

Auslesen einer Bitmap-Ressource: 16bpp BGR565

1	Grundlagen	2
1.1	Vorbemerkungen.....	2
1.2	Die Farbinformationen	2
1.3	Gewinnen der Farbinformationen.....	2
1.3.1	Farbinformationen Blau	2
1.3.2	Farbinformation Rot.....	3
1.3.3	Farbinformation Grün	3
2	Beispiel	4
2.1	Ausgangsdatei und Konvertierung	4
2.2	Die Binärdatei.....	4
2.3	Bemerkung	5
2.4	Die Farbinformationen der vier Beispielfarben.....	5
2.4.1	Der Wert 0x 00 F8.....	5
2.4.2	Der Wert 0x FF FF	6
2.4.3	Der Wert 0x 00 00	7
2.4.4	Der Wert 0x E0 07.....	8

1 Grundlagen

[\(Inhalt\)](#)

1.1 Vorbemerkungen

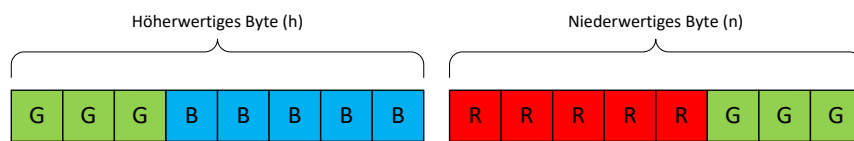
[\(Thema\)](#)

Bei der Arbeit mit Mikrocontrollern steht nur ein relativ geringer Speicherplatz zur Verfügung. Deshalb kommen bei der Anzeige von Bitmaps unter Verwendung von Displays oft nur Grafiken mit einer geringen Farbtiefe zum Einsatz. Ein mögliches Format ist High Color mit einer Farbtiefe von 16 Bit (**16bpp BGR565**). Dabei erfolgt die Farbcodierung für jeden Bildpunkt (Pixel) in 2 Bytes.

1.2 Die Farbinformationen

[\(Thema\)](#)

Die Bits (0 oder 1) der Farbinformationen haben folgende Bedeutung.



Wert Blau	Wert Grün	Wert Rot
5 Bits in h	3 Bits in h (niederwertige Information) 3 Bits in n (höherwertige Information)	5 Bits in n

Daraus ergeben sich $2^{16} = 65536$ mögliche Farben.

1.3 Gewinnen der Farbinformationen

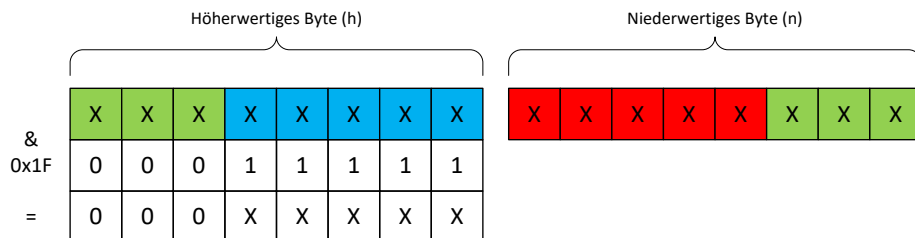
[\(Thema\)](#)

Das Gewinnen der Farbinformationen der drei Grundfarben erfolgt über Bitmaskierung und Bitverschiebung.

1.3.1 Farbinformationen Blau

[\(Thema\)](#)

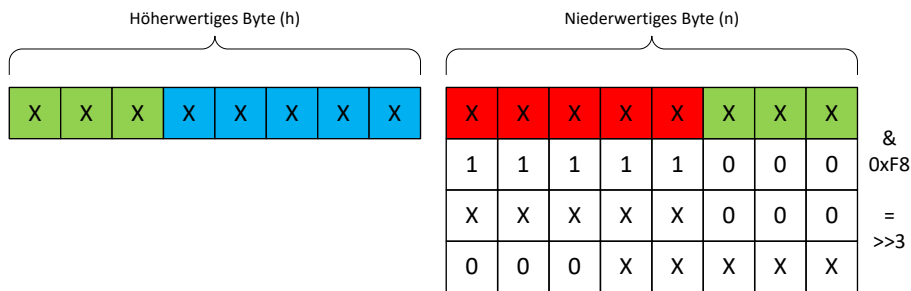
Dazu wird das höherwertige Byte h mit 0x1F maskiert



1.3.2 Farbinformation Rot

[\(Thema\)](#)

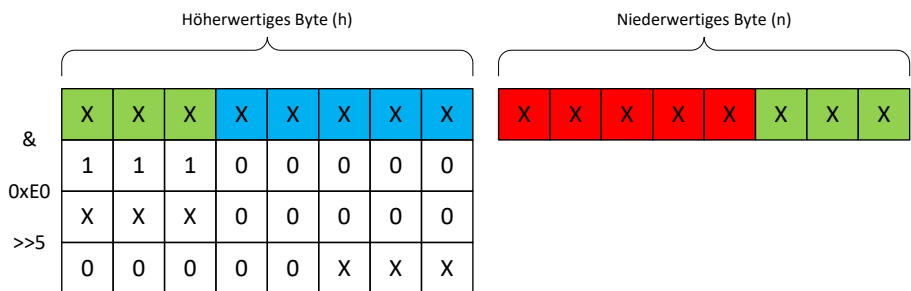
Dazu wird das niederwertige Byte n mit 0xF8 maskiert und das Ergebnis anschließend um drei Bits nach rechts verschoben.



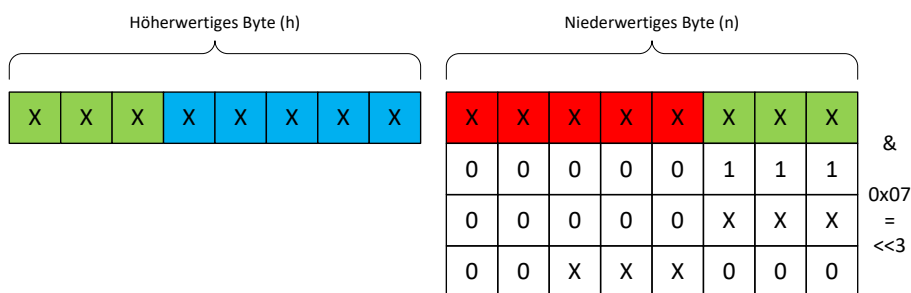
1.3.3 Farbinformation Grün

[\(Thema\)](#)

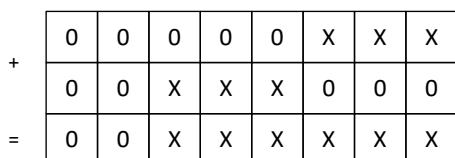
Hier basiert das Gewinn der Informationen auf der Summe von zwei Maskierungen. Zunächst wird das höherwertige Bit h mit 0xE0 maskiert und das Ergebnis um 5 Bits nach rechts verschoben.



Anschließend maskiert man das niederwertige Bit n mit 7 und verschiebt es um 3 Stellen nach links.



Die Farbinformation für Grün ergibt sich aus der Summe der beiden Teilinformationen.



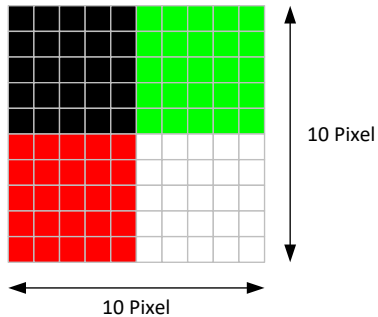
2 Beispiel

[\(Inhalt\)](#)

2.1 Ausgangsdatei und Konvertierung

[\(Thema\)](#)

Die Ausgangsdatei ist eine Grafik der Größe 10 x 10 Pixel, die aus vier farbigen Quadranten besteht.



Die Bitmap-Datei wird mit einem Zeichenprogramm als Windows-Bitmap erstellt. Die Konvertierung in 16bpp BGR565 kann z.B. vorteilhaft mit Microsoft Visual Studio erfolgen. Hier ist auch eine Ansicht als Binärdatei möglich.

2.2 Die Binärdatei

[\(Thema\)](#)

Der folgende Screenshot zeigt die Binärdatei des Beispiels.

```
00000000 42 4D 0A 01 00 00 00 00 00 00 42 00 00 00 28 00 BM.....B...(.
00000010 00 00 0A 00 00 00 0A 00 00 00 01 00 10 00 03 00 .....
00000020 00 00 00 00 00 00 12 0B 00 00 12 0B 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 F8 00 00 E0 07 00 00 1F 00 .....
00000040 00 00 00 F8 00 F8 00 F8 00 F8 00 F8 FF FF FF FF .....
00000050 FF FF FF FF FF FF 00 F8 00 F8 00 F8 00 F8 00 F8 .....
00000060 FF FF FF FF FF FF FF FF FF FF 00 F8 00 F8 00 F8 .....
00000070 00 F8 00 F8 FF FF FF FF FF FF FF FF FF FF 00 F8 .....
00000080 00 F8 00 F8 00 F8 00 F8 FF FF FF FF FF FF FF FF .....
00000090 FF FF 00 F8 00 F8 00 F8 00 F8 00 F8 FF FF FF FF .....
000000a0 FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 .....
000000b0 E0 07 E0 07 E0 07 E0 07 E0 07 00 00 00 00 00 00 .....
000000c0 00 00 00 00 E0 07 E0 07 E0 07 E0 07 E0 07 00 00 .....
000000d0 00 00 00 00 00 00 00 00 E0 07 E0 07 E0 07 E0 07 .....
000000e0 E0 07 00 00 00 00 00 00 00 00 00 00 00 E0 07 E0 07 .....
000000f0 E0 07 E0 07 E0 07 00 00 00 00 00 00 00 00 00 00 .....
00000100 E0 07 E0 07 E0 07 E0 07 E0 07 00 00 FF 00 00 FF .....
00000110 00 00 FF 00 74 61 00 00 00 00 00 00 00 00 00 00 .....ta.....
00000120 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00 00 FF .....
00000130 00 00 FF 00 74 61 00 00 00 00 00 00 00 00 00 .....ta.....
00000140 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00 00 FF .....
00000150 00 00 FF 00 74 61 00 00 00 00 00 00 00 00 00 .....ta.....
00000160 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00 00 FF .....
00000170 00 00 FF 00 74 61 .....ta
```

Interessant sind hierbei:

- Byte 0x0A = 0x42 (hier beginnen die Farbinformationen),
- Byte 0x12 = 0x0A (Breite Bitmap = 10),
- Byte 0x16 = 0x0A (Höhe Bitmap = 10),
- Byte 0x36 und 0x37 = 0x00F8 (Maske Rot),
- Byte 0x3A und 0x3B = 0xE007 (Maske Grün),
- Byte 0x3E und 0x3F = 0x1F00 (Maske Blau),
- 10 x 10 x 2 Farbinformationen (blau markiert).

2.3 Bemerkung

[\(Thema\)](#)

Anhand der Farben Weiß (0xFF) und Schwarz (0x00) lässt sich erkennen, dass die Informationen, beginnend bei der letzten Zeile, von unten nach oben zeilenweise angeordnet sind.

2.4 Die Farbinformationen der vier Beispielfarben

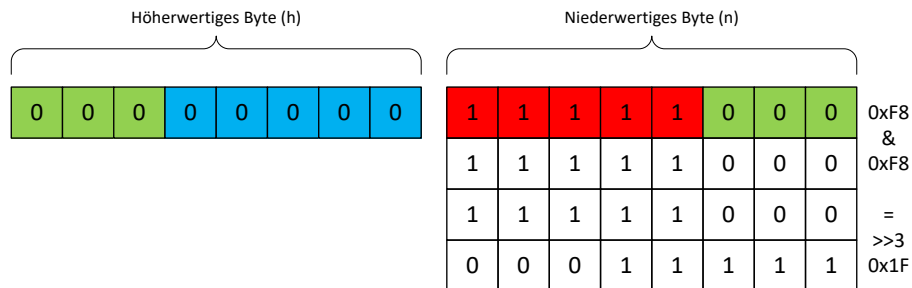
[\(Thema\)](#)

Bei den Farbinformationen sind, wie erwartet, nur vier Werte zu sehen: 00 F8, FF FF, 00 00, E0 07. Um daraus die Informationen für Rot, Grün und Blau zu erhalten, erfolgt für jede dieser Grundfarben eine Bitmaskierung mit den oben genannten Werten mit anschließender Bitverschiebung.

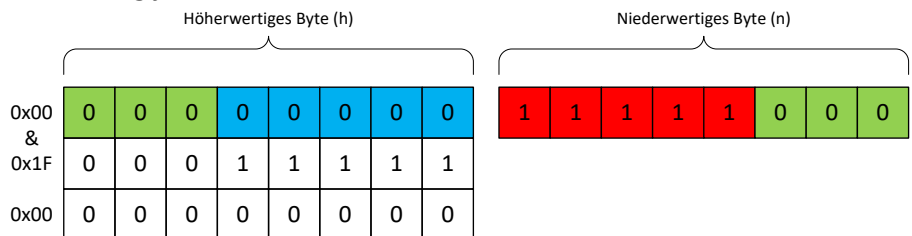
2.4.1 Der Wert 0x 00 F8

[\(Thema\)](#)

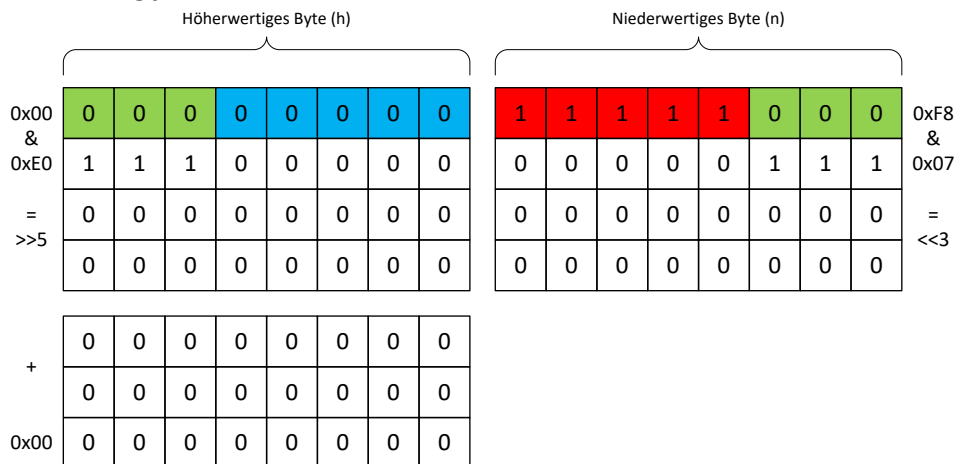
Maskierung für Rot



Maskierung für Blau



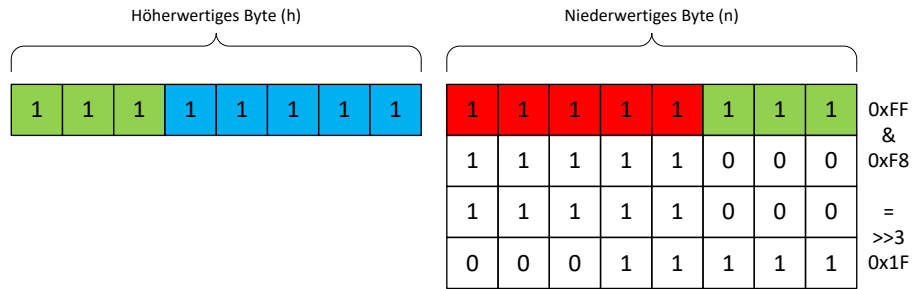
Maskierung für Grün



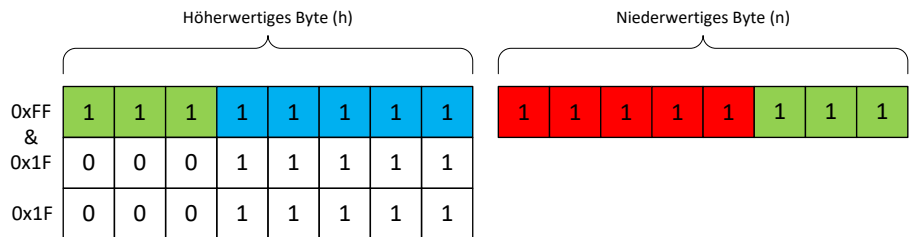
2.4.2 Der Wert 0x FF FF

[\(Thema\)](#)

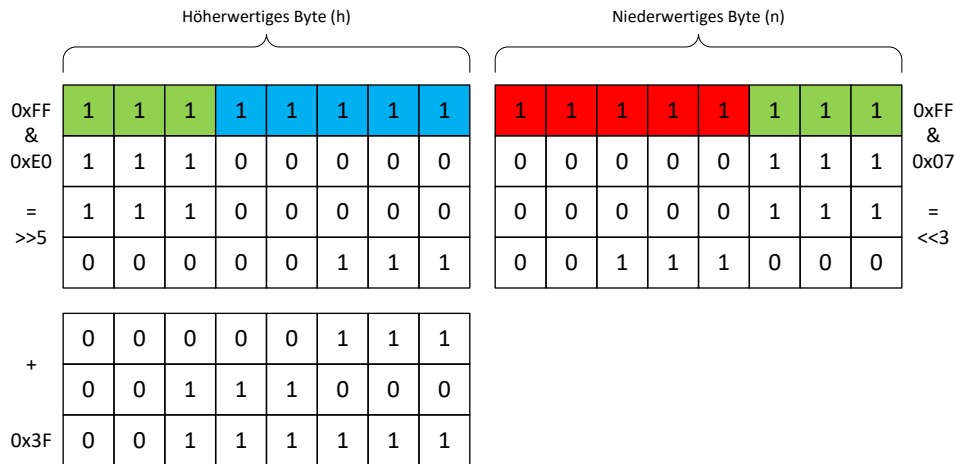
Maskierung für Rot



Maskierung für Blau



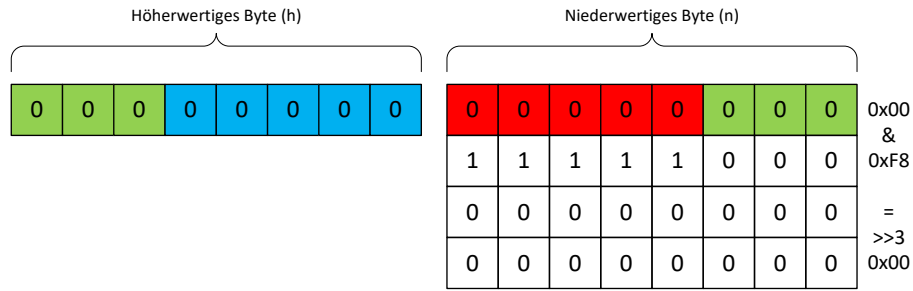
Maskierung für Grün



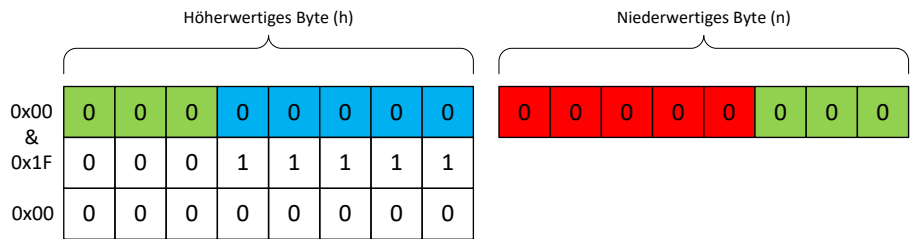
2.4.3 Der Wert 0x 00 00

[\(Thema\)](#)

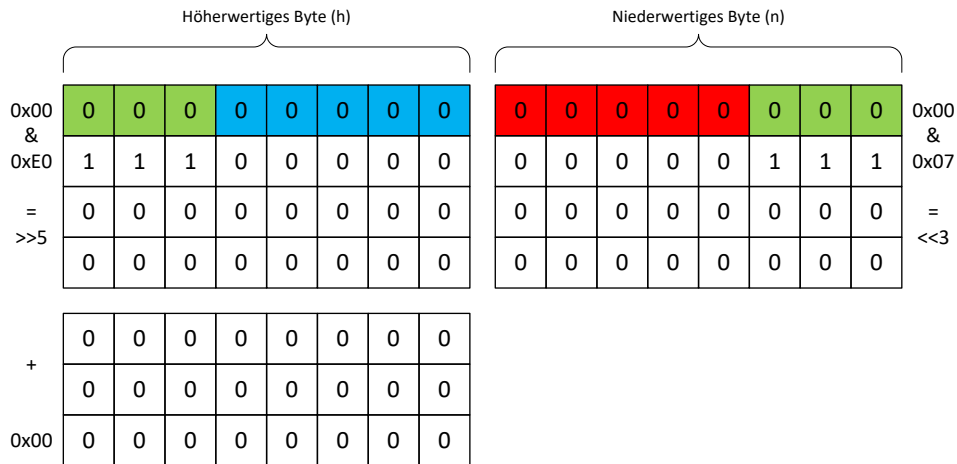
Maskierung für Rot



Maskierung für Blau



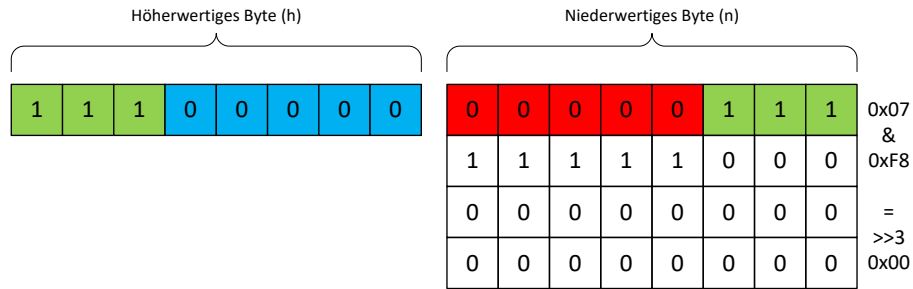
Maskierung für Grün



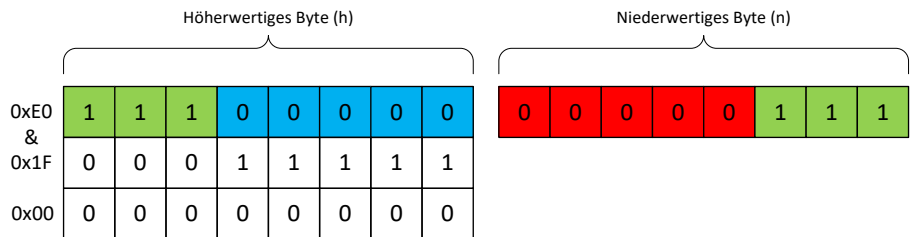
2.4.4 Der Wert 0x E0 07

[\(Thema\)](#)

Maskierung Rot



Maskierung Blau



Maskierung Grün

