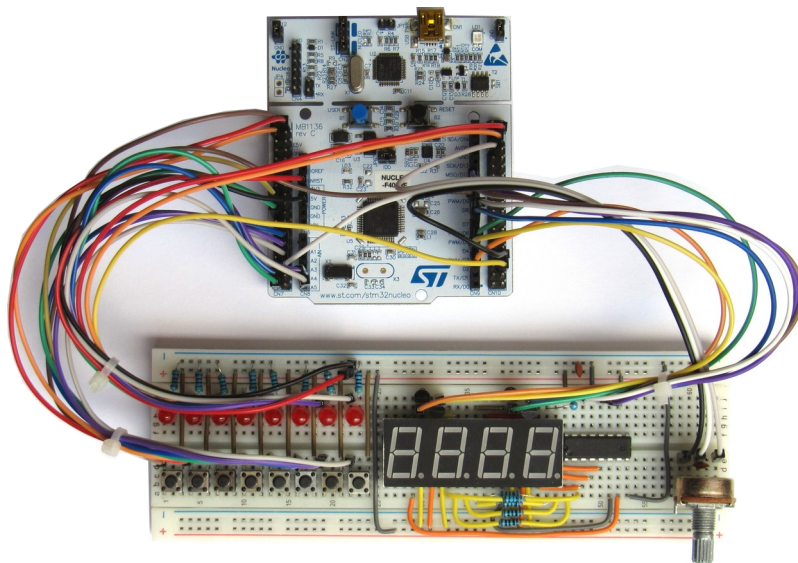


# Mikroprozessoren-Labor



## Versuch 3, USART und Messgerät

### Versuchsvorbereitung

Stand: 18. April 2021

# I. Versuchsbeschreibung

In diesem Versuch soll die serielle Schnittstelle in Betrieb genommen werden.

Danach wird mit allen jetzt bekannten Peripherieeinheiten eine komplette Anwendung „Messgerät“ realisiert.

## II. PuTTY

Um mit dem Mikrocontroller über die serielle Schnittstelle kommunizieren zu können, benötigen Sie ein Terminal mit serieller Schnittstelle (das hat aber keiner mehr) oder ersatzweise einen PC (Notebook o.ä.) mit einem Programm, das ein Terminal emuliert.

Der bekannteste, kostenlos verfügbare Terminal-Emulator für Windows ist PuTTY. Dieses Programm unterstützt nicht nur die Kommunikation über physikalisch tatsächlich vorhandene serielle Schnittstellen sondern auch über Netzwerkverbindungen, die seriellen Schnittstellen nachgebildet sind.

### 1 Installation

Laden Sie das Installationsprogramm von <https://www.putty.org/> herunter und führen Sie es aus.

### 2 Start

Starten Sie PuTTY indem Sie die Windows-Taste drücken und den Anfang des Namens „putt...“ eintippen. Klicken Sie auf das „App-Icon“.

PuTTY startet mit einem Fenster, in dem Sie die gewünschte Verbindung („Session“) konfigurieren können.

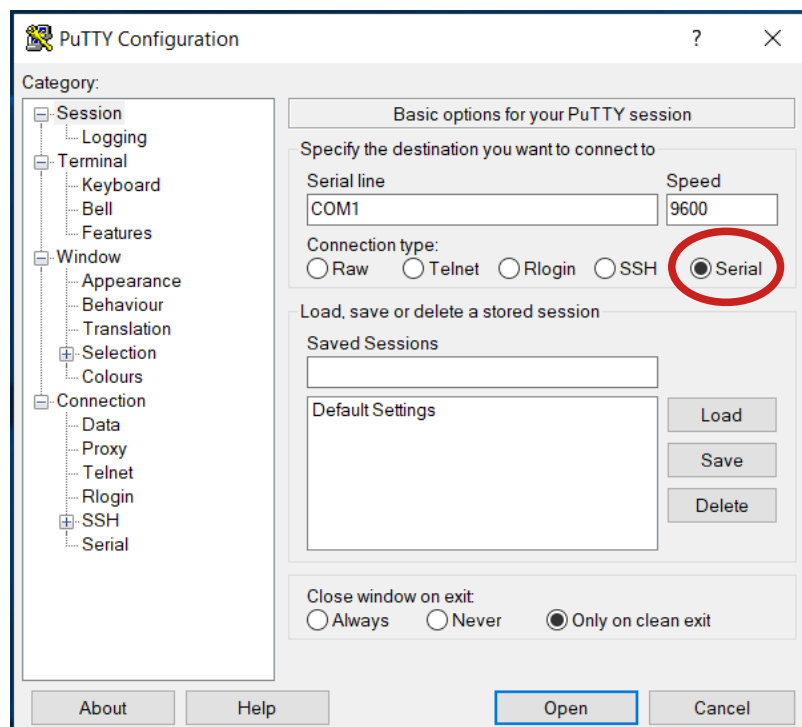
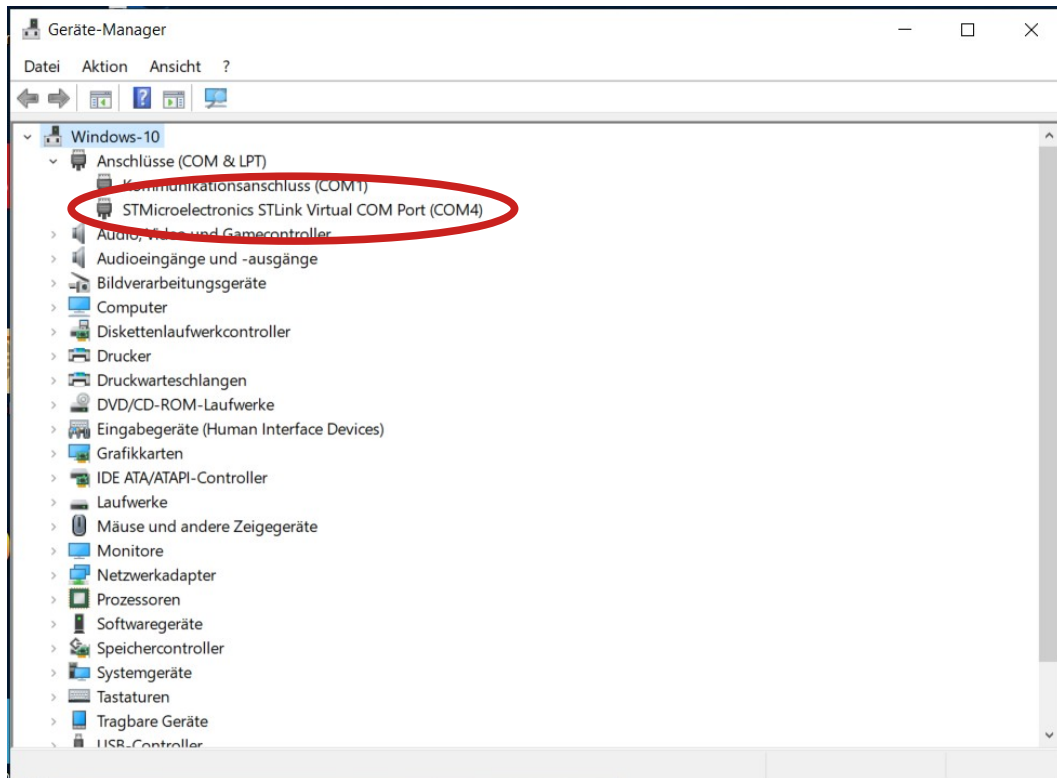


Bild 1: PuTTY Start-Fenster

Wählen Sie als Verbindungstyp („Connection Type“) „Serial“ aus. Die voreingestellten Werte für „Serial Line“ (Schnittstelle) und „Speed“ (Geschwindigkeit) müssen Sie anschließend noch anpassen.

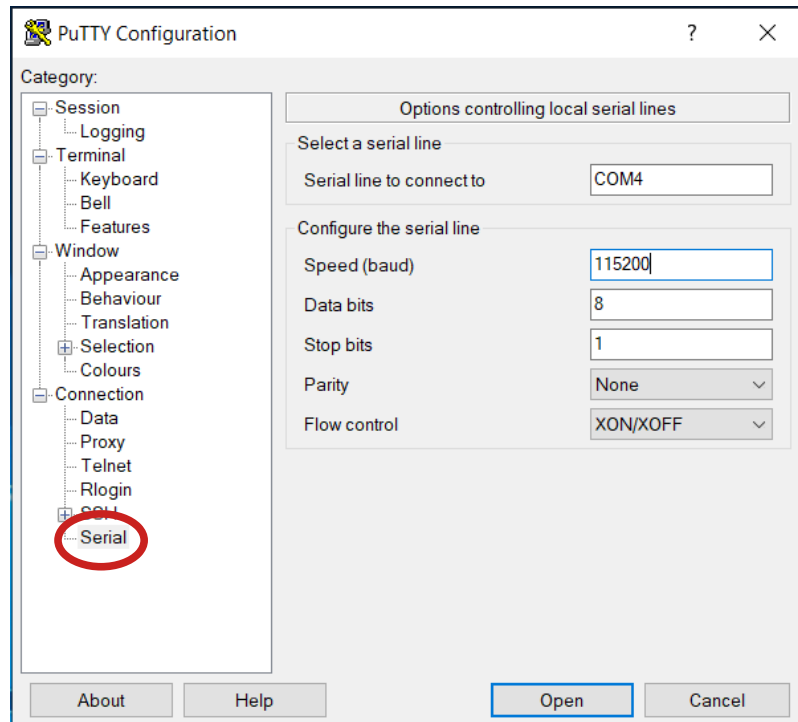
Um den Namen der durch die USB-Verbindung zum NUCLEO-Board getunnelten seriellen Schnittstelle (Virtual COM Port) herauszufinden, öffnen Sie den Gerätemanager (wieder Windows-Taste drücken und „gerä...“ eintippen). **Geräte Manager - Device Manager**



*Bild 2: Gerätemanager*

Suchen Sie unter „Anschlüsse“ nach „STMicroelectronics STLink Virtual COM Port“. Dahinter steht in Klammern der Name des Ports (im Bild „COM4“).

Setzen Sie den Namen des Ports in das Feld „Serial line“ des PuTTY-Fensters ein und ändern Sie die Geschwindigkeit („Speed“) auf den in Ihrem Programm verwendeten Wert.



*Bild 3: PuTTY Konfigurationsdialog*

Falls Sie Details der Schnittstelleneinstellungen (Anzahl Bits, Start-/Stop-Bits, Parity ändern wollen, müssen Sie links im Fenster auf Connection/Details gehen, dann sehen Sie rechts alle einstellbaren Werte.

## III. Aufgaben

### 1 Synchronisation

Das vorbereitete Laborprojekt enthält bereits eine Funktion `task1`, die wie in der Vorlesung gezeigt, immer wieder das selbe Zeichen ausgibt. In diesem Fall ist es (ein Stern `'*'`).

Starten Sie zuerst das Programm und dann erst den Terminal-Emulator. Mit einer gewissen Wahrscheinlichkeit sollten Sie statt des erwarteten Sterns im Terminal-Emulator ein `'%'` oder ein `'I'` sehen. Falls Sie doch den Stern sehen, beenden Sie den Terminal-Emulator wieder und starten ihn neu (bei Bedarf mehrmals). Wenn Sie das falsche Zeichen sehen, drücken Sie auf die Reset-Taste des Mikrocontrollers (schwarzer Taster). Dann werden im Terminal-Emulator auf jeden Fall Sterne angezeigt.

Schließen Sie den Logik-Analysator an GND und die Rx und Tx Pins auf dem Programmierteils an. Beachten Sie, dass die Pins auf dem Programmierteil aus Sicht des USB-Tunnels bezeichnet sind. Die Ausgabe des Mikrocontrollers messen Sie daher am Rx Pin, denn aus Sicht des USB-Tunnels handelt es sich um Daten, die von der seriellen Schnittstelle empfangen werden.

- ✎ **Schriftliche Aufgabe V-1:** Stellen Sie die Abtastfrequenz und die Anzahl zu messender Werte so ein, wie es für die Analyse einer seriellen Schnittstelle mit 115200 baud sinnvoll ist. Begründen Sie, warum Sie die von Ihnen gewählten Einstellungen für sinnvoll halten (vollständige, grammatikalisch korrekte Sätze).
- ✎ **Schriftliche Aufgabe V-2:** Zeichnen Sie in einen Screen-Shot des Messergebnisses ein, wie die Pegel an Rx von der seriellen Schnittstelle des Virtual COM Ports interpretiert werden um zu erklären, warum ein anderes Zeichen als der erwartete Stern empfangen wird.
- ✎ **Schriftliche Aufgabe V-3:** Verwenden Sie den gleichen Screen-Shot ein zweites Mal und zeichnen ein wie die gemessenen Pegel interpretiert werden müssten, um tatsächlich ASCII-Stern-Zeichen zu empfangen.

### 2 Zahlen empfangen und anzeigen

Erstellen Sie eine Funktion `task2` als Kopie von `task3` aus der Durchführung von Labor 2. Entfernen Sie das explizite Setzen der Werte der Anzeigestellen und des Dezimalpunkts, so dass nur die Definition von `ssd` (und der dafür erforderlichen Objekte), die Definition von `digitTimer` und die Endlosschleife mit der Abfrage des Timers und der Aktualisierung von `ssd` übrig bleiben.

Ergänzen Sie vor der Schleife ein Objekt vom Typ `BufferedSerial` für die Kommunikation mit dem Terminal-Emulator. Stellen Sie die Schnittstelle auf „non blocking I/O“.

Ergänzen Sie weiterhin vor der Schleife ein Objekt `application` vom Typ `CHEXShifter`. Die (unvollständige) Definition dieser Klasse finden Sie im Laborprojekt. In der Endlosschleife von

task2 rufen Sie die Methode poll der Klasse CHexShifter auf. Damit ergibt sich folgendes „Hauptprogramm“:

```

// ...
// ssd defintion etc.
// ...

CPolledTimer digitTimer(5);
BufferedSerial usart(PA_2, PA_3, 115200);
usart.set_blocking(false);
CHexShifter application(&ssd, &keys, &leds, &usart);
while (true) {
    if (digitTimer.timeReached()) {
        ssd.activateNextDigit();
    }
    application.poll();
}
    
```

Die Klasse CHexShifter repräsentiert die realisierte Anwendung. Die Implementierung nutzt die Hardware-Komponenten, deren Adressen die Klasse in den Parametern des Konstruktors übergeben bekommt.

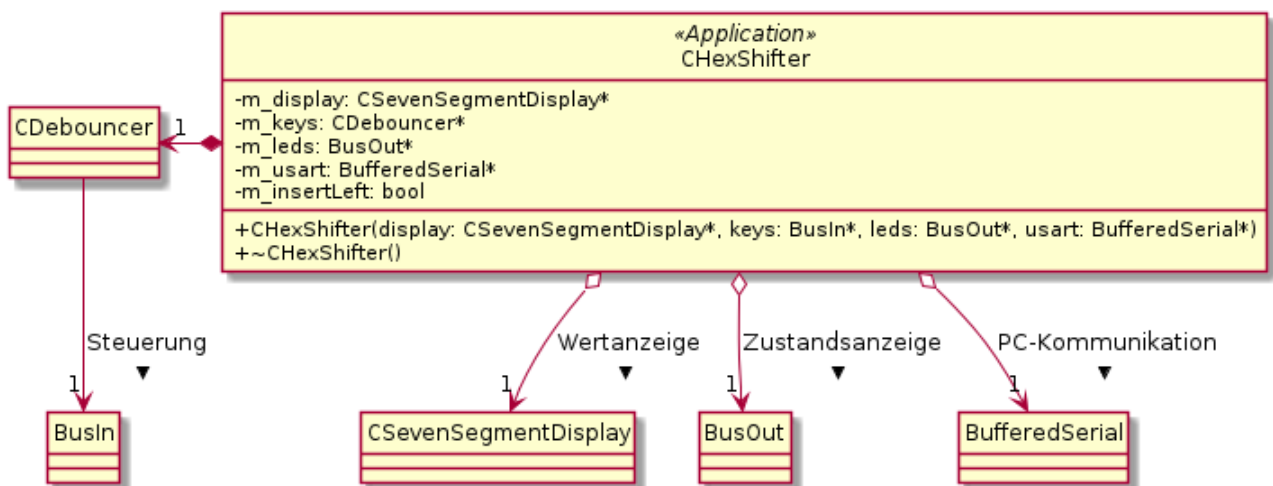


Bild 4: Klassendiagramm CHexShifter

Weitere Erläuterungen zur Klasse CHexShifter finden Sie in der Header-Datei.

## 2.1 Grundfunktionalität

Implementieren Sie die in den Kommentaren der Klassendefinition beschriebene Grundfunktionalität (Schieben nach links und Hinzufügen neuer Zahlen ganz rechts).

Testen Sie Ihre Lösung. Stellen Sie insbesondere sicher, dass sich bei Eingabe unzulässiger Zeichen im Terminal-Emulator die Anzeige nicht ändert.

## 2.2 Umschalten der Richtung

Ergänzen Sie die in der Klassendefinition beschriebene Funktionalität zum Umschalten der Schieberichtung und der Einfügeposition.

✎ **Schriftliche Aufgabe V-4:** Stellen Sie auf der Anzeige in der ersten Stelle Ihre Laborgruppe ein (,A‘, ,B‘, ...) und auf den folgenden drei Stellen die letzten drei Ziffern Ihrer Matrikelnummer. Dokumentieren Sie Ihren erfolgreichen Test durch eine Aufnahme, **auf der auch Ihr Studentenausweis** (lesbar!) zu sehen ist.

## 2.3 Fehleranzeige

(Freiwillige Ergänzung) Wird vom Terminal-Emulator ein unzulässiges Zeichen gesendet, soll die zweite LED von rechts für eine Sekunde leuchten. Hinweis: da ein `CPolledTimer` nicht aktiviert oder deaktiviert werden kann, müssen Sie den Timer für das Messen der Leuchtdauer dynamisch (auf dem Heap) anlegen.

## 2.4 Umschalten der Richtung über Tastatur

(Freiwillige Ergänzung) Die Schieberichtung und Einfügeposition soll auch über die Tastatur steuerbar sein. Die Eingabe von ,<‘ stellt die Schieberichtung auf links (Grundfunktionalität), die Eingabe von ,>‘ stellt die Schieberichtung auf rechts).

Wenn Sie diese Aufgabe bearbeiten, werden Sie feststellen, dass Sie Code duplizieren müssen. Das ist immer schlecht. Führen Sie daher eine neue private Methode `void setInsertLeft(bool)` ein, die das Attribut `m_insertLeft` entsprechend dem übergebenen Wert ändert und die LED ganz rechts ein oder ausschaltet.

Vergessen Sie nicht, die Fehleranzeige anzupassen (,<‘ und ,>‘ sind jetzt zulässige Zeichen).

## 2.5 Eingabe über Tasten

(Freiwillige Ergänzung) Es soll die Möglichkeit geschaffen werden, neue Werte – außer über die serielle Schnittstelle – auch über die Taster einzugeben. Die zusätzliche Funktionalität ist wie folgt definiert.

Beim Drücken einer der vier Tasten ganz links wechselt der Zustand der jeweils darüber angeordneten LED. Damit stellen die vier LEDs ganz links eine 4-stellige Binärzahl (ein Nibble) dar. Der Taster rechts daneben, also der fünfte von links, hat die Funktion einer „Enter-Taste“. Wird er gedrückt, wird die gerade dargestellte Zahl von links oder rechts in die Anzeige „geschoben“ so als wäre das entsprechende Zeichen über die serielle Schnittstelle empfangen worden. Weiterhin werden die vier LEDs ganz links ausgeschaltet. Hinweis: es sind nur Ergänzungen in der Methode `pollKeys` erforderlich.