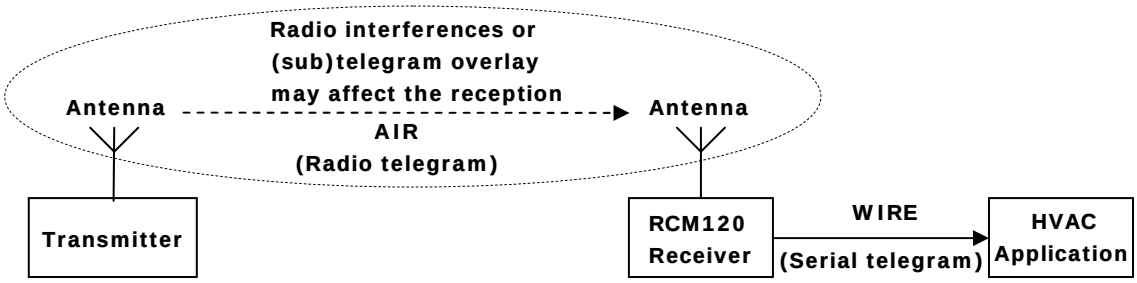
PHANTOM TELEGRAMS –
Cause and remedy of faulty telegrams at serial output

1. What are „Phantom Telegrams“?

“Phantom telegrams" are received telegrams which have a valid telegram structure, but contain a corrupt (unknown) ID or DATA. These faulty telegrams are caused by hazardous radio interferences and can occasionally be present at the RCM and TCM serial receiver interface output.

2. Causes

To understand the possible causes, it is important to know some details regarding the radio transmission. A typical EnOcean radio telegram consists of three identical (redundant) ASK modulated subtelegrams, transmitted within an around 60 milliseconds time slot. Within this 60 ms time received redundant sub-telegrams are combined again by the receiver logic to one serial output telegram. During the decoding of the received radio sub-telegrams (analogue data stream through air) into a digital serial (wired) telegram, a decision must be made due to the signal strength, whether the received radio signal is to be rated digitally as "1" or "0" on the serial wired output. The decoding accuracy (security in the 1/0 distinction) depends on the signal-to-noise ratio and becomes smaller with smaller signal-to-noise ratio. Background: Because of limited resources, only a simple checksum is actually implemented into the RCM receivers, not a comprehensive CRC. If more than one bit is incorrectly decoded and in dependence on the bit location of the affected bits, a (sub)telegram can be regarded as valid by the simplified internal checksum, although it is corrupted. Crucially for the appearance of such “phantom telegrams” is always the signal-to-noise ratio of the received signal.



With lower signal-to-noise ratio the probability for phantom telegrams rises. If for example the signal strength due to range limit distances (also due thick walls or screens) sinks in the background noise, more phantom telegrams are to be expected. A too small signal-to-noise ratio can be also caused naturally by higher background noise, for example by the strong irradiation of a disturber e.g. too close vicinity of the device to a strong source of interference, e.g. a switched (noisy) power supply, a wireless earphone or a mobile weather station using the same frequency, or even unfavorable antenna situations.

Theoretically also the temporal overlay of different (sub)telegrams in the air (collisions) could lead to phantom telegrams. However, compared with the mentioned range limit

cases, this happens very rarely due to the short sub-telegram duration and because this only occurs with strong transmitter irradiated power density.

3. Phantom telegrams appearance and their effects

Due to the described effects and since each telegram consists of 3 identical sub-telegrams, it may be possible that a single sub-telegram is corrupted and interpreted as a new, "phantom" telegram at the receiver. Depending on propagation conditions and location of the corrupted bits, following faulty cases can then statistically occur:

3.1 Sub-telegrams with just 1 bit error (highest probability) are 100% recognized as incorrect from the EnOcean internal checksum logic and are not passed through the serial interface. But in case of at least two corrupted bits this checksum logic can fail and a faulty (sub)telegram will be passed through (false ID and/or false DATA):

3.2 If one of the corrupted bits affect the ID, a "phantom (sub)telegram" (with faulty 32 bit ID, not really existing in this environment) will be "received" and passed through the serial interface. But please note, only with very low probability this faulty ID is known by the application (number of learned IDs over 4 billion error possibilities).

3.3 More problematic is the case when one of the corrupted bit affects the telegram DATA, but not the ID. It means a valid telegram with known ID but faulty DATA will be passed through the serial. This effect is illustrated in the following example (RCM120 received 4BS sensor telegram with ID = D9AC, passed on the wired serial interface):

	Time			ID	DATA	Temp. °C			Chk		Time diff. (s)
13.12.2007	19:16:14	723	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	100.006
13.12.2007	19:17:54	693	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	99.970
13.12.2007	19:19:34	677	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	99.984
13.12.2007	19:21:14	677	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	100.000
13.12.2007	**19:22:54**	**663**	**7**	**D9AC**	**0000690E**	**23,5294118**	**0**	**NA**	**0E**	**OK**	**99.986**
13.12.2007	**19:22:54**	**687**	**7**	**D9AC**	**0080E90E**	**3,45098039**	**0**	**NA**	**0E**	**OK**	**0,024**
13.12.2007	19:24:34	675	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	99.988
13.12.2007	19:26:14	622	7	D9AC	0000690E	23,5294118	0	NA	0E	OK	99.947
13.12.2007	19:27:54	615	7	D9AC	0000690E	23,5294118	0	NA	0E	OK	99.993
13.12.2007	19:29:34	599	7	D9AC	0000690E	23,5294118	0	NA	0E	OK	99.984
13.12.2007	19:31:14	583	7	D9AC	0000680E	23,6862745	0	NA	0D	OK	99.984

This example easily shows (yellow/red marked) that the normally 100 second transmission cycle of the sensor is disturbed by one (sub)telegram coming from the same sensor, following the previous telegram within few milliseconds only (Time: 19:22:54, time difference between both < 60 ms). This is obviously a sub-telegram, has the same ID (D9AC), but other data and therefore is handled as "new" telegram from the same sensor. The two fields marked in red in the DATA show both a "0" bit which is tilted over to a "1" in comparison with the previous "correct" telegram. (e.g. 0000=0 becomes 1000=8 and 0110=6 becomes 1110=E) This last bit alone leads to a dramatic virtual temperature fall (in this case 20°C, from 23°C to 3°C) within few milliseconds. Because such errors are obvious, this error type is the most observed case in-field. Please note that the checksum "0E" represents the renewed calculated checksum of the (wired) serial interface and so has nothing to do with the "on-air radio" transmitted checksum.

4. Measures

4.1 Hardware Improvements

Always improve first of all the HW system performance. Performance depends on several factors in the environment. Factors include: antenna orientation, height, intentional or unintentional interferences. Improve the signal-to-noise ratio: **Make sure the transmitted radio signal intensity is strong enough at the receiver antenna (test with EPM), e.g. optimize the propagation conditions using repeaters**.

4.2 Software Improvements

4.2.1 An effective way to reduce phantom telegrams is to **use a simple SW filter that forwards known IDs only** (because the ID with its 32 bit is relatively long, statistically the ID bits are priory affected).

4.2.2 Please note that in the case 3.3 described above (installation at range limit: Two telegrams with same ID in shortest time interval, but one telegram with corrupted data) the ID filtering only would not be sufficient (only data bits are affected). A suited **additive SW filter for this case is to drop both consecutive (sub)telegrams, when they follow within a very short (milliseconds) time slot**, e.g. within 60 ms (duration time for all redundant sub-telegrams transmitted).

Background: In the example mentioned all observed cases have a remarkable common behavior: Contrary to regular cyclic single transmissions (e.g. every 100 seconds), there are always two different (sub)telegrams of the same telegram coming from the same transmitter ID and received within a time slot of milliseconds only. It is obviously that one of both sub-telegrams is corrupt and since we can't know which one, we could either check their integrity (maybe like a "tolerated" expected range around the previous values) or better simply drop both subtelegrams. So at critical range limits would be anyway generally better to filter out a probably wrong telegram as to have obviously wrong data values. Such a SW filter could be implemented within the application SW following the serial output of the EnOcean module (eg. RCM 120 or TCM 3x0).

Please note that the "comparison time slot" of this filter should not be too large (60 ms should be fine) to avoid the filtering of probably intended valid radio transmitters like e.g. rapid switching telegrams from PTM devices. If such a quick PTM switch timing shall be processed by the application, the filter could be also implemented for 4BS sensor telegrams only.