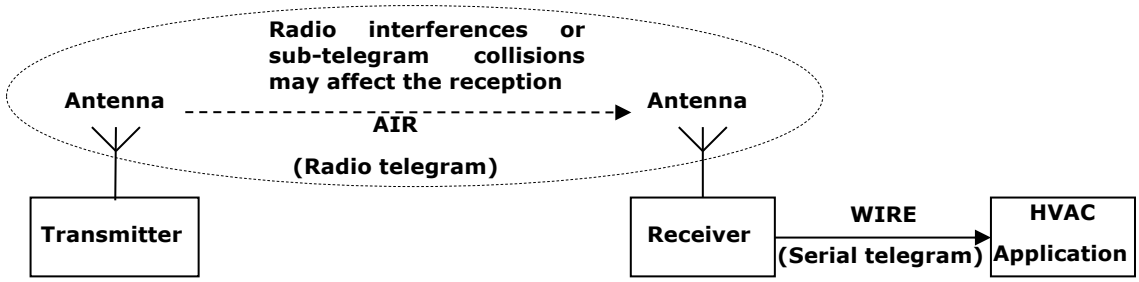
PHANTOM TELEGRAMS –
Cause and remedy of faulty telegrams at serial output

1. What are „Phantom Telegrams"?

"Phantom telegrams" are received telegrams which have an obviously valid telegram structure, but contain a corrupt (inexistent) ID or false DATA. These faulty telegrams caused by hazardous radio interferences can occasionally be present at receiver output as valid ones.

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 sub-telegrams, transmitted within an around 40 milliseconds time slot. Within this period, received redundant sub-telegrams are combined 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 lower signal-to-noise ratio. Background: Because of limited resources, just a simple checksum is implemented in the receivers, not a comprehensive CRC. If more than one bit is incorrectly decoded, 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 by higher background noise, for example by the strong irradiation of a disturber, 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, long range RFID readers or unfavorable antenna situations.

Theoretically, the temporal overlay of more different sub-telegrams in the air (collisions) could lead to phantom telegrams. However, compared with the mentioned range limit cases, this happens rarely due to the short sub-telegram duration and potentially limited number of devices emitting at the same time. On the other side, the use of more repeaters in same environment, all triggered by one single sub-telegram would lead to such collisions.

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 can be passed through (false ID and/or false DATA):

3.2 If occasionally one of the corrupted bits affect the ID, a "phantom telegram" (with faulty ID, not really existing in this environment) will be "received" and passed through the serial interface. Such singular telegrams would however never cause issues, because of their unknown ID.

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 is illustrated in the following example (real received 4BS temperature sensor telegram from ID = D9AC, passed on the wired serial interface):

Time				ID	DATA	Temp. °C					ChkSum	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 24 milliseconds only! (Time: 19:22:54:6xx, time difference between both "different" Telegrams < 40 ms). This time difference is very important to make the distinction between correct and phantom telegram in this case; the second one has same ID (D9AC), but different DATA and therefore is handled as a "new" telegram coming 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" received telegram; **0**000=**0** becomes **1**000=**8** and **0**110=**6** becomes **1**110=**E**. This last bit alone leads to a dramatic "virtual" temperature fall (in this case 20°C, from 23.5°C to 3.5°C) within few milliseconds which could lead e.g. to immediate full activation of heating. Because such errors are obvious, this error type is by far the most observed case in field. Please note that the checksum "0E" represents the renewed calculated checksum of the (wired) serial interface and has nothing to do with the "on-air, radio" transmitted checksum.

4. Measures

4.1 Hardware Improvements

Always improve first the HW system performance. Performance depends on several factors in the environment. Factors include antenna/position optimization, avoiding intentional or unintentional radio interference sources in near receiver environment (incl. integrated switch mode power supplies see EnOcean AN101). Increase the signal-to-noise ratio by optimizing propagation conditions, considering use of maximum two optional 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 (ms) time slot, e.g. within 100 ms receiver maturity time.

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 with different DATA, subsequent sub-telegrams of the same telegram, coming from the same transmitter ID and received within a time slot difference of milliseconds only. It is obviously that one of both sub-telegrams is corrupt and since we cannot know which one, we could either check their data integrity (maybe like a "tolerated" expected range around the previous one) or better simply drop both sub-telegrams. 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 specific end device FW. Please note that the "comparison time slot" of this SW filter should not be too large (100 ms should be fine), to avoid the filtering of probably intended valid radio transmitters like e.g. rapid switching RPS telegrams from PTM devices. If however such quick PTM switch timing is required in the application, above mentioned SW filter could be e.g. implemented for periodical sensor telegram types only.

4.2.3 In case of phantom telegrams obviously generated mostly by too much repeaters (sub-telegrams collision "in the air") we can generally assume that the very first received (original) telegram has the correct DATA, because at the first moment there are no repeated/collision yet in the air. That means automatic drop of eventually following telegram(s) coming from same ID within next 100 ms. (Reducing the repeater number in same environment by optimizing the relative device positions in field would be in this case still the better option).