge.

`git pull` Aktualisierung und zusammenbringen des lokalen und entfernten Repositories in einem Kommando.

`git push` Das lokale Repository veröffentlichen.

Handling von lokalen Änderungen

- 1 Code anpassen / Dateien hinzufügen
- 2 Änderungen hinzufügen:
`git add [--patch] [<file>]`¹
- 3 Testlauf
- 4 Die passenden Änderungen als Commit übernehmen:
`git commit -s -m <Message>`²

¹see `git-add(1)`

²see `git-commit(1)`

Handling von Remote Repositories

Hinzufügen neuer Repositories

```
git remote add <name> <url>
```

Ändern einer Url

```
git remote set-url [--push] <name> <url>
```

Fetch der Repositories

```
git remote update [<name>]
```

Handling von Änderungen in Remote Repositories

- 1 Update der lokalen Kopie des Repositories:
`git fetch <name>`¹
- 2 Status des Repositories erfragen:
`git status`
- 3 Einpflegen der Änderungen:
`git merge <refspec>`²
- 4 Ggf. Veröffentlichen des neuen Status:
`git push <repository> <refspec>`³

¹Alternativ `git remote update`

²Alternativ `git pull`

³see `git-push(1)`

Update eines lokalen Zweiges

Status erfragen

```
% git status
On branch avr32
Your branch is behind 'avr32/master' by 2 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
nothing to commit, working directory clean
```

Änderungen einspielen

```
% git pull
Updating f4c6f93..e25b369
Fast-forward
 configs/uniphier_ld20_defconfig | 2 ++
 include/configs/xilinx_zynqmp.h | 4 ----
 2 files changed, 2 insertions(+), 4 deletions(-)
```

Basiskonfiguration

Meine Basiskonfiguration

```
% git config --global -l
user.name=Andreas Bießmann
user.email=andreas@biessmann.de
log.decorate=short
color.ui=auto
```

user.name Echtname, wird für Author verwendet.

user.email Mailadresse, wird für Author verwendet.

log.decorate Zeige Tags und Zweige im log.

color.ui Farbige Ausgaben ...

U-Boot Setup

Remotes für mein U-Boot Repository

```
% git remote
arm
atmel
origin
```

Remote 'atmel'

```
% git remote show atmel
* remote atmel
Fetch URL: git://git.denx.de/u-boot-atmel.git
Push URL: gu-atmel@git.denx.de:u-boot-atmel
HEAD branch: master
Remote branches:
  fix-at91cap9 tracked
  master         tracked
  old-next       tracked
Local branch configured for 'git pull':
  atmel merges with remote master
Local ref configured for 'git push':
  master pushes to master (local out of date)
```

Inhalt

- 1** Einführung
 - Modelle der Zusammenarbeit
- 2** git: Grundlagen
 - Wichtige Kommandos
 - Globale Konfiguration
 - Mein U-Boot Repository
- 3** git: Erweiterte Funktionen
 - Rebase des Repositories
 - Arbeiten mit Patches
 - Reviewprozess
- 4** Patman
 - Was ist Patman
 - Wie funktioniert Patman

Übersicht

`git rebase` Geschichte des Repository neu schreiben.

`git format-patch` Patch Datei(en) aus dem Repository erstellen.

`git send-email` Patch Datei(en) per Mail verschicken.

`git am` Patch Datei(en) in das Repository einbinden.

`git request-pull` Eine Mail mit der Anfrage um einen Pull erzeugen.

Rebase des Repositories

Update eines lokalen Zweiges mit Änderungen

Status erfragen

```
% git status
On branch avr32
Your branch and 'avr32/master' have diverged,
and have 1 and 2 different commits each, respectively.
   (use "git pull" to merge the remote branch into yours)
nothing to commit, working directory clean
```

Änderungen einspielen

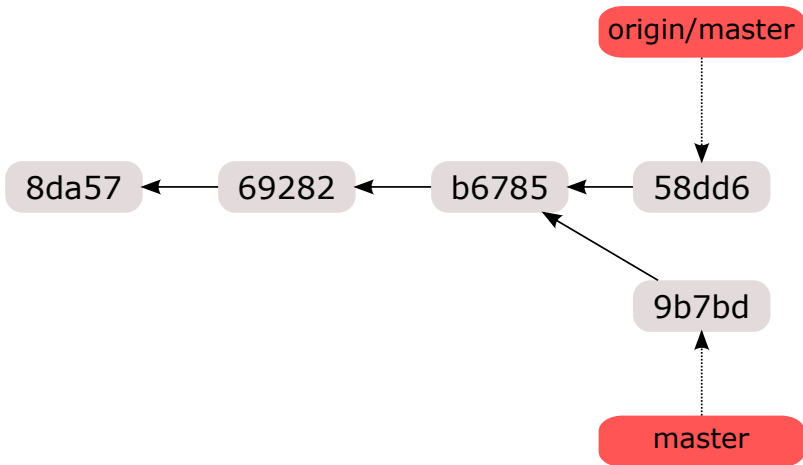
```
% git pull --rebase
First, rewinding head to replay your work on top of it...
Applying: Makefile: name it 'test'
```

Status überprüfen

```
On branch avr32
Your branch is ahead of 'avr32/master' by 1 commit.
   (use "git push" to publish your local commits)
nothing to commit, working directory clean
```

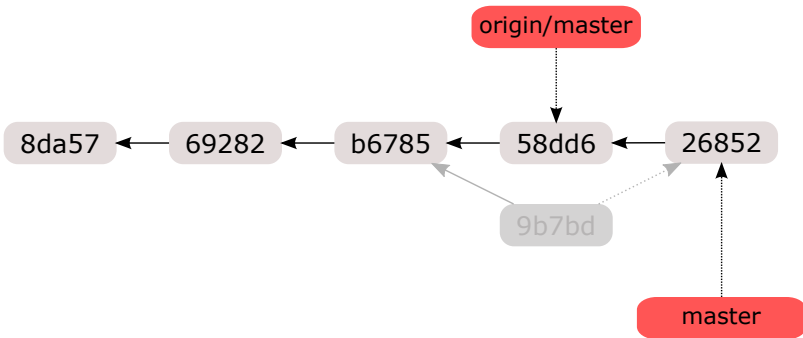
Rebase des Repositories

Zustand vor git pull --rebase



Rebase des Repositories

Zustand nach git pull --rebase



Rebase des Repositories

git rebase

Was macht das?

Verändert die Geschichte eines Repositories!

Wann braucht man das?

- Lokal und Remote haben sich auseinander bewegt.
- Vorwärts Portierung einer Featurebranch.
- Backport eines Fixes.
- Commits hübsch machen.
- Review Kommentare in Commit einarbeiten.
- ...

Interaktiver Rebase: `git rebase --interactive`

Konfiguration des Rebase

```
pick 58dd6bf git-patman: update centralizedWorkflow image
pick 9b7bd76 git-patman: update integrationManagerWorkflow image
pick 26852ce git-patman: introduce pro/con list
pick 328200a git-patman: add dictator workflow
pick 866b1aa git-patman: add minipage for images

# Rebase b67856c..866b1aa onto b67856c
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

Patch erzeugen

```
% git format-patch HEAD~
0001-drivers-video-am335x-fb-Add-support-for-16bpp-format.patch
0002-drivers-video-am335x-fb-Properly-point-framebuffer-b.patch
% head -20 0001-drivers-video-am335x-fb-Add-support-for-16bpp-format.patch
From 7d045170ac1bf512ed148e97432b7a8d772943e9 Mon Sep 17 00:00:00 2001
From: Martin Pietryka <martin.pietryka@chello.at>
Date: Wed, 27 Apr 2016 21:39:15 +0200
Subject: [PATCH 1/2] drivers/video/am335x-fb: Add support for 16bpp format
```

To support 16bpp we just need to change the raster_ctrl register accordingly. Also 32bpp mode should work as well, but was not tested. According to the TRM the uppermost byte will be ignored when LCD_TFT_24BPP_UNPACK is set.

The switch logic is based on the Linux kernel tilcdc driver: drivers/gpu/drm/tilcdc/tilcdc_crtc.c: lines 407 through 419 (kernel was checked out at commit: bcc981e9ed8)

```
Signed-off-by: Martin Pietryka <martin.pietryka@chello.at>
Reviewed-by: Hannes Schmelzer <oe5hpm@oevsv.at>
Tested-by: Hannes Schmelzer <oe5hpm@oevsv.at>
---
```

```
drivers/video/am335x-fb.c | 22 ++++++-----
1 file changed, 19 insertions(+), 3 deletions(-)
```

Basiskonfiguration Mail

Meine Basiskonfiguration für Mail

```
% git config --global -l | grep sendemail
sendemail.google.from=Andreas Bießmann <andreas.devel@googlemail.com>
sendemail.google.smtpserver=/usr/sbin/sendmail
sendemail.google.smtpserveroption=-f andreas.devel@googlemail.com
sendemail.identity=biessmann
sendemail.assume8bitencoding=UTF-8
sendemail.composeencoding=UTF-8
sendemail.biessmann.from=Andreas Bießmann <andreas@biessmann.de>
sendemail.biessmann.smtpserver=/usr/sbin/sendmail
sendemail.biessmann.smtpserveroption=-f andreas@biessmann.de
```

- Mehrere Konfigurationen (google, biessmann)
- Auswahl über `sendemail.identity`
- Einstellungen für Non-ASCII Buchstaben

Patch versenden

```
% git send-email 0001-drivers-video-am335x-fb-Add-support-for-16bpp-format.patch
0001-drivers-video-am335x-fb-Add-support-for-16bpp-format.patch
(mbox) Adding cc: Martin Pietryka <martin.pietryka@chello.at> from line 'From: Martin Pietryka <martin.pietryka@chello.at>'
(body) Adding cc: Martin Pietryka <martin.pietryka@chello.at> from line 'Signed-off-by: Martin Pietryka <martin.pietryka@chello.at>'
```

```
From: =?UTF-8?q?Andreas=20Bie=C3=9Fmann?= <andreas.devel@googlemail.com>
To: U-Boot ML <u-boot@lists.denx.de>
Cc: Martin Pietryka <martin.pietryka@chello.at>
Subject: [PATCH 1/2] drivers/video/am335x-fb: Add support for 16bpp format
Date: Fri, 29 Apr 2016 22:18:33 +0200
Message-Id: <1461961113-11129-1-git-send-email-andreas.devel@googlemail.com>
X-Mailer: git-send-email 2.6.4 (Apple Git-63)
```

The Cc list above has been expanded by additional addresses found in the patch commit message. By default send-email prompts before sending whenever this occurs. This behavior is controlled by the sendemail.confirm configuration setting.

For additional information, run 'git send-email --help'. To retain the current behavior, but squelch this message, run 'git config --global sendemail.confirm auto'.

Send this email? ([y]es|[n]o|[q]uit|[a]ll):

Mail Konfiguration

Mailserver Konfiguration

```
% git config -l | grep sendemail
sendemail.identity=google
sendemail.to=U-Boot ML <u-boot@lists.denx.de>
sendemail.aliasfile=doc/git-mailrc
sendemail.aliasfiletype=mutt
```

- Überschreiben der Mailserver Konfiguration.
- To: Header konfigurieren.
- Projekt Aliase für Abkürzungen der Empfänger in Mails.

Mail Aliase

Aufbau der Alias Datei

```
# To use this file, run in your u-boot tree:
# git config sendemail.aliasfiletype doc/git-mailrc
# git config sendemail.aliasfiletype mutt
#
# Then when sending patches, you can use:
# git send-email --to u-boot --cc i2c ...

alias uboot u-boot@lists.denx.de
alias u-boot uboot

...

# Architecture aliases
alias arch arc, arm, avr32, bfin, m68k, microblaze, mips, nds32, nios2, powerpc, ↵
        sandbox, superh, sparc, x86
alias arches arch
...
```

Vereinfachtes Handling über Kürzel für einzelne Benutzer und Gruppen.

Patch dem Repository hinzufügen

Patch anwenden

```
% git am ~/Downloads/*.eml
Applying: pinctrl: at91-pio4: add pinctrl driver
Applying: atmel: bring in at91 pio4 device tree file and bindings
```

Repository danach

```
% git log --oneline
ea680a4 (HEAD -> atmel) atmel: bring in at91 pio4 device tree file and bindings
902a9a5 pinctrl: at91-pio4: add pinctrl driver
fccdb28 (origin/master, avr32/master, atmel/master, avr32) Merge branch 'master' of git://git.
denx.de/u-boot-video
87ef765 Merge branch 'master' of http://git.denx.de/u-boot-sunxi
3d47b2d drivers/video/am335x-fb: Properly point framebuffer behind palette
7d04517 drivers/video/am335x-fb: Add support for 16bpp format
2489a7e omap3: Reduce logic/overo SPL max image size
ad14166 sunxi: Enable LD03 at 3.3V on A13-OLinuxino board
bfb33f0 sunxi: mctl_mem_matches: Add missing memory barrier
e25b369 ARM64: zynqmp: Cleanup config file after CMD move
266c63c ARM: uniphier: revive some commands lost by Kconfig re-sync
f4c6f93 Merge branch 'master' of git://git.denx.de/u-boot-samsung
a398e7a (tag: v2016.05-rc3) Prepare v2016.05-rc3
```

Pull Request verschicken

Pull Request Mail erzeugen

```
% echo "Subject: [PULL] u-boot-atmel/master -> u-boot/master" > pull.patch
% echo "" >> pull.patch
% git request-pull origin/master atmel atmel:master >> pull.patch
```

Pull request Mail

```
Subject: [PULL] u-boot-atmel/master -> u-boot/master
```

The following changes since commit 727a25b2418031e1b94ba603e0eb61657024091b:

```
    Prepare v2016.03-rc2 (2016-02-15 21:44:30 +0000)
```

are available in the git repository at:

```
    git://git.denx.de/u-boot-atmel.git master
```

for you to fetch changes up to 2a0b9ea3067b00366455b0504056cb6728a0b865:

```
    vinco: add Maintainers file (2016-02-20 21:40:02 +0100)
```

git Commit Tags

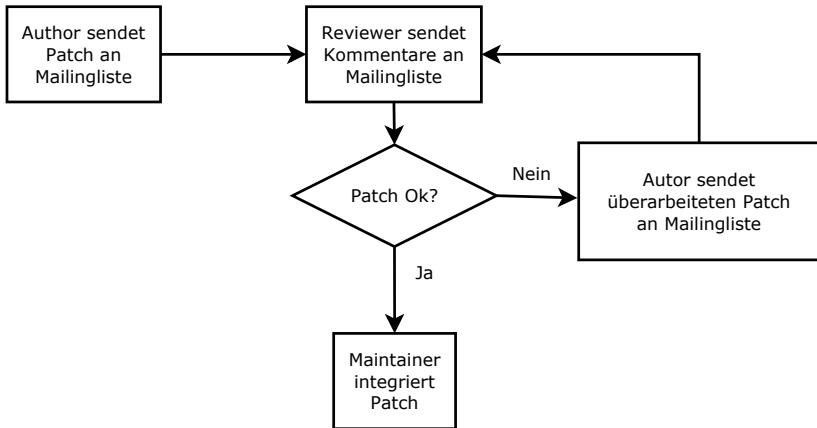
Signed-off-by: (Mit-)Autor der Änderung.

Reviewed-by: Reviewer der Änderung.

Tested-by: Tester einer Änderung.

Reported-by: Ursprünglicher Entdecker des Fehlers, der mit dieser Änderung repariert wird.

Patch Review



Konventionen für Patch Historie

Konventionen

- Subject Header bekommt eine Version [PATCH v2].
- Nach dem Mail-Trenner folgt ein Changelog.
- Mail In-Reply-To senden.

Patch Changelog

```
Signed-off-by: Eddy Petrisor <eddy.petrisor@nxp.com>
```

```
---
```

```
Changes in v2:
```

- Added support for device model. Compatibility with non-DM code is kept for easier synchronization with the code on the vendor branch where the conversion to DM is not done for all boards.
- remove TODO-s
- remove '#if 0'-ed code
- switched to SPDX headers
- removed uselss guard around 'DECLARE_GLOBAL_DATA_PTR;'
- replaced 'CONFIG_SYS_FSL' with 'S32V234' in defines for MMU bases and sizes

```
...
```

Inhalt

- 1 Einführung
 - Modelle der Zusammenarbeit
- 2 git: Grundlagen
 - Wichtige Kommandos
 - Globale Konfiguration
 - Mein U-Boot Repository
- 3 git: Erweiterte Funktionen
 - Rebase des Repositories
 - Arbeiten mit Patches
 - Reviewprozess
- 4 Patman
 - Was ist Patman
 - Wie funktioniert Patman

Patman: Eine Hilfestellung

- Anfang 2012 in U-Boot eingebaut.
- Von Simon Glass (Chromium Entwickler) geschrieben.
- Generiert Patches aus einem Branch.
- Wandelt spezielle Tags in Metainformationen um.
 - Cover Letter
 - Patch Historie
 - ...
- Prüft die Patches vor dem Versenden.
- Quelle: <U-Boot>/tools/patman [1]

Neue Tags für Patman

Series-to Sende die Patch Serie an weitere Empfänger.

Series-cc Sende die Patch Serie in Cc an weitere Empfänger.

Series-version Die Version dieser Serie.

Series-prefix Prefix wie z.B. RFC.

Series-notes Fügt weitere Kommentare zu dem Cover Letter hinzu.

Commit-notes Fügt dem Patch Kommentare hinzu, die aber nicht in der Commit Nachricht stehen.

Series-changes: n Fügt den Changelog Eintrag für eine bestimmte Version an einen Patch an.

Cover-letter Sendet eine Zusammenfassung vor der Patch Serie.

Wie funktioniert Patman

Beispiel Commit

```
commit 8a6f896d2a62289dd4f44942788eb7f6cc354d94 (HEAD -> fix_mkimage)
Author: Andreas Bießmann <andreas.devel@googlemail.com>
Date: Sat Apr 30 08:42:24 2016 +0200
```

```
mkimage: fix argument parsing on BSD systems
```

```
The getopt(3) optstring '-' is an GNU extension which is not available on BSD
systems like OS X.
```

```
Remove this dependency by implementing the feature in another way.
```

```
Signed-off-by: Andreas Bießmann <andreas.devel@googlemail.com>
Series-cc: sjg
```

Wie funktioniert Patman

Patman in Aktion

Senden des Patches

```
./tools/patman/patman -n --no-tags -c 1
Cleaned 1 patches
0 errors, 0 warnings, 2 checks for 0001-mkimage-fix-argument-parsing-on-BSD-systems.patch:
check: tools/mkimage.c,257: Alignment should match open parenthesis
check: tools/mkimage.c,263: Alignment should match open parenthesis

checkpatch.pl found 0 error(s), 0 warning(s), 2 checks(s)
Not sending emails due to errors/warnings
Dry run, so not doing much. But I would do this:

Send a total of 1 patch with no cover letter.
0001-mkimage-fix-argument-parsing-on-BSD-systems.patch
Cc: "Andreas Bießmann" <andreas.devel@googlemail.com>
Cc: William Cohen <wcohen@redhat.com>
Cc: u-boot@lists.denx.de
Cc: Marek Vasut <marex@denx.de>
Cc: Joe Hershberger <joe.hershberger@ni.com>
Cc: Philippe De Swert <philppedeswert@gmail.com>

Cc: Simon Glass <sjg@chromium.org>
Version: None
Prefix: None
Email would not be sent
```

Ende

Vielen Dank für Ihre Aufmerksamkeit ...

Bei Fragen ... gerne Mail an:

andreas@biessmann.de

Vortrag online unter:

<http://andreas.biessmann.org>

Quellen



Simon Glass. *Patman README*. 2016. URL:
<http://git.denx.de/?p=u-boot.git;a=blob;f=tools/patman/README>.



Ben Straub Scott Chacon. *Pro Git Book: Working with Remotes*. 2016. URL:
<https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes>.