
Graphics Interchange Format

Georg Kusch

11.06.2003

Inhalt :

1. Einführung
2. Aufbau
3. Beispiel
4. LZW - Algorithmus im Überblick
5. Zusammenfassung / Quellen

1. Einführung - Entstehungsgeschichte

- Anfang der 80er Jahre wachsender Bedarf an Grafikformaten
- frühe Bildformate waren kaum standardisiert und bestanden häufig nur aus einer Kopie des Bildschirmspeichers
- Veröffentlichung von GIF in der Spezifikation der ersten Version '87a' am 15.06.1987 (Steve Wilhite)
- entwickelt für Online-Transfer und Austausch von Rastergrafiken, unabhängig von der benutzten Hardware
- Der LZW-Kompressionscode ist allerdings patentiert (Unisys 1984)
- 1989 Erweiterung zu heutiger Version '89a'
- 1996 - Entwicklung einer TrueColor-GIF-Variante ('GIF24') die keinen LZW-Code benutzen sollte.
=> PNG Format

- Sequentieller Aufbau (Aufteilung in fest definierte Blöcke), dadurch möglich :
 - Verarbeitung von bereits empfangenen Daten
 - Hinzufügen von Zusatzinformationen (z.B. bei Animationen)
 - Speicherung von mehreren Bildern in derselben Datei
- Farbenpalette auf 256 Farben beschränkt
- keine Übertragungs-Fehlererkennung / -korrektur
- verlustfreie Kompression durch LZW-Algorithmus

2. Aufbau - GIF89a Grammatik

<GIF Datei> ::= <Dateikopf> {< Daten >}* <Anhang>

<Dateikopf> ::= <Signatur> <Logischer Bildschirm> [<globale Farbtabelle>]

<Signatur> ::= 'GIF87a' | 'GIF89a'

<Logischer Bildschirm> ::= <Breite> <Höhe> <Farbtabelleninformation>
<Hintergrundfarbe> <Pixelgrößenverhältnis>

[<globale Farbtabelle>] ::= (RGB)^{Anzahl der Farben}

Beispiel : 47 49 46 38 39 61 10 00 10 00 B3 00 00 = GIF89a?·?·|··

Version 89a / Breite = 16 / Höhe = 16

Farbtabelleninformation = 10110011

(globale Farbtabelle(unsortiert) mit 16 Einträgen folgt,
4 Bits je Farbe in der Ursprungsgrafik)

Hintergrundfarbe = 0 (d.h. keine Transparenzfarbe definiert)

Pixelgrößenverhältnis = 0 (d.h. quadratisch(Standard))

<Daten> ::= <Grafik Block> | [<Spezialblock>]

<Grafik Block> ::= [<Grafikkontrollerweiterung>] <Grafikdaten> | [<Text-Erweiterung>]

<Grafikkontrollerweiterung> ::= 0x21 0xF9 <Datenblockgrösse>
<Bildanzeige-Beendigungsverfahren>
<Verzögerungszeit> <Transparentfarbe> 0x00

Beispiel : 21 F9 04 04 00 00 08 00 = !"??..?.

0x21 = ein Erweiterungsblock folgt

0xF9 = Kennung der Grafikkontrollerweiterung

04 = Blockgrösse in Byte

04 = 00000100 (Beseitigungsmethode : Grafik nicht beseitigen,
Benutzerreaktion nicht abwarten)

0x00 0x00 = Verzögerungszeit(entfällt hier)

0x08 = Index der Transparenzfarbe

0x00 = Block-Abschluss-Zeichen

<Grafikdaten> ::= <Bildkopf> [<lokale Farbtabelle>] < Bilddaten >

<Bildkopf> ::= 0x2C <Startposition x-Wert> <Startposition y-Wert>
<Bildbreite> <Bildhöhe> <Zusatzinformation>

[<lokale Farbtabelle>] ::= (RGB)^{Anzahl der Farben}

< Bilddaten > ::= < Minimale LZW-Code-Grösse > {< Datenblock >}*⁺

< Datenblock > ::= < Blockgrösse > < LZW-codierte Farbindizes > 0x00

Beispiel : 2C 00 00 00 00 10 00 10 00 00 04 = , ? . ? . . ?

0x2C = Kennung für Grafikdaten

0x00 0x00 0x00 0x00 = Startpunkt im definierten logischen Bildschirm (x,y)

0x10 0x00 0x10 0x00 = Breite = 16 / Höhe = 16

0x00 = Zusatzinformationen (hier: keine lokale Farbtabelle, nicht Interlaced ...)

0x04 = minimale LZW-Code-Grösse

$\langle \text{Text-Erweiterung} \rangle ::= 0x21\ 0x01\ \langle \text{Textinformationen} \rangle\ \{\langle \text{Textblock} \rangle\}^*{}^+ 0x00$

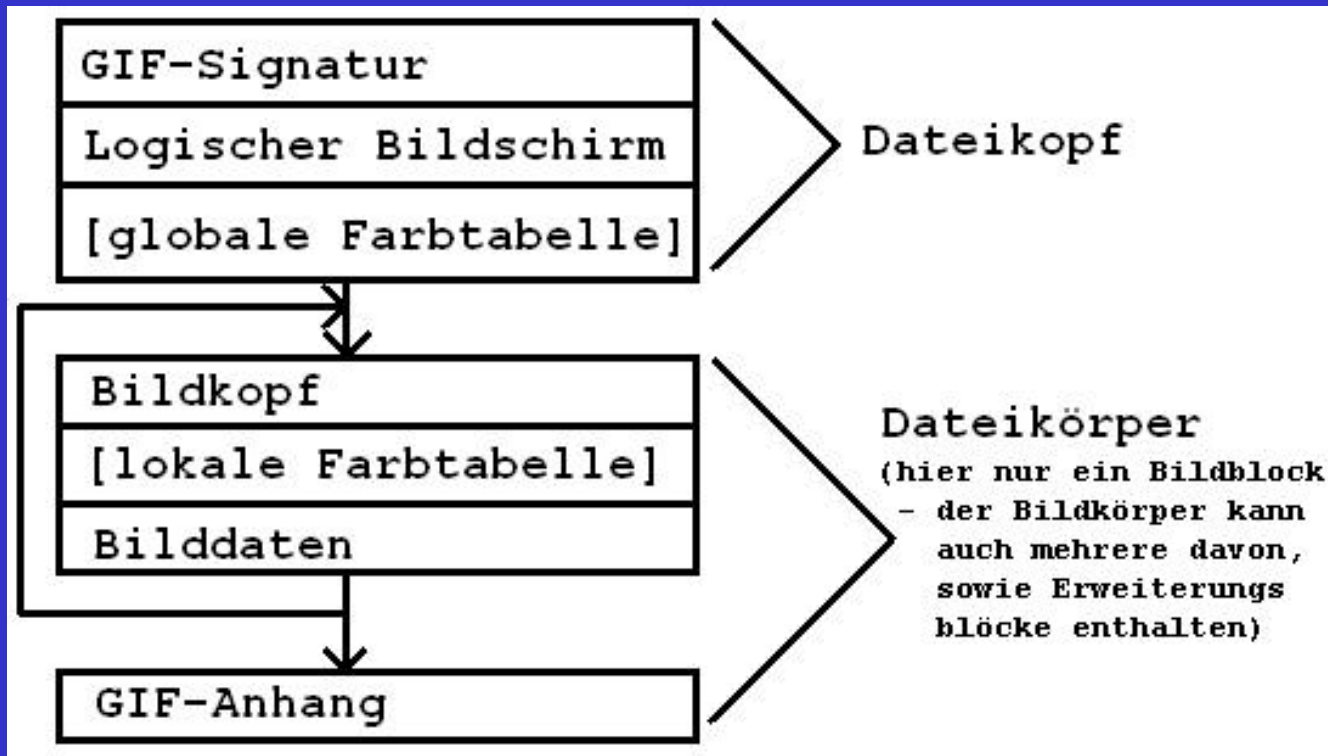
$\langle \text{Spezialblock} \rangle ::= \langle \text{Anwendungs-Erweiterung} \rangle \mid \langle \text{Kommentar-Erweiterung} \rangle$

$\langle \text{Anwendungs-Erweiterung} \rangle ::= 0x21\ 0xFF\ \langle \text{Anwendungsinformationen} \rangle\ \{\langle \text{Anwendungs-Datenblock} \rangle\}^*{}^+ 0x00$

$\langle \text{Kommentar-Erweiterung} \rangle ::= 0x21\ 0xFE\ \{\langle \text{Kommentar-Datenblock} \rangle\}^*{}^+ 0x00$

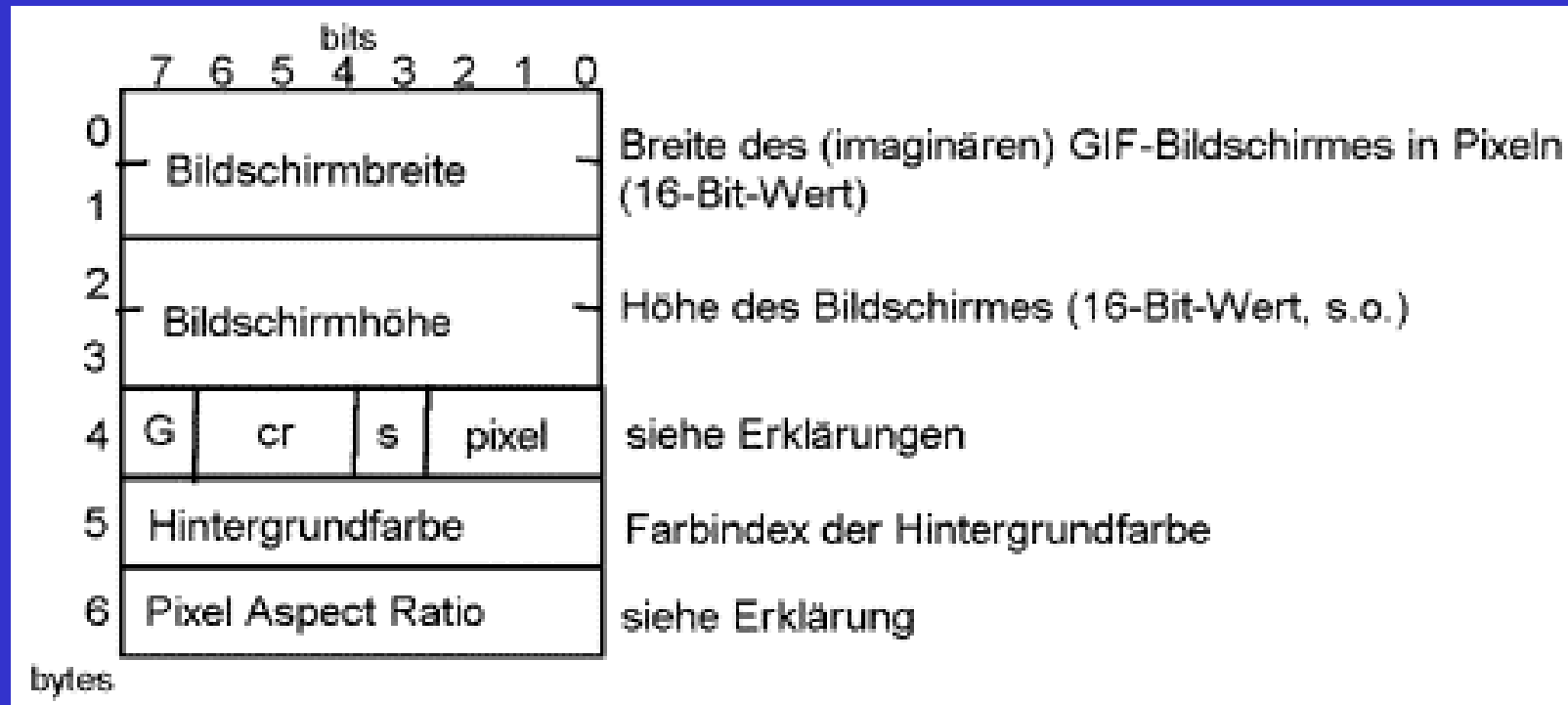
$\langle \text{Anhang} \rangle ::= 0x3B$

Veranschaulichung der sequentiellen Blockstruktur :



- Dateikopf
 - Logischer Bildschirm
 - Farbtabelle
- Dateikörper
 - Grafikblock
 - *Grafikkontrollerweiterungsblock*
 - Bildblock
 - *Bildkopf*
 - Farbtabelle
 - Bilddaten
 - Text-Erweiterungsblock
- Anwendungs-Erweiterung
- Kommentar-Erweiterung

2. Aufbau - Logischer Bildschirm



`G` : ist dieses Bit gesetzt , folgt dem logischen Bildschirm eine globale Farbtabelle

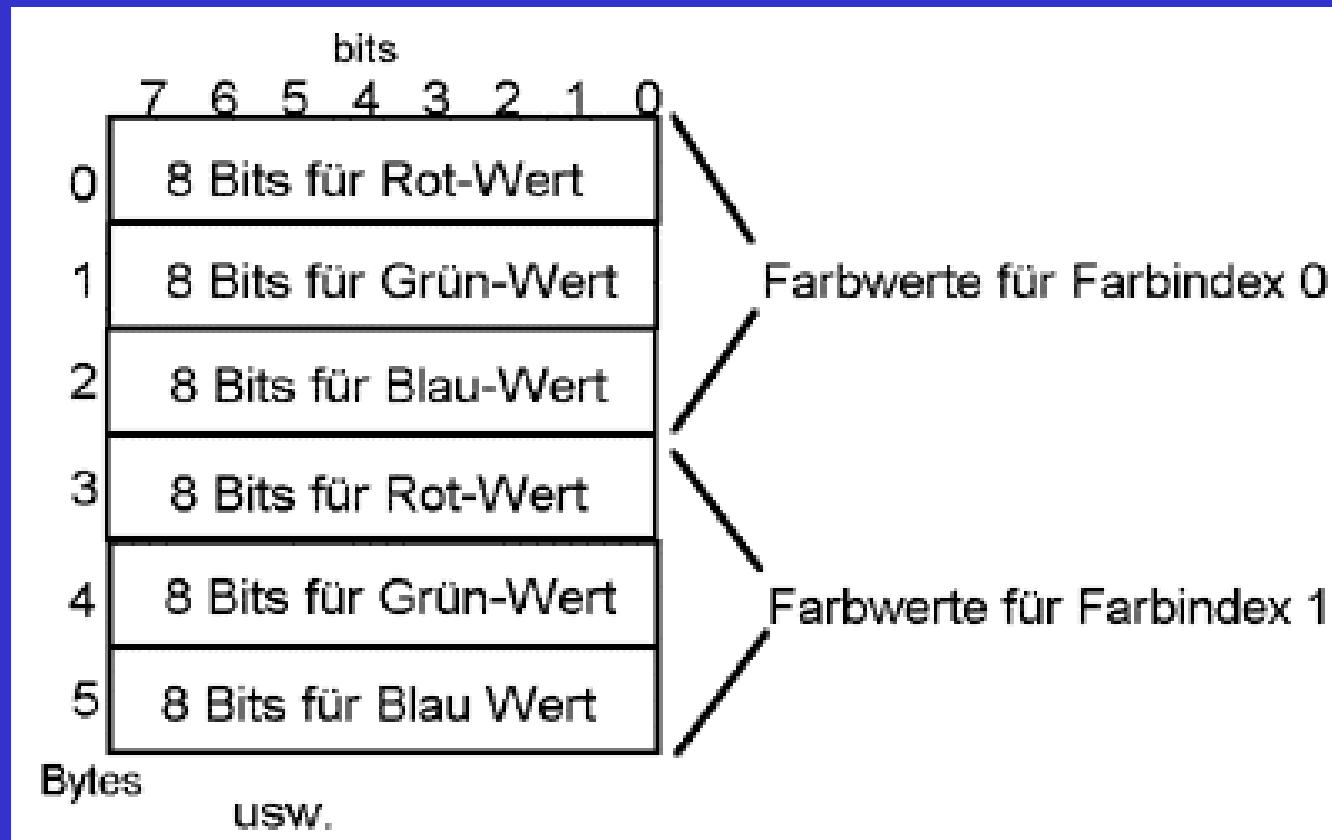
`cr` (color resolution) : Farbtiefe der Ursprungsgrafik ($2^{\{cr+1\}}$ Farben)

`s` (sort flag) : ist eine globale Farbtabelle vorhanden und dieses Bit gesetzt, so ist die Tabelle absteigend nach der Häufigkeit der Farben sortiert

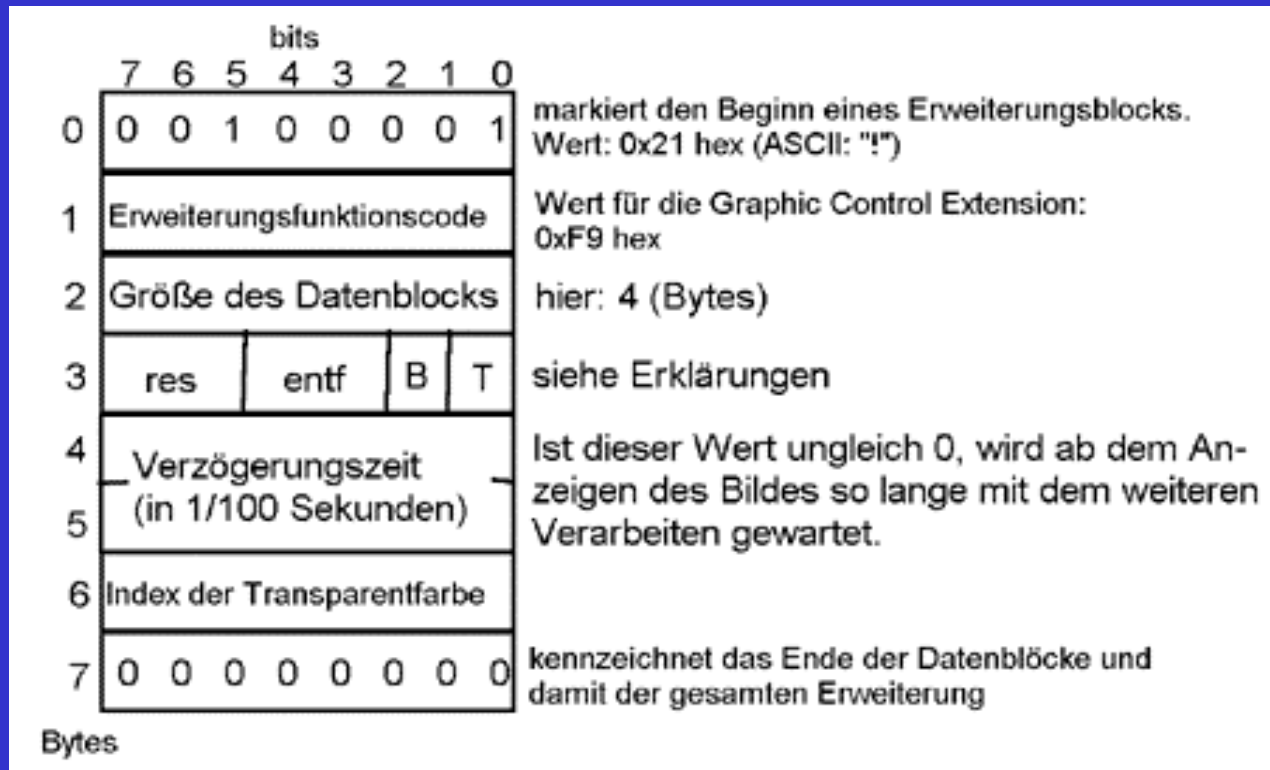
`pixel` : Anzahl der Bits pro Pixel ($2^{\{pixel+1\}}$) = Größe der Farbtabelle in RGB

`Pixel Aspect Ratio` : Verhältnis Pixelbreite : Pixelhöhe im Ursprungsbild

2. Aufbau - Globale / Lokale Farbtabelle



2. Aufbau - Grafikkontrollererweiterung



res : reserviert, haben den festen Wert 0

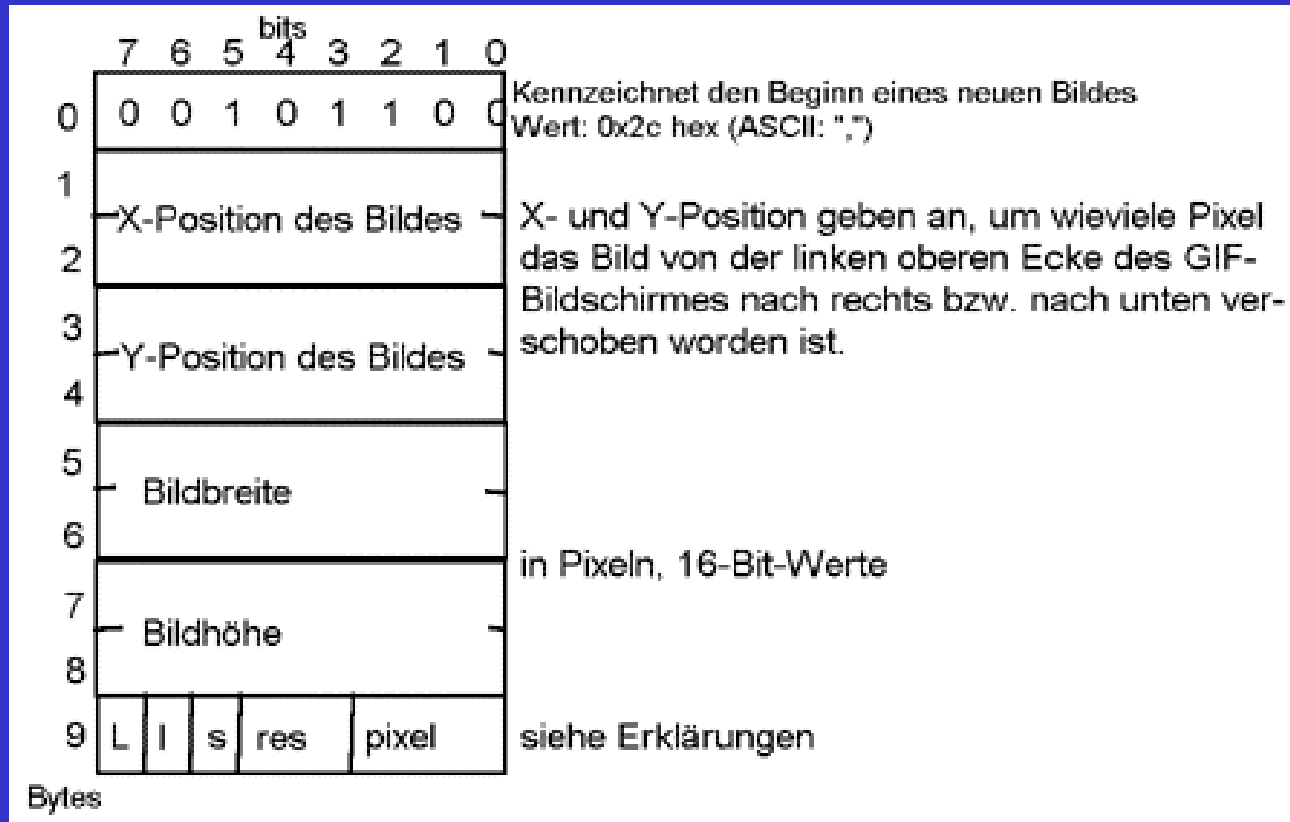
entf : Der Wert dieser drei Bits gibt an, wie mit der Grafik nach der Anzeige auf dem Bildschirm verfahren werden soll :

- 0 - kein bestimmtes Verfahren
- 1 - Grafik nicht entfernen
- 2 - Fläche wieder mit der ursprünglichen Hintergrundfarbe ausfüllen
- 3 - früheren Hintergrund wiederherstellen
- 4-7 - noch nicht definiert

B : wenn gesetzt, dann vor dem Anzeigen des nächsten Bildes entsprechend lange warten

T : wenn gesetzt, dann ist in Byte 6 der Index der Transparentfarbe angegeben

2. Aufbau - Bildkopf



L : wenn gesetzt, so folgt nach dem Bildkopf eine lokale Farbtabelle

I : wenn gesetzt, so ist die Grafik "interlaced" gespeichert :

jede 8-te Zeile (beginnend mit der 0. Zeile)

jede 8-te Zeile (beginnend mit der 4. Zeile)

jede 4-te Zeile (beginnend mit der 2. Zeile)

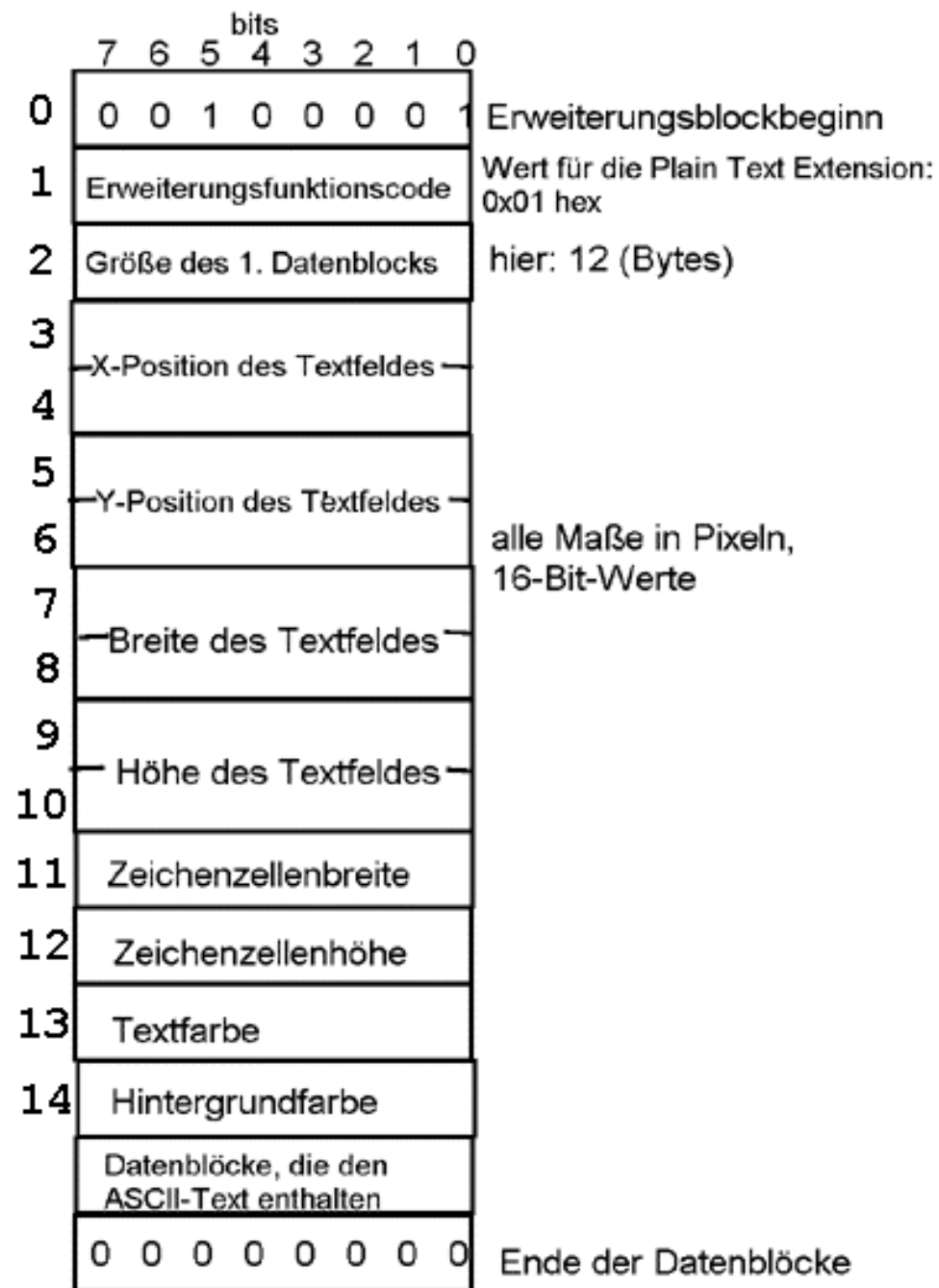
jede 2-te Zeile (beginnend mit der 1. Zeile)

s (sort flag): Ist eine lokale Farbtabelle vorhanden und dieses Bit gesetzt, so ist die Tabelle absteigend nach der Häufigkeit der vorkommenden Farben sortiert

res : reserviert, haben den festen Wert 0

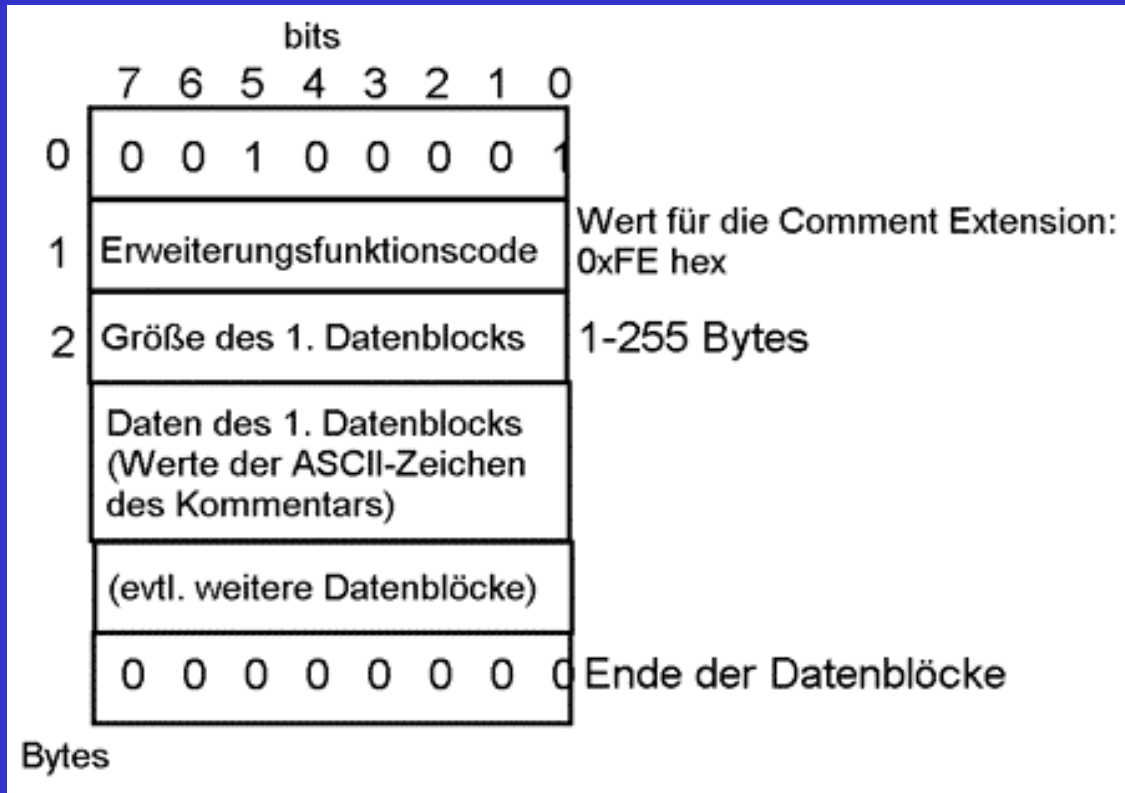
pixel : Größe der lokalen Farbtabelle, sofern vorhanden

2. Aufbau - Text-Erweiterung

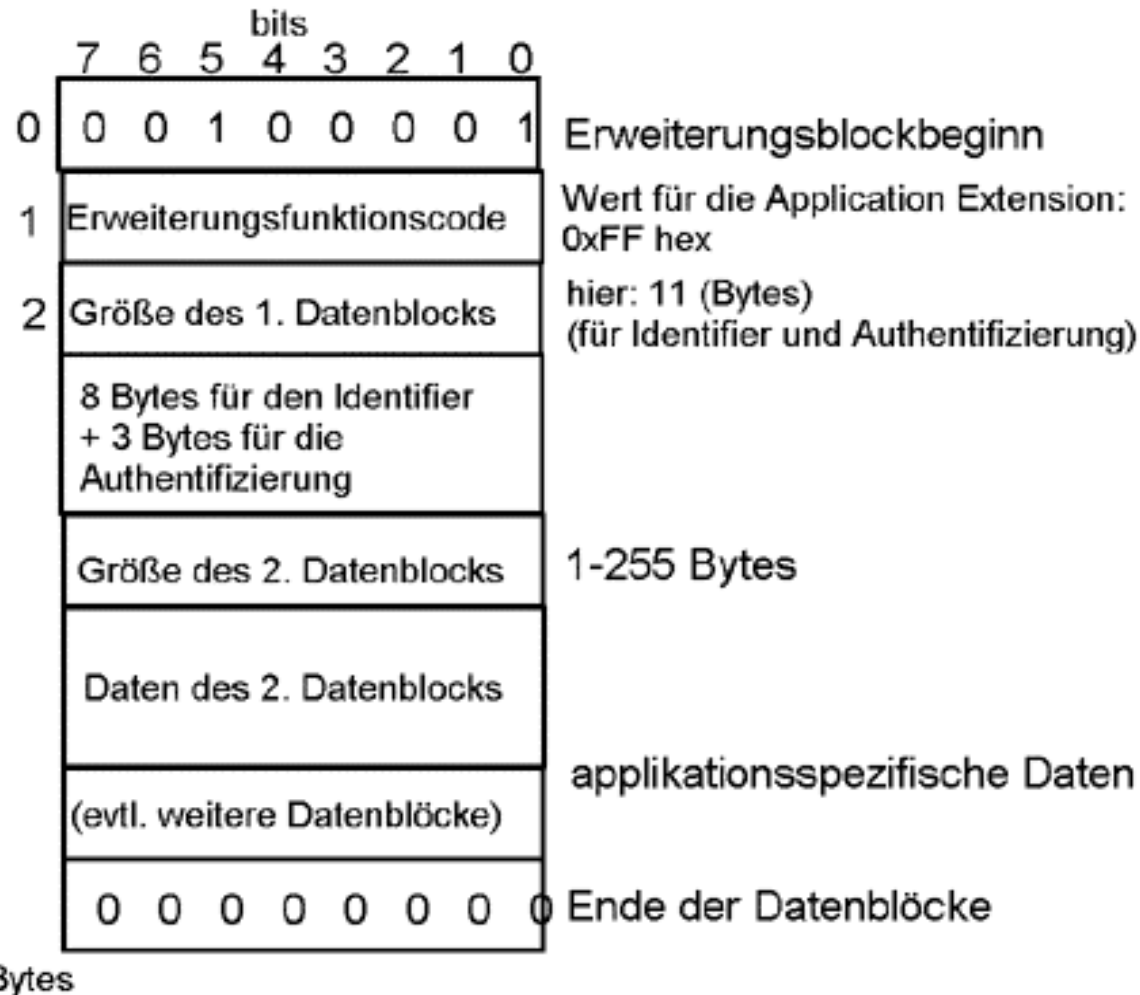


Bytes

2. Aufbau - Kommentar-Erweiterung



2. Aufbau - Anwendungs-Erweiterung



3. Beispiel

Adr. Dez	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII -----
0	47	49	46	38	39	61	10	00-10	00	B3	00	00	FF	FF	00		GIF89a?·?· ··__·
16	C0	C0	C0	00	FF	00	80	80-80	80	80	00	00	80	00	FF		+++· ·ÇÇÇÇÇ··Ç·
32	00	00	00	00	00	FF	FF	FF-00	00	00	00	00	00	00	00	
48	00	00	00	00	00	00	00	00-00	00	00	00	00	21	F9	04	!""?
64	05	14	00	08	00	2C	00	00-00	00	10	00	10	00	00	04		?¶·?·,.....?·?·?·?
80	4F	10	C9	49	AB	BD	74	E8-CD	F9	41	43	21	8E	45	20		O?+I½çtP-~AC!ÄE
96	14	42	50	7C	83	11	84	C1-51	96	47	10	1F	E1	A1	EB		¶BP â?ä-QûG??ßíÛ
112	B3	AE	EA	AD	17	6D	26	B2-01	57	3B	99	6D	89	84	C1		«Û;?m& ?W;Ömëä-
128	94	B3	1F	0E	B9	EB	AD	56-20	D2	48	45	FA	24	77	03		ö ?? Û;V ÊHE·\$w?
144	00	41	BC	C3	84	C7	84	0F-06	F1	55	AF	DF	F0	49	04		·A++äÃäα?±U»^-I?
160	00	3B	EOF	--	--	--	--	-----	--	--	--	--	--	--	--		·;?

----- globale Farbtabelle -----

16*3 Bytes für Farbtabelle

Byte 13-60

- Farbe00 = RGB(0xFF, 0xFF, 0x00)
- Farbe02 = RGB(0x00, 0xFF, 0x00)
- Farbe04 = RGB(0x80, 0x80, 0x00)
- Farbe06 = RGB(0xFF, 0x00, 0x00)
- Farbe08 = RGB(0xF0, 0xF0, 0x00)
- Farbe10 = RGB(0x00, 0x00, 0x00)
- Farbe12 = RGB(0x00, 0x00, 0x00)
- Farbe14 = RGB(0x00, 0x00, 0x00)

- Farbe01 = RGB(0xC0, 0xC0, 0xC0)
- Farbe03 = RGB(0x80, 0x80, 0x80)
- Farbe05 = RGB(0x00, 0x80, 0x00)
- Farbe07 = RGB(0x00, 0x00, 0x00)
- Farbe09 = RGB(0x00, 0x00, 0x00)
- Farbe11 = RGB(0x00, 0x00, 0x00)
- Farbe13 = RGB(0x00, 0x00, 0x00)
- Farbe15 = RGB(0x00, 0x00, 0x00)

3. Beispiel

Adr.	Dez	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII	-----
0		47	49	46	38	39	61	10	00-10	00	B3	00	00	FF	FF	00		GIF89a?	·?· ·· ·
16		C0	C0	C0	00	FF	00	80	80-80	80	80	00	00	80	00	FF		+++·	·ÇÇÇÇÇ··Ç·
32		00	00	00	00	00	FF	FF	FF-00	00	00	00	00	00	00	00		·	·
48		00	00	00	00	00	00	00	00-00	00	00	00	00	00	21	F9	04		!?"
64		05	14	00	08	00	2C	00	00-00	00	10	00	10	00	00	04		?¶·?·	, ····?·?·?·?
80		4F	10	C9	49	AB	BD	74	E8-CD	F9	41	43	21	8E	45	20		O?+I½çtP-	"AC!ÄE
96		14	42	50	7C	83	11	84	C1-51	96	47	10	1F	E1	A1	EB		¶BP â?ä-QûG??ßíÛ	
112		B3	AE	EA	AD	17	6D	26	B2-01	57	3B	99	6D	89	84	C1		«Û;?m& ?W;Ömëä-	
128		94	B3	1F	0E	B9	EB	AD	56-20	D2	48	45	FA	24	77	03		ö ?? Û;V ÊHE·\$w?	
144		00	41	BC	C3	84	C7	84	0F-06	F1	55	AF	DF	F0	49	04		·A++äÃäα?±U»	^-I?
160		00	3B	EOF	--	--	--	--	-----	--	--	--	--	--	--	--		·;?	

----- Grafikdaten Bildkopf -----

- Byte69 = 0x2C -> Grafikdaten
- Byte70/71 = 0x00 0x00 x-Position im logischen Bildschirm
- Byte72/73 = 0x00 0x00 y-Position im logischen Bildschirm
- Byte74/75 = 0x10 0x00 = 16 , Bildbreite
- Byte76/77 = 0x10 0x00 = 16 , Bildhöhe
- Byte 78 = 0x00 , Zusatzinformationen (keine lokale Farbtabelle, nicht Interlaced)
- Byte79 = 4 , Minimale LZW-Code-Grösse (in Bit)

4. Lempel-Ziv-Welch Algorithmus im groben Überblick

- Erstellung eines Wörterbuchs
- unkomprimierter Datenstrom wird in Zeichenketten zerlegt
- Vergleich mit den bereits vorhandenen Zeichenketten im Wörterbuch
 - wenn bereits vorhanden, so wird nur der Wörterbuchindex ausgegeben
 - ansonsten neuer Wörterbucheintrag
- => Die Kompressionsrate hängt von der Datenstruktur und von der Kunst des Programmierers ab.

Vorteile von GIF :

- Erweiterbarkeit und Spezialfunktionen durch die Erweiterungs-Blöcke
- spezielle Abstimmung auf Datenfernübertragung durch die Interlaced-Darstellung
- weite Verbreitung auf allen gängigen Plattformen
- gute Packraten ohne Datenverlust durch die LZW-Kompression
- Definition einer transparenten Farbe möglich
- jeder gängige browser kann GIF verarbeiten (ohne Plug-in)
allerdings werden nicht alle Funktionen von GIF unterstützt (UserInput, PlainText)

Nachteile :

- Niedrige Anzahl an Farben
- Datenkomprimierung ist lizenzpflichtig

5. Zusammenfassung - Die Zukunft von GIF

1984 LZW-Algorithmus wird entwickelt und von Unisys patentiert

1994 Lizenzgebühren für Software-Entwickler (1.5%)

1999 Lizenzgebühren für Betreiber von kommerzielle Websites (2500\$ je Server)

Aufgrund des Patentproblems und dem veralteten Funktionsumfang dürfte die Verbreitung von GIF in Zukunft zurückgehen.

Den freien Platz dürfte dann vor allem PNG einnehmen.

- www.w3.org/Graphics/GIF/spec-gif87.txt (GIF-Spezifikationen)
- www.w3.org/Graphics/GIF/spec-gif89a.txt
- <http://goethe.ira.uka.de/seminare/rftk/> (zip/gif/jpg/png/lzw/huffman...)
- http://lpf.ai.mit.edu/Patents/Gif/gif_lic.html (offizielle Lizenzvereinbarung)
- <http://www.comrecht.de/deutsch/recht/pub/gifkontroverse.htm> (LZW-Patent)
- <http://burnallgifs.org/> (Patent-Gegner)