

Per GIT am LoxBerry Core entwickeln

Dieser Text wurde für LoxBerry V0.2.x entwickelt. Mit LoxBerry V1.0 wurde LoxBerry Update eingeführt. Mit LoxBerry Update kann das System - je nach Einstellung auch automatisch - aktualisiert werden. Das kann eure Änderungen ohne Vorwarnung überschreiben, wenn ihr direkt im System entwickelt! Der Autor dieses Artikels entwickelt daher aktuell nicht mehr auf diese Art, sondern legt sich ein separates Development-Verzeichnis direkt auf dem LoxBerry an, aus dem heraus er Änderungen committet.

Wer am LoxBerry Core entwickeln möchte, macht dieses am einfachsten direkt auf einem Entwicklungs-RaspberryPi und veröffentlicht seine gemachten Änderungen per GIT. Das hat den Vorteil, dass das Entwicklungssystem immer auf dem gleichen Stand wie das GIT Repository ist und man mit seinem System zudem auch auf dem identischen Stand aller anderen Entwickler ist.

Die nachfolgende Anleitung beschreibt die Schritte, die dazu notwendig sind.



Mittlerweile werden Datei-Berechtigungen und Dateibesitzer vom System bei einigen Aktionen gesetzt (z. B. Update, Plugin-Installation). Verantwortlich dafür ist das Skript "`~/sbin/resetpermissions.sh`". Sobald dieses Skript ausgeführt wird, werden zahlreiche Dateien von GIT als "geändert" angesehen. Ändert ein anderer Entwickler nun etwas an diesen Dateien, verweigert GIT jegliche Aktionen wie push oder pull!

Daher ist es sinnvoll, vor Änderungen an Dateien jeweils das aktuelle Repository zu klonen und die Berechtigungen nicht neu zu setzen!

1. Fork generieren

Ist man kein LoxBerry Hauptentwickler und hat somit Schreibzugriff auf das [Original-Repository](#), legt man als aller erstes einen "Fork" des Original-Repositories an. Vereinfacht gesagt kopiert man sich damit das Original-Repository in den eigenen GIT-Account und kann dort dann sämtliche Änderungen hineinschreiben ("committen"). Ist man mit seinen Änderungen fertig, macht man einen sogenannten Pull-Request, was soviel bedeutet wie eine Anfrage an die Hauptentwickler, den eigenen Code zu übernehmen.

Wer zu den Stammentwicklern gehört kann diesen Schritt überspringen.

2. Image installieren

Jetzt installiert man das letzte Entwicklungsbild, welches verfügbar ist auf einer freien SD-Karte. Das Entwicklungsbild findet man auf den Downloadservern im Unterverzeichnis `/testing/images`. Es macht Sinn den Setup-Assistenten nach dem ersten Booten **nicht** zu durchlaufen und zu überspringen.

3. GIT-Sourcen aktualisieren

Meist enthält das Image nicht die allerletzten Änderungen des GIT-Repositories. Daher müssen zunächst die im Image enthaltenen Dateien aktualisiert werden. Dazu verbindet man sich per SSH mit dem LoxBerry, wechselt zum user "root" und führt folgende Befehle aus:

```
KONSOLE cd /opt
```

```
KONSOLE rm -r /opt/loxberry
```

```
KONSOLE git clone https://github.com/mschlenstedt/Loxberry.git -b  
loxberry-0.3.0-saruman loxberry
```

```
KONSOLE /opt/loxberry/sbin/resetpermissions.sh
```

Die oben genannte GIT-Adresse funktioniert nur, wenn man zu den Core-Entwicklern gehört und direkt in das Original-Repository schreiben darf. Ansonsten gibt man hier die GIT-Adresse zum eigenen Fork im eigenen GIT-Account an. Zudem muss natürlich der entsprechend gerade aktuelle Branch (Parameter -b) angegeben werden.

3. Änderungen committen

Nun muss man noch GIT den eigenen Namen und die eigene Emailadresse bekannt geben:

```
KONSOLE git config --global user.email "DEINE EMAIL ADRESSE"
```

```
KONSOLE git config --global user.name "DEIN NAME"
```

Ab jetzt kann man Änderungen an den Quelltexten vornehmen. Alle Änderungen werden dabei lokal von GIT "getrackt". Geänderte Dateien kann man sich zu jeder Zeit anzeigen lassen:

```
KONSOLE git status
```

Alle geänderten Dateien, die man nun veröffentlichen möchte (committen), fügt man nun mit folgendem Befehl dem Commit hinzu:

```
KONSOLE git add DATEINAME
```

Oft wird in GIT-Tutorials empfohlen, einfach alle geänderten Dateien mittels `git add .` zu committen. **Das sollte man nie tun:** Sehr häufig passiert es dabei, dass man so aus Versehen Konfigurationsdateien oder sonstige private Daten aus Versehen ins GIT Repository committet. Einmal veröffentlicht sind sie so gut wie nicht mehr löscher! Zudem würden eigene Konfigurationsdateien das Repository unbrauchbar machen. Wenn man sich unsicher ist, was man dem Commit hinzugefügt hat, hilft ein `git status` weiter.

Ist man zufrieden erzeugt man den Commit, gibt einen ausführlichen Text der gemachten Änderungen ein (in Englisch) und lädt die Änderungen mit einem "Push" zu GIT hoch:

```
KONSOLE git commit (Es öffnet sich der Standardeditor, meist nano, zur Eingabe einer
```

ausführlichen Beschreibung)

oder

```
KONSOLE git commit -m "A short Description"
```

```
KONSOLE git push
```

Um von anderen Entwicklern gemachte Änderungen von GIT zu holen, führt man einen Pull aus:

```
KONSOLE git pull
```

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/entwickler/advanced_developers/per_git_am_loxberry_core_entwickeln

Last update: **2022/10/07 06:44**