

RADOLAN-Unterstützungsdokument

für Programmierer

Lesen des RADOLAN-Binärformats

Deutscher Wetterdienst
- Abteilung Hydrometeorologie -

Internet:
<http://www.dwd.de/RADOLAN>

Allgemeine Mailadresse:
radolan@dwd.de



- Dieses Dokument dient der Programmiererunterstützung für das Lesen **binärer quantitativer (2-Byte-codierter) RADOLAN-Komposits**. Es ergänzt die RADOLAN/RADVOR-Kompositformat-Beschreibung (Link am Ende des Dokuments).

```
0101 1101 0000 1111  
0111 0001 1111 0000  
1100 1011 1100 0010
```



1 Aufbau des binären RADOLAN-Formats

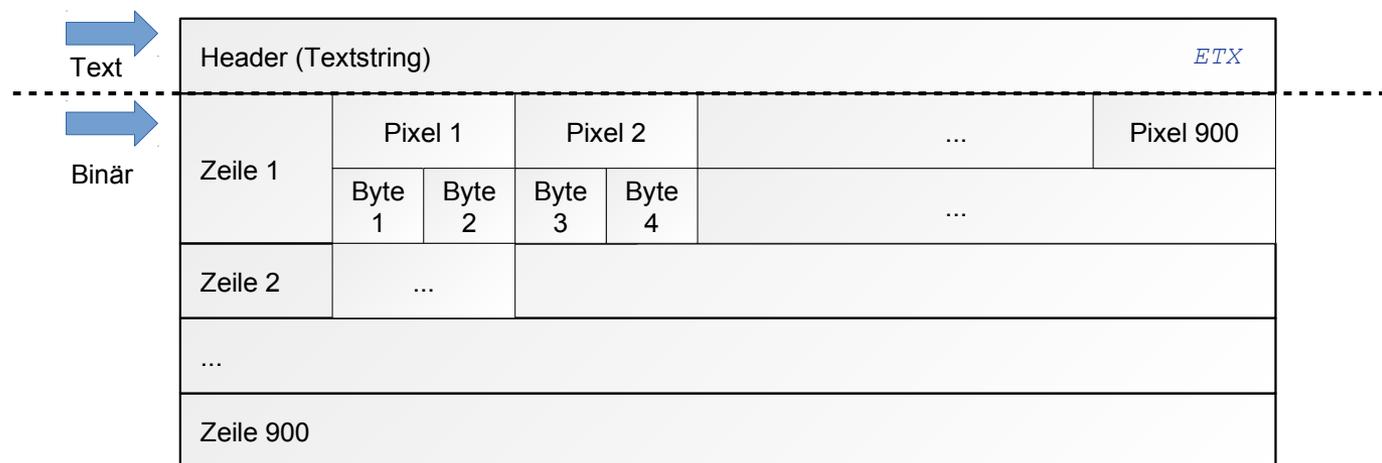
2 Flags

3 Praxis (Prinzip des binären Lesens)

1 Aufbau des binären RADOLAN-Formats

- Header
- Binärteil

- Der Datensatz beginnt mit einem ASCII-Header und wird von einem Binärteil gefolgt. Der Header kann in der Länge variieren, da hier die verwendeten Radarstandortkürzel angegeben werden. Der Header schließt mit dem Zeichen *ETX* ab.



900 x 900-Komposit

- Jedes binäre RADOLAN-Produkt gibt über die Dimension seines Wertefeldes im Header an der Stelle **GP** (grid points) Auskunft. Diese Angabe kann genutzt werden, ein Leseprogramm dynamisch für das jeweilige Produkt zu initialisieren.
Zum unten gezeigten Beispiel:
Es handelt sich um das kleinste (nationale) quantitative Komposit¹ mit 900×900 Pixeln und somit 900 Zeilen mit jeweils 900 Werten.
- Der Header gibt über die in das Komposit eingegangenen² **Radarstandorte** (Datenquellen) in Form von Abkürzungen Auskunft. Beispiel: `oft` = Radar Offenthal.
In der RADOLAN/RADVOR-Kompositformatbeschreibung finden sich Tabellen zu den in- und ausländischen Radaren.

```
RW241150100001011BY1620134VS 3SW 2.9.0PR E-01INT 60GP 900x900  
MS62<ham, ros, emd, han, umd, bln, ase, fld, drs, nhb, oft, eis, tur, muc, fbg> ETX  
(nicht darstellbar)
```

hier folgen die binär codierten Werte

Beispiel für einen Header aus dem RW-Produkt

¹ Weitere Kompositgrößen verfügbar:

1100×900 aus der Radarklimatologie, 1500×1400 für das mitteleuropäische Komposit und in Zukunft sicherlich noch weitere Varianten.
Daher ist die dynamische Einstellung eines Programms auf das Wertefeld eines Produkts sicherlich sinnvoll.

² Aufgrund von Wartungen oder technischen Störungen an den Radaranlagen können Standorte fehlen. Die Länge des Headers ist daher variabel. Man sollte also für den Teil der Radarstandorte nicht mit festen Positionen arbeiten.

1 Aufbau des binären RADOLAN-Formats

Binärteil: Codierung der Werte

- Im Binärteil sind alle Datenpunkte (Pixel) codiert, zu je 2 Byte.
- Die Codierung der Niederschlagswerte erfolgt mit 12 Bit. Da in 2 Byte codiert wird, sieht die Darstellung des Wertes 1 (Dezimal) in 16 Bit folgendermaßen aus (es wird hier die Little-Endian-Darstellung verwendet, d.h. gelesen von rechts nach links, rechts steht das niederwertigste Bit):
- Somit ergibt sich bei Vollbelegung (alle 12 Bits auf 1) ein Wert von 4.095 als größter codierbarer dezimaler Zahlenwert ($2^{12} \rightarrow$ Bereich von 0 bis 4.095).

Byte 2	Byte 1
0000 0000	0000 0001

Darstellung von Dezimal 1 mit 12 Bit

Byte 2	Byte 1
0000 1111	1111 1111

Vollbelegung aller 12 Bits:
Maximalwert 4095 im Dezimalsystem

2 Flags

- Als eigentlicher Niederschlagswert sind nur die ersten 12 Bits zu interpretieren (ergeben im Maximum 4095). Hinter dem 12. Bit beginnt der für Flags vorgesehene Bereich (4 Bits = ein Halb-Byte).
- Flags sind Zusatzinformationen, die „über“ den Werten (codiert in den 12 Bits eines Bytes) liegen. Sie sind im höherwertigen Byte codiert und liefern so zusätzliche Informationen zur Interpretation des Wertes.

Byte 2	Byte 1
0000 0000	0000 0001

Bit 13-16: ein Halbbyte für Flags

Byte 2	Byte 1
0001 0000	0000 0000

Bit 13 gesetzt

Byte 2	Byte 1
0001 0000	0000 1010

Niederschlag und Bit 13 gesetzt

Bit 13 (nur interpolierte Stationsdaten)

Bit 13 kommt nur in angeeichten Produkten, wie dem RW, und hiervon abgeleiteten Produkten (Summen, wie z.B. dem SF) vor.

Dieses Bit zeigt an, dass im betreffenden Bereich keine Radar-, sondern nur Stationsdaten vorlagen. Der Grund hierfür kann eine Abschaltung des betreffenden Radars aufgrund einer Störung oder einer Wartung sein. Im RW-Produkt können die so entstehenden Lücken dennoch mit Messdaten aufgefüllt werden. Dies geschieht aufgrund von online-messenden Niederschlagsstationen (Ombrometer), deren Messungen jeweils über einen gewissen Radius in die Fläche interpoliert werden. Das bedeutet für die Interpretation des Niederschlagswertes, dass das Bit 13 (wenn man alle Bits in einen Wert im Dezimalsystem umgerechnet hat) abgezogen werden muss.

Beispiel:

gelesener Wert: 4097

$4097 - 4096 (\text{Bit } 13) = 1 \rightarrow 0,1 \text{ mm}$ aus interpolierter Stationsmessung.

Die Wirkung eines gesetzten Bit 13 in einem ausgelesenen Produkt zeigt das Bild auf der nächsten Seite auf der linken Seite (RW oben, SF unten). Die Flächen mit gesetzten Flags sind jeweils schwarz dargestellt.

Folgende Phänomene lassen sich ebenfalls im Bild erkennen:

- Im RW- oder SF-Bild sind häufig „Interpolationskreise“ außerhalb von Deutschland zu sehen. Die Ursache sind ausländische Stationsmessungen, die auch in das nationale Aneichprodukt eingehen, wobei hier keine Radarinformation aus dem DWD-Radarverbund mehr vorliegt.
- Zu Unterschieden im RW und aufsummierten Produkten wie z.B. dem SF bezüglich Bit 13: Stationsmessungen ohne Radar führen zum gesetzten Bit 13 in einer RW-Stunde. Das SF wird durch eine 24h-Aufsummierung der RWs gebildet. Wann immer ein RW Bit 13 gesetzt hat, wird dieses in das SF „weitergereicht“. Somit kann das SF größere Bereiche mit diesem Flag zeigen, da jedes RW Flags in die Summe tragen kann.

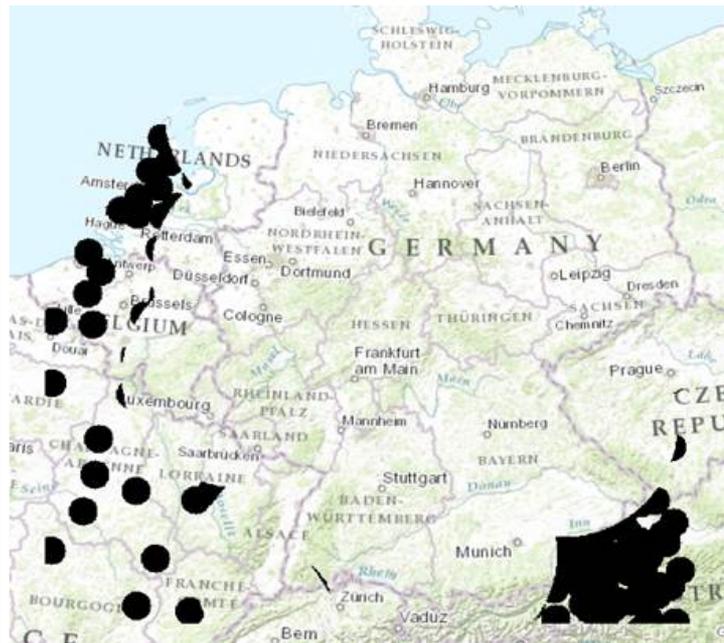
Bit 13 rw1510201150.tif



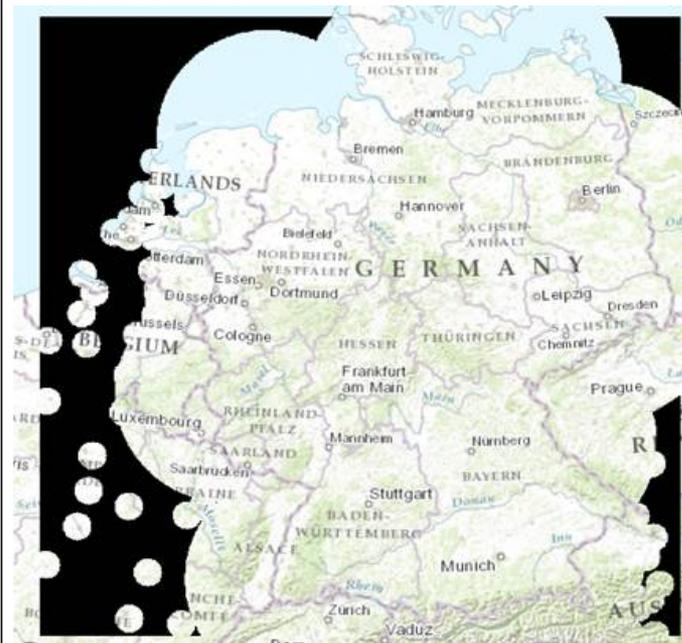
Bit 14 rw1510201150.tif



Bit 13 sf1510201150.tif



Bit 14 sf1510201150.tif



2 Flags

→ Bit 14 + 2500 (Fehlerkennung)

Die Außenbereiche des RADOLAN-Gebiets, die vom Radarverbund nicht mehr abgedeckt werden, sind mit Fehlwerten belegt (s. unten stehendes Bild, rechte Seite). Ebenso Gebiete, die durch einen Radarausfall nicht abgedeckt werden und in denen gleichzeitig keine Stationsdaten verfügbar waren (Stationsdaten nur in angeeichten Produkten wie RW und SF).

Der Fehlwert setzt sich zusammen aus dem Bit 14 (8192) und dem Wert 2500. Zusammen ergibt das den Dezimalwert 10692. Vermutlich genügt es beim Lesen und Feststellen der gültigen Werte einfach auf \geq Bit14 zu prüfen (ignoriert aber die folgenden beiden Flags).

Das Bild zeigt die Wirkung des Bit 14 in ausgelesenen Produkten auf der rechten Seite (RW oben, SF unten).

→ Bit 15 (negativer Wert)

Bei gesetztem Bit 15 ist der gelesene Wert als negativer Wert zu interpretieren. Negative Werte sollten nur in den Produkten der interpolierten Aneichdifferenzen vorkommen. In den angeeichten Endprodukten wie RW und SF kommt dieses Bit nicht vor.

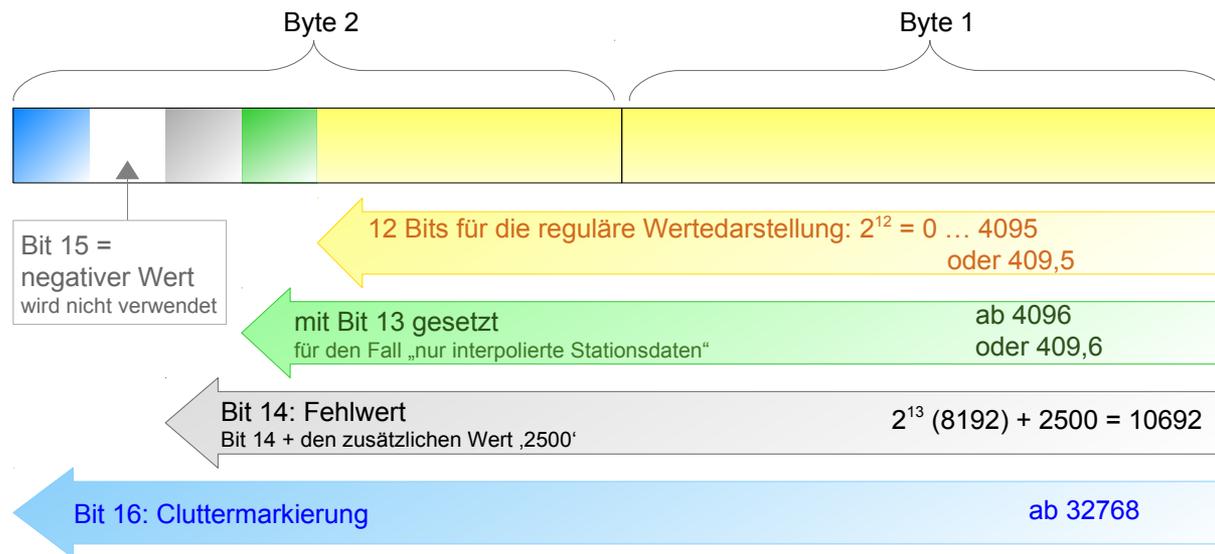
→ Zu Bit 16 (Clutterflag)

Bit 16 bezeichnet ein sog. Clutterflag. Der Begriff Clutter bezeichnet Falschechos im Radarprodukt. Eine Cluttermarkierung erfolgt, wenn in den vergangenen 72 Stunden des RH-Produkts (Vorstufenprodukt zum RW) Pixel immer Niederschlag anzeigten.

Das Vorgehen ist hier ähnlich der Behandlung des Bit 13: Ein als Clutter markierter aber möglicherweise brauchbarer Wert muss um dieses Flag „korrigiert“ werden, um den Niederschlagswert zu erhalten. Der Dezimalwert ist 32768. Das Clutterflag sollte in den angeeichten RW- und SF-Produkten nicht enthalten sein.

2 Flags

Dezimalwerte in Abhängigkeit der gesetzten Bits



Als eigentlicher Niederschlagswert sind nur die ersten 12 Bits zu interpretieren. Dies ergibt im Maximum 4095 - im RW geteilt durch 10, da die Werte in Zehntel codiert sind = 409,5 mm. Die Flags machen eine weitere Interpretation dieses Wertes möglich.

3 Praxis: Prinzip des binären Lesens

- Header
- Binärteil
- Besonderheiten
- Praxisbeispiele

- Das RADOLAN-Format hat einen Text-Header und einen nachfolgenden Binärteil (die eigentlichen Pixelwerte).
- **Header lesen**
Header- und Binärteil sind durch das nicht druckbare Zeichen *ETX* (End Of Text) voneinander getrennt. Das Zeichen *ETX* markiert das Header-Ende und ist ein nicht druckbares Zeichen an der Position 4 der ASCII-Tabelle. Um an den Binärteil zu gelangen, muss bis zu diesem Zeichen (inklusive) gelesen werden.
- **Binärteil lesen**
Vorgehensprinzip: Lese jedes Pixel bestehend aus 2-Byte.

Es ist in der Regel nicht erforderlich, jedes **Bit** zu lesen und zu interpretieren. Die Programmiersprachen verfügen über eingebaute Lesefunktionen, die jeweils ein ganzes Byte lesen.

Die Interpretation kann später anhand der aus dem Binär- in das Dezimalsystem umgerechneten Zahlenwerte erfolgen.

Zeilencodierte Werte liest man mit einer typischen kaskadierten Schleife ein, wobei man ein 2D-Array erzeugt. Dabei spricht eine äußere Schleife jede Zeile und eine Innere jede Spalte an. Über die Indizes der beiden Schleifenzähler kann gezielt das Pixel angesteuert werden - meist in der Form `pixel[row_index][col_index]`.

→ veranschaulicht:

Lesevorgang 1:

Header	ETX
--------	-----

Lesevorgang 2.1:

Zeile 1	Pixel 1		Pixel 2		...	Pixel 900
	Byte 1	Byte 2	Byte 3	Byte 4	...	

Lesevorgang 2.2:

Zeile 1	Pixel 1		Pixel 2		...	Pixel 900
	Byte 1	Byte 2	Byte 3	Byte 4	...	

Lesevorgang 2.n

...

Bei der Programmierung eines RADOLAN-Lese/Verarbeitungsprogramms, muss auf folgende Dinge eingegangen werden:

- Datenbeginn:
Die erste Datenzeile ist nicht der Norden, sondern die Daten wurden von der südwestlichen Ecke ausgehend codiert. Dazu gibt es am Ende des Dokuments noch eine Veranschaulichung.
- Byte-Order:
Das RADOLAN-Binärformat ist kein systemunabhängiges Format (wie z.B. NetCDF). Das bedeutet, es muss die Byte-Order berücksichtigt werden! Je nach verwendeter Rechner(CPU)-Architektur stehen die beiden gelesenen Bytes in unterschiedlicher Reihenfolge im Speicher. Evtl. muss dann das Byte-Array getauscht werden.
- Flags (wurde bereits behandelt):
Zusätzliche Informationen zur Interpretation eines Wertes. Da das Flag-Bit nicht zur codierten Niederschlagshöhe beiträgt (nur Markierungscharakter), muss der aus beiden Bytes errechnete Dezimalwert um den dezimalen Wert dieses Bits korrigiert werden:
Gesamtwert *minus* Flag

3 Praxis: Besonderheiten

Südwestliche Ecke ist Beginn der Wertecodierung

- Bei der Interpretation des gelesenen Feldes muss beachtet werden, dass die erste Datenzeile der Datei nicht den Norden, sondern den Südwesten angibt.
- Der Dateizeiger wandert beim Lesen durch die Datei von Süd nach Nord. Daher muss man bei späteren Ausgaben (z. B. als Text) das gelesene Array beginnend vom Ende bis zum Anfang ausgeben (z. B. über eine rückwärts zählende verschachtelte Schleife).

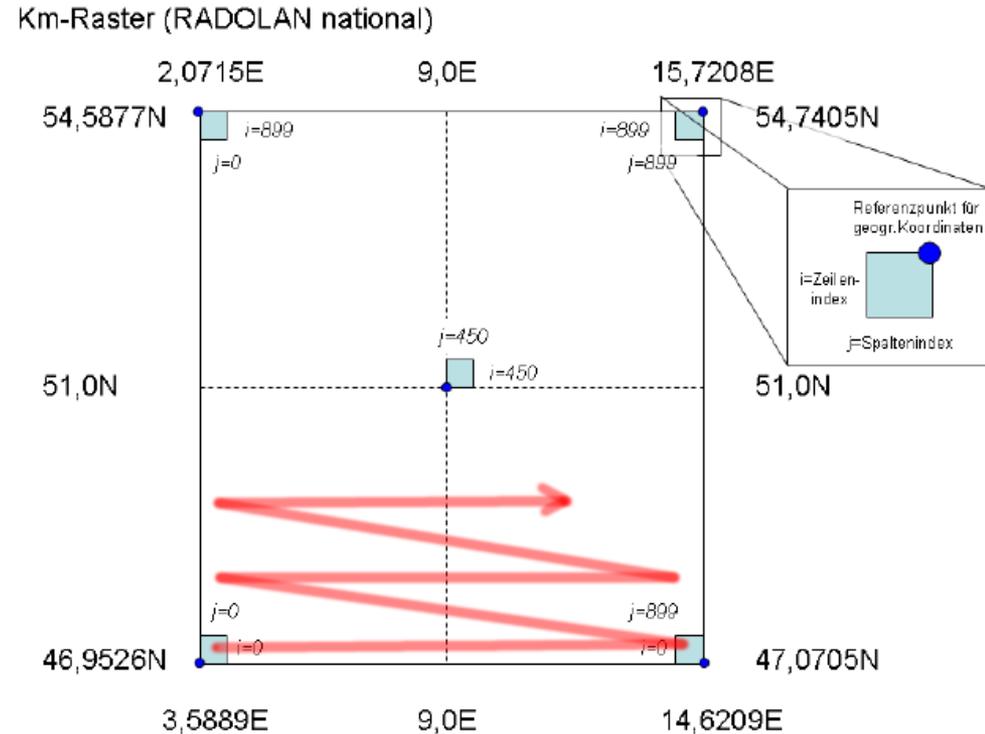


Bild: 1 × 1 km-Raster des nationalen Komposits.
Beginn der Daten ist der Südwesten.

ETX definieren

In Java kann *ETX* folgendermaßen definiert werden:

```
char ETX = (char)0x03;    // Hexadezimaler ASCII-Code
```

Header

Ein Byte (Zeichen) lesen bis *ETX*:

```
int ch;    // Zeichen
// Zeichen(Byte)weise Lesen bis zur Ende-Markierung...
while ( (ch = radolan.read()) != ETX ) {
    ...
}
```

Binärteil

```
// Zeilen von S nach N
for(int y = 0; y < max_rows; y++) {
    // Spalten (einer Zeile) von W nach E
    for(int x = 0; x < max_cols; x++) {
```

Für eine korrekte Nord-Süd-Ausrichtung der Werte direkt beim Lesen bietet sich eine rückwärts zählende verschachtelte Schleife an (so nicht im obigen Beispiel zu sehen), sodass der Index `[0][0]` (Beginn des 2D-Arrays) mit den nördlichen Pixeln belegt ist. So muss nicht nachträglich umgespeichert (geswapt) werden.

In der richtigen Byte-Order einlesen / Swappen

Wie liest man 2 Bytes in der richtigen Byte-Order? Hier ein Beispiel anhand eines Java-Schnipsels:

Da Java die Big-Endian-Byteorder verwendet und die RADOLAN-Daten in Little-Endian-Codierung gespeichert sind, muss das Byte-Array „von hinten nach vorne“ belegt oder später getauscht werden:

```
int BYTES = 2; // "Record-/Bytelaenge" des Produkts
byte[] bytes = new byte[BYTES];

for (int i = BYTES-1; i >= 0; i--)
    bytes[i] = (byte)radolan.readUnsignedByte();

unsignedShortToInt(bytes); // byte[] -> int
```

→ Weg zur RADOLAN-RADVOR-Kompositformatbeschreibung:

1) <http://www.dwd.de/RADOLAN>

2) Seitenleiste rechts: „RADOLAN-RADVOR-Kompositformatbeschreibung“