

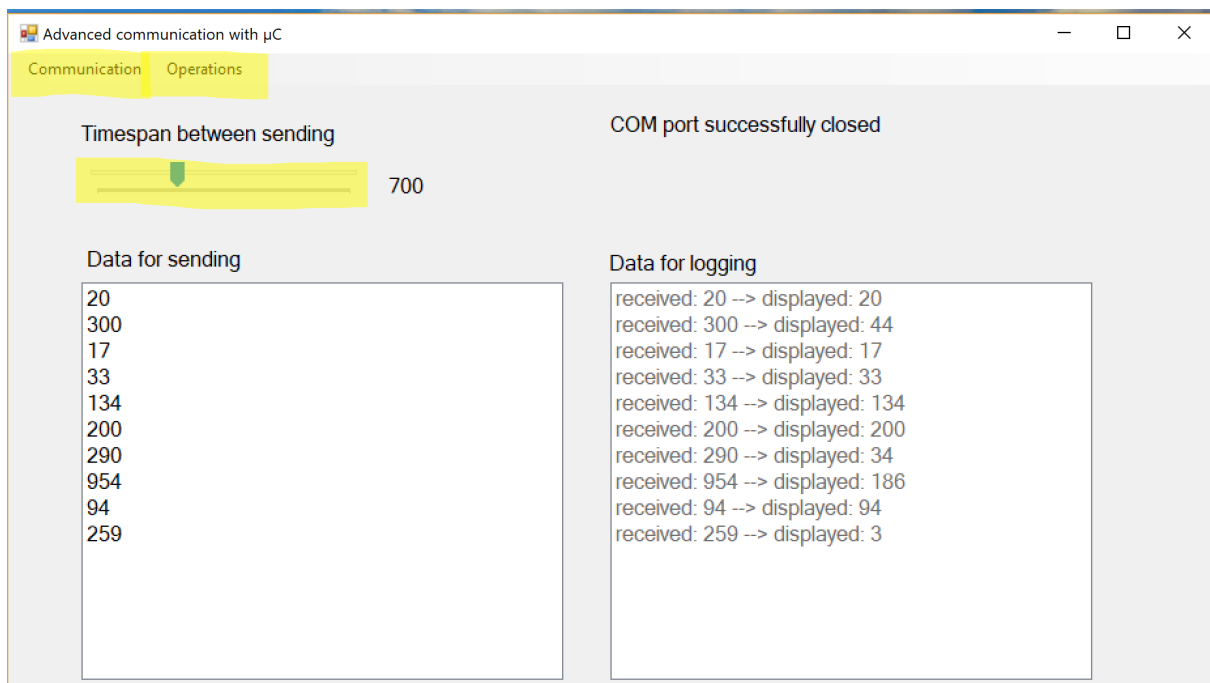
Beispiel Advanced Serial Communication

Mit dem nachfolgenden Beispiel wird eine zeitgesteuerte seriellen Kommunikation zwischen Mikrocontroller und C#-Visualisierung, und der Umgang mit Dateien und Menüs geübt.

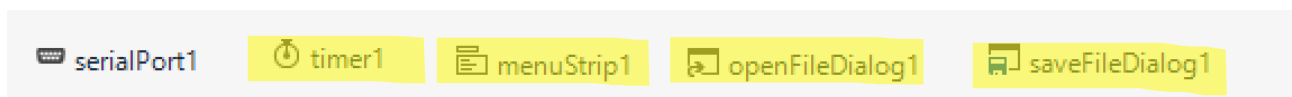
Entpacken Sie B5_AdvancedCommunication.zip in jenem Ordner, in dem sich Ihre AII-Projekte befinden. Die Visual Studio Solution enthält zwei Projekte: die Windows Forms App (.Net Framework) AdvancedCommunicationVisu und das Arduino-Projekt uCAdvancedCommunication mit der c-Datei

Visualisierung mit C#

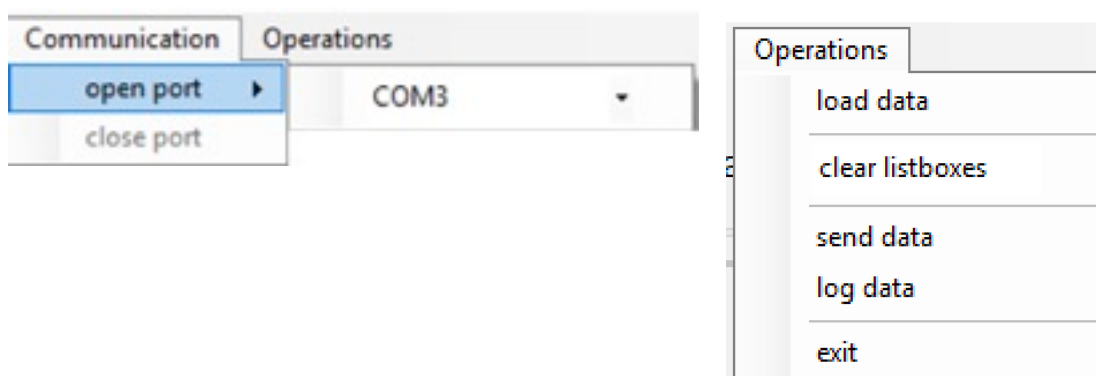
Für das Formular werden eine ListBox für den zu sendenden Text, eine ListBox für den empfangenen Text und eine TrackBar zum Einstellen der Verzögerungszeit, sowie einige Labelfelder benötigt. Im Communication-Menü ist eine ComboBox für die Anzeige der COM-Ports vorzusehen. Das vorliegende Formular enthält die bereits bekannten Elemente, und muss um die neuen Elemente (gelb markiert) ergänzt werden.



Zusätzlich benötigen Sie die gelb markierten Komponenten



In den MenuStrips Communication und Operations befinden sich nachfolgende Einträge



Nach dem Start des Programms sind folgende Einstellungen erforderlich

- Die ComboBox im Menü wird mit der Liste der aktuellen COM-Ports befüllt
- Die TrackBar wird mit einer oberen und unteren Grenze (z.B. 100 und 2000) für die Verzögerungszeit in ms, sowie Werten für das Inkrement initialisiert
- Im LabelFeld neben der TrackBar wird deren aktueller Wert angezeigt
- Im Communications-Menü kann nur der Eintrag für open port ausgewählt werden, close port ist zu Beginn deaktiviert

Funktionalität (Events) des Communication-Menüs

- open port: Die serielle Schnittstelle kann nur geöffnet werden, wenn ein COM-Port aus der ComboBox ausgewählt wurde, andernfalls erscheint eine Fehlermeldung. Ein erfolgreiches Herstellen der Verbindung wird in der Statuszeile angegeben. Der Menü-Eintrag zum Schließen wird aktiv, jener für das Öffnen wird inaktiv
- close port: Eine geöffnete Verbindung wird geschlossen und in der Statuszeile angezeigt. Außerdem sind die beiden Einträge im Communications-Menü wieder entsprechend zu aktivieren bzw. deaktivieren

Funktionalität (Events) des Operations-Menüs

- load data: Der openFileDialog erscheint und der Inhalt der Datei wird in die linke ListBox übernommen
- clear listboxes: Der Inhalt beider ListBoxes wird gelöscht
- send data: Falls sich keine Einträge in der linken ListBox befinden, erscheint eine Fehlermeldung. Ansonsten werden alle Einträge der linken ListBox zeilenweise zum Mikrocontroller geschickt und die Rückmeldungen vom Mikrocontroller werden in der rechten ListBox angezeigt. Zwischen dem Senden der einzelnen Zeilen wird die in der TrackBar eingestellte Zeit (in Millisekunden) eingehalten. Nutzen Sie zum Senden und Empfangen den Timer-Event.
- log data: Der saveFile-Dialog erscheint und der Inhalt der rechten ListBox wird in die angegebene Datei geschrieben. Ist die rechte ListBox leer, erscheint eine Fehlermeldung.
- exit: Bevor die Anwendung beendet wird, soll ein noch geöffneter Port geschlossen werden.

Mikrocontroller-Programm

Das Mikrocontroller Programm benötigt die Bibliothek USART.h der DriverLibHW-Bibliothek, um die serielle Schnittstelle für die Kommunikation mit dem C#-Programm einzurichten.

Zur Initialisierung der Schnittstelle wird Anschluss 0 mit einer Baudrate von 9600 verwendet, das Senden und Empfangen ist aktiviert, beide ohne Verwendung von Interrupts.

Die Kommunikation zwischen Mikrocontroller und C#-Programm erfolgt mit den Funktionen scanf() und printf().

Von C# wird eine Ganzzahl gesendet, die binär am LED-Balken dargestellt wird. Das Mikrocontroller-Programm sendet eine entsprechende Rückmeldung an die C#-Visualisierung (siehe rechte Listbox)

Das µC-Projekt entspricht dem Projekt aus dem Beispiel SimpleCommunication und ist bereits einsatzbereit.