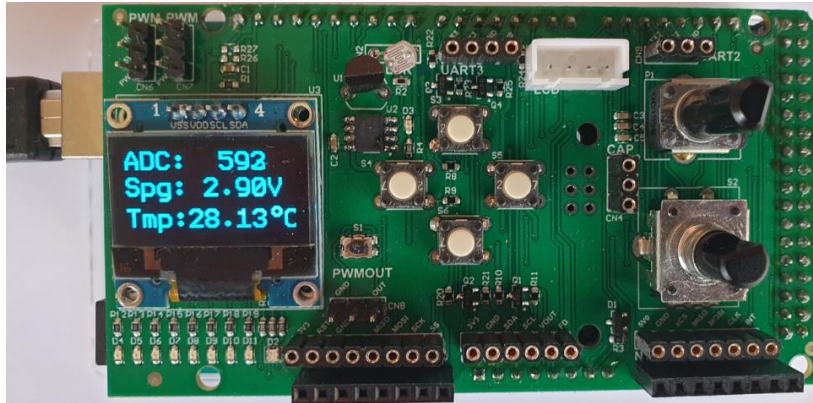


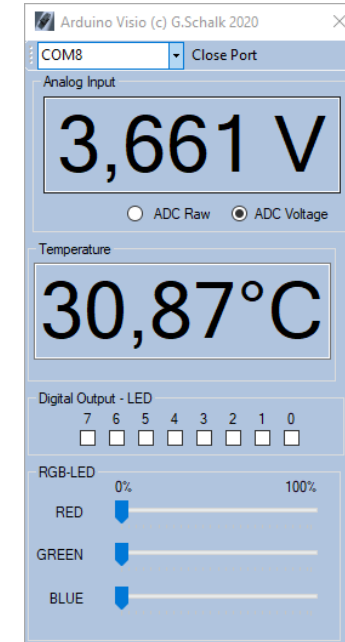
# SOFTWARE- ENTWICKLUNG ÜBUNGSBEISPIEL

Übungsbeispiel – Arduino Board  
Visualisieren und Steuern über RS-232

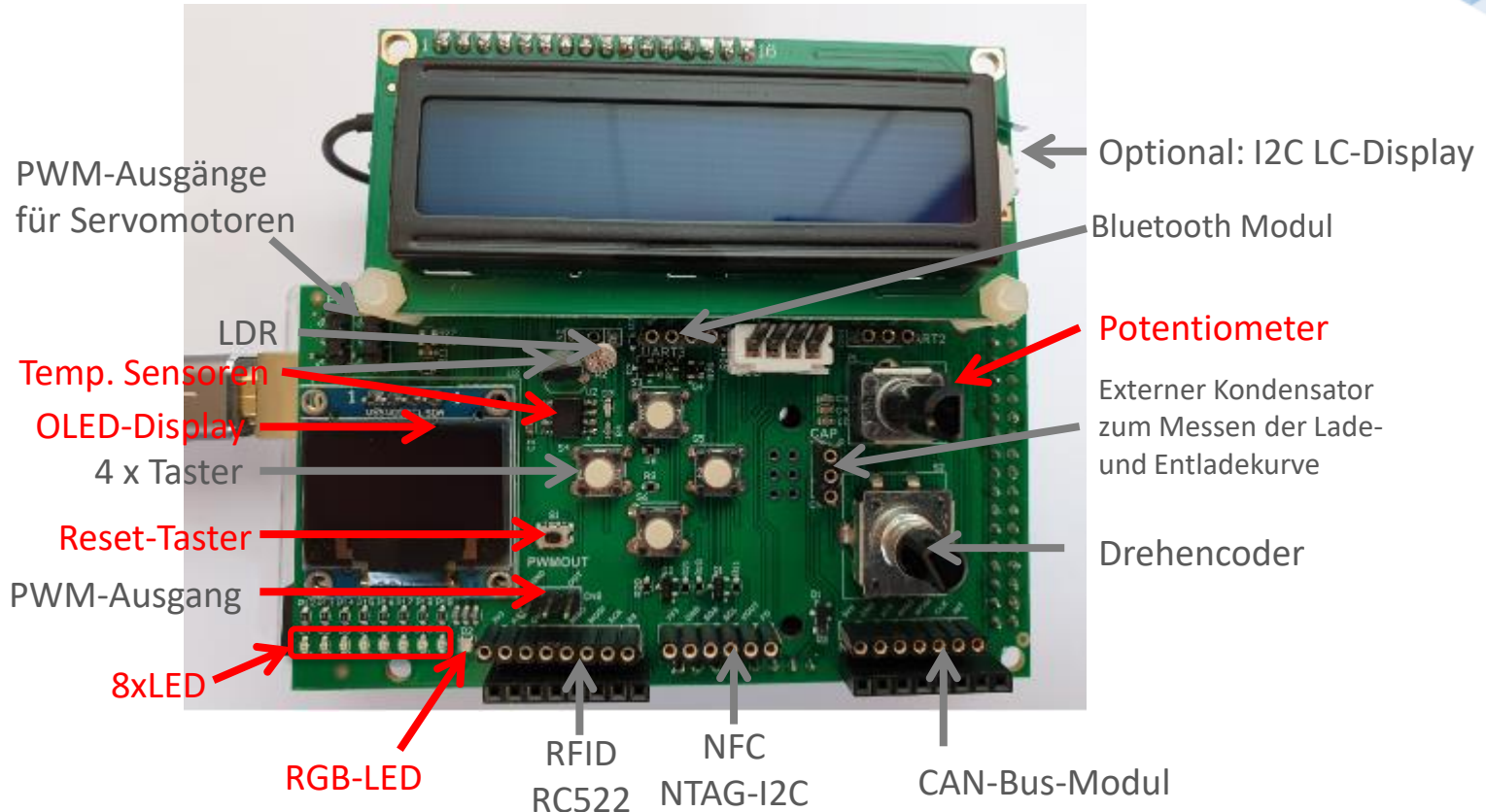
# Visualisieren und Steuern über RS-232



USB  $\leftrightarrow$  RS232

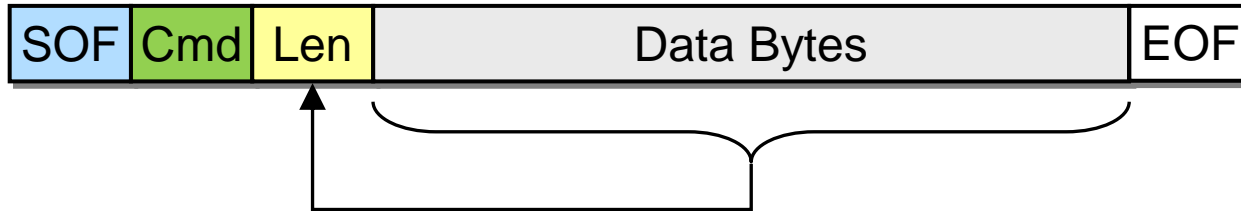


# Campus 02 Arduino Shield



# 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 <sup>[1]</sup>	EOF
-----	-----	-----	---------------------------	-----

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

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

SOF	Cmd	Len	Data Bytes <sup>[1]</sup>	EOF
-----	-----	-----	---------------------------	-----

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

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

SOF	Cmd	Len	Data Bytes <sup>[1]</sup>	EOF
-----	-----	-----	---------------------------	-----

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



# Protokolldefinition Arduino ← PC

Digital Output (8 x LED)

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

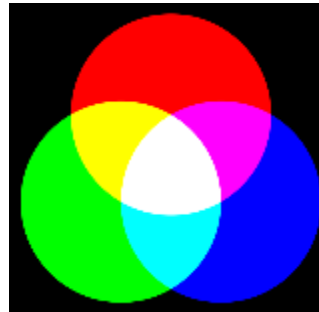
0xA5 **0x10** 0x01 LED value

RGB-LED

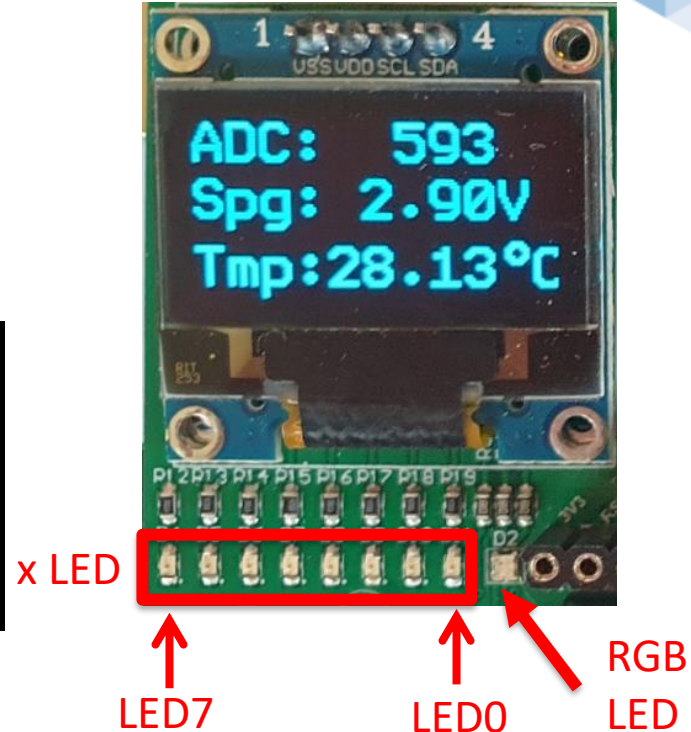
SOF	Cmd	Len	R	G	B
-----	-----	-----	---	---	---

0xA5 **0x12** 0x03 0xXX 0xXX 0xXX

RGB Farbenmischer

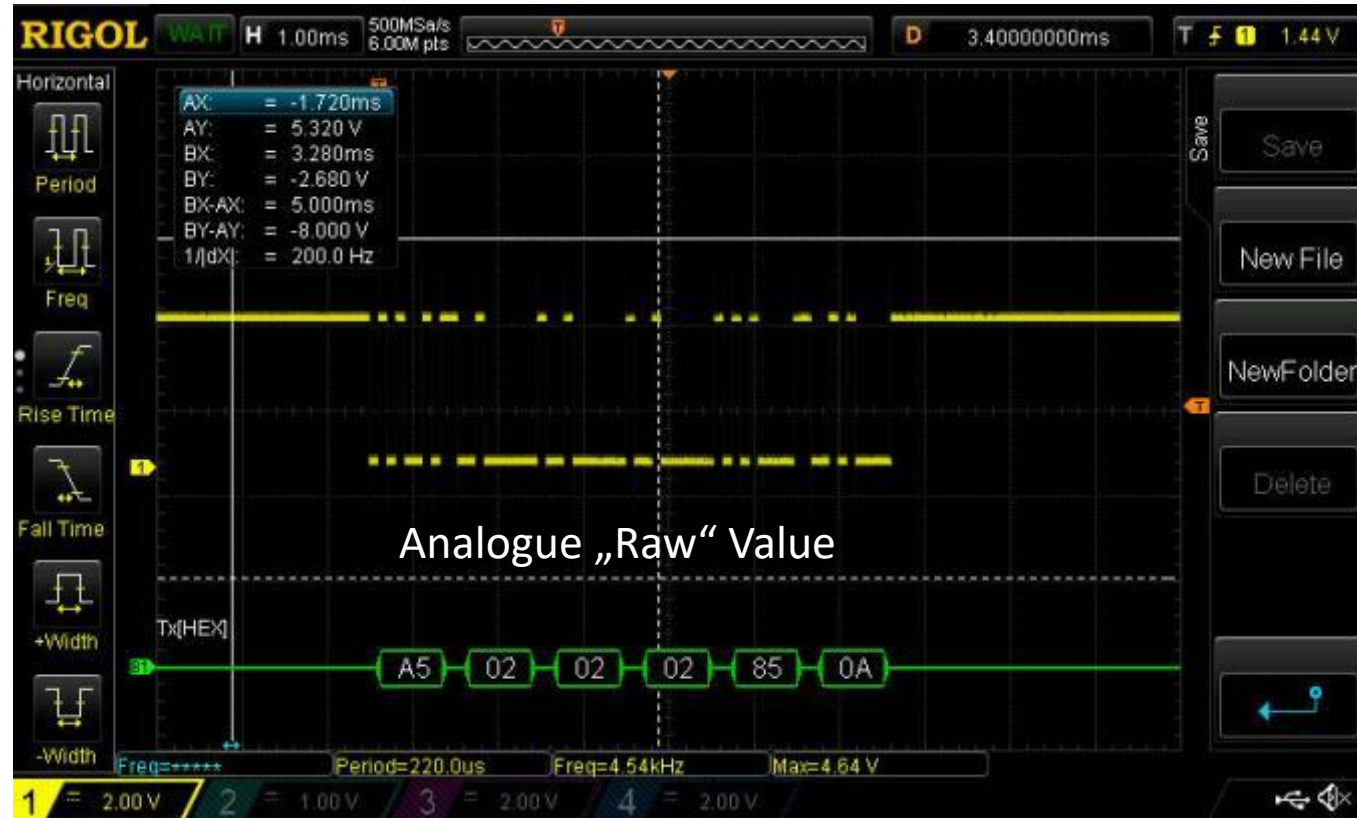


0xXX .. 0..255 entspricht 0 .. 100%





# Analoger Roh-Wert am Oszilloskop

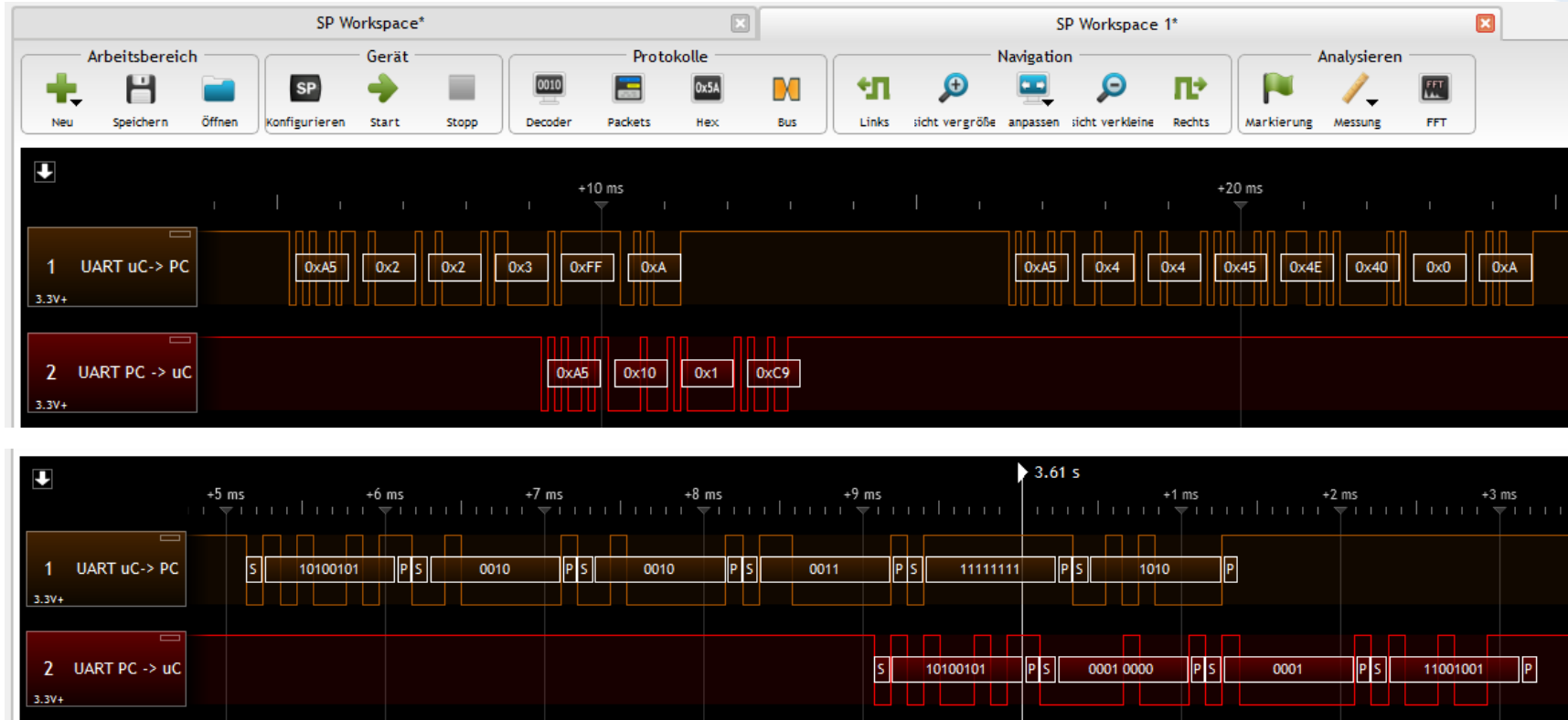


# Analoger Roh-Wert am Oszilloskop – Zoom

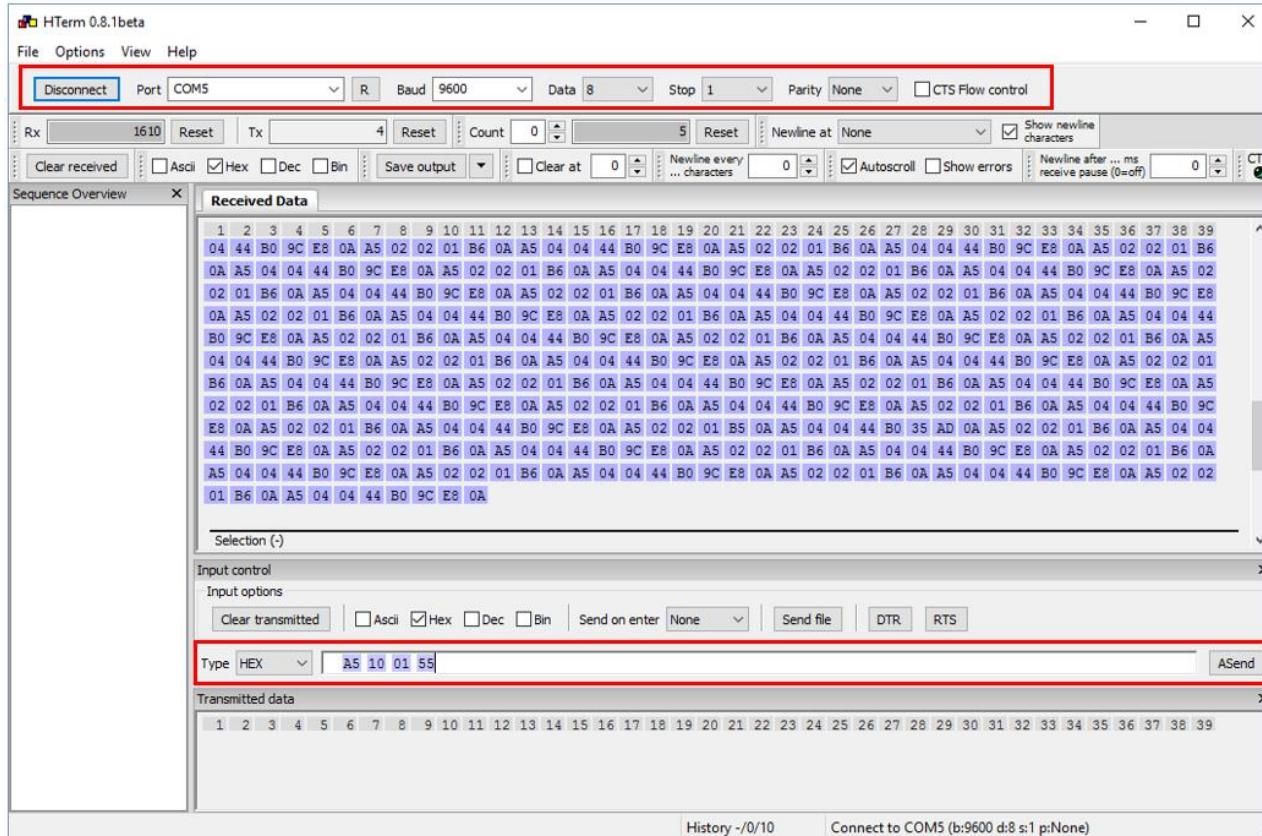




# Betrachtung mit dem Logik-Analysator



# 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](http://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;
```

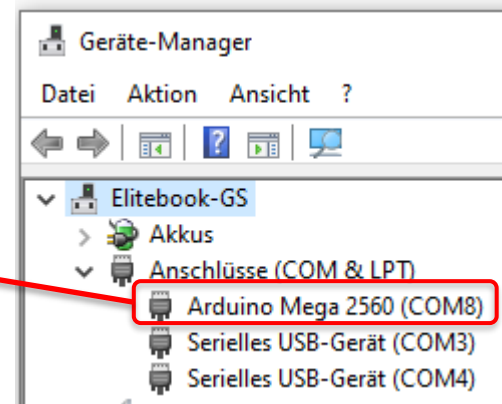
```
serialPort1.StopBits = StopBits.One;
```

```
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 += SerialPort1_DataReceived;
try{
    serialPort1.Open(); // Open the serial port connection
    while (Console.KeyAvailable == false)
    {
        SendLedCmd(0x55);
        Thread.Sleep(500);
        SendLedCmd(0xAA);
        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( ...
    }
}
```



# Beispiel der empfangen Daten

A5

020203FF

0A

A5040445

9C40000A

A5060441F6

00000A

A50202

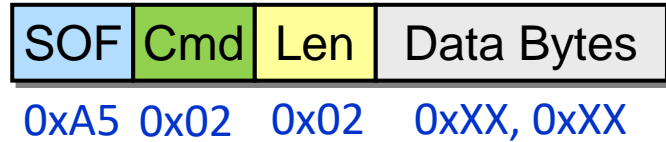
03FF0A

A50404459C40

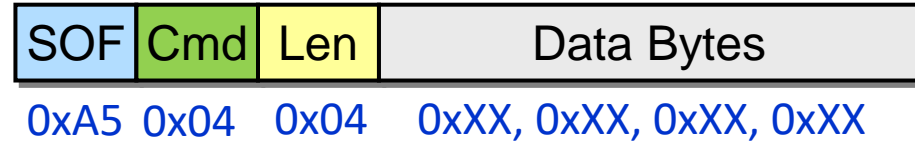
000AA5060441F600000A

# Receive - State Machine (PC-Applikation)

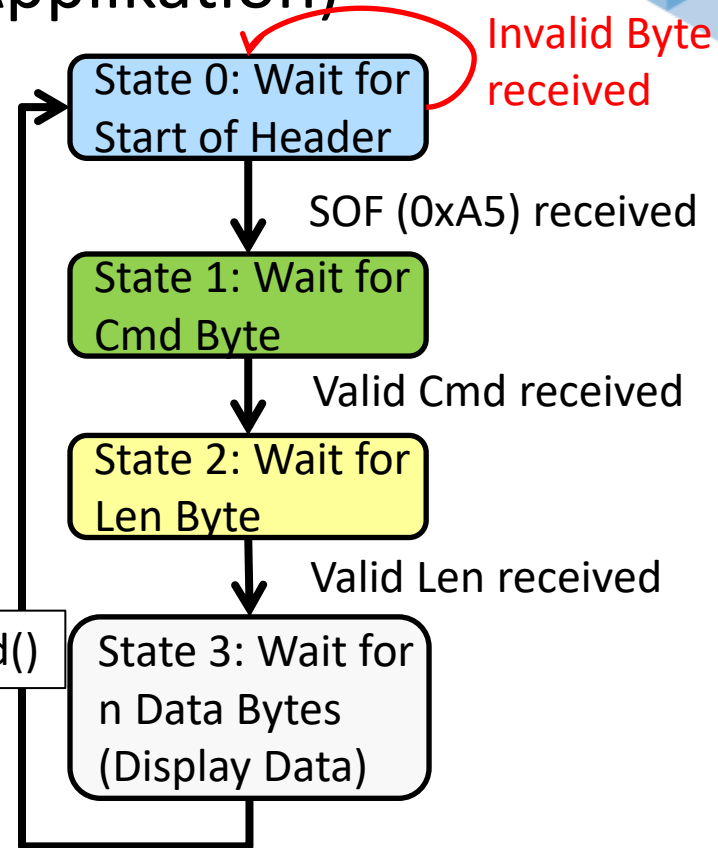
**ADC:** Analog „Raw“ Value [0 .. 1023]



**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)

# Aufgabe : Ansteuern des CAMPUS 02-Boards

## Konsolen und Windows Forms Anwendung

8-LEDs via serieller Schnittstelle steuern

RGB-LED Farben via serieller Schnittstelle „mischen“

Daten via serieller Schnittstelle empfangen (Analogwert und Temperatur)

Anzeige des aktuellen Analogwerts

Anzeige des aktuellen Temperatur Messwertes

Anzeige des minimalen und des maximalen Temperaturwertes

Temperaturwerte periodisch in einer CSV-Datei abspeichern (siehe nächste Seite)

# Datei - Ausgabeformat

TempValues.csv - Editor

Datei Bearbeiten Format Ansicht Hilfe

Date,Time,Temp.

22.06.2022,20:58:15,32.75

22.06.2022,20:58:15,32.75

22.06.2022,20:58:15,32.75

22.06.2022,20:58:15,32.63

22.06.2022,20:58:15,32.63

22.06.2022,20:58:15,32.75

22.06.2022,20:58:15,32.75

22.06.2022,20:58:15,32.75

22.06.2022,20:58:16,32.75

22.06.2022,20:58:17,32.75

22.06.2022,20:58:18,32.63

22.06.2022,20:58:19,32.63

22.06.2022,20:58:20,32.63



AutoSave Off

File Home Insert Draw Page Layout Formulas

Clipboard Font

D28

	A	B	C
1	Date	Time	Temp.
2	22.06.2022	20:58:15	32.75
3	22.06.2022	20:58:15	32.75
4	22.06.2022	20:58:15	32.75
5	22.06.2022	20:58:15	32.63
6	22.06.2022	20:58:15	32.63
7	22.06.2022	20:58:15	32.75
8	22.06.2022	20:58:15	32.75
9	22.06.2022	20:58:15	32.75
10	22.06.2022	20:58:15	32.75
11	22.06.2022	20:58:16	32.75
12	22.06.2022	20:58:17	32.75
13	22.06.2022	20:58:18	32.63
14	22.06.2022	20:58:19	32.63