9 LocoNet

From Z21 FW Version 1.20.

As mentioned in the introduction, the Z21 can be used as **an Ethernet/LocoNet gateway**, where the Z21 is also the LocoNet master refreshing the slots and generating the DCC packets.

The LAN client can subscribe to the corresponding LocoNet messages using ***2.16*** LAN_SET_BROADCASTFLAGS in order to receive also the messages from LocoNet.

Messages received by the Z21 from the LocoNet bus are forwarded to the LAN client with the LAN header ***LAN_LOCONET_Z21_RX***.

Messages sent by the Z21 onto the LocoNet bus are also forwarded to the LAN client using the LAN header ***LAN_LOCONET_Z21_TX***.

With the Z21-LAN command ***LAN_LOCONET_FROM_LAN*** the LAN client itself can write messages onto the LocoNet bus. If there are other LAN clients with LocoNet subscriptions at the same time, they will also be notified with a message ***LAN_LOCONET_FROM_LAN***. Only the actual sender will not be notified.

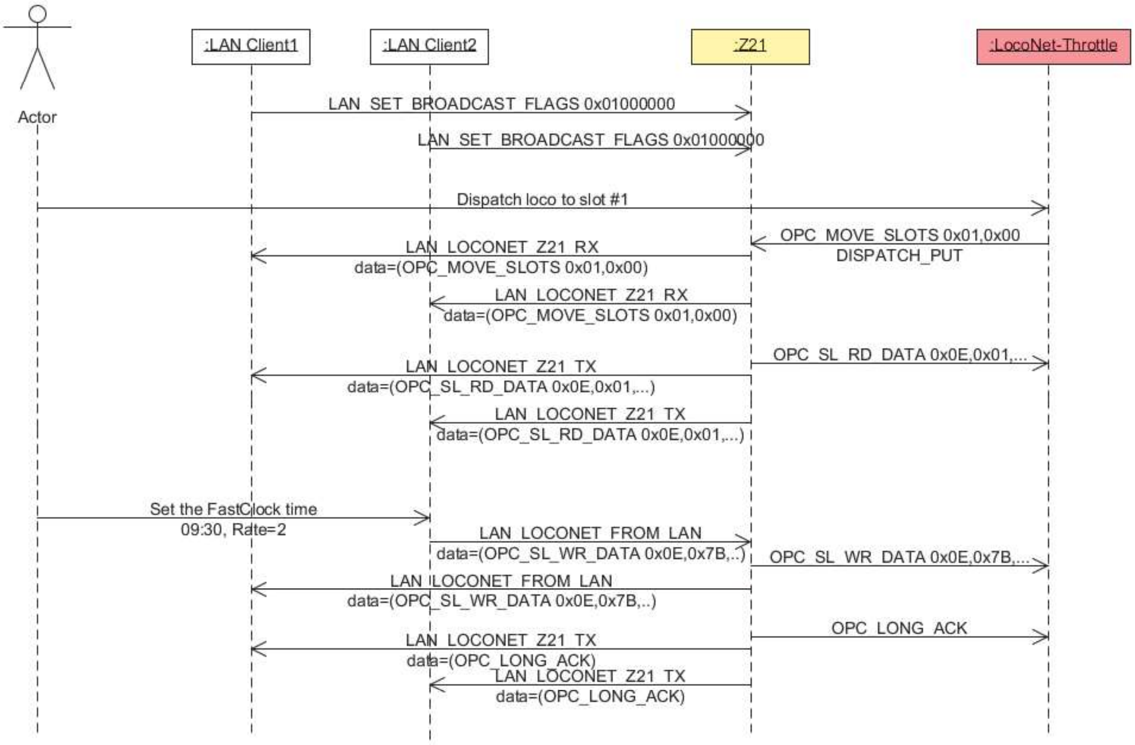

Figure 8 Example Sequence: Ethernet/LocoNet gateway

This example shows that even with trivial processes on the LocoNet bus, considerable network traffic can simultaneously occur on the Ethernet or Wi-Fi.

Please note that this Ethernet/LocoNet Gateway functionality has primarily been created for PC control SW as an additional tool for communicating with e.g. LocoNet feedback devices, etc.

When subscribing to the LocoNet messages, you should carefully consider whether the broadcast flags 0x02000000 (LocoNet locomotives) and 0x04000000 (LocoNet switches) are really necessary for your application. For conventional driving and switching, in particular, you should better use the LAN commands already described in chapters ***4*** Driving, ***5*** Switching and ***6*** Reading and writing Decoder CV. The actual LocoNet protocol is not described in more details in this specification. Please directly contact Digitrax or the manufacturer of the respective LocoNet hardware, especially if that manufacturer has extended the LocoNet protocol for e.g. configuration purposes etc.

9.1 LAN_LOCONET_Z21_RX

From Z21 FW Version 1.20.

This message is asynchronously reported to the client by the Z21 when the client

- activated the corresponding broadcast, see ***2.16*** LAN_SET_BROADCASTFLAGS, Flags 0x01000000, 0x02000000 or 0x04000000.
- and a message has been received by the Z21 from the LocoNet bus.

Z21 to Client:

DataLen		**Header**		**Data**
				LocoNet message incl. CKSUM
0x04+n	0x00	**0xA0**	0x00	n Bytes

9.2 LAN_LOCONET_Z21_TX

From Z21 FW Version 1.20.

This message is asynchronously reported to the client by the Z21 when the client

- activated the corresponding broadcast, see ***2.16*** LAN_SET_BROADCASTFLAGS, Flags 0x01000000, 0x02000000 or 0x04000000.
- and a message has been written to the LocoNet bus by the Z21.

Z21 to Client:

DataLen		**Header**		**Data**
				LocoNet message incl. CKSUM
0x04+n	0x00	**0xA1**	0x00	n Bytes

9.3 LAN_LOCONET_FROM_LAN

From Z21 FW Version 1.20.

This message allows a LAN client to write a message to the LocoNet bus.

This message is also asynchronously reported by the Z21 to a client when the client

- activated the corresponding broadcast, see ***2.16*** LAN_SET_BROADCASTFLAGS, Flags 0x01000000, 0x02000000 or 0x04000000.
- and **another** LAN client has written a message to the LocoNet bus via the Z21.

LAN client to Z21, or Z21 to LAN client:

DataLen		**Header**		**Data**
				LocoNet message incl. CKSUM
0x04+n	0x00	**0xA2**	0x00	n Bytes

9.3.1 DCC Binary State Control Instruction via LocoNet OPC_IMM_PACKET

From Z21 FW Version 1.42 on, the new command *4.3.3 LAN_X_SET_LOCO_BINARY_STATE* is recommended for switching binary states instead of using the method described below.
However, the following paragraph text, which is now somewhat outdated, remains for the sake of completeness:

From FW Version V1.25 any DCC packets can be generated at the track output using LAN_LOCONET_FROM_LAN and the LocoNet command OPC_IMM_PACKET, among them the Binary State Control Instruction (also called "F29...F32767"). This also applies to the white z21, which has no physical LocoNet interface, but however has a virtual LocoNet stack inside.

For the structure of the OPC_IMM_PACKET see LocoNet Spec (also in “personal edition” for learning purposes). For the structure of the Binary State Control Instruction see NMRA S-9.2.1 Section “Feature Expansion Instruction”.

9.4 LAN_LOCONET_DISPATCH_ADDR

From Z21 FW Version 1.20.

Prepare a loco address for the LocoNet dispatch.

This message allows a LAN client to prepare a specific locomotive address for the LocoNet dispatch. This corresponds to a "DISPATCH_PUT" and means that at the next "DISPATCH_GET" (triggered by handset controller) the slot belonging to this loco address is reported back by Z21. If necessary, the Z21 automatically occupies a free slot for this purpose.

Request to Z21:

DataLen		**Header**		**Data**
0x06	0x00	**0xA3**	0x00	16 bits Loco address (**little endian**)

Reply from Z21:
Z21 FW Version < 1.22: none
Z21 FW Version ≥ 1.22:

Z21 to Client:

DataLen		**Header**		**Data**	
0x07	0x00	**0xA3**	0x00	16 bits Loco address (**little endian**)	**Result** 8 bit

Result 0 The "DISPATCH_PUT" for the given address failed.
This can happen, for example, if the Z21 is operated as a LocoNet slave and the LocoNet master has rejected the dispatch request because this locomotive address is already assigned to another handset.

>0 The "DISPATCH_PUT" was executed successfully. The loco address can now be transferred to a handset controller (e.g. FRED). The value of Result corresponds to the current LocoNet slot number for the given loco address.

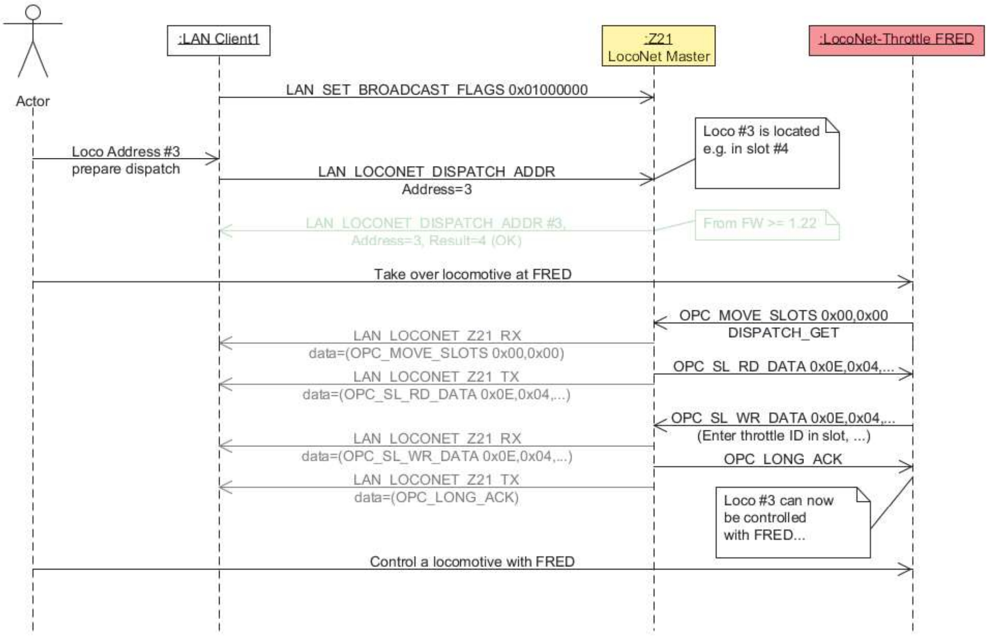

Figure 9 Example Sequence: LocoNet Dispatch per LAN-Client

9.5 LAN_LOCONET_DETECTOR

From Z21 FW Version 1.22.

If LAN client application wants to support a LocoNet track occupancy detector, there are two ways. The first would be to receive the LocoNet packets via ***9.1*** *LAN_LOCONET_Z21_RX* and process the corresponding LocoNet messages directly. However, this requires an exact knowledge of the LocoNet protocol and it would produce a lot of network traffic.

Therefore the following alternative was created, with which you can **poll** the occupied status **as well as** be **asynchronously informed** about a change of the occupied status, without having to go into the depths of the LocoNet protocol.

Information: please note the following essential difference between the Roco Feedback Module 10787 on the R-BUS (see ***7*** Feedback **–** R-BUS) and LocoNet Track Occupancy Detectors:

- 10787 is normally connected with mechanically operated switching contacts, which can be closed and reopened per axis of the train running over it.

- LocoNet track occupancy detectors are usually based on exact current measurement at the monitored track section or on advanced technologies (transponder, infrared, RailCom, ..) in order to reliably determine the occupancy state of the track. During normal operation, ideally only one message is generated when the occupied state changes.

The following command can be used to poll the status of one or more track occupancy detectors.

Request to Z21:

DataLen		**Header**		**Data**	
0x07	0x00	**0xA4**	0x00	**Type** 8 bit	16 bits report address (**little endian**)

Type **0x80** Request via "Stationary Interrogate Request" (SIC) according to Digitrax procedure. This procedure also can be used for the occupancy detectors from Blücher-Elektronik. The parameter “report address” is 0 (not relevant).

0x81 Request via so-called **report address** for Uhlenbrock occupancy detector.
This report address can be configured by the user e.g. UB63320 via LNCV 17 in the occupancy detector. The default value there is 1017.
With type 0x81, this report address is only used for polling and should **not be confused** with the **feedback address**.
Note: At the LocoNet bus this query is implemented via turnout switching commands, therefore the value according to LocoNet has to be decremented by 1. Example:

```
0x07 0x00 0xA4 0x00 0x81 0xF8 0x03
```

means: "request status of all occupancy detectors with report address 1017
(Report address = 1017 = **0x03F8** +1 = 1016 + 1)"

0x82 **Status request for LISSY, from Z21 FW Version 1.23**
On the other hand, for Uhlenbrock LISSY the report address corresponds to the feedback address however. The type of the subsequent feedback message(s) strongly depends on the configured operating mode of the LISSY receiver. In the LISSY manual you can find out more about the extensive setting options of the LISSY receiver.

Please note that in the case of a single request, several occupancy detectors may be addressed at the same time, and therefore multiple responses are to be expected. Depending on the manufacturer of the occupancy detector, the status of the same input can be also reported several times after one request.

Reply from Z21:

Z21 to Client:

DataLen		Header		Data		
0x07 + ***n***	0x00	**0xA4**	0x00	**Type** 8 bit	**Feedback address** 16 bits (**little endian**)	**Info[*n*]**

This message is asynchronously reported to the client by the Z21 when the client

- has activated the corresponding broadcast, see ***2.16*** LAN_SET_BROADCASTFLAGS*,* Flag 0x08000000
- and the Z21 has received a corresponding message from a track occupancy detector **due to a status change on its input**, or **due to an explicit status-request (polling) by a LAN client** using the commands described above.

Feedback address Each input of an occupancy detector has its own feedback address, which can be configured by the user (e.g. for Uhlenbrock and Blücher via LNCV), and which describes the monitored block with an unique address.

Info[*n*] Byte-Array; content and length ***n*** depending on **Type**, see below

Type **0x01** For occupancy detector types like Uhlenbrock 63320 or Blücher GBM16XL reporting only the status "occupied" and "free" (by using LocoNet OPC_INPUT_REP, X=1).

n=1

Status of the input belonging to the feedback address can be found in **Info[0]:**
Info[0]=**0** ... sensor is **LOW** ("free")
Info[0]=**1** ... sensor is **HIGH** ("occupied")

0x02 **Transponder Enters Block**
0x03 **Transponder Exits Block**
For occupancy detectors such as Blücher GBM16**XN** etc. reporting also the information about the vehicle (e.g. locomotive address) inside the block to the command station (by using LocoNet OPC_MULTI_SENSE transponding encoding from Digitrax).
In addition to the feedback address, also a so-called transponder address is transmitted. The **transponder address** identifies the vehicle in the block. In the case of the GBM16**XN**, this is the **locomotive address** which was determined by the occupancy detector by using RailCom.

n=2

The transponder address is located in **Info[0]** und **Info[1]**, 16 Bit little endian:
Info[0] ... transponder address low byte
Info[1] ... transponder address high byte

Remark: due to lack of specification inside the LocoNet spec, the value ranges in OPC_MULTI_SENSE is not quite clear, which leaves the manufacturers of the occupancy detectors sometimes in the dark. Therefore in the case of GBM16XN the following must be observed according to our experience:

- You have to add +1 to the feedback address to get the feedback address configured in GBM16**XN**.
- Depending on the configuration of the GBM16**XN**, the direction of the vehicle on the track can also be coded in the bit under the mask 0x1000. Such a configuration is not recommended because this bit collides with the address space of long locomotive addresses!

0x10 LISSY Loco address from Z21 FW 1.23.
This message is sent to the Z21 LAN Client when an Uhlenbrock LISSY receiver reports a vehicle equipped with a LISSY transmitter and this LISSY receiver is configured to the "Transfer format (ÜF) Uhlenbrock" (LNCV 15=1). Furthermore, this message strongly depends on the configured operating mode (LNCV2, ...) of the LISSY receiver.
See also LISSY manual.

n=3

The Loco address is located in **Info[0]** und **Info[1]**, 16 Bit little endian:
Info[0] ... loco address low byte
Info[1] ... loco address high byte
Locomotives have a value range from 1..9999
Wagons have a value range from 10000 to 16382

Info[2] ... Additional info according to following bits: 0 **DIR1 DIR0** 0 **K3 K2 K1 K0**
DIR1=0: **DIR0** is to be ignored
DIR1=1: **DIR0**=0 is forwards, **DIR0**=1 is backwards
K3..K0: 4 bit "class information" stored in the LISSY sender.

Example Configuration for Lissy Receiver 68610:

LNCV	Value	Comment
2	98	optional module reset: sets all LNCV to 0, except LNCV 0 and 1 (address)
2	0	Basic function: readout locomotive data and direction information by using double sensor
15	1	Send using "Transfer format (ÜF) Uhlenbrock" onto LocoNet

0x11 LISSY block status from Z21 FW 1.23.
This message is sent to the Z21 LAN Client when an Uhlenbrock LISSY receiver sends a block occupancy status message using the "Transfer format (ÜF) Uhlenbrock". See also LISSY manual.

n=1

Status of the block belonging to the feedback address is in **Info[0]**:
Info[0]=0 ... block is free
Info[0]=1 ... block is occupied

Example Configuration for Lissy Receiver 68610:

LNCV	Value	Comment
2	98	optional module reset: sets all LNCV to 0, except LNCV 0 and 1 (address)
2	22	Automation function with block status message: Time-controlled waiting station
3	2	Automation active in both driving directions
4	3	Wait time 3 seconds
10	2	Block option: Block status change to "free" after 2 seconds
15	1	Send using "transfer format (ÜF) Uhlenbrock" onto LocoNet

0x12 **LISSY Speed from Z21 FW 1.23.**
This message is sent to the Z21 LAN Client when a Uhlenbrock LISSY receiver is configured for speed measurement.
See also LISSY manual.

n=2

The speed is located in **Info[0]** and **Info[1]**, 16 bit little endian:
Info[0] ... speed low byte
Info[1] ... speed high byte

Example Configuration for Lissy Receiver 68610:

LNCV	**Value**	**Comment**
2	98	optional module reset: sets all LNCV to 0, except LNCV 0 and 1 (address)
2	0	Basic function: readout locomotive data and direction information by using double sensor
14	**15660**	**Velocity Scaling factor =** **1566 (H0-scale) * 10mm (sensor distance)**
15	1	Send transfer format Uhlenbrock to LocoNet

Note: **Type** will be extended by further IDs in the future as required.