



Technische
Universität
Braunschweig

Institut für
Flugführung



Serielle Kommunikation und API-Anleitung

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Andreas Dekiert M. Sc.,
19. Juni 2018

Agenda

- 03. April Einführung
- 10. April Softwareprojektmanagement
- 17. April Entwicklungstools
- 24. April GitHub
- 08. Mai Einführung Arduino/Fundduino
- 15. Mai Dateieingabe und -ausgabe
- 22. Mai Exkursionswoche
- 29. Mai Dokumentation und Bug-Reporting
- 05. Juni Einführung von Qt
- 12. Juni GUI-Erstellung
- 19. Juni Serielle Kommunikation und API-Anleitung**
- 26. Juni Projektarbeit
- 03. Juli Vorbereitung der Abgabe
- 10. Juli Abgabe

Serielle Kommunikation und API-Anleitung

Als Teilnehmer soll ich am Ende dieser Übung...

- ☐ wissen, wie die serielle Schnittstelle angesteuert wird
- ☐ die Anleitung für die Projektmappe verfassen können

Serielle Schnittstelle



Einführung

QtSerial

Arduino

Serielle Schnittstelle

- Datenübertragung zwischen verschiedenen Geräten
- Bits werden nacheinander übertragen (seriell)
- Bekannte Standards:
 - RS-232
 - Serial ATA (SATA)
 - Universal Serial Bus (USB)



**TX/RX-Ports des Arduino sind für maximal 5V ausgelegt.
Bei RS232 kann die Spannung bis zu 12 V betragen.**

Baudrate

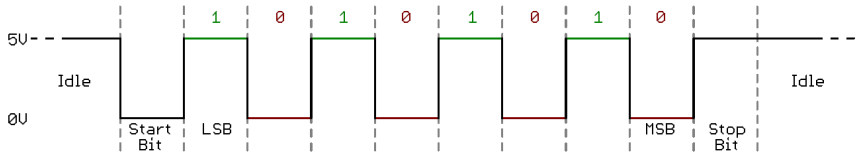
	Baud-Rate	max. Länge
■ Die Baudrate gibt an, mit welcher Schrittgeschwindigkeit Daten übermittelt werden.	2.400	900 m
■ Einheit: $\frac{\text{Symbol}}{\text{Sekunde}}$	4.800	300 m
	9.600	152 m
■ Bei Übertragungen mit 2 Spannungen	19.200	15 m
gilt die Einheit $\frac{\text{Bits}}{\text{Sekunde}}$	57.600	5 m
	115.200	< 2 m

Wichtig:

Empfänger und Sender benötigen die gleiche Baudrate!

Start-/Stop-Bit

- Asynchrone Kommunikation
- Start-Bit kündigt Nachricht an
- Stop-Bit schließt Nachricht ab
- Dazwischen wird ein Datenset mit 5-9 Bits übertragen
- Häufig wird eine Länge von 8 Bit verwendet



Paritätsbit (engl. Parity Bit)

- Das Paritätsbit folgt auf die Nachrichten-Bits
- Gibt an, ob die Anzahl der 1-Bits gerade oder ungerade ist
- Zwei Varianten üblich:
 - Parity Bit = 1 wenn gerade Anzahl von 1-Bits
 - Parity Bit = 1 wenn **ungerade** Anzahl von 1-Bits
- Empfänger validiert empfangene Daten durch Zählen der 1-Bits und Vergleich mit dem Parity Bit

Daten-Bits	Anzahl 1-Bits	a) Gerade	b) Ungerade
0000 0000	0	0000 0000 1	0000 0000 0
0000 0111	3	0000 0111 0	0000 0111 1
0100 0111	4	0100 0111 1	0100 0111 0

Gibt es Fragen oder Anmerkungen zu dem Unterthema
Serielle Schnittstelle?



QSerialPort als Arduinoschnittstelle

- Abgeleitet aus QIODevice (Input/Output-Device)
- Kann zum Lesen und Schreiben von Daten verwendet werden
- Seit Qt 5.1 in Qt eingebunden
 - Verfügbar über das Modul serial
- ```
Qt += serial
```

## Vorbedingung

- Der Rechner muss Zugang zu einer seriellen Schnittstelle haben.  
Das Arduino-Board verfügt über eine serielle Schnittstelle, welche auch per USB angesteuert werden kann.
- Der Portname dieser Schnittstelle muss identifiziert werden.

# Nutzung des QSerialPort-Objekts

1. Erstellung des Objekts QSerialPort unter Angabe des Serial-Portnamens
2. Öffnen der Schnittstelle über die Methode `open()`
  - `open(QIODevice::ReadOnly)`
  - `open(QIODevice::WriteOnly)`
  - `open(QIODevice::ReadWrite)`
3. Versand von Daten erfolgt über die Methode `write()`
4. Wenn Daten empfangen wurden, wird das Signal `readyRead()` emittiert
5. Über `read()` können die Daten ausgelesen werden

Die Methode `read()` sollte über eine Schleife aufgerufen werden, bis der Puffer leer ist.

# Übung: Öffnen der seriellen Schnittstelle

1. Füge in der Projektdatei das Modul `serial` hinzu
2. Erstelle die Klasse `ArduinoInterface` (Basisklasse: `QObject`)
3. Lege die Membervariable `m_serialPort` des Typs `QSerialPort` an
4. Nehme im Konstruktor für die Instanz `m_serialPort` folgende Einstellungen vor:
  - 4.1 Stelle über die Methode `setPortName()` die richtige Portnummer ein
  - 4.2 Stelle über die Methode `setBaudRate()` die richtige Baudrate ein
  - 4.3 Öffne über die Methode `open()` den Port zum Schreiben von Daten

# Übung: Schreiben von Daten

1. Erstelle in der Klasse `ArduinoInterface` den Slot `writeColorToArduino()` mit zwei Eingangsparametern:
  - `colorIdent` des Typs **char**
  - `colorValue` des Typs **char**
2. Leite die eingegangenen Daten über die Methode `write()` an `m_serialPort` weiter
3. Damit die Daten sofort versendet werden, rufe in der Instanz die Methode `flush()` auf
4. Verbinde das Signal der GUI `sendColor()` mit dem hier angelegten Slot `writeColorToArduino()`

# Gibt es Fragen oder Anmerkungen zu dem Unterthema QtSerial?



```
void setup() {
 // Beim Oeffnen wird die Baudrate festgelegt
 Serial.begin(9600);
}
```

Tabelle 1: Standardwerte

| Parameter   | Einstellung        |
|-------------|--------------------|
| Datenset    | 8 Bits             |
| Paritätsbit | Keins (engl. none) |
| Stop-Bit    | 1 Bit              |
| ⇒           | 8-N-1              |

## Arduino

## Initialisierung serielle Schnittstelle

```
// Speicher fuer das ankommende Byte
char incomingByte = '\0';

void setup() {
 // Beim Oeffnen wird die Baudrate festgelegt
 Serial.begin(9600);
}

void loop() {
 if (Serial.available() > 0){
 incomingByte = Serial.read();

 // Zuruecksenden des empfangenen chars
 Serial.print(incomingByte);
 }
}
```



# LED-Farbwahl

# Übertragungsprotokoll

- Übertragung erfolgt seriell
- 0-100 sind reserviert für den Farbwertbereich
- 101-103 wählen die Farbe aus
  - 101 Rot
  - 102 Grün
  - 103 Blau
- 2 Bytes werden für eine Farbeinstellung gesendet

|      | Byte 0 | Byte 1 |             |
|------|--------|--------|-------------|
| Wert | 101    | 0      | → 0 % Rot   |
| Wert | 101    | 100    | → 100 % Rot |
| Wert | 102    | 50     | → 50 % Grün |
| Wert | 103    | 20     | → 20 % Blau |

# Übung: LED-Einstellung Arduino

1. Erstelle ein Arduino-Projekt
2. Nutze die `setup()`-Funktion, um die Pins für die RGB-LED einzustellen
3. Richte die serielle Schnittstelle mit der passenden Baudrate ein
4. Lese mit jedem Durchgang - sofern vorhanden - ein Byte aus
5. Für die Werte 101 - 103 soll die aktive Farbe geändert werden
  - 101 → rot
  - 102 → grün
  - 103 → blau
6. Die Werte 0 - 100 sollen auf den Bereich 0 - 255 interpoliert und der aktiven LED zugewiesen werden

Gibt es Fragen oder Anmerkungen zu dem Thema  
**Serielle Schnittstelle?**



# Abgehakt

## Serielle Kommunikation und API-Anleitung

Als Teilnehmer soll ich am Ende dieser Übung...



- ☒ wissen, wie die serielle Schnittstelle angesteuert wird
- ☐ die Anleitung für die Projektmappe verfassen können

## Anleitung



Voraus-  
setzung

Inbetrieb-  
nahme

Verwendung

# Voraussetzung

In der Anleitung sollen kurz die nötigen Voraussetzungen zum Bauen und Ausführen des Softwareprojekts aufgelistet werden.

- Betriebssystem  
Nur Betriebssysteme angeben, mit denen die Software getestet wurde
- Erforderliche Sensoren
- Mit welchem Compiler kann die Software gebaut werden  
Bspw.: MinGW, clang, MSVC, AVR-g++ (Arduino)
- Erforderliche Bibliotheken  
Verwendete Qt-Module (QtGui, QtCore, QtSerialport,...) oder externe Bibliotheken



# Inbetriebnahme

In der Anleitung soll die Inbetriebnahme der Software beschrieben werden. Welche Schritte sind erforderlich, um die Software verwenden zu können?

- Müssen oder können Einstellungen in Dateien vorgenommen werden?
  - Dateinamen und Pfade angeben
  - Einstellungsmöglichkeiten auflisten
  - Beispiele nennen
- Benötigt das Programm Eingangsparameter?
  - Nennung der Eingangsparameter
  - Mögliche Werte angeben
- Werden weitere Ressourcen benötigt?
  - Datenbanken
  - Skripte
  - ...



# Verwendung

Jede User-Story soll in der Anleitung ein Kapitel erhalten.  
Falls vorhanden, sollten die folgenden Punkte in der Anleitung dokumentiert sein:

- Beschreibung der Nutzung und Bedienung sämtlicher Funktionen der User-Stories
- Screenshots bei GUI-Applikationen
  - Fenster
  - Reiter
  - Sonstige Ansichten
- Sämtliche Auswahlmöglichkeiten eines Menüs
- Die Status eines Programms (play, pause, ff)
- Beschreibung der Programmlogik bei automatisierten Abläufen  
Z.B. über Ablaufdiagramme

# Form der Anleitung

Die Anleitung muss entweder als pdf-Dokument erstellt oder direkt in das Projekt-Wiki eingetragen werden.

## pdf-Anleitung

- Das pdf-Dokument muss sich im doc-Ordner des Code-Repositories befinden.
- Das pdf-Dokument muss zusätzlich im Projekt-Wiki unter Punkt 11: Anleitung verlinkt werden.

## Wiki-Anleitung

- Die Anleitung kann zusätzlich Unterkapitel enthalten.  
Siehe `_Sidebar.md`

Gibt es Fragen oder Anmerkungen zu dem Thema  
**Anleitung?**



# Abgehakt

## Serielle Kommunikation und API-Anleitung

Als Teilnehmer soll ich am Ende dieser Übung...

- ☒ wissen, wie die serielle Schnittstelle angesteuert wird
- ☒ die Anleitung für die Projektmappe verfassen können

Jetzt besteht die Möglichkeit, das Sprintmeeting durchzuführen.

Protokolliert bitte

- die bearbeiteten Aufgaben der Vorwoche.
- die Zwischenstände der geplanten Aufgaben.
- die in der kommenden Woche zu bearbeitenden Aufgaben.

# Ende

Vielen Dank für eure Aufmerksamkeit!