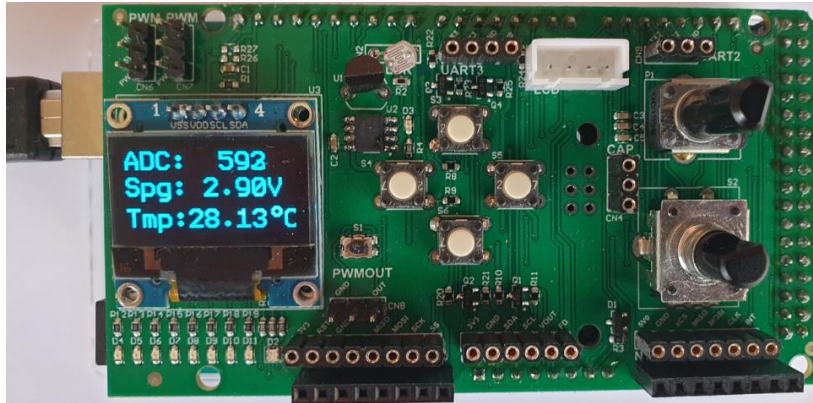


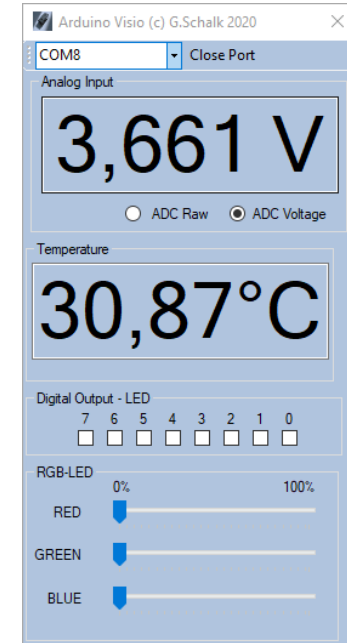
SOFTWARE- ENTWICKLUNG ÜBUNGSBEISPIEL

Übungsbeispiel – Arduino Board
Visualisieren und Steuern über RS-232

Visualisieren und Steuern über RS-232



USB \leftrightarrow RS232



Aufgabe 3: Ansteuern des CAMPUS 02-Boards

Serielle Schnittstelle einbinden und abfragen

Verwendung von Windows Forms für die Darstellung

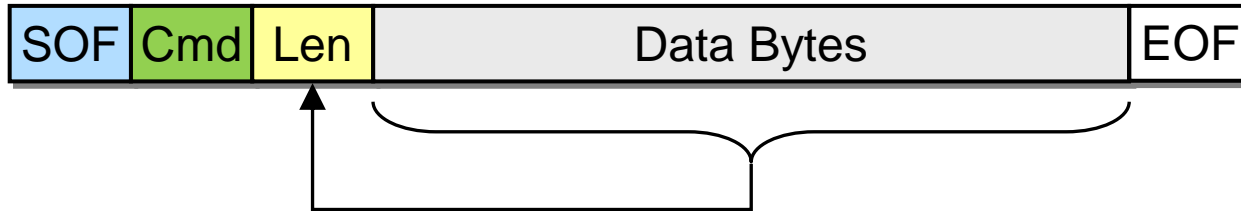
Test mit dem Echtsystem

Optionale Erweiterungen (für Fortgeschrittene):

- Verwendung von WPF (Windows Presentation Foundation) statt Windows Forms
- Implementation weiterer Befehle (Mikrocontroller-Board und PC-Applikation)

Das Übertragungsprotokoll bei RS-232

Prinzipieller Aufbau eines Kommandos



SOF (Start of Frame)	→ Synchronisierungsbyte (Wert 0xA5)
CMD (Command)	→ Kommandos
Len (Length)	→ Anzahl der Nutzdaten (Data Bytes)
Data Bytes	→ Nutzdaten
EOF (End of Frame)	→ Line Feed (Wert 0x0A) für H-Term

Protokolldefinition Arduino → PC

ADC: Analog „Raw“ Value [0 .. 1023]

SOF	Cmd	Len	Data Bytes ^[1]	EOF
-----	-----	-----	---------------------------	-----

0xA5 **0x02** 0x02 0xXX, 0xXX 0x0A

Spg: Analog „Voltage“ Value [0 .. 5000 mV]

SOF	Cmd	Len	Data Bytes ^[1]	EOF
-----	-----	-----	---------------------------	-----

0xA5 **0x04** 0x04 0xXX, 0xXX, 0xXX, 0xXX 0x0A

Tmp: Analog „Temperature“ Value [0.00 .. 100.00 °C]

SOF	Cmd	Len	Data Bytes ^[1]	EOF
-----	-----	-----	---------------------------	-----

0xA5 **0x06** 0x04 0xXX, 0xXX, 0xXX, 0xXX 0x0A



Protokolldefinition Arduino ← PC

Digital Output (8 x LED)

SOF	Cmd	Len	Data Byte
-----	-----	-----	-----------

0xA5 **0x10** 0x01 0xFF

RGB-LED

SOF	Cmd	Len	R	G	B
-----	-----	-----	---	---	---

0xA5 **0x12** 0x03 0xFF 0xFF 0xFF

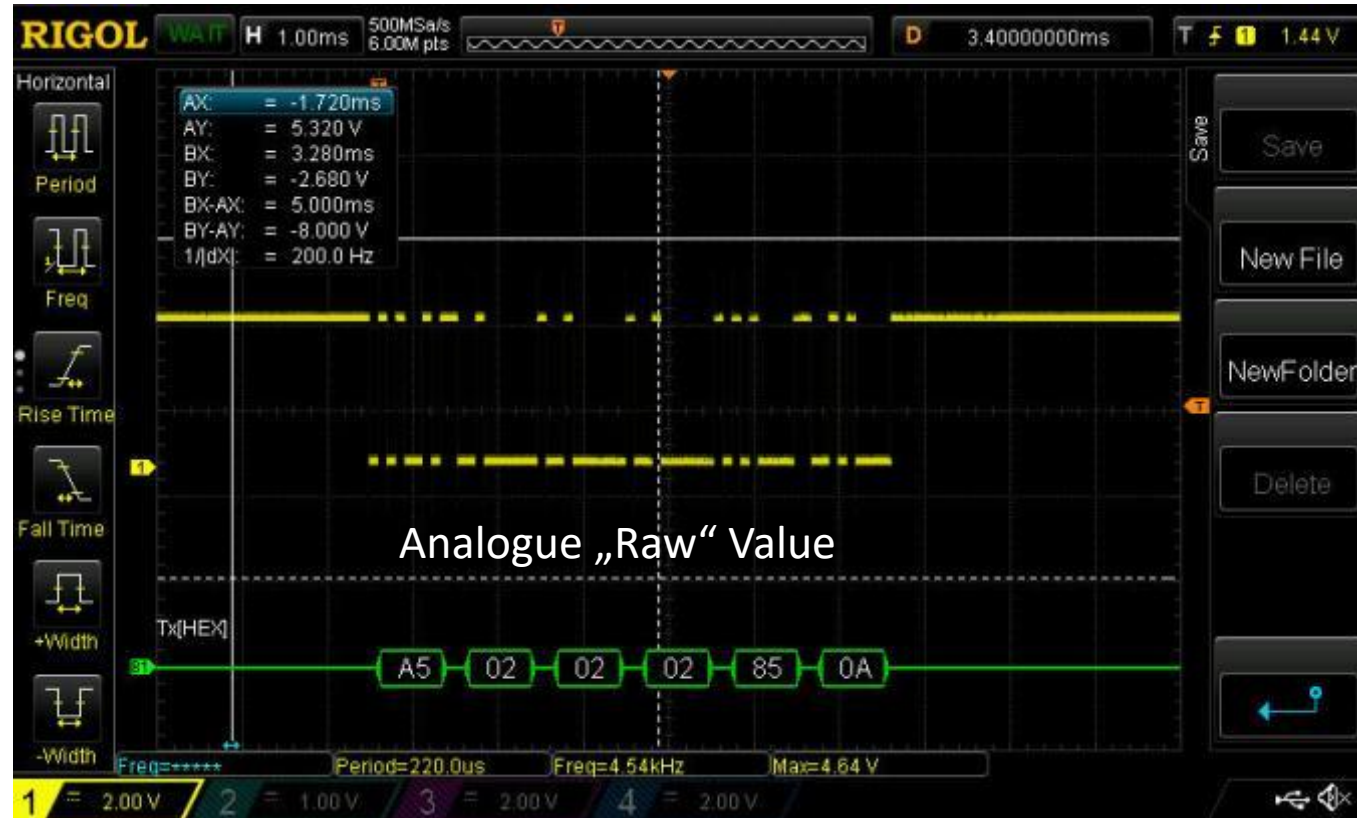
0xFF .. 0..255



8 x LED

RGB - LED

Analoger Roh-Wert am Oszilloskop

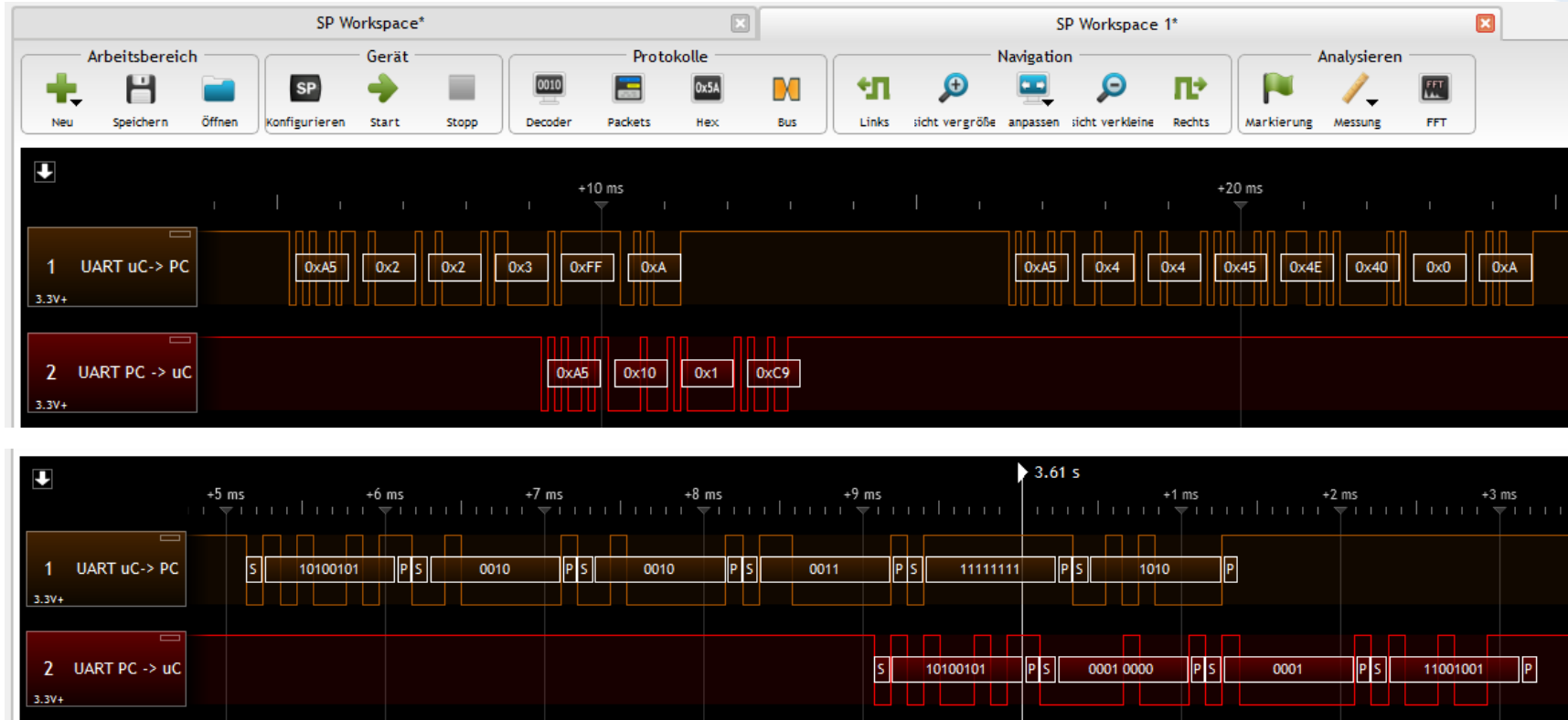


Analoger Roh-Wert am Oszilloskop – Zoom

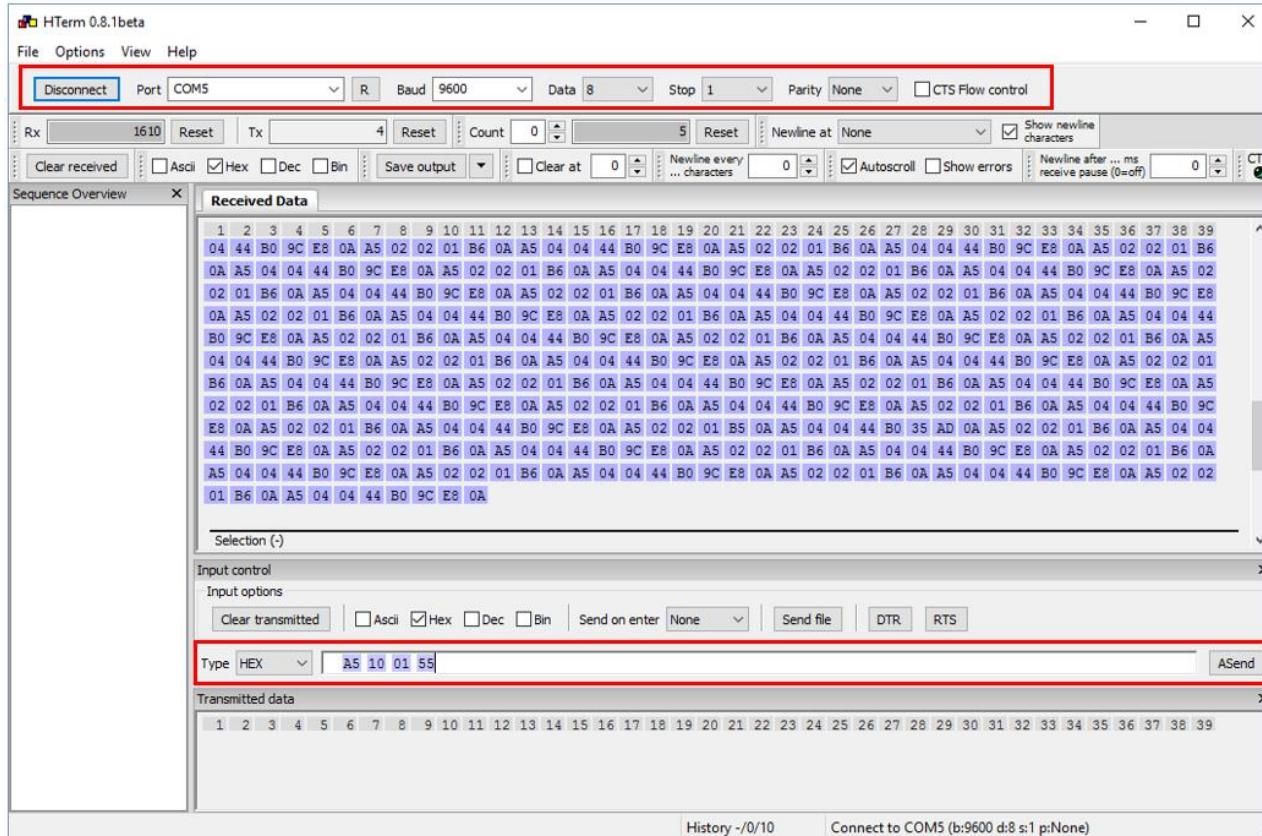


Analogue „Raw“ Value - Zoomed

Betrachtung mit dem Logik-Analysator



Terminal-Programm Hterm (I)



Terminal-Programm Hterm (II)

 HTerm 0.8.2

File Options View Help

[illegible]

<http://www.h->

schmidt.net/FloatConverter/IEEE754de.html

Serial Port-Klasse in .NET - Beispiel (I)

```
using System;
using System.IO.Ports; ← Namespace
using System.Diagnostics;
```

```
namespace SerialPortConsole
```

```
{
```

```
class Program
```

```
{
```

```
static SerialPort serialPort1;
```

```
static void Main(string[] args)
```

```
{
```

```
serialPort1 = new SerialPort();
```

```
// Configure serial port
```

```
serialPort1.PortName = "COM8";
```

```
serialPort1.BaudRate = 9600;
```

```
serialPort1.DataBits = 8;
```

```
serialPort1.Parity = Parity.None;
```

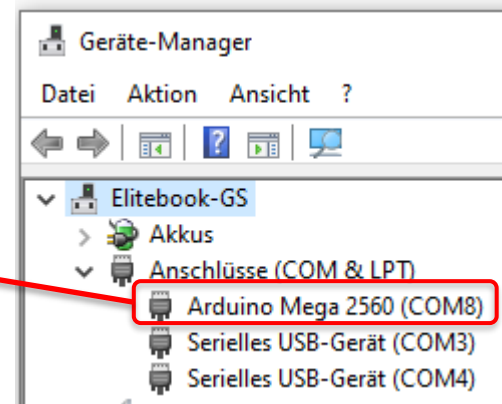
```
serialPort1.StopBits = StopBits.One;
```

```
serialPort1.ReadBufferSize = 30;
```

```
serialPort1.BytesToReadThreshold = 1;
```

statisches Feld

SerialPort-Objekt mit
Standardeinstellungen erzeugen



[https://msdn.microsoft.com/de-de/library/system.io.ports.serialport\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/system.io.ports.serialport(v=vs.110).aspx)

Serial Port-Klasse in .NET - Beispiel (II)

```
//Adds a SerialDataReceivedEventHandler to DataReceived to read
// all the available data received on the serial port.
serialPort1.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);
try{
    serialPort1.Open(); // Open the serial port connection
    while (Console.KeyAvailable == false)
    {
        SendLedCmd(0x5A);
        Thread.Sleep(500);
        SendLedCmd(0xA5);
        Thread.Sleep(500);
    }
    serialPort1.Close();
}
catch (Exception e)
{
    Console.WriteLine("ERROR: " + e.Message);
    return;
}
}
```

Serial Port-Klasse in .NET - Beispiel (III)

```
static void SendLedCmd(byte ledValue)
{
    if (serialPort1.IsOpen)
    {
        // 0      1          2          3
        //-----
        // SOH | Cmd | Len | Data[0] .. Data[Len-1] |
        //-----

        // ToDo ...

        serialPort1.Write( ...
    }
}
```

Daten - Empfangen

A5

020203FF

0A

A5040445

9C40000A

A5060441F6

00000A

A50202

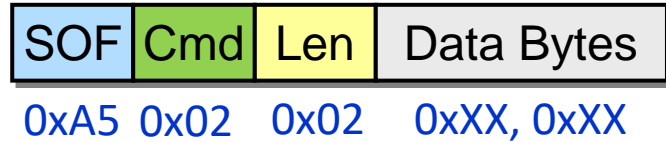
03FF0A

A50404459C40

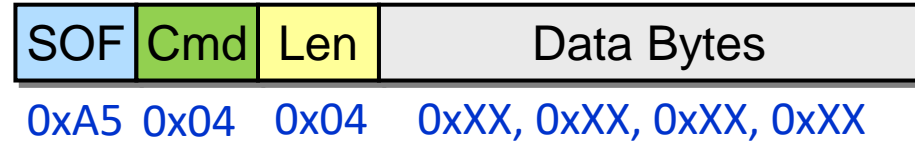
000AA5060441F600000A

Receive - State Machine (PC-Applikation)

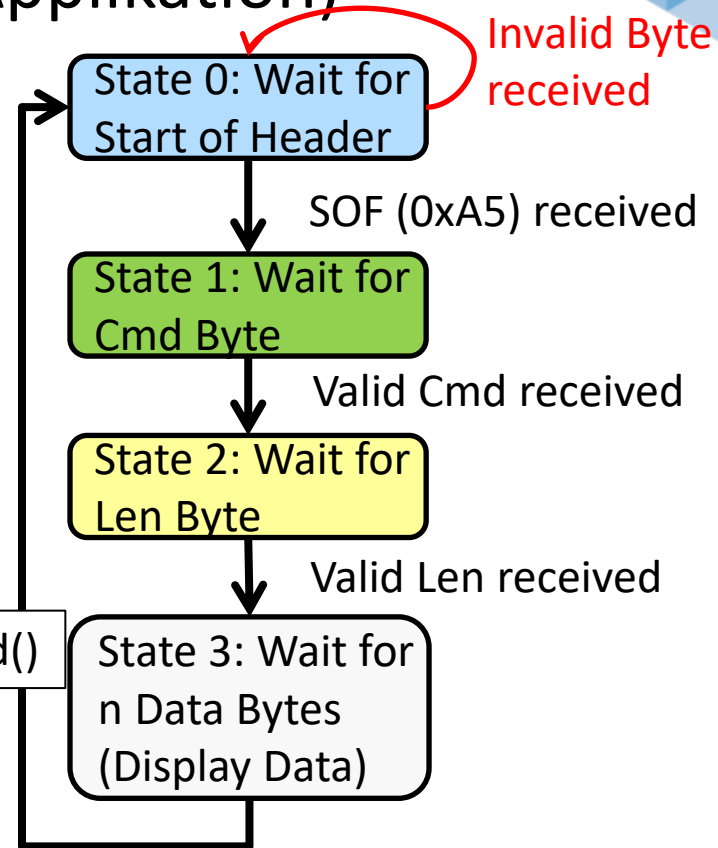
ADC: Analog „Raw“ Value [0 .. 1023]



Spg: Analog „Voltage“ Value [0 .. 5000 mV]



ExecuteRecSerialCommand()



Serial Port-Klasse in .NET - Beispiel (VI)

```
private static void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
{
    // Any serial data received...
    // Gets the number of bytes of data in the receive buffer.
    int recCount = serialPort1.BytesToRead;

    // read all received bytes
    byte[] baActRecData = new byte[recCount];

    serialPort1.Read(baActRecData, 0, recCount);

    // ToDo
```

Steuerelemente in Windows Forms

Die Steuerelemente eines Windows Forms- bzw. WPF-Programms (Eigenschaften und Methoden) sind nicht threadsicher. Erfolgt der Zugriff auf eine Windows Forms bzw. WPF-Komponente aus einem anderen Thread als dem Anwendungs-Thread, so wird eine Invalid Operation Exception ausgelöst.

Lösung: Verwendung der Invoke Methode:

```
textBox1.Invoke(new EventHandler(delegate  
{  
    textBox1.Text = "Hello World";  
}  
));
```

[https://msdn.microsoft.com/de-de/library/ms171728\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/ms171728(v=vs.110).aspx)