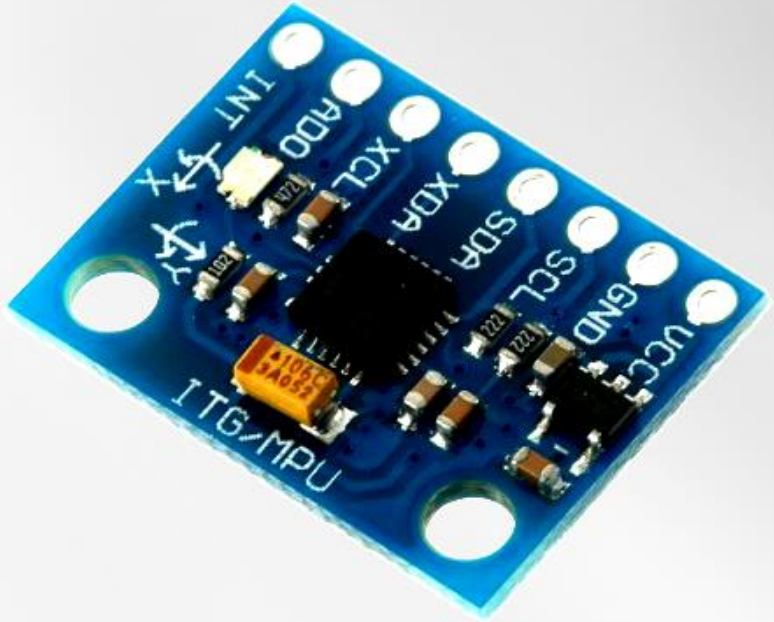
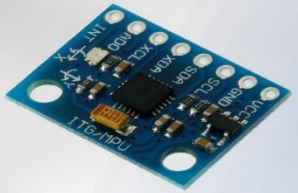
Kommunikation

Telekommandos

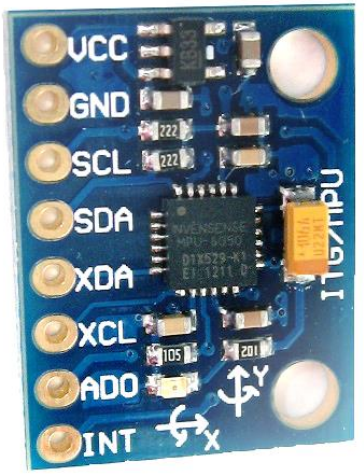


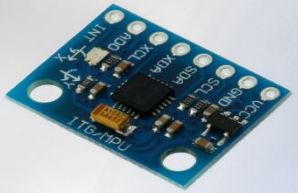


Inhalt

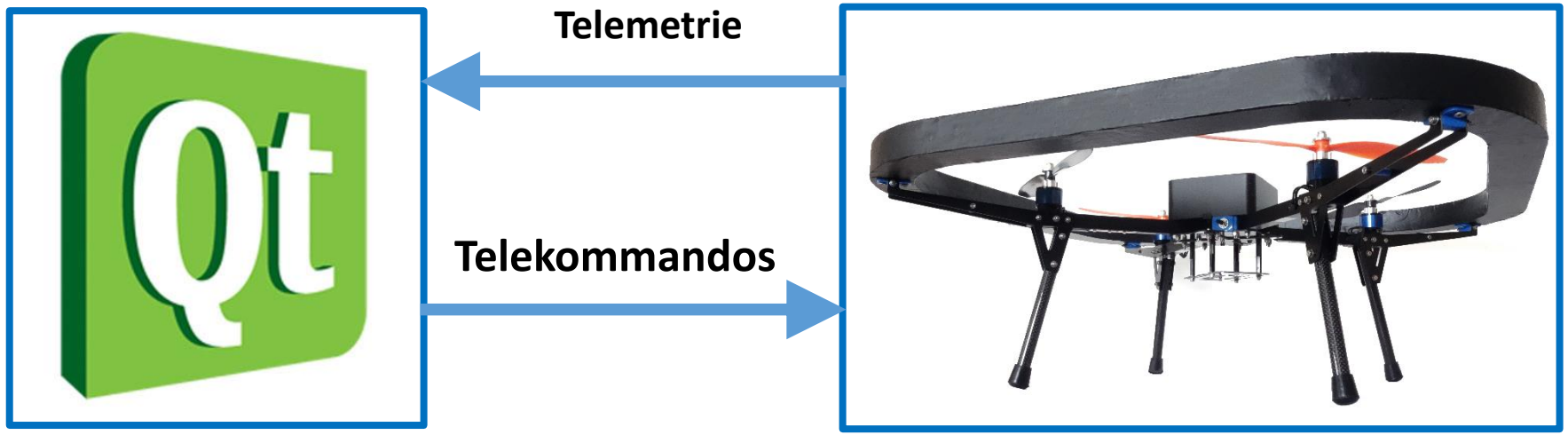
Umfang: ca. 2-4 Zeitstunden

- Telekommandos senden
- Telekommandos empfangen
- Bearbeitung der Benutzeroberfläche
- Aufgaben



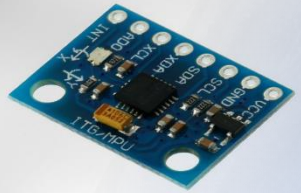


Telekommandierung



Pilot / Bodenstation

Quadrokopter

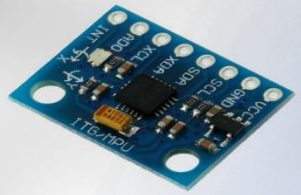


Telekommandierung

Das Qt – Programm wird verwendet, um den Quadrokopter zu steuern / kontrollieren

Allgemeine Vorgehensweise:

1. Daten entsprechend eines Protokolls zu einer Nachricht formulieren
2. Erstellte Nachricht über USART versenden
3. Nachricht auf AVR – Seite empfangen
4. Anhand des Protokolls die Nachricht interpretieren und Daten extrahieren
5. Daten verwenden, um Quadrokopter zu steuern / kontrollieren



Telekommandos senden

1. Daten entsprechend eines Protokolls zu einer Nachricht formulieren + 2. Nachricht senden

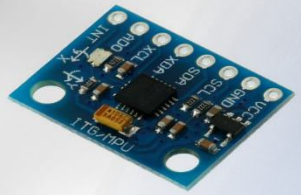
- `void` **`createFrameMessageAndSend`**`(`**`int`** `type, `**`int`** `data_length, `**`float`** `data[])`
  in ***mainwindow.c*** erstellt entsprechend dem Kommunikationsprotokoll einen Frame und sendet ihn ab.

`int` `type` — wird von der Enumeration *TelecommandTypes* aus ***protocol.h*** *entsprechend der gewünschten Nachricht ausgewählt:*

```
35  // Telecommandos: IDs
36  #define         TC_NOTHING                  -1
37  #define         TC_LED                      0
38  #define         TC_TRANS                    1
39  #define         TC_AUTOFLUG                 2
40  #define         TC_MOTOR                    3
41  #define         TC_IMU                      4
```

`int` `data_lenght` — Anzahl der Daten, die in diesem Frame mitgesendet werden.

`float` `data []` — Daten, die in diesem Frame mitgesendet werden.



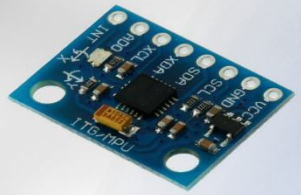
Telekommandos empfangen

3. Nachricht auf AVR – Seite empfangen. + 4. Daten extrahieren

- `int` **`extract_type_and_data_from_telecommand_frame`**`()`
  - In ***telecommand.c***
  - ließt vom USART
  - extrahiert einen ankommenden Telekommando-Frame
  - schreibt die relevanten Daten in my_*data*
  - gibt den Frametyp des empfangenen Frames zurück.

5. Daten verwenden, um Quadrokopter zu steuern / kontrollieren

- `void` **`my_read_telekommand()`**
  - In ***telecommand.c***
  - wird alle TC_UPDATE_RATE Millisekunden aufgerufen
  - Auslesen und Verarbeiten der Telekommandierung.

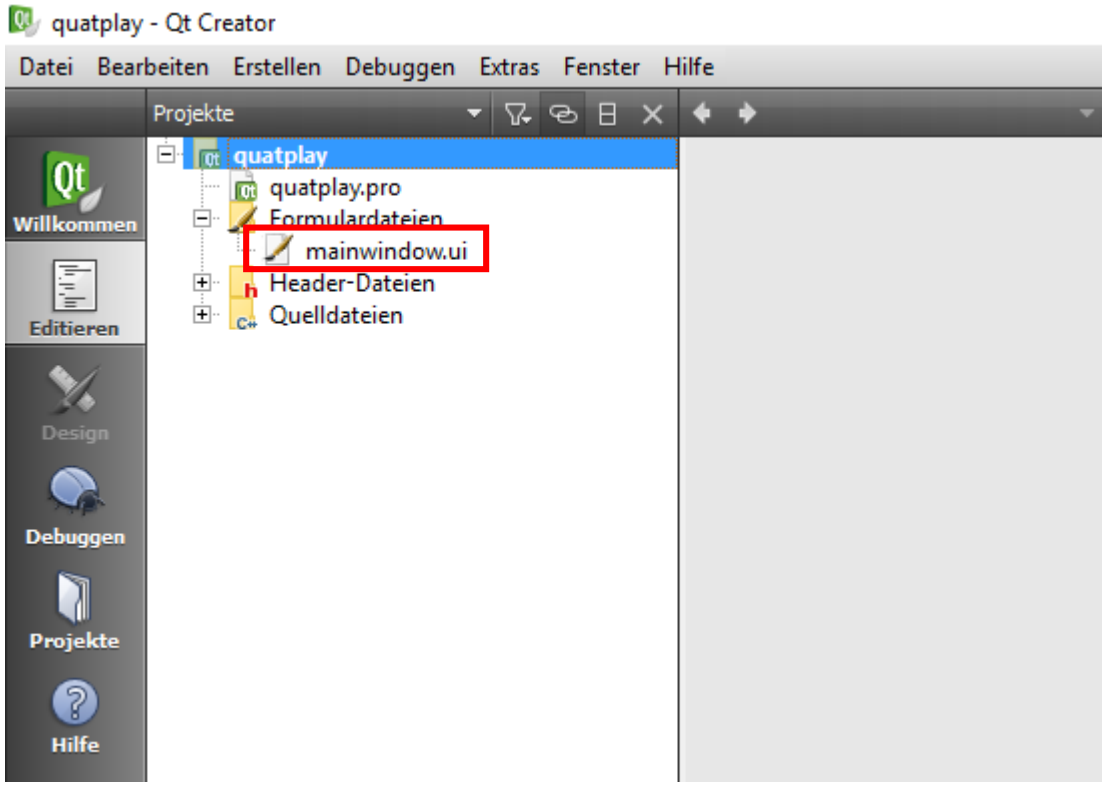


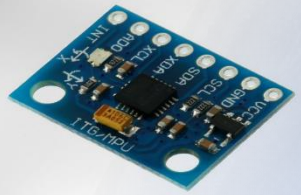
Bearbeitung der Benutzeroberfläche

Bedienelemente hinzufügen und bearbeiten

Doppelklick auf *Formulardateien/mainwindow.ui*

→ Öffnet die Bearbeitungsansicht der Benutzeroberfläche

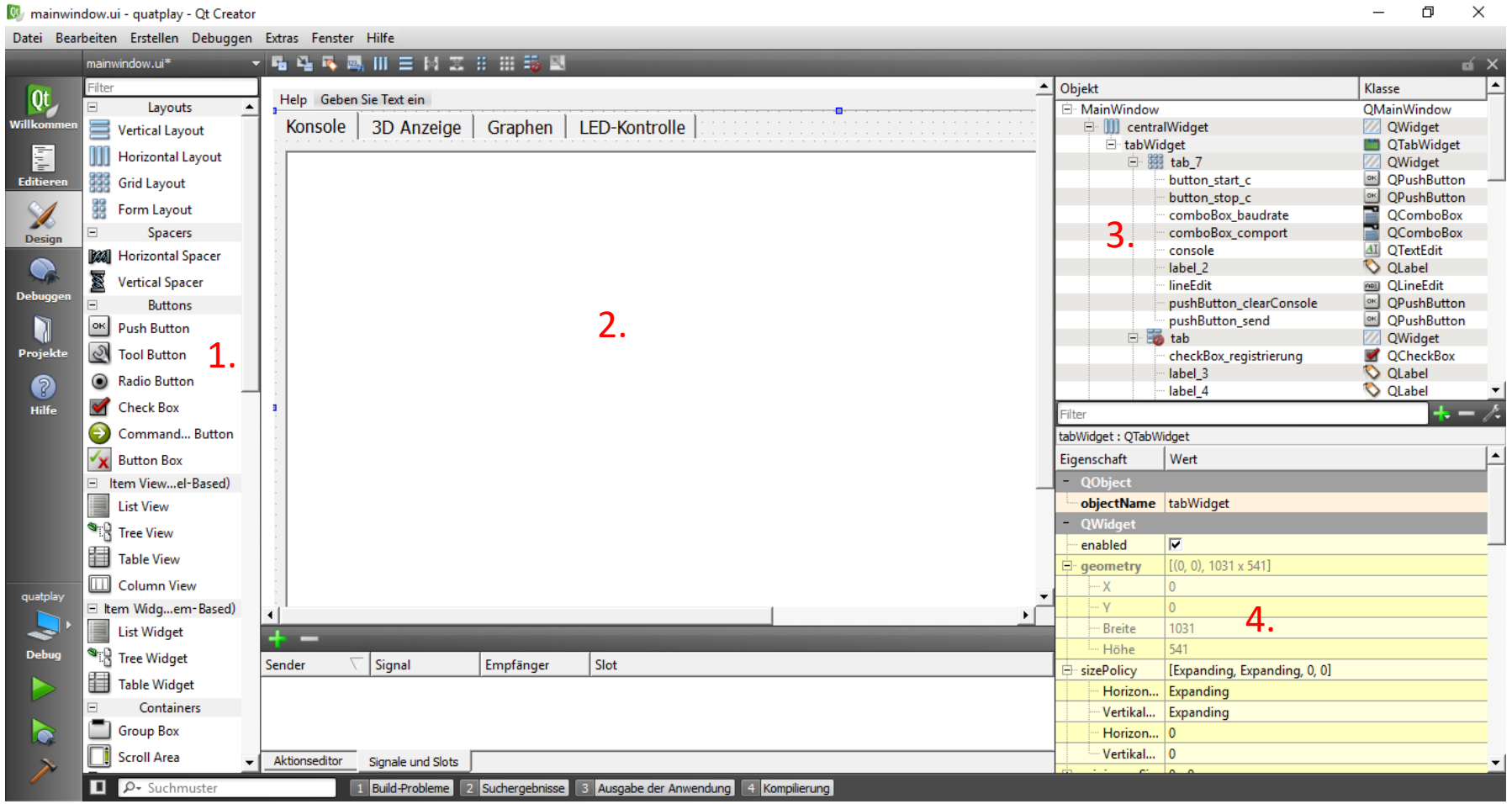


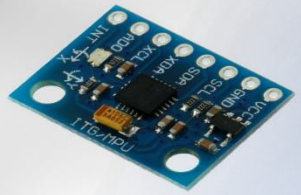


Bearbeitung der Benutzeroberfläche

Die Bearbeitungsansicht

1. Elementbibliothek
2. UI – Vorschau
3. UI – Elemente
4. Element - Details





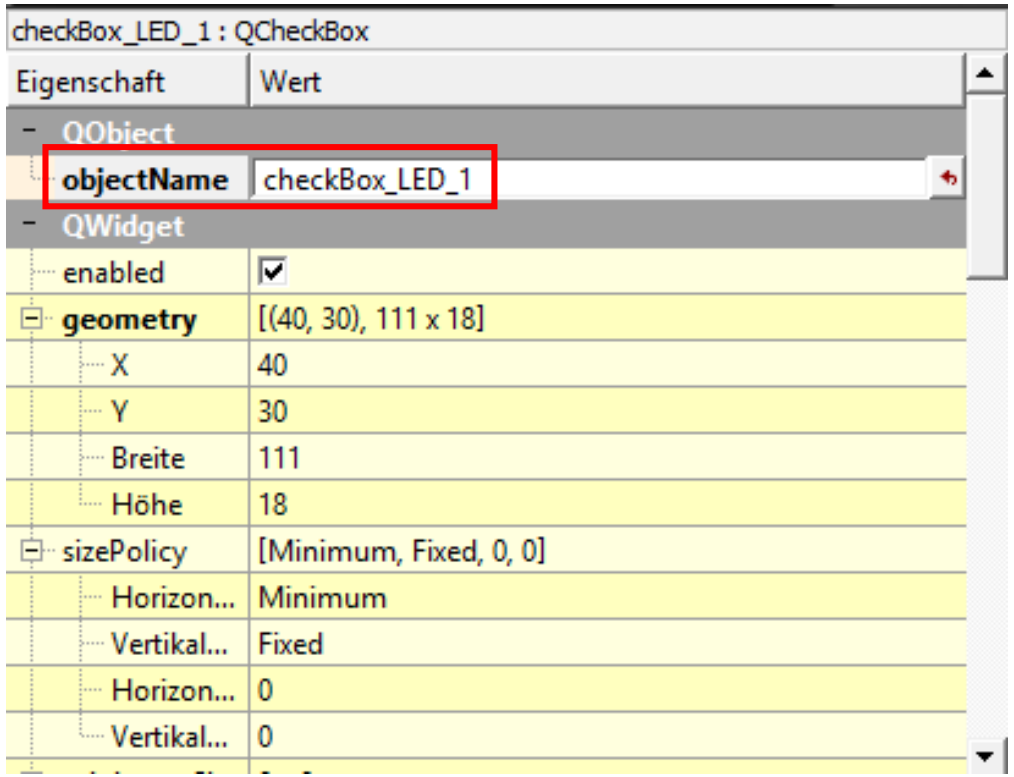
Bearbeitung der Benutzeroberfläche

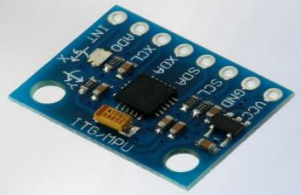
Element hinzufügen:

- Einfach über Drag und Drop aus Elementbibliothek einfügen
- **Wichtig:** Nach Einfügen gleich den Namen unter *Element – Details / objectName* aussagekräftig anpassen:
- Alle Elemente der Benutzeroberfläche können in *mainwindow.cpp* mit ui->objectName aufgerufen werden
- Die Funktionen von Methoden können über ui->objectName-> … aufgerufen werden

  Hier z.B.: `ui->checkBox_LED_1->isChecked()`

- Mit <strg> + <leer> nach dem ‚->‘ öffnet sich eine Liste möglicher Ergänzungen. So lassen sich gewünschte Funktionen schnell finden.



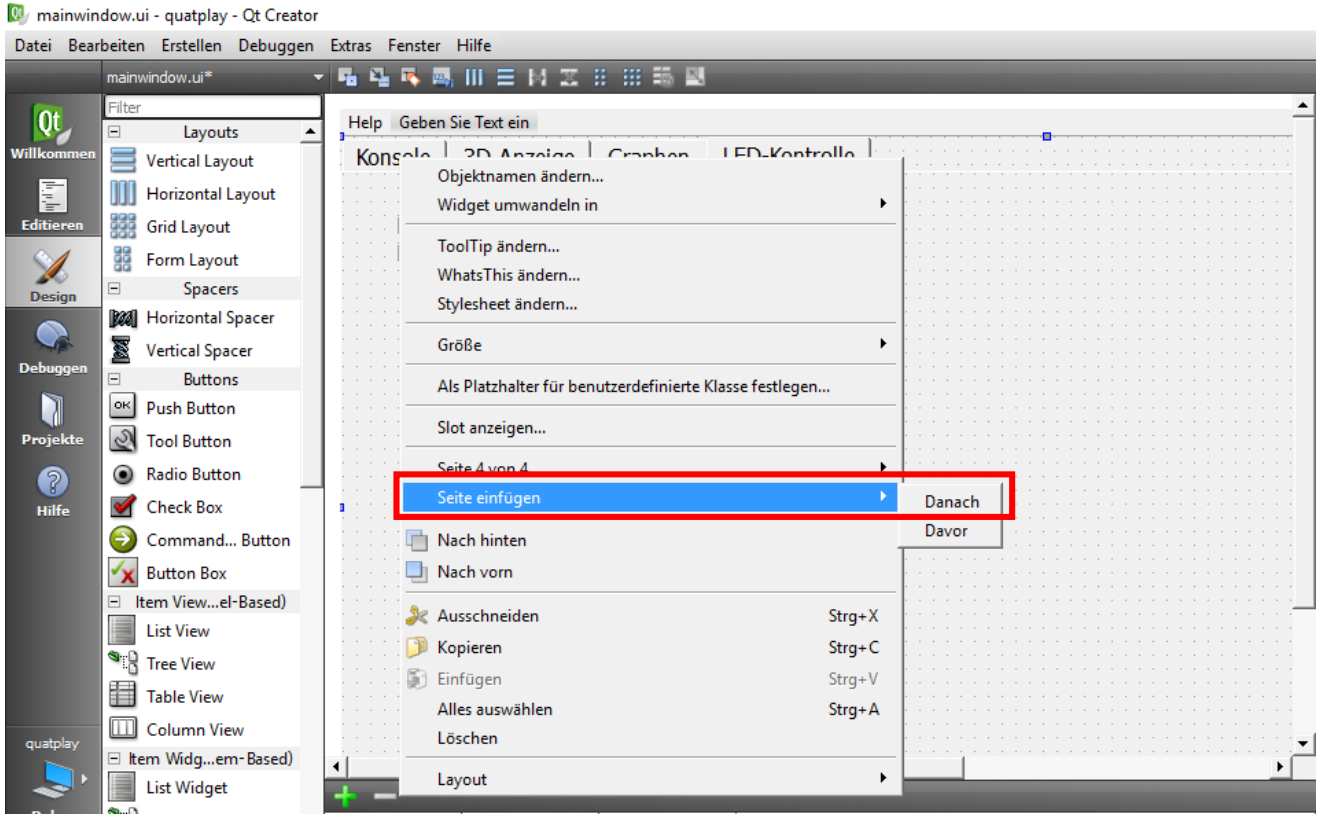


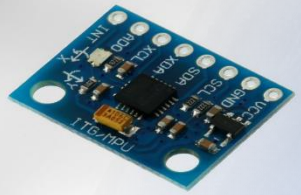
Bearbeitung der Benutzeroberfläche

Spezialfall: Hinzufügen von Tabs:

1. Rechtsklick auf bestehenden Tab
2. Seite Einfügen -> Danach auswählen
3. *currentTabText* unter *Element-Details* eingeben
4. *currentTabName* unter *Element-Details* ändern

Eigenschaft	Wert
tabPosition	North
tabShape	Rounded
currentIndex	4
iconSize	16 x 16
elideMode	ElideNone
usesScrollButtons	☑
documentMode	☐
tabsClosable	☐
movable	☐
currentTabText	Seite
currentTabName	tab_3
currentTabIcon	
currentTabToolTip	
currentTabWhats...	



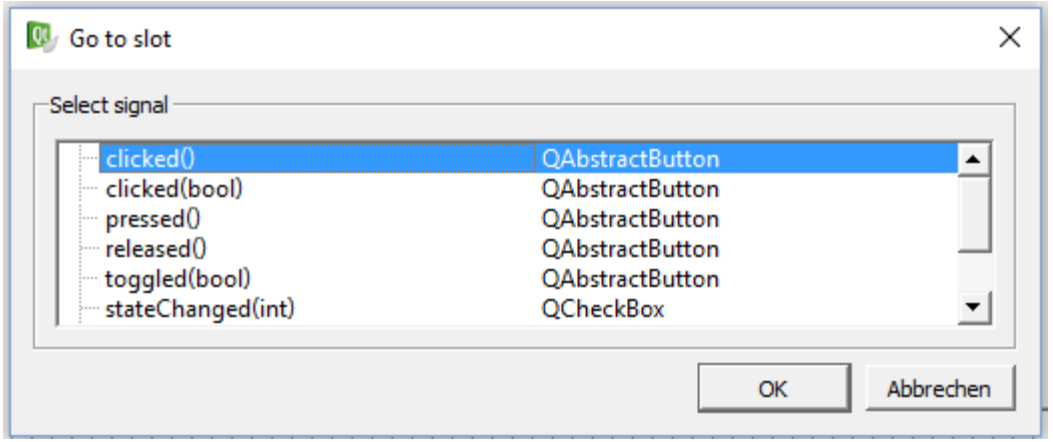


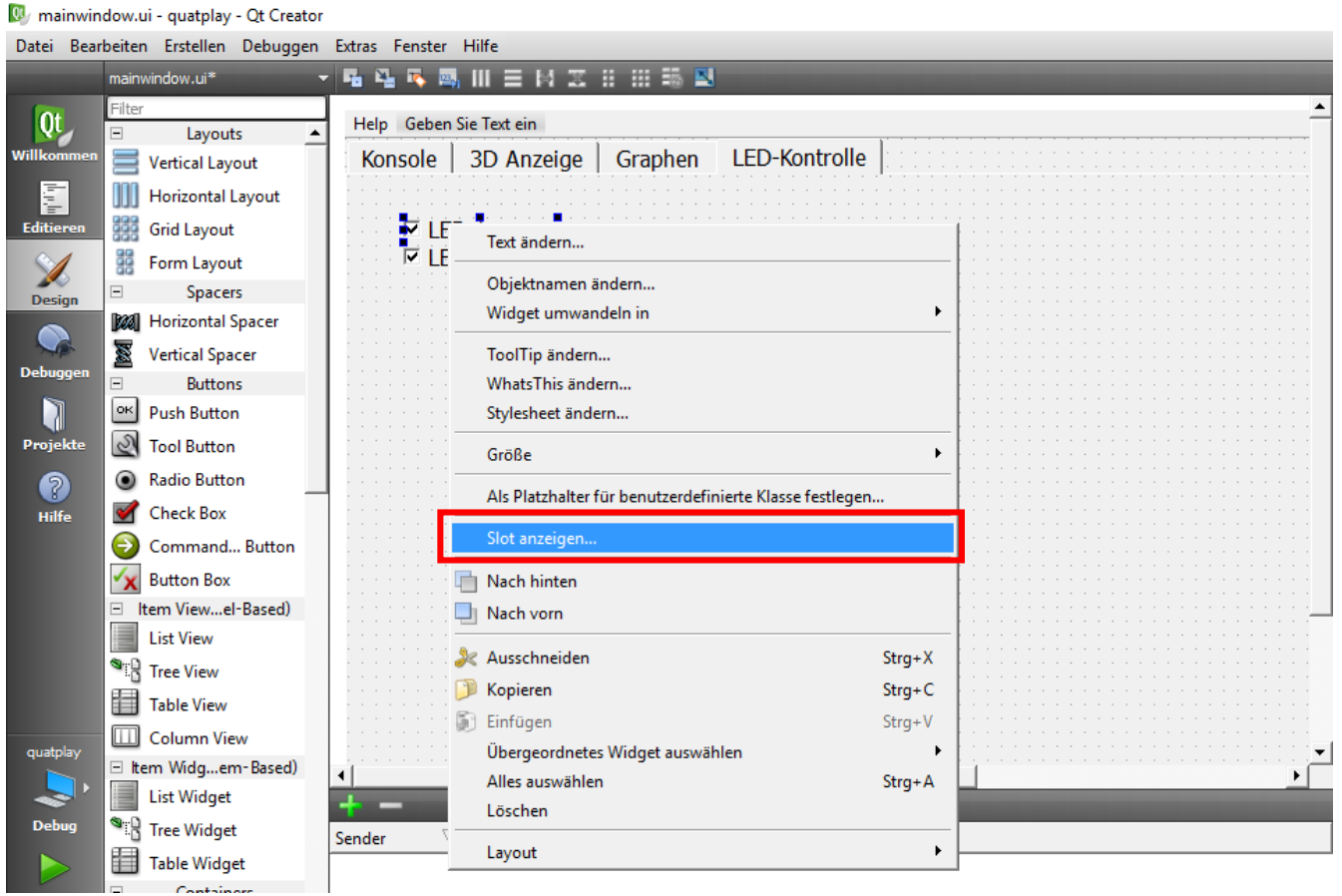
Bearbeitung der Benutzeroberfläche

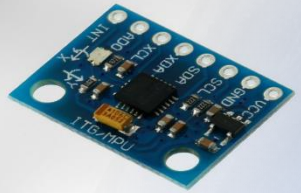
Funktionalität programmieren:

1. Rechtsklick auf Element
2. *Slot anzeigen…* auswählen
3. Gewünschte Aktion auswählen und mit *OK* bestätigen

→ Entsprechende Funktion wird unter *mainwindow.cpp* angelegt







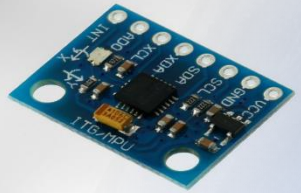
Aufgaben

Notwendige Hardware:

- EMQ3000
- Mini USB Kabel für Strom und zum Flashen
- Mini USB Kabel für Kommunikation: RS232 auf PC (RS232 oder USB)
- QCS / QCSF

Notwendige Software:

- AVR Studio 32 installiert (mit Tool Chain und Flip Treiber)
- Qt SDK Version 4.8.1 oder 4.7.4.
- Quatplay Qt Framework (Code)



Aufgaben

Aufgabe 1:

Erstellen Sie einen neuen Reiter neben dem Tab Graphen mit der Beschriftung „LED-Kontrolle“. Fügen Sie im neu erstellten Bereich zwei CheckBoxen LED 1 und LED 2 hinzu, die bei Programmstart aktiviert sind.

Aufgabe 2:

Programmieren Sie die neuen CheckBoxen so, dass man damit die LEDs 1 und 2 des EVK ansteuern kann. `void` **`createFrameMessageAndSend`**`(`**`int`** `type,` **`int`** `data_length,` **`float`** `data []`) in *mainwindow.cpp* sendet einen Frame über *USART* zum *EVK1100.* Um die Telekommandos zu verarbeiten, sind die Funktionen `void` **`my_read_telecommand`**`()` und `void` **`my_read_LED_Command`**`()` in *telecommand.c* zu implementieren. Die Funktion `int` **`extract_type_and_data_from_telecommand_frame`**`()` hilft Ihnen, den Frametyp des Telekommandos auszulesen. Vergleichen Sie dazu die *defines* in *protocol.h* .