

Daten & Information

...oder wie kann der Computer überhaupt
Daten speichern?

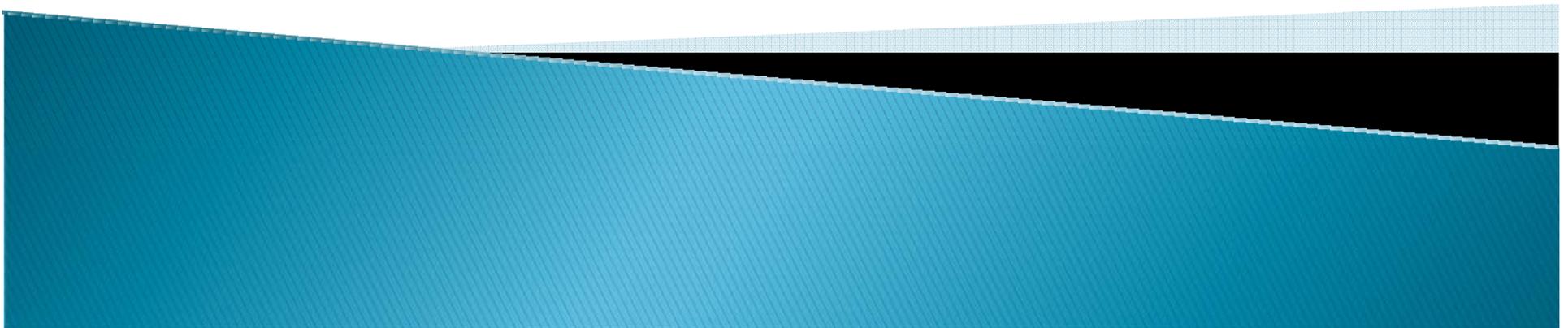
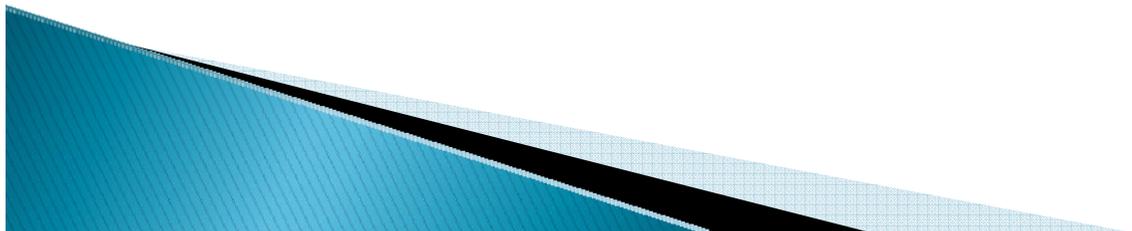


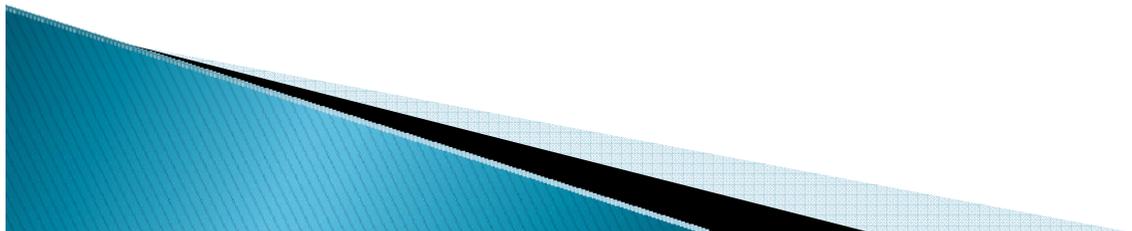
Bild ist nicht gleich Bild

- ▶ Im Tausch Ordner (Tausch-Schule/InfoBauer) findet ihr 2 Dateien.
- ▶ Öffnet diese mit einem entsprechenden Programm
- ▶ Beschreibt zu zweit die Unterschiede hinsichtlich Darstellung und Speicherung



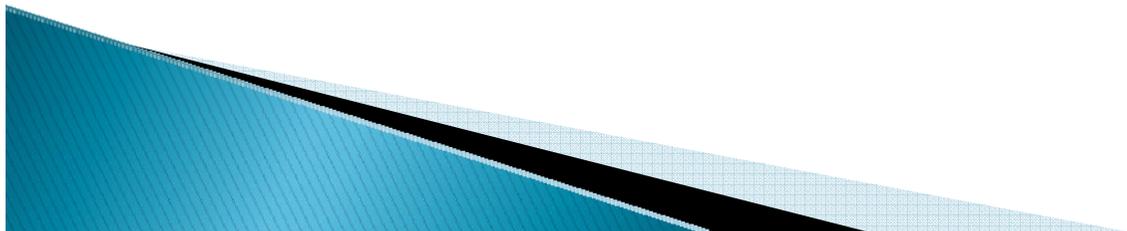
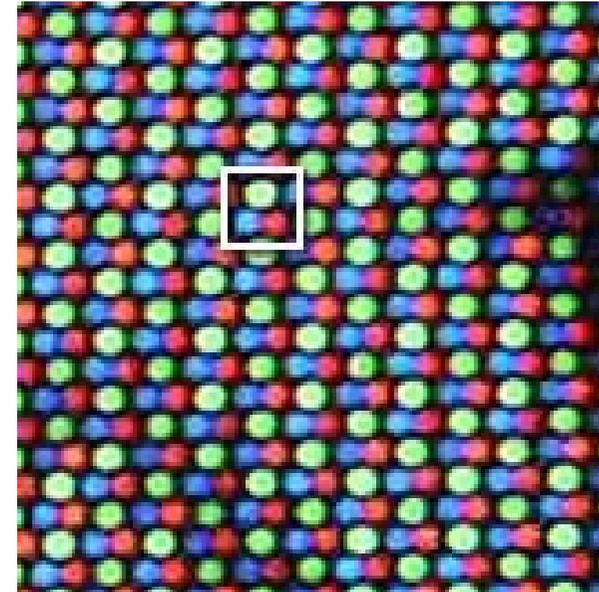
Grafik (Rastergraphik)

- ▶ Bilder im Computer bestehen aus einzelnen Pixeln (= Picture Element, „Bildpunkt“)
- ▶ Ein Pixel ist auf einem Monitor etwa $1/3$ mm groß
- ▶ Mit einer Lupe kann man die einzelnen Pixel sehen



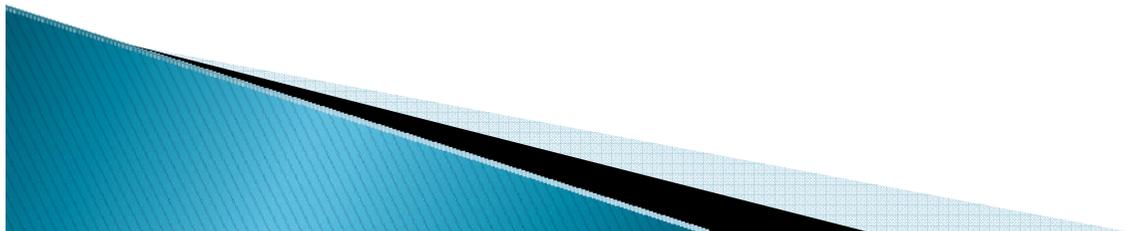
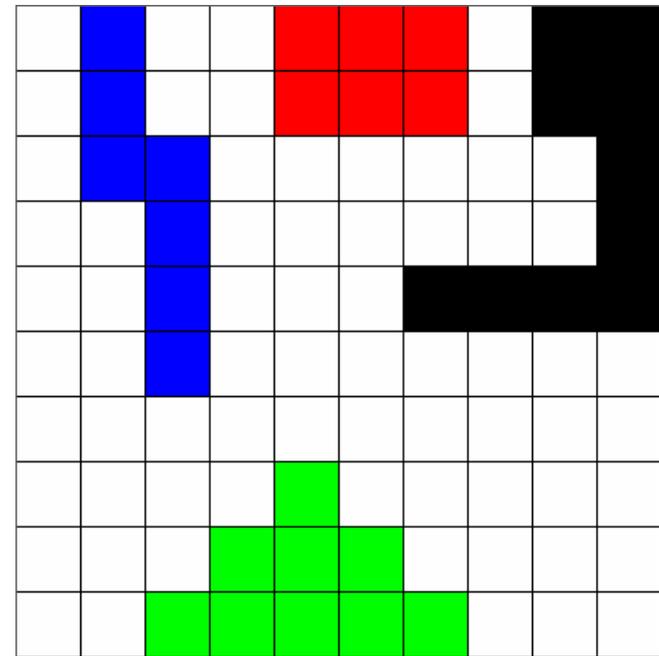
Grafik

- ▶ Wie entstehen Farben?
- ▶ Jedes Pixel besteht aus roten, grünen und blauen Subpixel
- ▶ Die Farbe eines Pixels ergibt sich dann aus den jeweiligen Farbanteilen der Subpixel
- ▶ Wie viele verschiedene Farben sind wohl möglich?
 - $256^3 = 16,7$ Millionen Farben

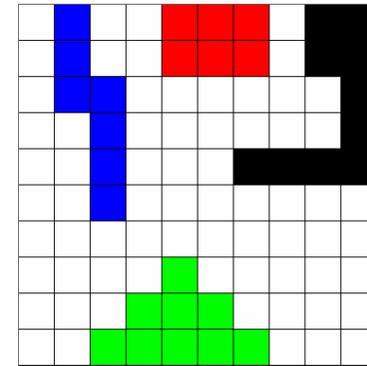


Grafik

- ▶ Um ein Bild als Datei abspeichern zu können, muss die Farbe jedes Pixels auf irgendeine Weise gespeichert werden.
- ▶ Wie könnte dieses Bild wohl so kurz wie möglich beschrieben werden?

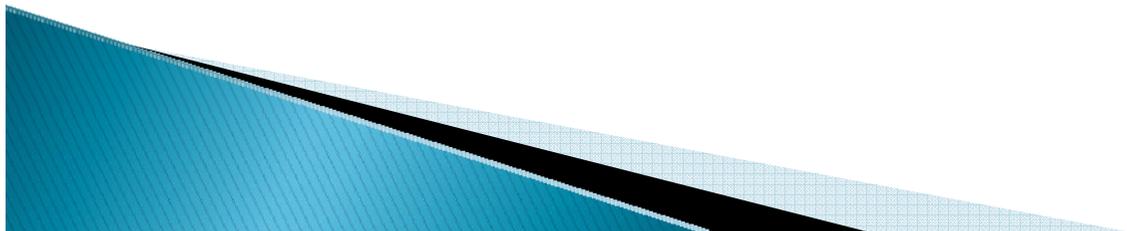


Grafik



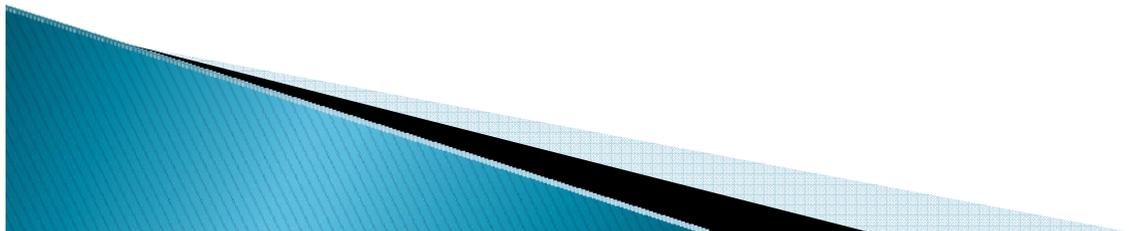
▶ Beispiele:

- 10*10 Pixel
1 weiß, 1 blau, 2 weiß, 3 rot, 1 weiß, 2 schwarz,
1 weiß, 1 blau, 2 weiß, 3 rot, 1 weiß, 2 schwarz,
1 weiß, 2 blau, 6 weiß, 1 schwarz, ...
- 10*10 Pixel, Hintergrund weiß
blau: (1,2),(2,2),(3,2),(3,3),(4,3),...
rot: (1,5),(1,6),(1,7), (2,5),(2,6),(2,7)
grün: (8,5),(9,4),(9,5),(9,6)
- Verbesserung des letzten Systems



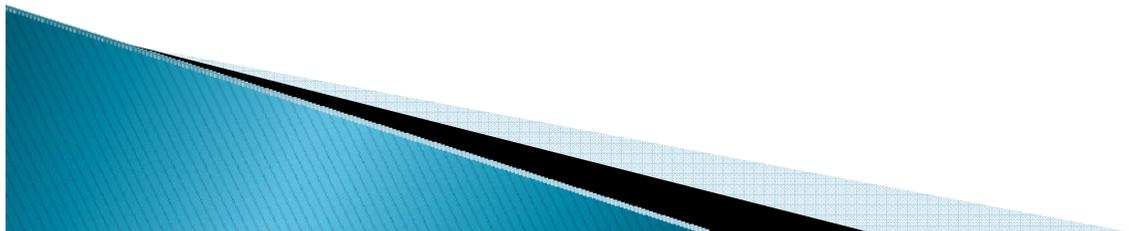
Grafikformate

- ▶ Im Ordner Tausch-Schule\InfoBauer liegt ein Programm namens `ImageEdit.exe`, erstelle damit ein Bild aus 8x8 Pixeln in 2 Farben (monochrom)
- ▶ Speichere das Bild unter dem Namen „`monochrom.bmp`“



Grafikformate

- ▶ Dieses Bild soll auf Papier an deinen Nachbarn übermittelt werden. Dazu brauchen wir zwei Teile:
- ▶ Teil 1: Eine Reihe aus Bits (0 und 1), die alle Informationen des Bildes enthält. (Zeilenumbrüche dürfen nicht verwendet werden)
- ▶ Teil 2: Eine genaue umgangssprachliche Beschreibung, wie aus der Bitreihe das Bild rekonstruiert werden kann.

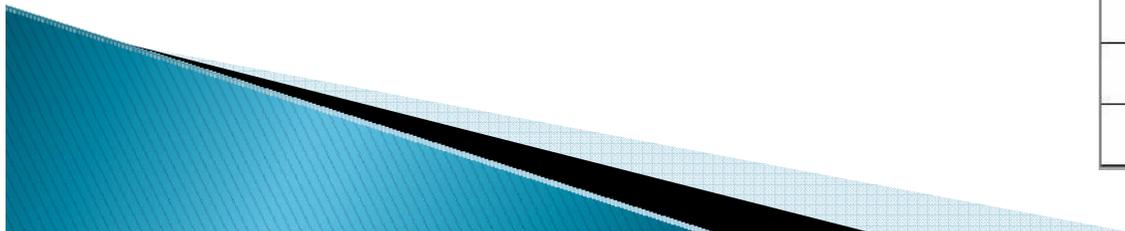
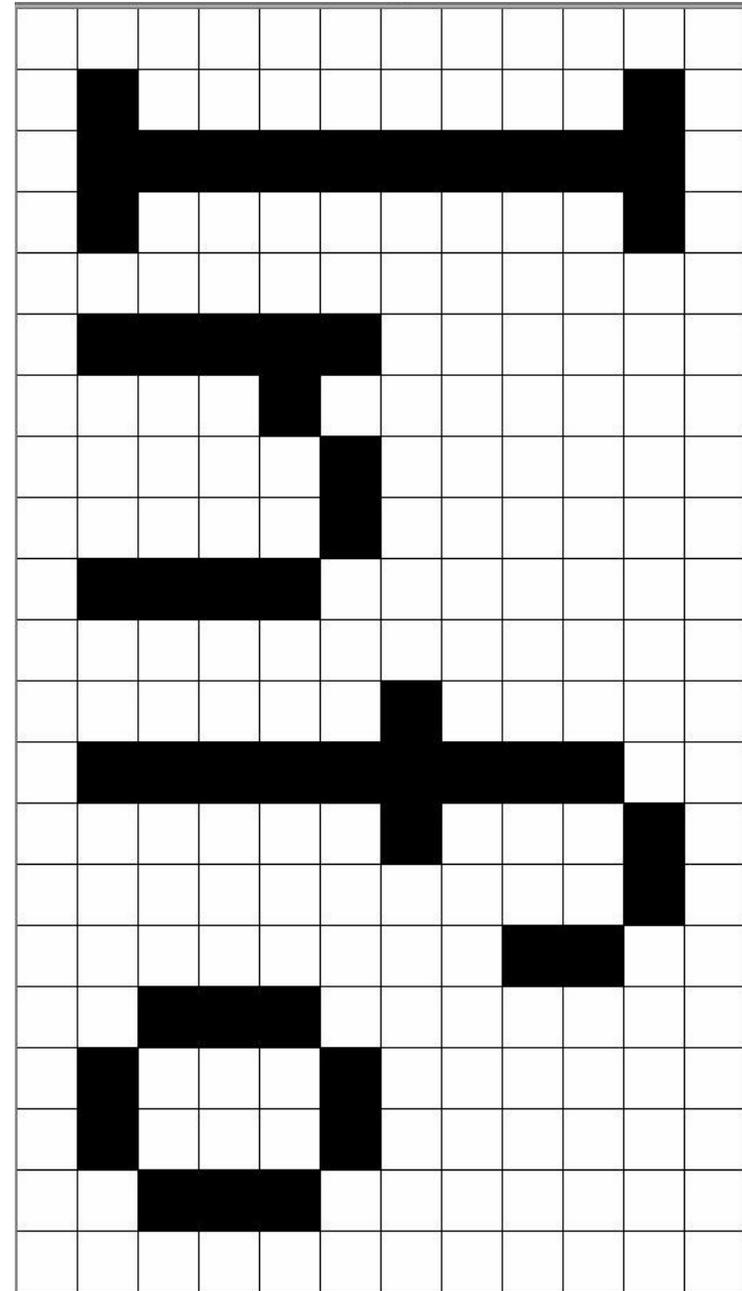


Grafikformate

- ▶ Für Profis: Wie sieht das Bild aus, das durch diese Byteschlange codiert wird?
- ▶ 0C 15 00 0D 01 08 01 02 0A 02 01 08 01
0E 05 0A 01 0C 01 0B 01 07 04 19 01 06
09 08 01 03 01 0B 01 09 02 04 03 08 01
03 01 07 01 03 01 08 03 13
- ▶ Das erste Byte codiert die Breite, das zweite Byte die Höhe des Bildes
- ▶ Wir speichern pro Byte die Anzahl der Pixel in einer Farbe
- ▶ Schwarz ist die erste Farbe, dann weiß, schwarz



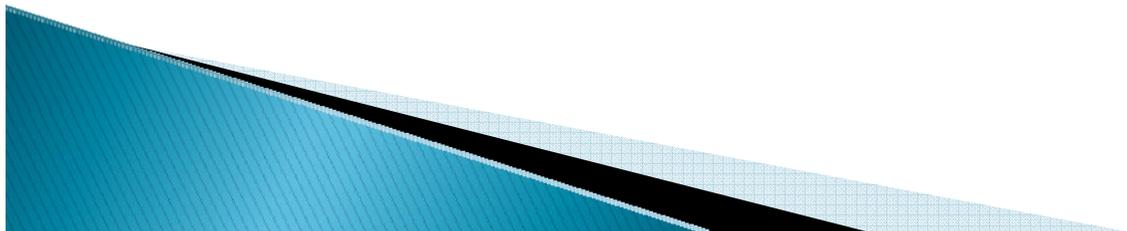
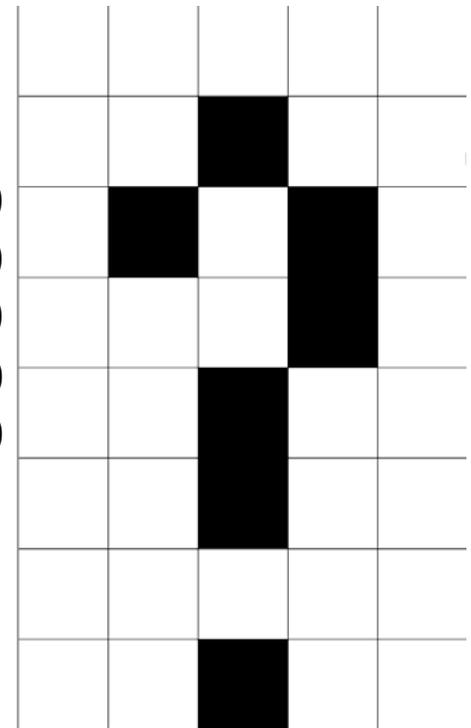
Grafikformate



Grafikformate: BMP

- ▶ Wie wird ein BMP-Bild abgespeichert?
- ▶ HexEdit liefert für das Bild rechts:

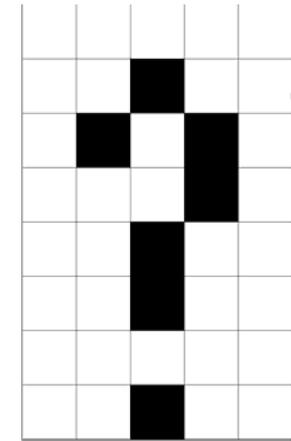
```
42 4D 5E 00 00 00 00 00 00 00 00 3E 00 00 00 28 00
00 00 05 00 00 00 08 00 00 00 01 00 01 00 00 00
00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 FF FF FF 00 D8 00
00 00 F8 00 00 00 D8 00 00 00 D8 00 00 00 E8 00
00 00 A8 00 00 00 D8 00 00 00 F8 00 00 00
```



Grafikformat: BMP

```

42 4D 5E 00 00 00 00 00 00 00 3E 00 00 00 28 00
00 00 05 00 00 00 08 00 00 00 01 00 01 00 00 00
00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 FF FF FF 00 D8 00
00 00 F8 00 00 00 D8 00 00 00 D8 00 00 00 E8 00
00 00 A8 00 00 00 D8 00 00 00 F8 00 00 00
  
```

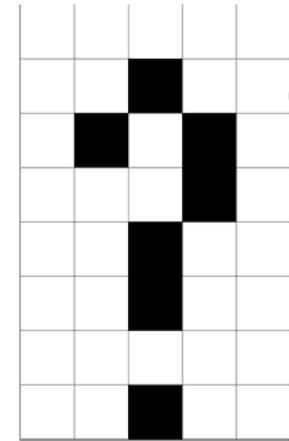


- ▶ 42 4D = „BM“: Magic Byte für BMP-Format
- ▶ 5E_h = 94: Länge der Datei
- ▶ 3E_h = 62: Position, an der die Bildinfos beginnen
- ▶ 05_h: Breite des Bildes
- ▶ 08_h: Länge des Bildes
- ▶ 01_h: Anzahl der Bits pro Pixel
- ▶ 00 00 00_h und FF FF FF_h: schwarz und weiß, die beiden Farben (BGR wegen Little Endian)



Grafikformat: BMP

```
42 4D 5E 00 00 00 00 00 00 00 00 3E 00 00 00 28 00
00 00 05 00 00 00 08 00 00 00 01 00 01 00 00 00
00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 FF FF FF 00 D8 00
00 00 F8 00 00 00 D8 00 00 00 D8 00 00 00 E8 00
00 00 A8 00 00 00 D8 00 00 00 F8 00 00 00
```



- ▶ Bilddaten: Zeilenweise von unten nach oben
- ▶ Jedes Bit entspricht einem Bildpunkt
 - Bitwert 0 → Farbe 1 (also schwarz)
 - Bitwert 1 → Farbe 2 (also weiß)
- ▶ Jede Zeile wird mit 0-Bits auf ein Vielfaches von 4 Byte aufgefüllt
- ▶ Unterste Zeile:
D8 00 00 = 1101 1000 ... = Farbe 2, Farbe 2,
Farbe 1, Farbe 2, Farbe 2

Grafikformate: BMP

- ▶ 42 4D 5E 00 00 00 00 00 00 00 3E 00 00 00 28 00
00 00 05 00 00 00 08 00 00 00 01 00 01 00 00 00
00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 | 00 00 00 00 FF FF FF 00 | D8 00
00 00 F8 00 00 00 D8 00 00 00 D8 00 00 00 E8 00
00 00 A8 00 00 00 D8 00 00 00 F8 00 00 00 |
- ▶ Welche Bytes muss man ersetzen, um folgendes Bild zu erhalten?

