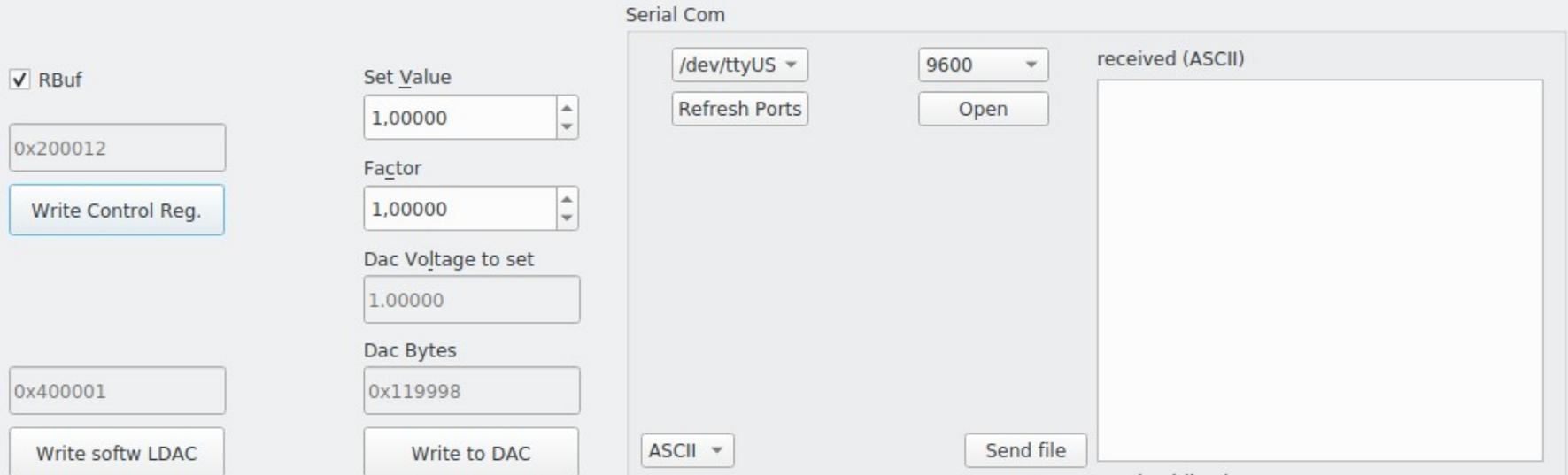


Plattformübergreifende Benutzeroberflächen mit Python und Qt

Und die Geschichte der Programmversionen archivieren mit git
Einführung anhand eines Beispiels

Institut für Prozessdatenverarbeitung und Elektronik (IPE)



The screenshot shows a Qt-based GUI for a DAC control application. The interface is divided into several sections:

- Control Panel (Left):** Contains a checked checkbox for "RBuf", a text input field with "0x200012", a "Write Control Reg." button, another text input field with "0x400001", and a "Write softw LDAC" button.
- Parameter Settings (Middle-Left):** Includes "Set Value" (1,00000), "Factor" (1,00000), "Dac Voltage to set" (1,00000), and "Dac Bytes" (0x119998), each with a corresponding input field and a "Write to DAC" button.
- Serial Com (Right):** Features a dropdown menu for the serial port (set to "/dev/ttyUS"), a baud rate dropdown (set to "9600"), a "Refresh Ports" button, an "Open" button, a "received (ASCII)" text area, an "ASCII" dropdown menu, and a "Send file" button.

Motivation Python und Qt



- Python und Qt: **Einmal Programmieren, in verschiedenen Betriebssystemen ausführen** (ohne Neukompilierung)
- Gestalten von grafischen Oberflächen mit **QtDesigner**; Implementieren der **Funktionen in Python**
- Python: **große Verbreitung, auf vielen Plattformen, viele Pakete, freie Lizenz**
- **kommerzielle** und nicht-kommerzielle **Verwendung erlaubt**. Python (Python-Software-Foundation-Lizenz / BSD-ähnlich); **PySide2** und Qt (**GPL**)

(https://de.wikipedia.org/wiki/Python_Software_Foundation#Python-Software-Foundation-Lizenz

[https://de.wikipedia.org/wiki/Qt_\(Bibliothek\)#Andere_Anbindungen](https://de.wikipedia.org/wiki/Qt_(Bibliothek)#Andere_Anbindungen))

- Buch: **Rapid GUI Programming with Python and Qt**; Mark Summerfield (aktualisierte Codebeispiele auf der Internetseite des Autors) (PyQt / GPL → PySide2 / LGPL)



Grafische Benutzeroberfläche (GUI) mit Python und Qt: Vorteile gegenüber anderen Sprachen

■ Vorteile gegenüber Labview:

- Lizenz (auch für kommerzielle Anwendungen frei)
- alle Programmbausteine frei verfügbar

■ Vorteile gegenüber JavaScript:

- Hardware-naher Zugriff einfach möglich (Gegensatz: -> Browser-Sandkasten)
- Programmierfehler werden bei Python nicht so leicht übergangen (mit entsprechender Fehlermeldung)

■ Vorteile gegenüber Qt/C++:

- Weniger Zeilen Code (als bei C++)
- Ohne Neukompilierung ausführbar

Grafische Benutzeroberfläche (GUI) mit Python und Qt: Installation

■ Python 3

- Linux, Windows: installiere python 3
- Mac: installiere homebrew; dann: `brew install python3`

■ Python und Qt

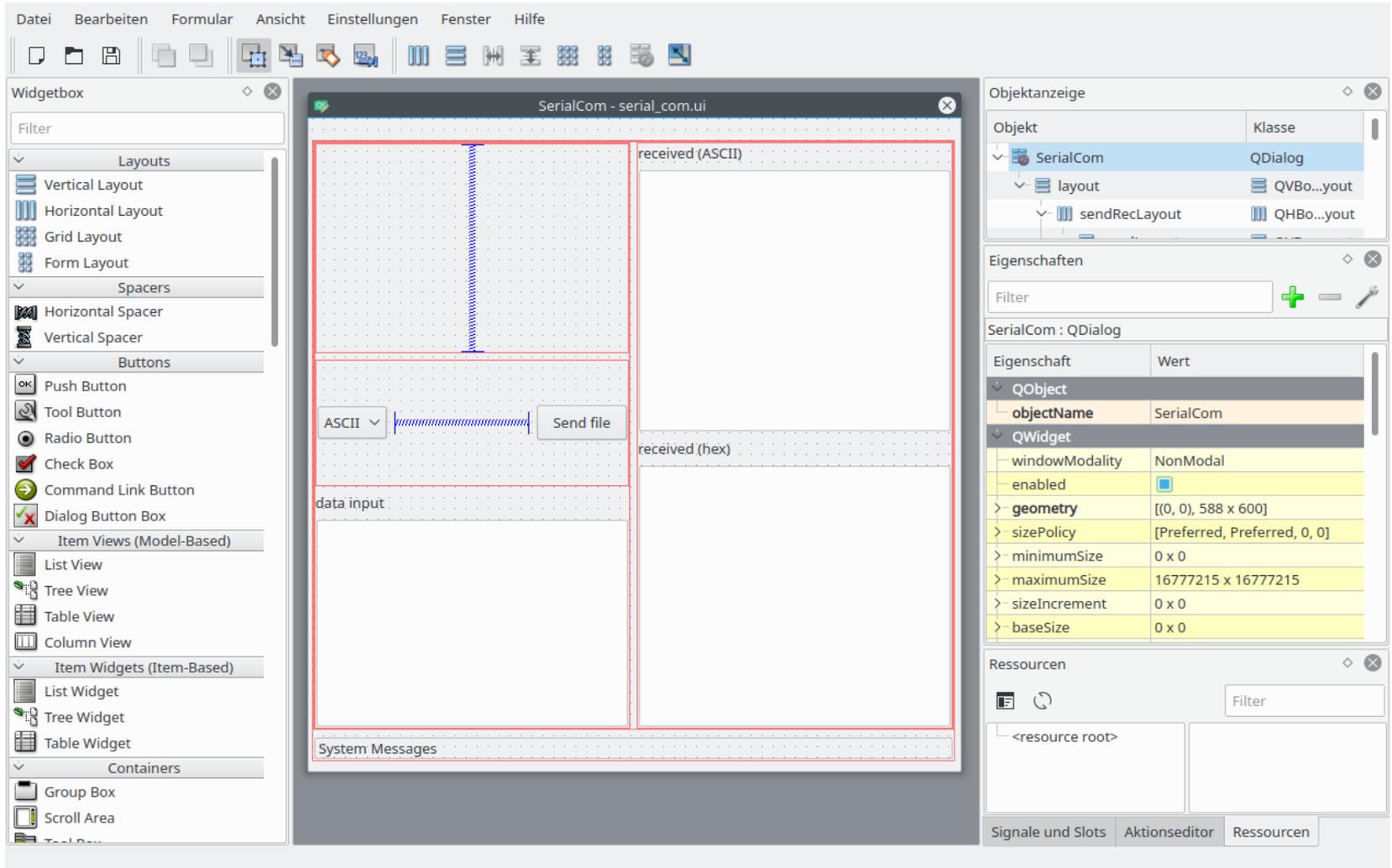
- In einem terminal eintippen: `pip3 install --user pyside2`

■ Weitere python Pakete

- Serial (RS232) access: `pip3 install --user pyserial`

■ Installiere Qt Designer

Grafische Benutzeroberfläche (GUI) mit Python und Qt: QtDesigner



The screenshot displays the Qt Designer application window titled "SerialCom - serial_com.ui". The interface is divided into several panels:

- Widgetbox:** A sidebar on the left containing various widget categories such as Layouts (Vertical, Horizontal, Grid, Form), Spacers, Buttons, Item Views, Item Widgets, and Containers.
- Design Canvas:** The central workspace showing a grid-based layout. It features a "data input" field, a "Send file" button, and two text areas labeled "received (ASCII)" and "received (hex)". A "System Messages" area is visible at the bottom.
- Objektanzeige (Object Inspector):** A panel on the right showing a hierarchical tree of the GUI's object model. The root is "SerialCom" (class QDialog), which contains a "layout" (class QVBoxLayout) and a "sendRecLayout" (class QHBoxLayout).
- Eigenschaften (Properties):** A panel below the Object Inspector showing the properties of the selected object. The selected object is "SerialCom : QDialog".
- Ressourcen (Resources):** A panel at the bottom right showing the resource files used in the GUI.

Eigenschaft	Wert
QObject	
objectName	SerialCom
QWidget	
windowModality	NonModal
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 588 x 600]
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0 x 0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0

Python und Qt: Einbindung der Oberfläche in eigenes Programm

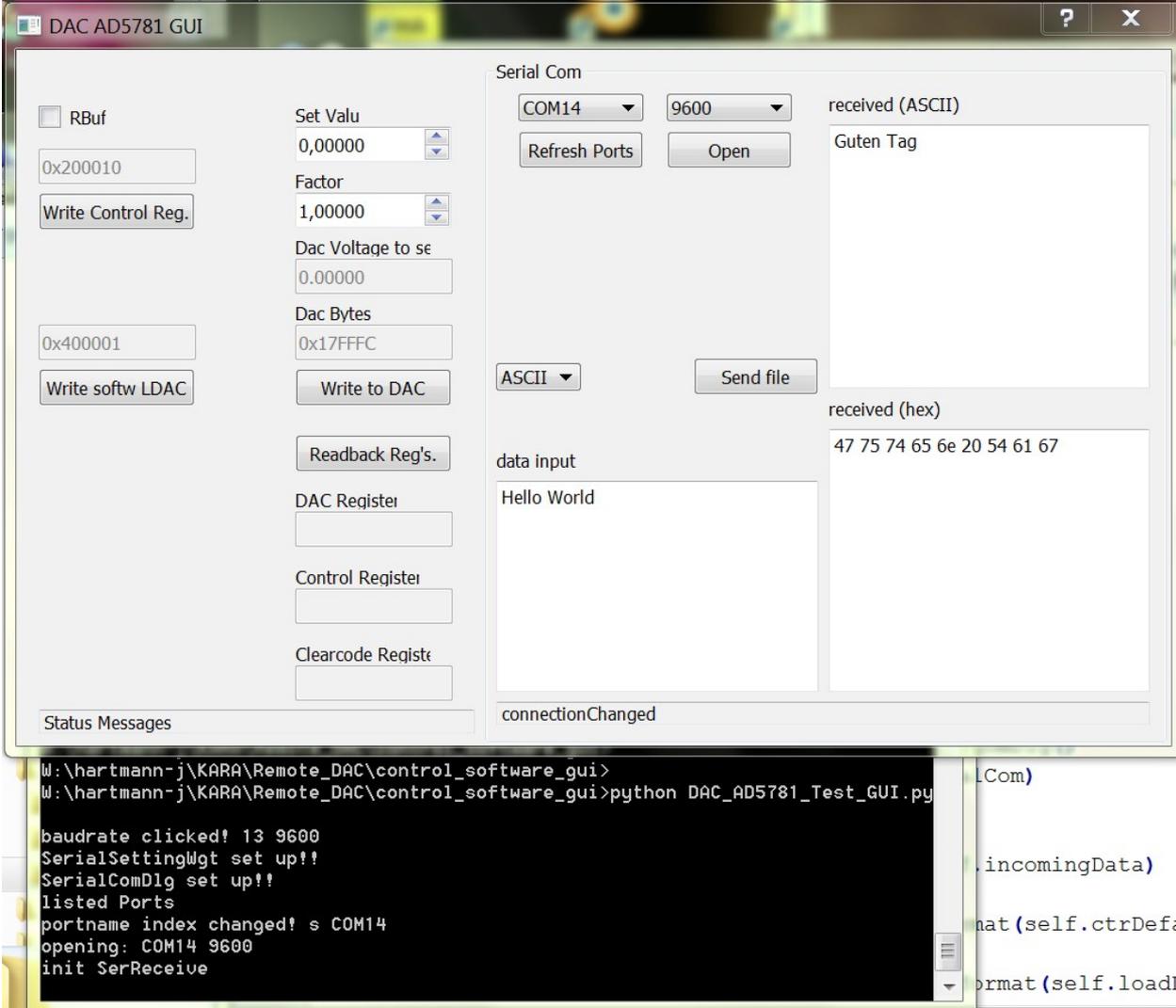
■ Konvertieren der .ui -Datei in eine .py -Datei:

- `pyside2-uic serial_settings.ui -o ui_serial_settings.py`
- (Pfad unter Linux nach `pip3 install: .local/bin/`)

■ Einbinden der .py -Datei (GUI-Schablone) in eine eigene .py -Datei (Funktionen des Programms):

```
try:  
    from .ui_serial_settings import *  
except (SystemError, ImportError):  
    from ui_serial_settings import *
```

Grafische Benutzeroberfläche (GUI) mit Python und Qt: Programm unter Windows



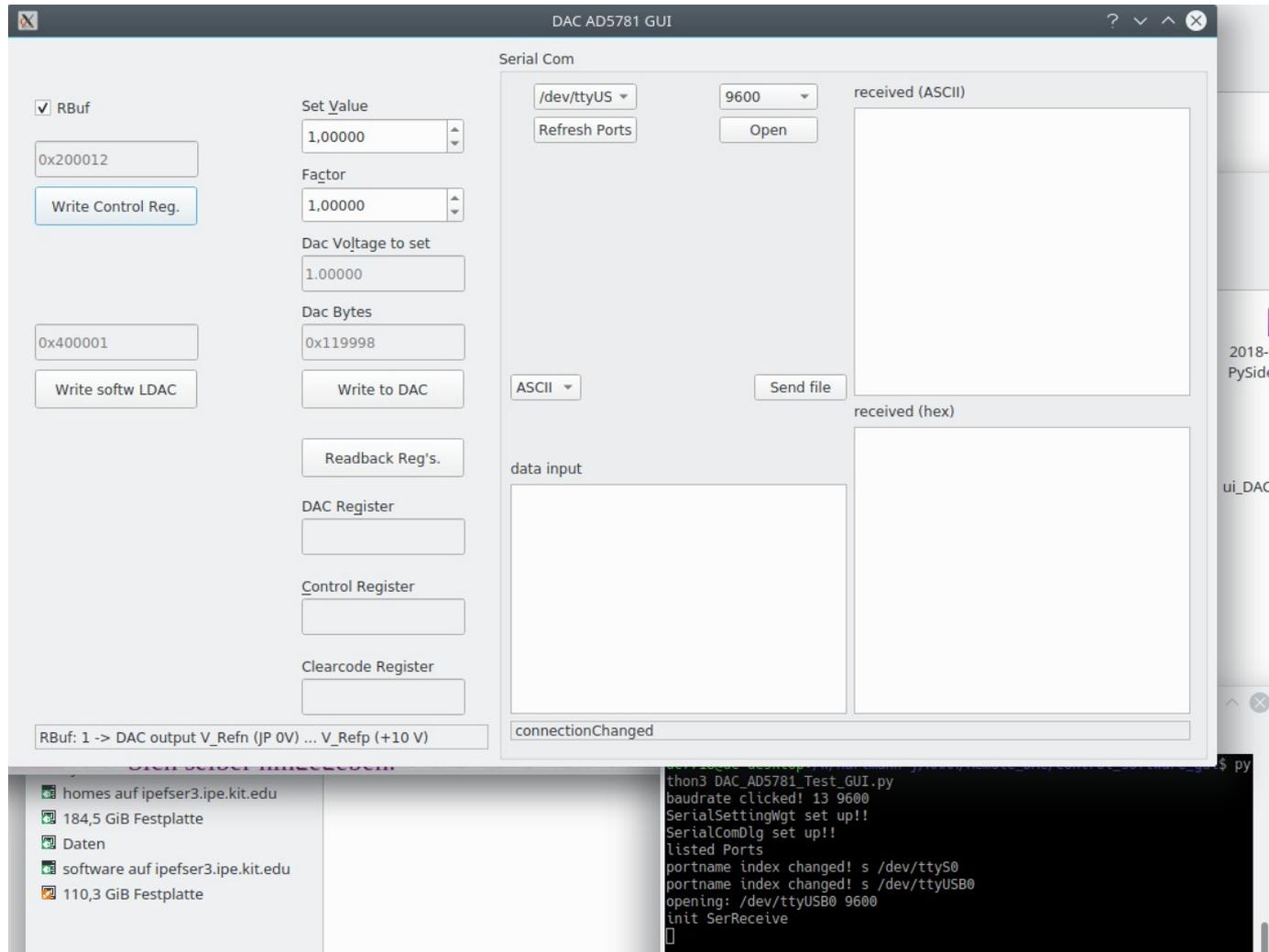
The image shows a screenshot of a Windows application titled "DAC AD5781 GUI". The application interface is divided into several sections:

- Left Panel:** Contains a checkbox for "RBuf", a text input field with "0x200010", a "Write Control Reg." button, another text input field with "0x400001", and a "Write softw LDAC" button.
- Center Panel:** Includes a "Set Value" spinner set to "0,00000", a "Factor" spinner set to "1,00000", a "Dac Voltage to se" text input with "0.00000", a "Dac Bytes" text input with "0x17FFFC", and a "Write to DAC" button. Below these are "Readback Reg's.", "DAC Register", "Control Register", and "Clearcode Regist" text input fields.
- Serial Com Panel:** Features a dropdown menu set to "COM14", a baud rate dropdown set to "9600", "Refresh Ports" and "Open" buttons, a "received (ASCII)" text area containing "Guten Tag", a dropdown menu set to "ASCII", and a "Send file" button. Below this is a "data input" text area containing "Hello World".
- Bottom Panel:** A "Status Messages" area with the text "connectionChanged".

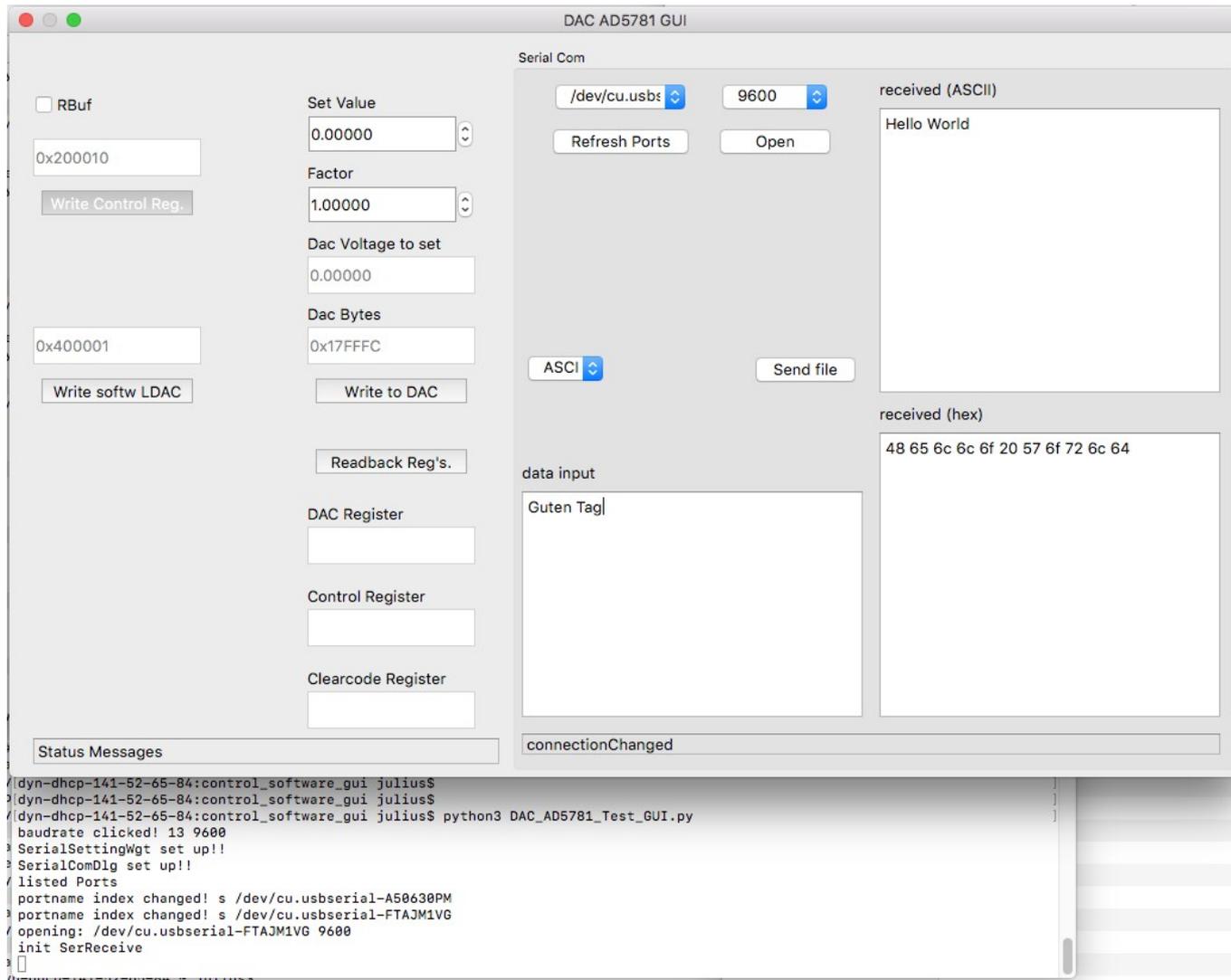
Below the GUI window, a terminal window shows the command prompt and the execution of the Python script:

```
W:\hartmann-j\KARA\Remote_DAC\control_software_gui>
W:\hartmann-j\KARA\Remote_DAC\control_software_gui>python DAC_AD5781_Test_GUI.py
baudrate clicked! 13 9600
SerialSettingWgt set up!!
SerialComDlg set up!!
listed Ports
portname index changed! s COM14
opening: COM14 9600
init SerReceive
```

Grafische Benutzeroberfläche (GUI) mit Python und Qt: Programm unter Linux



Grafische Benutzeroberfläche (GUI) mit Python und Qt: Programm unter MacOS



Motivation (Lokale) Versionsverwaltung mit git

- Etappen der Versionsgeschichte werden festgehalten
- Änderungen bleiben nachvollziehbar
- Änderungen können rückgängig gemacht werden (Vorsicht: vorher eine Kopie des gesamten Ordners anlegen!)
- Git kann sowohl in einem lokalen Ordner als auch auf einem Server gemeinsam benutzt werden

Git: Nutzungsbeispiel (Lokal)

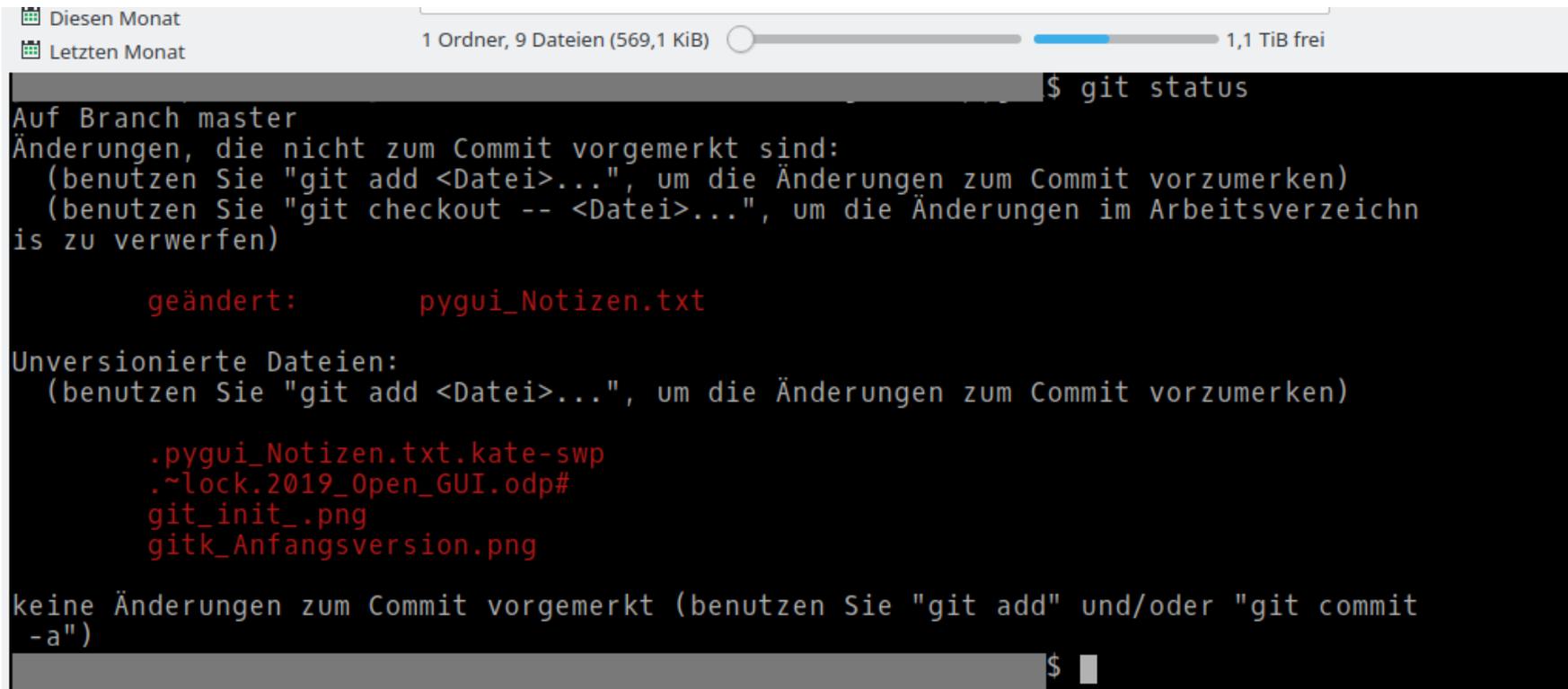
- Für ein Projekt mit Ordner und Unterordner soll die Versionsgeschichte im Laufe der Arbeit dokumentiert werden
- **git init** legt einen versteckten `.git` Ordner im aktuellen Verzeichnis an. Darin wird die Versionsgeschichte gespeichert.

> KIT > Seminarvorträge_etc > pygui

Name	Größe
> - .git	8 Elemente
- pygui_Notizen.txt	839 B
- 2019_Open_GUI.odp	19,2 KiB
- 2018-09_Serial_Test_Windows.PNG	186,5 KiB
- 2018-09_Serial_Test_Mac.png	136,7 KiB
- 2018-09_Serial_Test_Linux.png	107,8 KiB
- .pygui_Notizen.txt.kate-swp	82 B
- ~/.lock.2019_Open_GUI.odp#	79 B

Git: Nutzungsbeispiel (Lokal)

- `git add *` schaltet die Überwachung für alle aktuell im Ordner (und Unterordner) enthaltenen Dateien ein.
- `git status` zeigt den aktuellen Status des Ordners (neue Dateien...)



```
Diesen Monat
Letzten Monat
1 Ordner, 9 Dateien (569,1 KiB) 1,1 TiB frei

$ git status
Auf Branch master
Änderungen, die nicht zum Commit vorgemerkt sind:
 (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
 (benutzen Sie "git checkout -- <Datei>...", um die Änderungen im Arbeitsverzeichnis zu verwerfen)

    geändert:      pygui_Notizen.txt

Unversionierte Dateien:
 (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)

    .pygui_Notizen.txt.kate-swp
    ~lock.2019_Open_GUI.odp#
    git_init_.png
    gitk_Anfangsversion.png

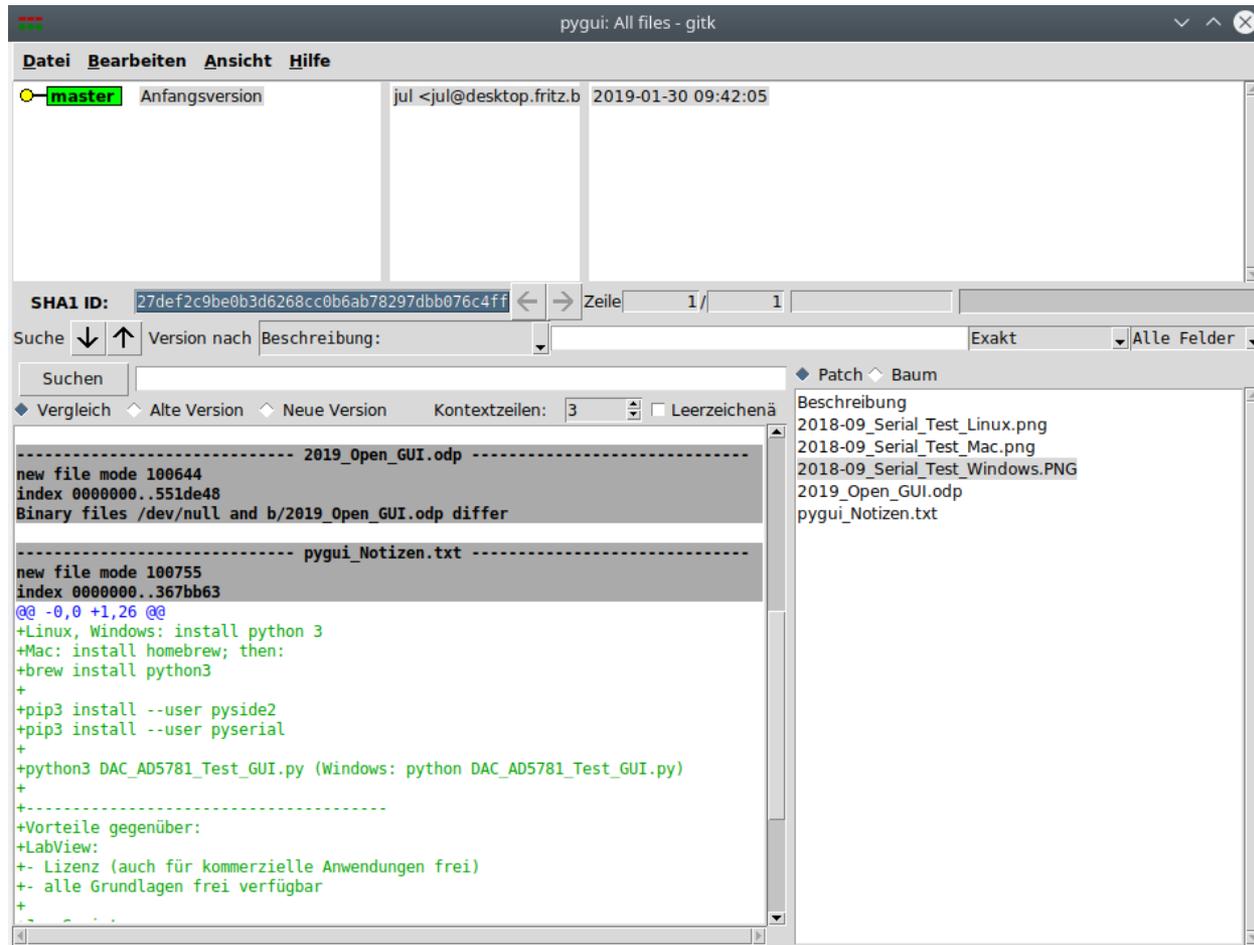
keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
$
```

Git: Nutzungsbeispiel (Lokal)

- `git diff` zeigt die Änderungen der beobachteten Dateien seit der letzten festgehaltenen Version an.
- `git commit -a -m "Versionsbeschreibung"` speichert den aktuellen Stand der beobachteten Dateien in der Versionsgeschichte.

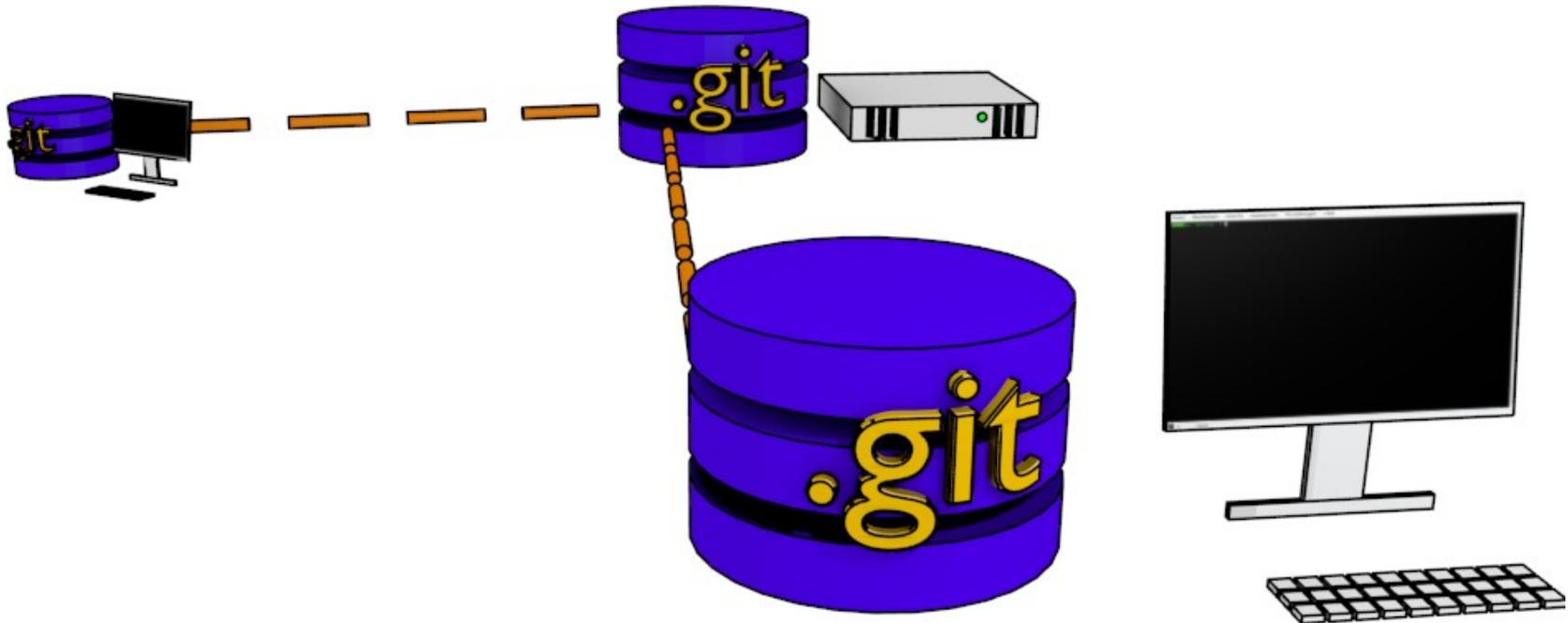
Git: Nutzungsbeispiel (Lokal)

- **gitk** startet die standard-Grafikoberfläche von git für den lokalen Ordner.



`git add *` und `git commit -a -m "jenes"`

- `git pull` und `git push` : als Gruppe mit zusätzlicher zentraler Datenbank arbeiten



Git: Nutzungsbeispiel (Lokal)

- `git add *` und `git commit -a -m "Neue Bildschirmfotos und Änderungen an den Vortragsfolien"` erzeugen eine weitere Marke in der Versionsgeschichte.

```
$ git commit -a -m "Neue Bilds
chirmfotos und Änderungen an den Vortragsfolien"
[master cace137] Neue Bildschirmfotos und Änderungen an den Vortragsfolien
  Committer: jul <jul@desktop.fritz.box>
Ihr Name und E-Mail Adresse wurden automatisch auf Basis
Ihres Benutzer- und Rechnernamens konfiguriert. Bitte prüfen Sie, dass
diese zutreffend sind. Sie können diese Meldung unterdrücken, indem Sie
diese explizit setzen. Führen Sie dazu den folgenden Befehl aus und folgen
Sie den Anweisungen in Ihrem Editor, um die Konfigurationsdatei zu bearbeiten:

  git config --global --edit

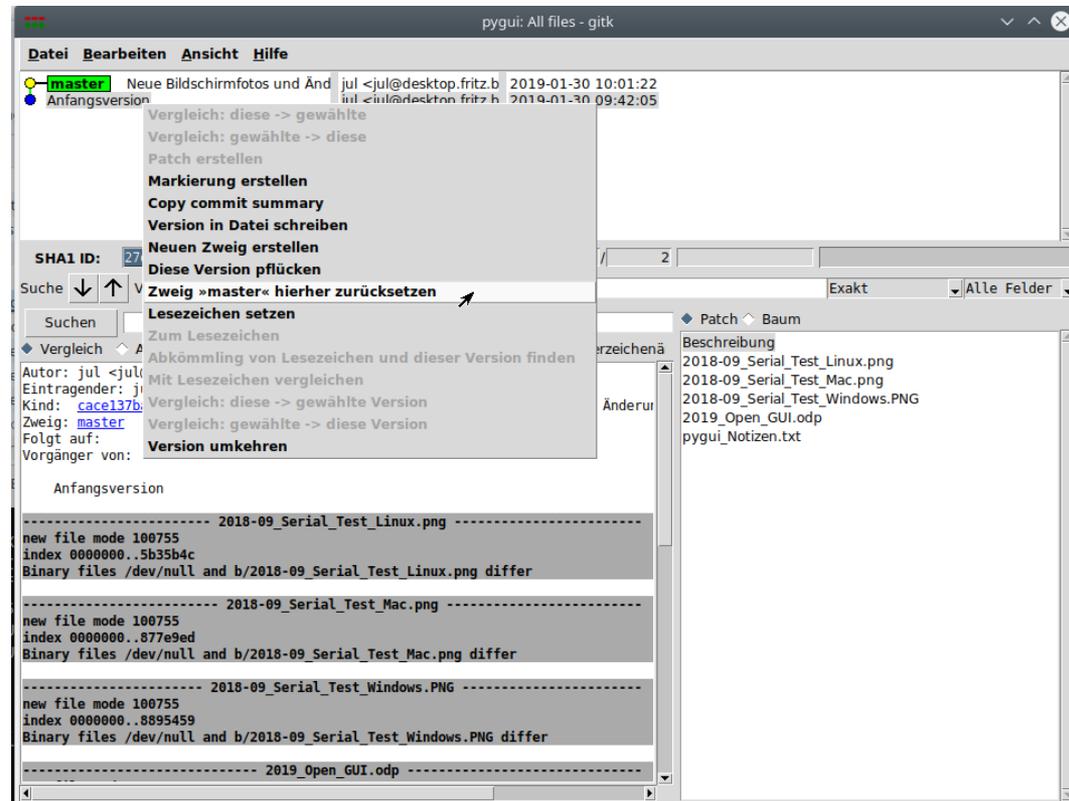
Nachdem Sie das getan haben, können Sie Ihre Identität für diesen Commit
ändern mit:

  git commit --amend --reset-author

4 files changed, 8 insertions(+)
create mode 100644 git_init.png
create mode 100644 git_status.png
create mode 100644 gitk_Anfangsversion.png
$ █
```

Git: auf einen alten Versionsstand zurücksetzen

- Von dem gesamten Ordner eine Sicherungskopie erstellen!
- In gitk: Rechtsklick auf Versionsstand und zurücksetzen. Neue Dateien und Änderungen werden gelöscht! Kann ohne eine zusätzliche Sicherungskopie in gitk nicht rückgängig gemacht werden!



GUI mit Python und Qt; Versionsverwaltung mit git

Danke für Ihr Interesse. Vortragsteil zuende.

Fragen und Diskussion



Bildquelle: Wikipedia;
Raffael