Note: Transmission format has changed. I have not yet worked that out. KB, March 2007

EFR – Europäische Funk-Rundsteuerung

Telecontrol by transmitting audio tones over power lines has been a well-established technique for decades. Known as "ripple control", it is used by electric power utilities mainly for tariff snot transmitted on the power grid, but over radio.

A longwave-based telecontrol service is provided in Germany by "EFR Europäische Funk-Rundsteuerung GmbH" (loosely translates as "European Radio Telecontrol Ltd"). See *http://www.efr.de* for company details.

Rather than operating their own network, several power utilities now submit their data to a central control, where it is coded and passed to the transmitters. The service costs are based on the amount of data and on the message priority. The higher the priority, the shorter the reaction time until the data is sent out over radio. However, fast reaction is not necessary in most cases, since the receivers are fitted with switching clocks. Usually clock updates rather than immediate switching commands are transmitted.

High receiving levels are necessary, as the receivers are often installed in noisy environments. A field strength of least 1 mV/m in all parts of Germany is ensured with two transmitters. A third one has gone into operation in 2006 Near Budapest. It covers Hungary, Slovakia, Slovenia and parts of Austria

129.1 kHz	DCF49	Mainflingen, near Frankfurt/Main	100 kW
135.6 kHz	HGA22	Lakihegy, near Budapest	100 kW
139.0 kHz	DCF39	Burg, near Magdeburg	50 kW

Data Format

Transmission is asynchronous (that is, with start and stop bit) at 200 baud and a shift of 550 Hz (formerly 340 Hz), with 8 data bits plus an even parity bit. Each telegram consists of 7 header bytes, a user data field of up to 16 bytes, and 2 trailing bytes:

Byte	Meaning / Value
1	68h (start character)
2	length; number of bytes 5…n
3	length (byte 2 repeated)
4	68h (byte 1 repeated)
5	bit 0…3 = reserved, bit 4…7 = telegram number
6	address byte A1
7	address byte A2
8…n	user data, 0…16 bytes
n + 1	check sum; LSB of sum of bytes 5…n
n + 2	16h (stop character)

The upper four bits of byte 5 are incremented for each telegram to a particular address. The two address bytes 6 and 7 are unique for each customer, who in most cases is a power utility. The format of the data field depends on the ripple control system used by that company.

Note: Transmission format has changed. I have not yet worked that out. KB, March 2007

Some addresses have special meanings. A1 = A2 = FFh is used to transmit information to all participants (broadcast). At times the string "DCF49" is transmitted in the data field, with A1 = A2 = FFh.

Time code

In pauses when no user data is transmitted, EFR provides a time synchronization service, with A1 = A2 = 0. A time stamp of 7 data bytes is sent every 12-15 seconds. As far as I could figure out, the 7 bytes D1..D7 are coded as follows, in binary representation:

Data byte	Data bit (bit 0 transmitted first)								Meaning
	7	6	5	4	3	2	1	0	
D1	0	0	0	0	0	0	0	0	
D2	s	s	s	s	s	s	0	0	second
D3	0	0	m	m	m	m	m	m	minute
D4	S	0	0	h	h	h	h	h	hour; bit 7 = summer time flag
D5	w	w	w	D	D	D	D	D	day of week (0 = Monday) and day of month
D6	0	0	0	0	M	M	M	M	month
D7	0	Y	Y	Y	Y	Y	Y	Y	year

The meaning of the fields currently set to zero is not known. Observations show that up to 90 percent of all data bursts heard are time code.

Sample Record

Only byte #5 (here called "C") to byte #n are shown here, in an arbitrary hexadecimal fromat. Characters in brackets are the ASCII equivalent of the data bytes. In case of time stamp data, a human-readable translation is given:

```
C/A1/A2= B7 00 00  D= 00 6C 20 15 84 01 01  Th 04-Jan-01 21:32:27
C/A1/A2= C7 00 00  D= 00 98 20 15 84 01 01  Th 04-Jan-01 21:32:38
C/A1/A2= D7 00 00  D= 00 DC 20 15 84 01 01  Th 04-Jan-01 21:32:55
C/A1/A2= 57 5A 5A  D= 5C 83 01 02 03 80  (\~~~~~)
C/A1/A2= 57 5A 5A  D= 5C 83 01 02 03 80  (\~~~~~)
C/A1/A2= 67 5A 5A  D= 5C 83 01 02 03 40  (\~~~~@)
C/A1/A2= 67 5A 5A  D= 5C 83 01 02 03 40  (\~~~~@)
C/A1/A2= 77 5A 5A  D= 5C 83 01 02 03 20  (\~~~~ )
C/A1/A2= 77 5A 5A  D= 5C 83 01 02 03 20  (\~~~~ )
C/A1/A2= 87 5A 5A  D= 6C 83 01 02 03 E0  (l~~~~~)
C/A1/A2= 87 5A 5A  D= 6C 83 01 02 03 E0  (l~~~~~)
C/A1/A2= F7 FF FF  D= 44 43 46 34 39 20 54 45 53 54  (DCF49 TEST)
C/A1/A2= E7 00 00  D= 00 60 22 15 84 01 01  Th 04-Jan-01 21:34:24
C/A1/A2= F7 00 00  D= 00 98 22 15 84 01 01  Th 04-Jan-01 21:34:38
C/A1/A2= 07 00 00  D= 00 C8 22 15 84 01 01  Th 04-Jan-01 21:34:50
C/A1/A2= 17 00 00  D= 00 1C 23 15 84 01 01  Th 04-Jan-01 21:35:07
C/A1/A2= 27 00 00  D= 00 4C 23 15 84 01 01  Th 04-Jan-01 21:35:19
C/A1/A2= 37 00 00  D= 00 94 23 15 84 01 01  Th 04-Jan-01 21:35:37
C/A1/A2= 47 00 00  D= 00 D0 23 15 84 01 01  Th 04-Jan-01 21:35:52
```

Note: Transmission format has changed. I have not yet worked that out. KB, March 2007

```
C/A1/A2= 57 00 00  D= 00 18 24 15 84 01 01  Th 04-Jan-01 21:36:06
C/A1/A2= 67 00 00  D= 00 48 24 15 84 01 01  Th 04-Jan-01 21:36:18
C/A1/A2= 77 00 00  D= 00 90 24 15 84 01 01  Th 04-Jan-01 21:36:36
C/A1/A2= 87 00 00  D= 00 C8 24 15 84 01 01  Th 04-Jan-01 21:36:50
C/A1/A2= A7 E3 E4  D= 83 80 78 88  (~~x~)
C/A1/A2= A7 E3 E4  D= 83 80 78 88  (~~x~)
C/A1/A2= B7 E3 E4  D= 83 80 78 48  (~~xH)
C/A1/A2= B7 E3 E4  D= 83 80 78 48  (~~xH)
C/A1/A2= C7 E3 E4  D= 83 80 78 28  (~~x()
C/A1/A2= C7 E3 E4  D= 83 80 78 28  (~~x()
C/A1/A2= 97 00 00  D= 00 EC 25 15 84 01 01  Th 04-Jan-01 21:37:59
C/A1/A2= A7 00 00  D= 00 34 26 15 84 01 01  Th 04-Jan-01 21:38:13
```

As can be seen from this example, messages to "real" addresses are sent twice. The telegram number is not incremented for the repetition. A 24-hour log starting 4-Jan-2001 at 2100 CET revealed

- 5059 x time stamp,
- 96 x "DCF49 TEST",
- 709 x user data telegrams (each sent 2 x) to 10 different addresses A1/A2.
  95% of all messages were sent to only two different adresses.

Klaus Betke
April 2001. Updated November 2006