The Way to DCC-A

Wolfgang Kufer, OpenDCC

The Way to DCC-A

Need to enhance DCC

- Automatic discovery of new decoders, permanent alive
- Larger data transmission in both direction
- Comprehensive readout and configure of decoder functions
- Firmware-Update of decoders
- More bandwidth → DCC 2.0 (next step, not covered here)

But:

- Compatible to existing equipment
- Reuse current hardware
- No change in physical layer, keep DCC bit encoding, keep railcom channels

The Way to DCC-A

Obstacles:

- DCC reception often optimized interrupt routines, nobody wants to change. (esp. in multiprotocol decoders)
- Demanding timing requirements (DCC endbit to start of railcom):
→ no immediate answer, railcom answer is precalculated.
- Limited memory space in decoders
- Limited feature support in decoders (no hardware CRC)
- Limited message length in DCC (booster support, queues in command stations)

The Way to DCC-A

Step 1: Automatic discovery of new decoders

- Need a unique identifier
- Need to transport this identifier as ‚atomic' object.
- Choosen: 40 bits = 12 bits manufacturer, 28 bits product+serial
- Consequence: downstream: longer DCC messages, upstream: coupling of railcom channel 1 and channel 2.
- Ability to know decoder properties before sign on → address mode with unique identifier required.
- Compact information summary in data space, transported as whole

The Way to DCC-A

Do it fast:

- Streaming required, no ping-pong
- Select a target (with unique identifier), use shortest possible DCC message for streaming

Do it safe:

- Avoid most error sources seen in practice by evaluating both half bits
- Target Selection requires not only checksum ok, but additionally complete 40 bit match of unique identifier.
- Secure subpackets with CRC.

The Way to DCC-A

Step 2: Transfer large data blocks

- Use same technique (select with identifier and transport)?
- Use same technique (select with classical address and transport)?
- Or use classical address space (and only channel 2)?
- Reuse same streaming technique as for sign on.
- Result: maximum bandwidth.

The Way to DCC-A

Step 3: readout and configure of decoder functions

- Build up a hierarchical mapping of physical effects to logical operations.
- Support min, max and default values for configuration
- Easy extendable
- Compact definition (similar to mdns/bonjour)
- Currently work in progress

Step 4: Firmware Update

- Use the streaming technique for firmware update …

The Way to DCC-A

Implementation

- Implemented in command station
- Implemented in feedback modules
- Implemented in test decoders
- Prove-of-concept presented at Intermodellbau Dortmund
- Testhardware and software available

The Way to DCC-A

Outlook: how to get more bandwidth?

- Use more timing granularity for bit coding
- Use second half bit for bit coding as well
- Coding of more symbols per state change (i.e. 8 values: 50µs=000, 60µs=001, 70µs=010, … 120µs=111)
- Use DC balancing bits to keep overall DC at zero
- Use FEC to get more messages with successful reception