



AppBuilder & ArduVIZ
Softwareprojekt - Verteidigung



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

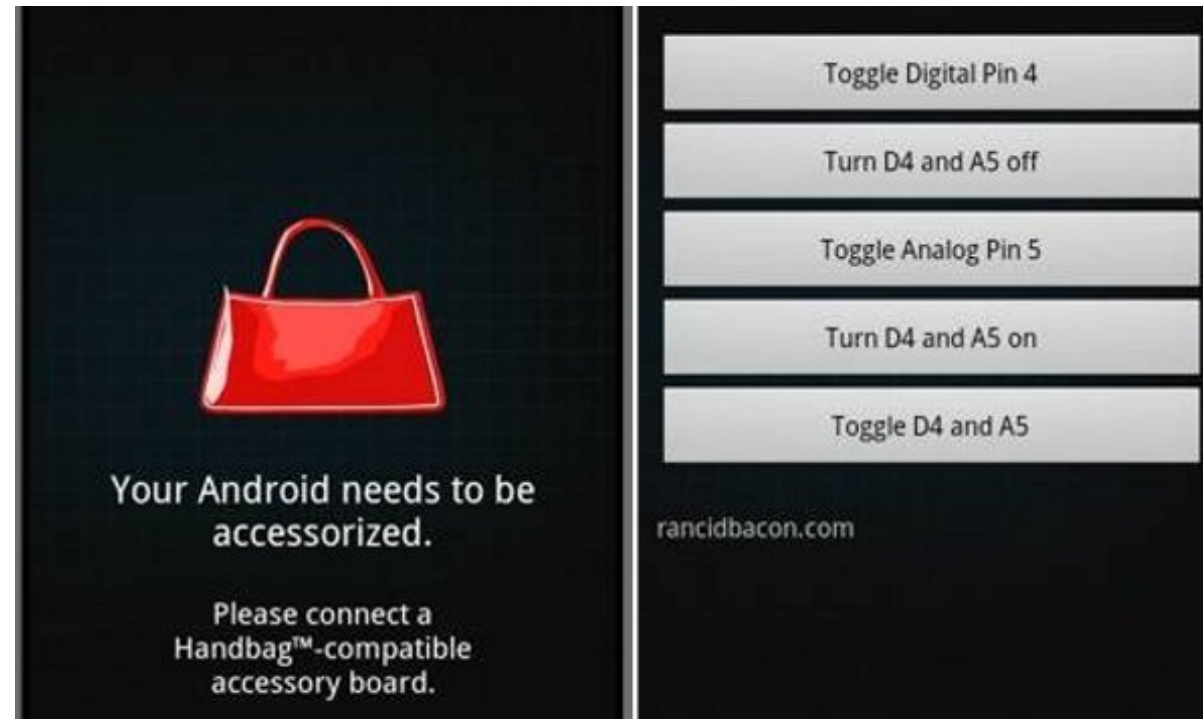
FAKULTÄT FÜR
INFORMATIK

AppBuilder & ArduVIZ
Softwareprojekt 2014 - Verteidigung
Holger Harzer & Leopold Johannes Ryll

Gliederung

1. Motivation und Zielsetzung
2. Bisherige Version
3. Aktuell Version (Änderungen, Neuerungen, Probleme)
4. Dokumentation
5. Aussichten
6. Fazit

Motivation und Zielsetzung



Motivation und Zielsetzung

Probleme mit Handbag

Instabil

Unflexibel

Unzuverlässig

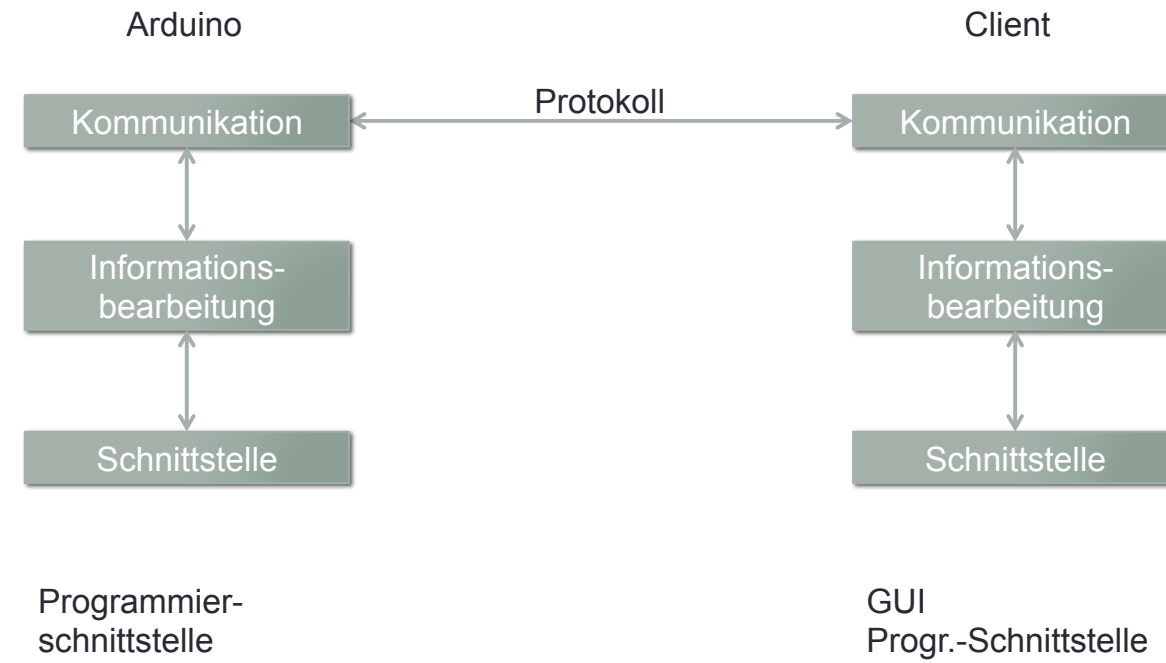
Stabile Verbindung

Universelle Kommunikation

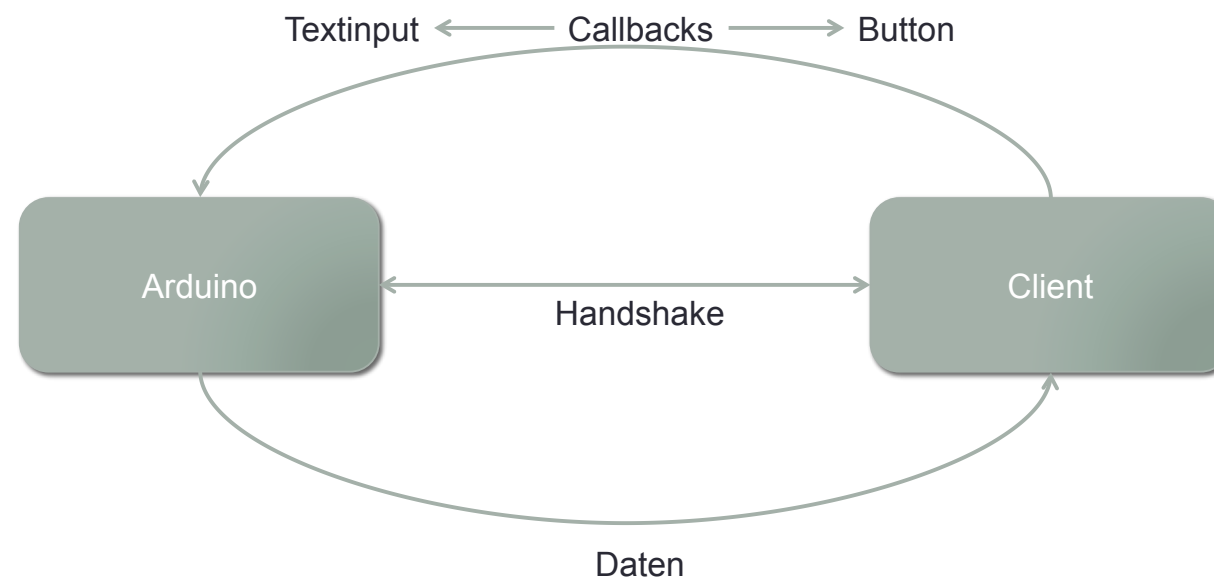
Neuinitialisierung zuverlässig

Offen für Erweiterungen

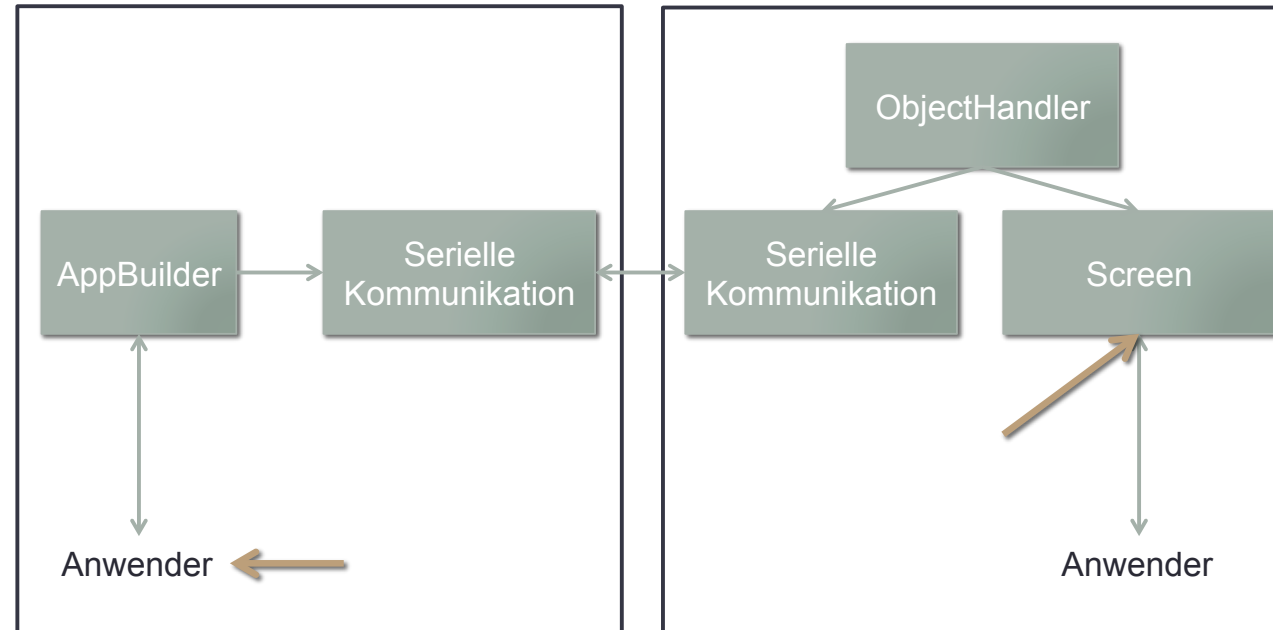
Motivation und Zielsetzung



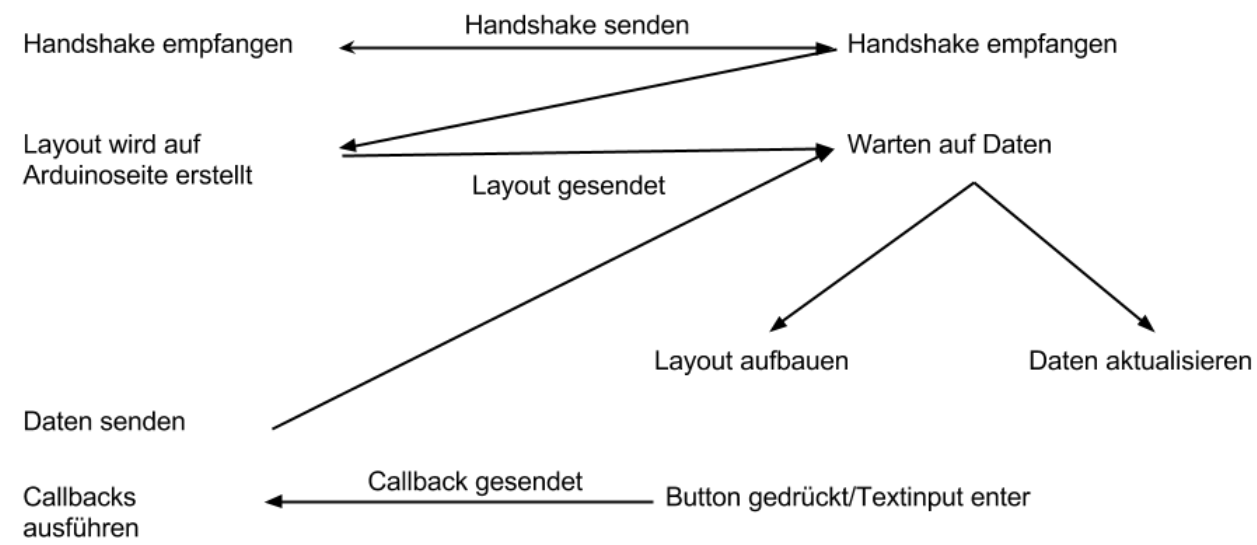
Motivation und Zielsetzung



Motivation und Zielsetzung



Bisherige Version



Bisherige Version

Public Member Functions

	AppBuilder ()
bool	sendApp () Sends App to client synchronous. More...
void	serialEvent () SerialEvent must be executed on SerialEvent of your Arduino
uid_8	start_layout ()
uid_8	end_layout (uid_8 layout)
uid_8	add_button ()
uid_8	add_label ()
uid_8	add_textinput ()
uid_8	add_diagram ()
char	get_type (uid_8 widget)
uid_8	set_text (uid_8 widget, char *text) Sets the text of a specified object. More...
uid_8	set_axes (uid_8 widget, char *x_axis, char *y_axis) Sets the identifiers of the axes of a given diagram. More...
uid_8	set_callback (uid_8 widget, app_callback cb) Sets callback for both text inputs and buttons. More...
char	send_data (char *identifier, int data) Sends data to client. More...
void	refresh () Refreshes the state of the AppBuilder object. More...

Verbindung ist einmaliger Vorgang

→ Warten auf Verbindung

→ Kein Reagieren auf Änderung

Wenig Beeinflussung von AppBuilder

- Puffer, Speicher, Verbin.Versuche

→ Mehr, bessere Anpassungen

→ Zugeschnitten auf Anwendung

Connect/Disconnect

→ Standard: Auto-Reset

→ Keine Reinitialisierung notwendig

→ Praktisch für Robby bspw.

Aktuelle Version (Änderungen)

Intuitivere Anpassbarkeit

```
AppBuilder::AppBuilder ( byte  num_com,  
                           byte  num_callbacks,  
                           String handshake,  
                           int    attempts  
                           )
```

Constructor of `AppBuilder`.

Don't forget to start the Serial connection before using `AppBuilder` (e.g. `Serial.begin(9600)`)


Parameters

<code>num_com</code>	Number of components that will be used
<code>num_callbacks</code>	Number of callbacks that will be used
<code>handshake</code>	The phrase that's used for handshakes
<code>attempts</code>	Attempts that are made to connect to the computer (not implemented yet)

- Zugeschnitten auf Anwendungsfall
- Geringerer Speicherbedarf
- Schnelle Übersicht & ggfs. Verständnis

Aktuelle Version (Änderungen)

On Connect/Disconnect



```
void AppBuilder::set_onconnect ( gen_callback onc )
```

Sets the function that's called when you're successfully connection.

This includes performing the handshake. Nothing will happen if the AppBuilder::handshake wasn't successful

Parameters

onc The function that will be called on connection. typedef ::gen_callback

```
void AppBuilder::set_ondisconnect ( gen_callback ondc )
```

Sets the function that's called when the connection is lost.

Parameters

ondc The function that will be called on disconnection

- Arbeiten vor dem Verbinden
- Reaktion auf Änderung der Verbindung
- Initialisierung/Deinitialisierung möglich
- System kann immer wieder verbunden/getrennt werden
- Reaktion auf Änderungen besser möglich

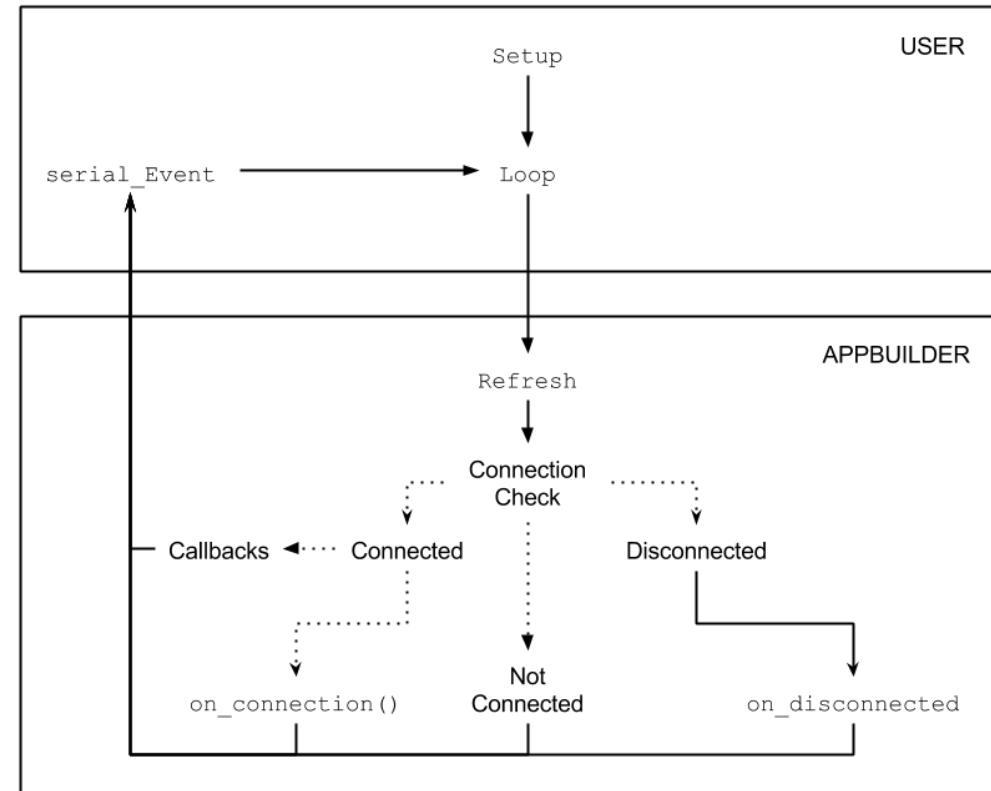
Aktuelle Version (Änderungen)

Integration der Schnittstelle in AppBuilder

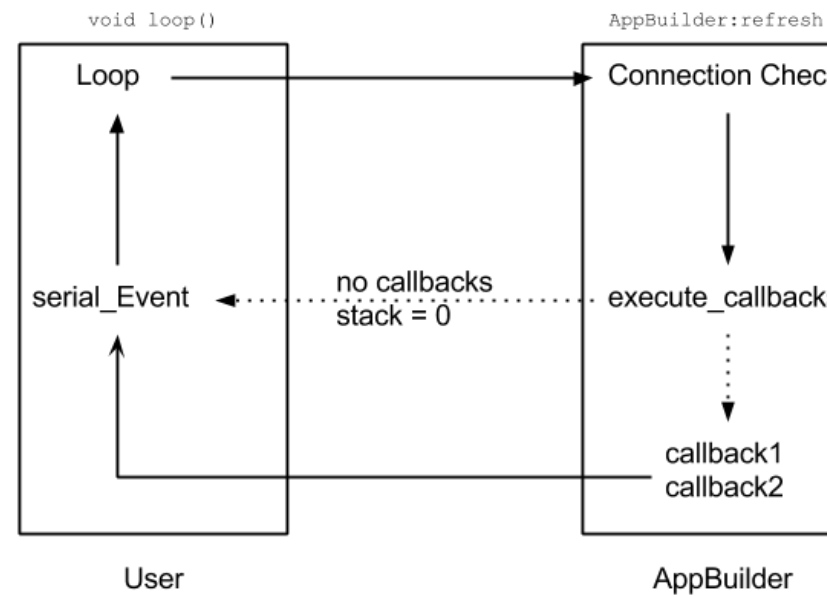


- Geringerer Speicherbedarf
- Einfachere Anpassung durch Benutzer
- Schneller, weniger Overhead
- Weniger Kapselung

Aktuelle Version (Ablauf)



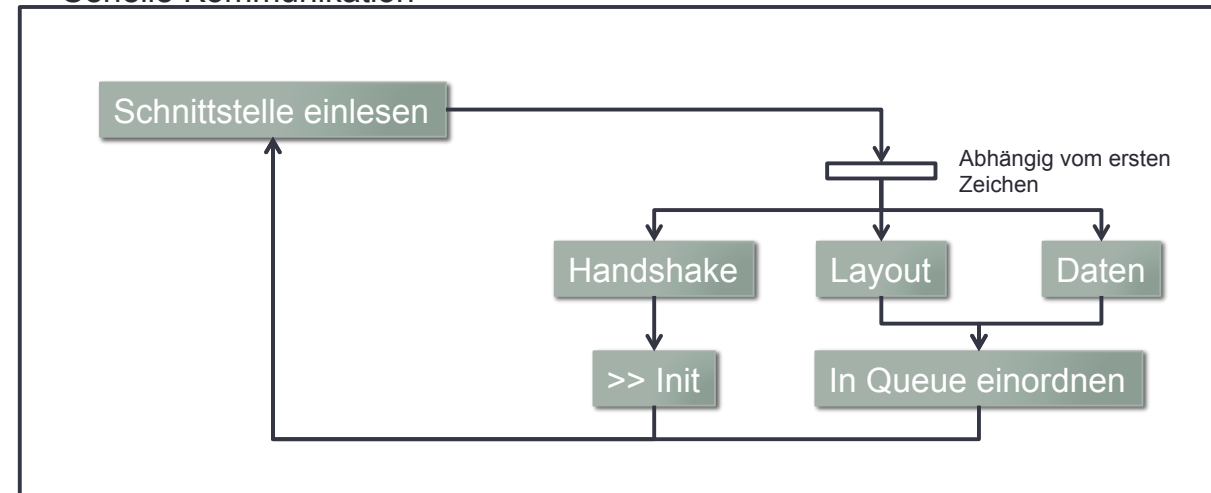
Aktuelle Version (Callbacks)



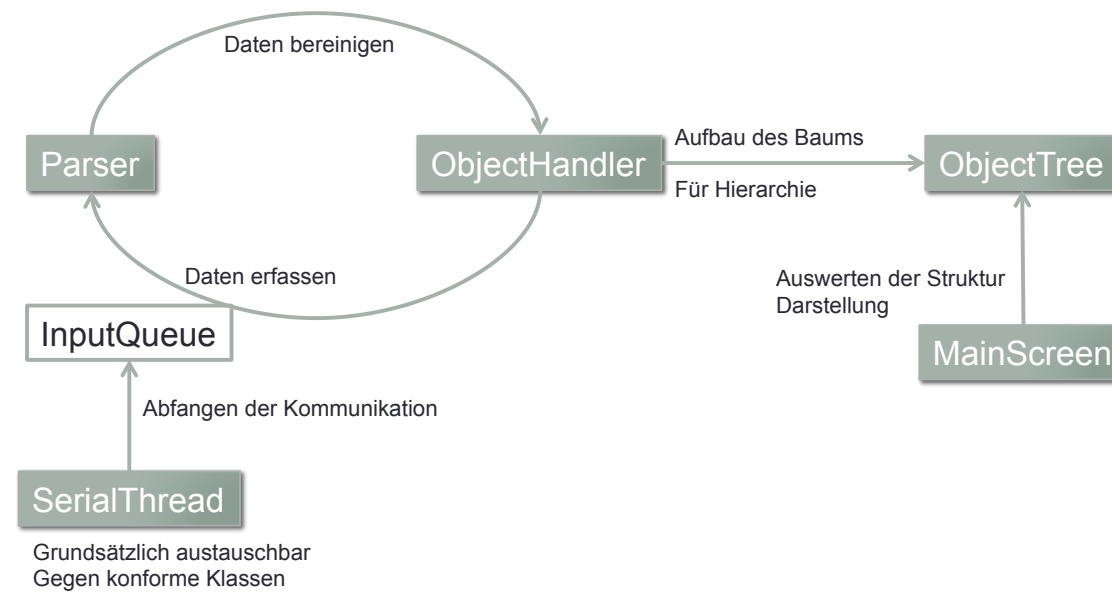
- Neue Callbacks erst nach abarbeiten von loop
- Callbacks werden nur bei bestehender Verbindung abgearbeitet

Aktuelle Version (Client)

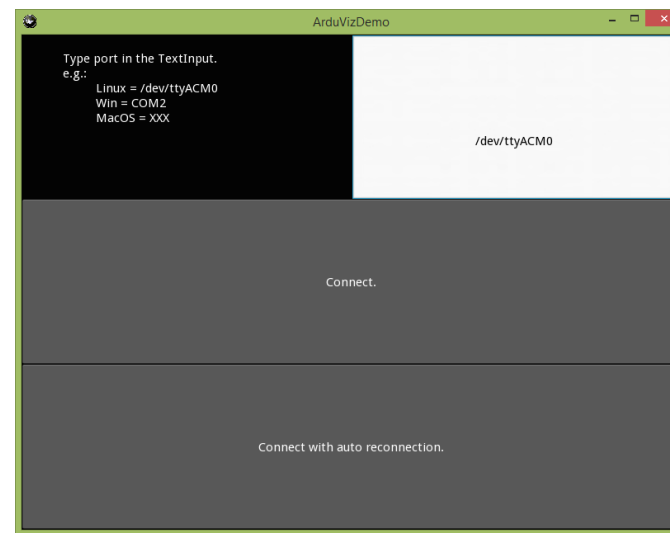
Serielle Kommunikation



Aktuelle Version (Client)



Aktuelle Version (Client)



- Threadsicherheit
- Zwei Connection-Modi
- Menü überarbeitet
- Auto-Port-Suche weggefallen

Aktuelle Version (Client)

Leichter erweiterbar

Zusammenfassung der Kivy-Widgets in Modul

```
class MyLayout(BoxLayout):
    def __init__(self, uid):
        super(MyLayout, self).__init__(orientation='vertical')
        self.uid = uid

    ##
    # Widget for Label
    #
    class MyLabel(Label):
        def __init__(self, uid, text="DEFAULT"):
            super(MyLabel, self).__init__(text=text)
            self.uid = uid

    ##
    # Widget for Button
    #
    class MyButton(Button):
        def __init__(self, uid, serialcom, text="DEFAULT"):
            super(MyButton, self).__init__(text=text)
            self.uid = uid
            self.bind(on_press = partial(serialcom.sendCB, self.uid))

    ##
    # Widget for TextInput
    #
    class MyInput(TextInput):
        def __init__(self, uid, serialcom, text="Default Eingabe"):
            super(MyInput, self).__init__(text=text, multiline=False)
            self.uid = uid
            self.bind(on_text_validate = partial(serialcom.sendTxt, self.text))

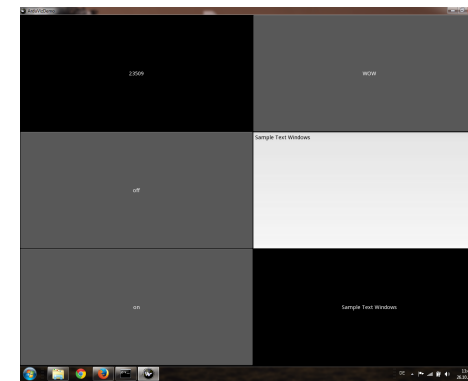
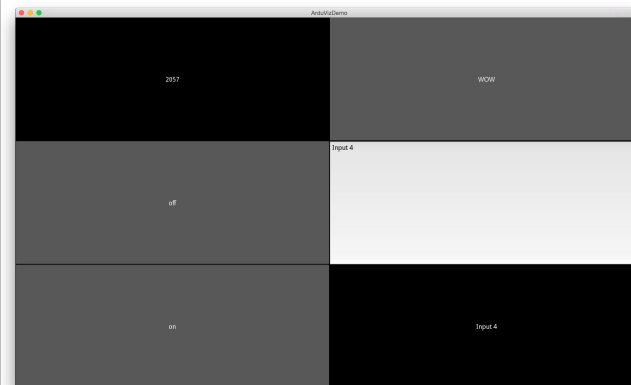
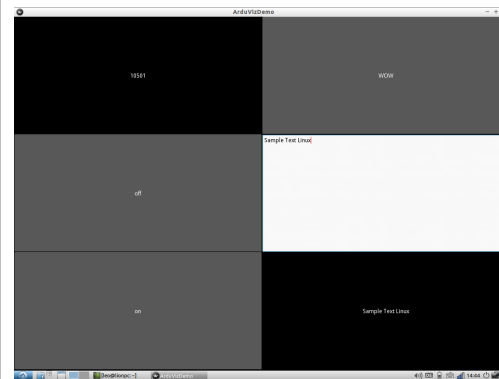
    ##
    # Widget for the port selection in the main menu.
    #
    class PortSelection(BoxLayout):
        def __init__(self, uid):
            super(PortSelection, self).__init__(orientation='horizontal')
            self.uid = uid
            self.Text_PortInfo = TextInput(text='Type port in the TextInput.\n e.g.: \\n \\t Linux = /dev/ttyACM0')
            self.Text_PortInput = TextInput(text='/dev/ttyACM0', padding=[150, 120, 150, 20])
            self.add_widget(self.Text_PortInfo)
            self.add_widget(self.Text_PortInput)
```

Ableitung aller
Widgets von
Kivy-Widgets

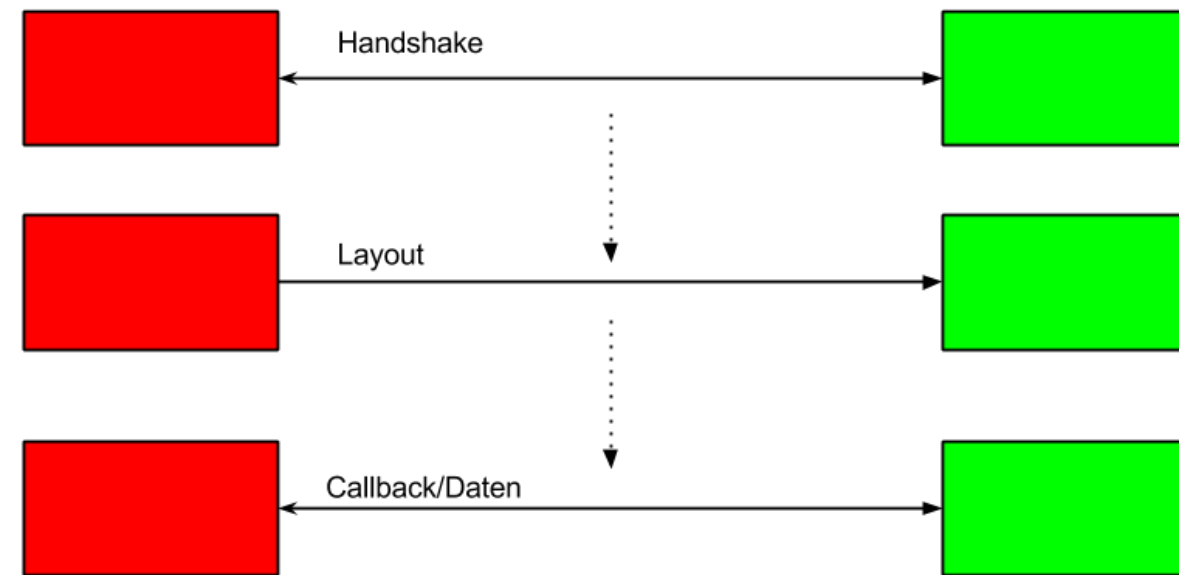
Kivy bietet gute Grundlage für Erweiterungen

Aktuelle Version (Portabilität)

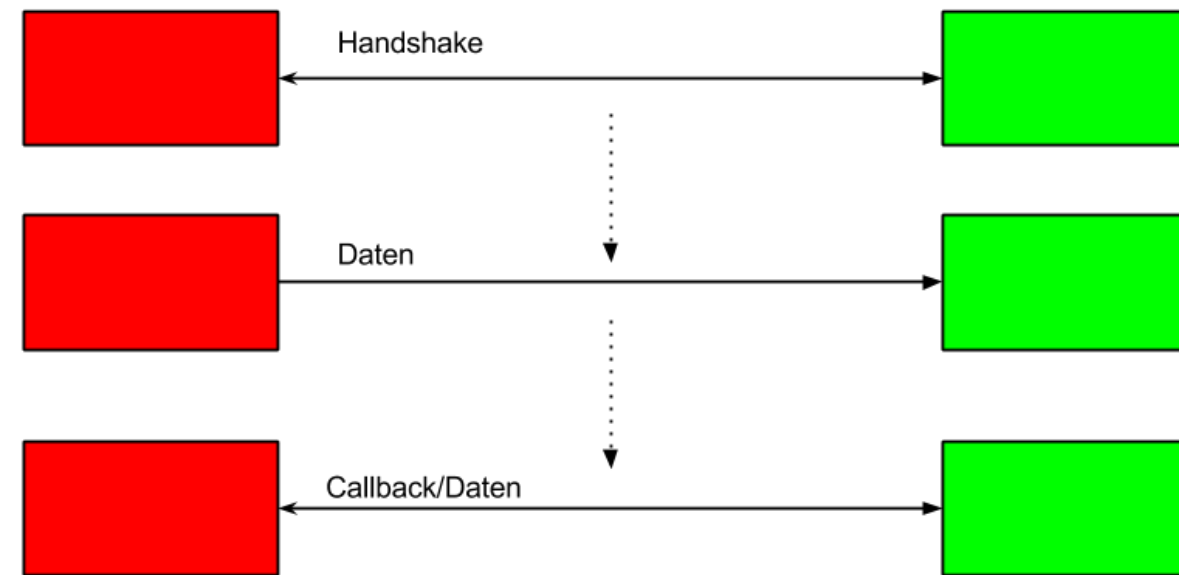
Lauffähig auf Windows, Mac OS und Linux



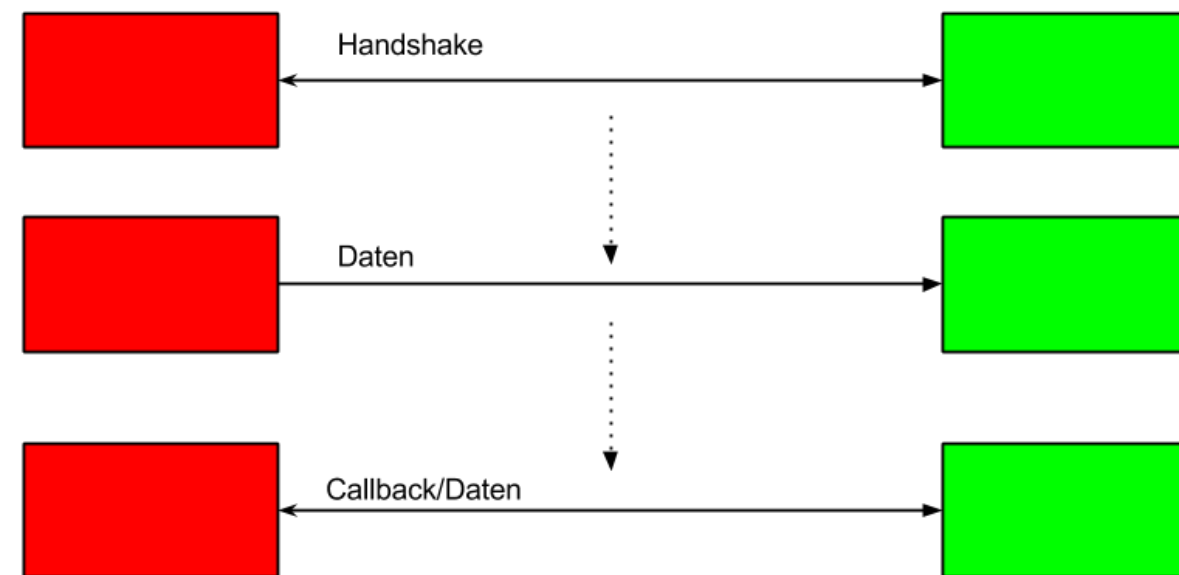
Aussichten (Verallgemeinerungen)



Aussichten (Verallgemeinerungen)



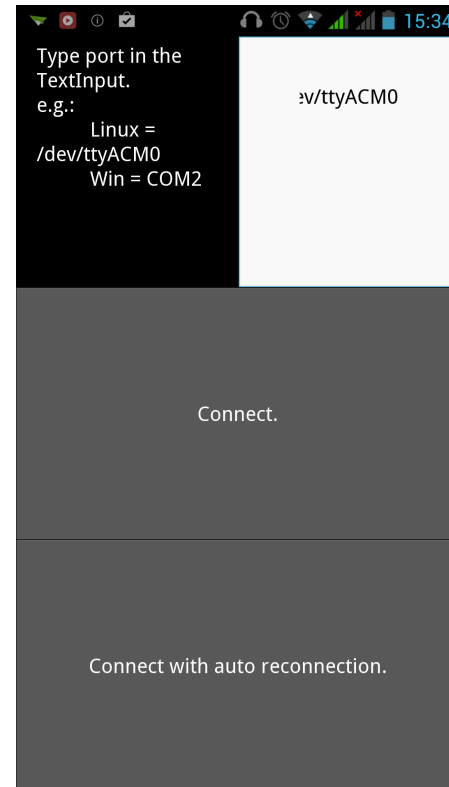
Aussichten (Verallgemeinerungen)



Universelle Verbindung mit Initialisierung

→ Nicht auf Layout per se beschränkt (Arduino-Arduino-Verbindung)

Aussichten (Android)



Kivy bereits lauffähig
Herausforderung mit serieller
Kommunikation

Erweiterung um Diagramme, Bilder, etc.

Dokumentation

AppBuilder

2.0rc1

Main Page
Classes
Files

▶ AppBuilder

AppBuilder Documentation

AppBuilder is a library that's designed to communicate with a compatible

Currently supported are the following functions:

- Connection detection (disconnection detection isn't currently supported)
- Handshake to ensure that you're connected to a compatible and trusted device
- Building layouts and sending them (You can nested layouts etc. The library will handle the rest)
- Setting, receiving and executing callback for both buttons and textinput
- Setting text

Allgemein:

- Installation & Benutzung des Clients
- Benutzung der Library
- Hinweise zu Stolperfallen und Abläufen

Aktuelle Version

- Integration von AppBuilder und Communication (Weniger Abhängigkeiten, kürzere Wege, effizientere Nutzung von Buffern, Speicher etc., Änderung der Klasse an sich)
- Keine Auto-Connect (Schönes Feature, kollidiert mit mehreren Arduinos)