

CANx protocol introduction

KNX was made 25 years ago. When newest specification was made **25 years ago**, such things as **data encryption/security, usability was not so important**. Nobody thought about ecosystem concept (not only protocol specification, but also universal way of commissioning)

Our idea was to choose the best from existing wired home and building automation standards (KNX, ModBus, BacNet)

Disadvantages of KNX

- **Low speed:** 9.6 kbit (on today this is way too low)
- **Security** (cannot be made without full redefinition of standard)
- **Not fully open standard** (meaning you can participate if you are part of the alliance and pay your yearly or product fees)
- Commissioning and operation with KNX is **centralized through ETS** which is a property of KNX Association. Only those companies who are members of KNX Association have possibility to manufacture devices for KNX
- KNX only targets home and building automation. **It is missing concept for modern IoT** approach (universal bus for any systems e.g. automation of machines etc.)
- KNX is also **missing** some modern features like **semantics**

Our idea

- Our idea was to improve drawbacks of KNX but leave same user-experience and compatibility.
- Our idea was to create **something new but based on history-proven technologies** e.g. *in aviation industry absolutely new plane is the one which has max 25% of changes. This is correct because you need to use proven solutions. We maximally use positive/negative experience of other protocols*

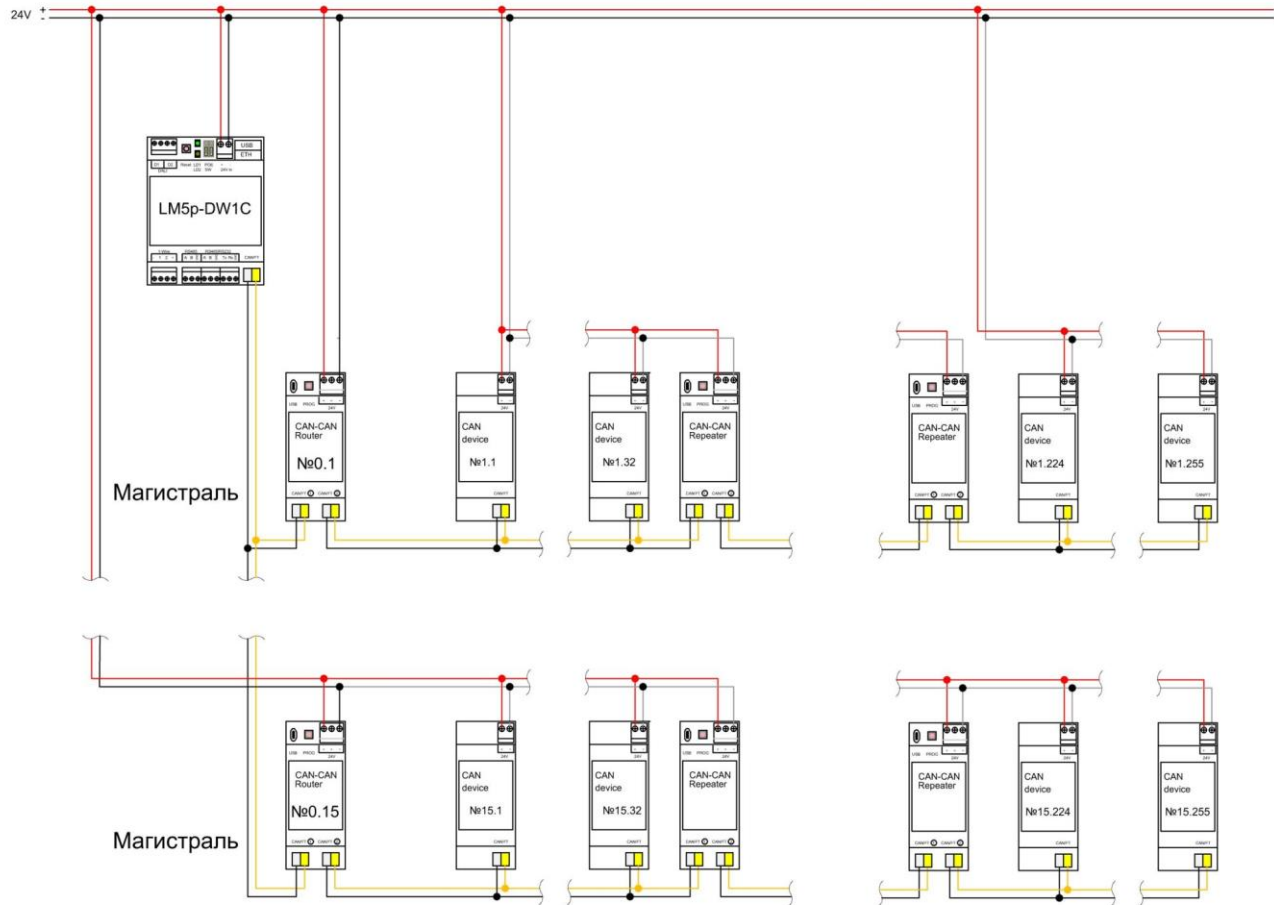
- Without any compromise in quality/functionality we have created our specification based on K.I.S.S. approach (**Keep it simple stupid**)
 - *Installed devices should be possible to discover and be identified in an uniform way*
 - *Devices should be able to be configured in a uniform way*
 - *Autonomous/distributed device functionality*
 - *Uniform way to update/maintain device firmware*

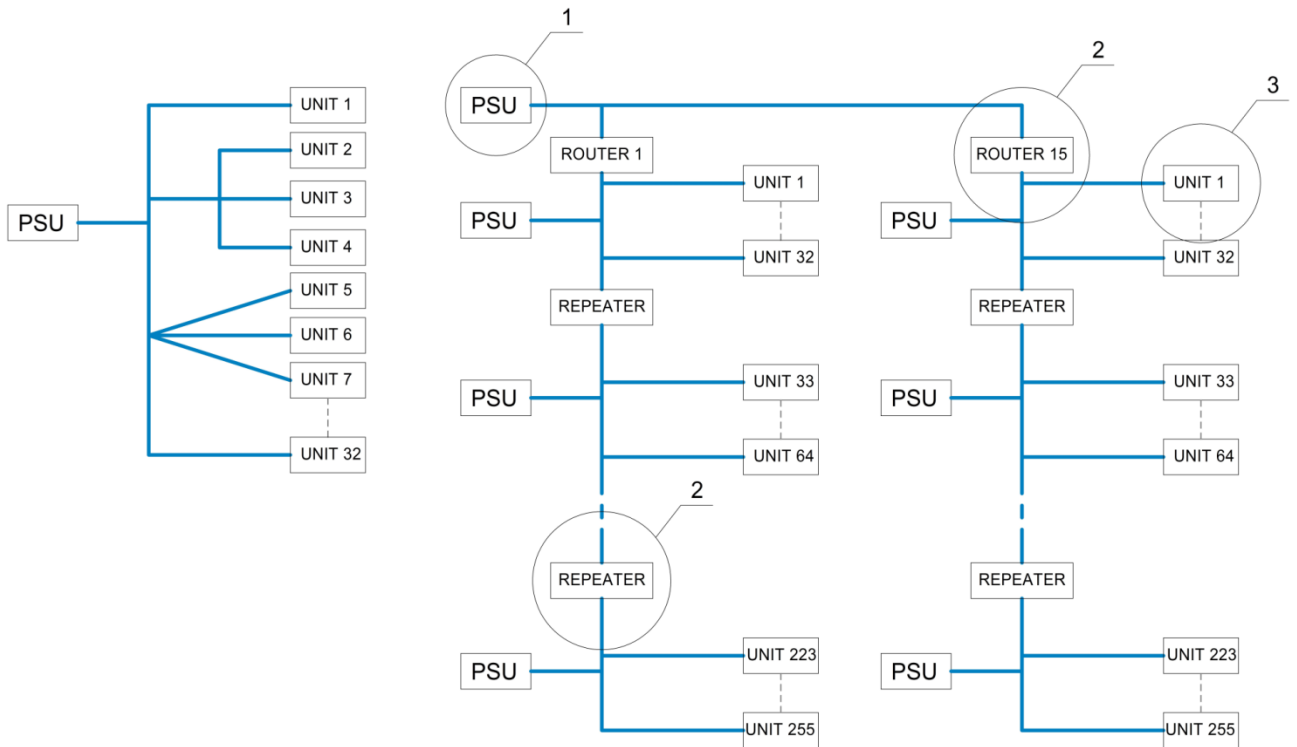
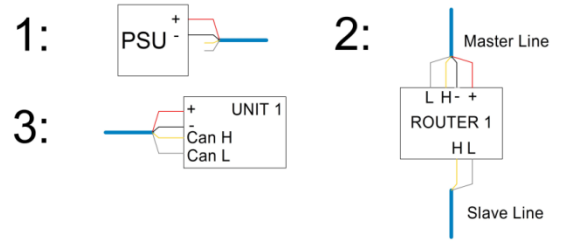
Our solution based on CAN FT specification

- We made **1:1 mapping to KNX** (you don't need to change group addresses, data-types), it will work plug&play
- We improved the bus **speed in >10 times** compared to KNX
- The system can work as **plug&play solution** – just plug the device to fieldbus on the fly and push the button:
 - the device will automatically send info about hw and sw installed in it
 - the system will know exact info about this device and provide address to it
 - the system is ready to work
 - we have two types of communication – direct and pub-sub
- We have chosen **CAN FT** as basic media. *Fault-Tolerant (FT) CAN physical layer maintains functionality in the case of a broken or shorted bus wire. This is of great importance in areas where such failures can occur of such a fault condition occurs the FT-CAN node simply switches to single-wire operation. Automatic detection of bus system recovery then automatically returns the transceiver to differential mode when the failure has been cleared.*
- CAN FT is the most fresh version of international standard **ISO 11898-3:2006**
- CAN is **time and harsh environment proven** technology
- CAN is popular fieldbus system in space, automotive, aerospace, railway industries
- We can use our CAN devices **without additional termination resistor**

Topology

We can use them in **any topology** (except loop) – Line, Star, Tree etc.





Router – CAN-CAN router/repeater with filtering table. LogicMachine with CAN FT interface acts as router as well.

Repeater – simple version of CAN-CAN, used to repeat signal for each 32 CANx devices in one line

- Group communication has communication objects (the same as KNX) with numbers 0..65535
- Direct communication has source and target addresses based on Line ID(0..15), Node ID(0..255), Object ID (0..255)
- Data per message can vary from 0 to 8 bytes
- Communication object has own EEPROM place with configuration and flags (1byte)
- Number of communication object equal the same number for EEPROM address (CommObj_15 -> EEPROM(15))
- The byte is divided to 2 parts (4 bites + 4bites). High part of byte for Configuration of communication object (0..15)
- 4 low bites for Flags -> 3 - is writable, 2 - is readable, 1 - is transmittable, 0 - is filtered

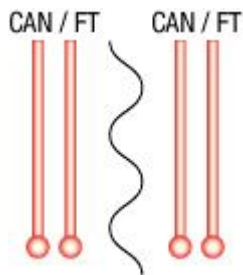
CAN FT bus powering

As the transfer speed is higher in CAN FT fieldbus, we use additional 2 wires for powering. It shouldn't be issue as all KNX installations are using 4-wire cable where 2 cables can be used to this task. Plus, it's much easier to make devices fully galvanically isolated in CAN.

We recommend to use KNX certified cable or fire alarm cable which features two twisted pairs of solid conductor with a foil screen 2x2x0.8.



There is no current limitation for 24V DC power supply to use for CAN FT device powering



Each 32 devices are galvanically separated with router / repeater and own power supply should be used for every 32 devices

Distance limitations

Max distance of one CANx line is 300m. You have to use max number of devices (32) in such line to operate it. If smaller number of devices is used in long distance, you might need to use Active Line filters which we supply. Also you can use line repeaters which increase line length twice.

We've created a calculator tool for max theoretical cable length (regular KNX cable is usually 100pF/m (or 100nF/km)):

<http://canx.info/calc/>

Media

- We have implemented in this specification also for IP network transport
- You can **use any IP networks like Ethernet, WiFi, Optics, 3G/4G** etc.
- We have also implemented **CaNX over BLE**. This can be used on any BLE compatible device from version 4.0 to 5.0
- Also we can use any realization of the **802.15.4 protocol** (6LoWPAN, ZigBee)

User experience

- We use the same **WAGO connector** as in KNX which is easy to plug, easy to maintain.
- Additional features are implemented like **ability to blink LED on device to find it**, locate devices in address programming state.
- We added **additional LED for indication of communication** (if a message is received, it blinks)
- CAN checks and works even when:
 - Interrupt on CAN_H
 - Interrupt on CAN_L
 - **Short circuit** between CAN_H and GND, short circuit between CAN_H and VCC, short circuit between CAN_L and GND, short circuit between CAN_L and VCC, short circuit between CAN_H and CAN_L

Self-learning

- In CAN bus if Line Repeater is installed, it can do auto physical and group address assignments to actuators and sensors
- Line repeater listens the bus all the time. When we press programming button on CAN actuator/sensor, Line Repeater send physical address to it and device overwrites its default physical address
- When physical addresses are programmed, we press again programming buttons on both actuator and input device. There are complementary pairs defined which means if complementary numbers are the same on both devices, their ports will be paired automatically with group addresses from grp address register

Security

- It is possible to **disable direct telegrams** (you cannot change parameters, addressing). Only group communication works. This can be enabled/disabled with one telegram
- You can check the **status of each device** (ping). Each device replies with count of errors (e.g. how many telegrams was not sent/received well)
- Health monitoring. You **can check each sensor's functionality and errors** (e.g. when you ping the object, it returns the error code – connection is broken, short-circuit etc.)

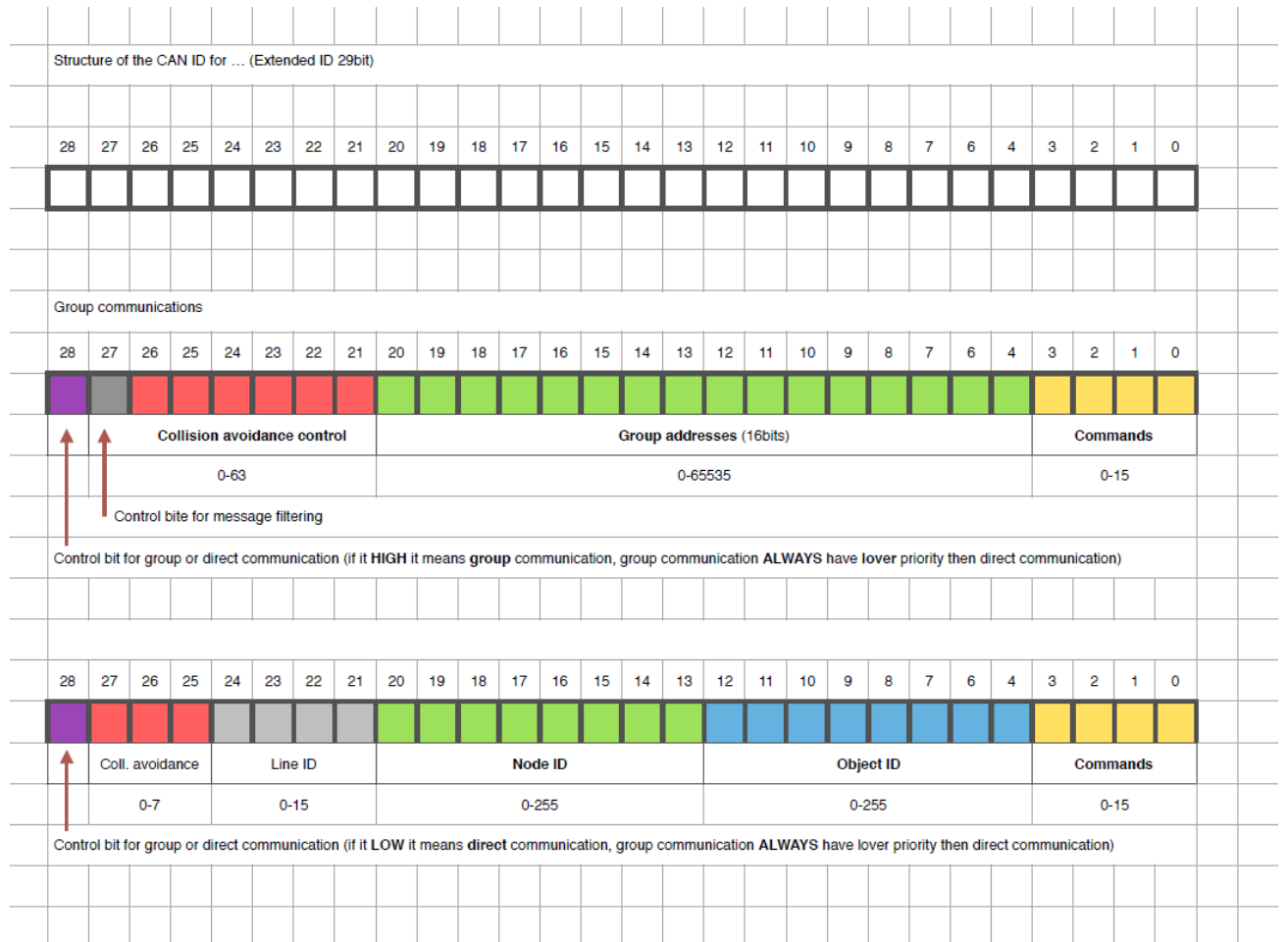
Commissioning tool

There is a common commissioning tool for CAN devices. It works as **web-application** (part of LogicMachine) or we plan also to implement **stand-alone application** for MAC, Windows and Linux.

- You **can import ETS project** and enrich this data with semantics.
- Each device has it's unique hw and sw ID with resource description file
- After a small interoperability test this file is included in a common list. You can **have online and local list of all device description files**. Online list updates automatically.

Telegram structure

Extended CAN frame with 31 bit header.



Direct device communication

DeviceIDRead	Read Hardware ID and Software ID for given device, device status and device Objects Quantity
DeviceIDResponse (8 bytes)	Response for ..., return ObjQty (byte), status (byte), 5 bytes for HW id and 1 byte for SW id
DeviceAddrRequest (8 bytes)	Request for address for this device (current Line ID and Node ID) with HW+SFT id as message DATA
DeviceAddrWrite	Return: Line ID (d[1]), Node ID (d[0]) and Master Code for device 6 bytes (d[2] - d[7])
DeviceConfigRead (2 bytes)	EEPROM read command for given Address (2 bytes DATA)
DeviceConfigResponse (8 bytes)	Return 8 byte of data from EEPROM for given start address
DeviceConfigWrite (8 bytes)	Write 6 bytes of data with 2 bytes address
DeviceConfigWriteResponse (1 byte)	1 if write is successful of 0 if failed
DeviceLedOn	Switch the device green LED to ON state
DeviceLedOff	Switch the device green LED to OFF state
DevicePing	Ping request to device with given Line ID and Node ID
DevicePong	Response on Ping request with CanBus TX and RX errors and maximum q-ty of GAs per Object

Group communication

GroupValueRead	Request of Value for given Group address
GroupValueResponse (1..8 bytes)	Response for GroupValueResponse command, return Value for given Group address
GroupValueWrite (1..8 bytes)	Write Value to given Group address

Control

DirectMsgOn (8 bytes)	Switch on direct communication with code (master_code = Fx(code))
DirectMsgOff (8 bytes)	Switch off direct communication with code (master_code = Fx(code))
SecureGAOn (8 bytes)	Switch to Secure GA mode with code (master_code = Fx(code))
SecureGAOff (8 bytes)	Switch Off Secure GA mode with code (master_code = Fx(code))

Connecting to KNX network

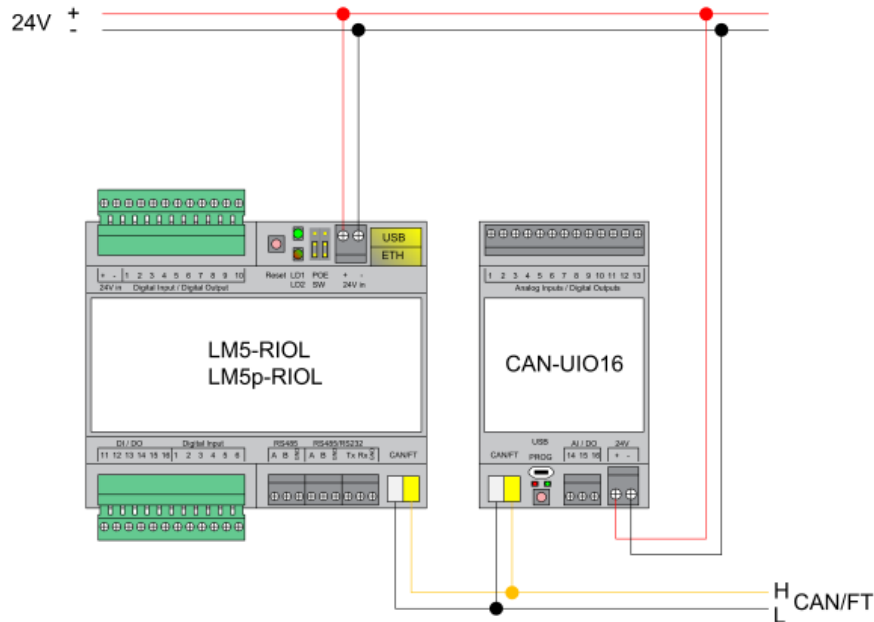
For connection CAN net you need to have gw witch will convert group telegrams media, direct telegrams must be blocked. Group telegrams will be transferred without changes exclude data-types with length of data more than 8 bytes.

Using online utility you can import ETS projects and later upload this file to KNX-CAN gateway.

Frequently Asked Questions

1. Q: Do I understand correctly that there is 4 wires needed – 2 x powering wires and 2 x fieldbus wires?

A: Yes, you need to power each CAN device with 24V DC power wire and need to have CAN FT fieldbus wires:



2. Q: Which elements are powered by PSU 1?

A: PSU 1 powers Unit 1 – 32 and Router 1.1

3. Q: Is Router 1 connected to its lower segment only via CAN FT twisted pair?

A: Yes. It is recommended to place Unit 1 to its upper Router as close as possible because here we can lose possibility to use power cables to transfer data (e.g. in case for short-current on H and L)

4. Q: Why there is 32 device limitation per one IP Router / PS

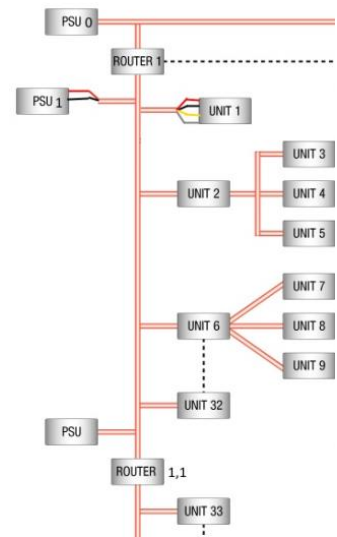
A: Only limitation is power requirement and additional galvanic isolation

5. Q: What is included in these 32 devices – Router too?

A: Yes, the Router is included in these 32 devices

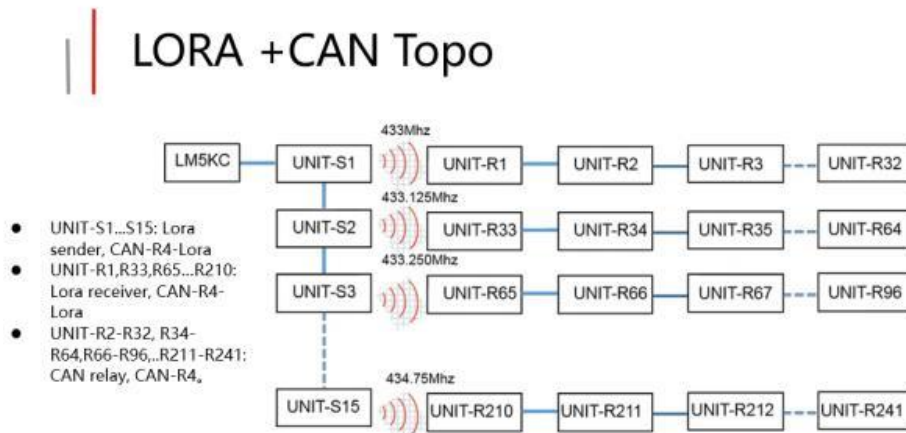
6. Q: Does the Router have physical address?

A: Yes



7. Q: Do I use CANx Router when I am creating different CAN FT lines? If I have only one CAN FT line, I don't have to use CANx Router? The CANx Repeater I use when I have more than 32 device in the same CAN FT Line?
A: Yes
8. Q: How many CANx Repeaters I Can Use in the same CAN FT line?
A: 8
9. Q: There is filtration Flag. If it is enabled, it means that the telegrams do not pass outside their segment?
A: The filtering on the router is made as follows. There are 3 attempts on the Router:
- Ignore filtering Flag
 - Block telegrams with filtering Flag
 - Allow only telegrams with filtering Flag
10. Q: Does LM have physical CANx address
A: In telegram there is no physical address sent, only group communication address or direct communication address
11. Q: Which is line number where Routers are placed?
A: 0
12. Q: Where can be LM placed in this diagram?
A: There is no limitation for LM installation location in this diagram
13. Q: I connected LM5 with its 24VDC PSU and CANx device with own 24VDC PSU, but I cannot find devices in Line scan. What is the reason?
A: You have to unite 0V of both PSUs, especially when PSUs are located far apart. Also, please make sure your power supplies support operation in parallel (.e.g Phoenix Contacts do support). Another option is to run LM and CANx devices from one PSU then you don't have to worry about GND loops. Also, if you have only one or two devices long distance away from LM, you might need to use special termination devices. 16

CANx and LoRa interrelation



Wired CANx bus is limited to 32-36 devices in total.

Each CANx line can have up to 255 devices. CANx-LoRa only accepts telegrams that have the same line ID. This only applies to CANx-LoRa devices. Wired communication does not take line ID into account only group address is used. This means that in your topology only S1-R1 units and so on must be on the same line. Line id does not matter for R2, R3 etc.

If you want to separate networks by frequency, then you must use larger gap than 0.125MHz. From our tests only 0.5MHz and higher guarantee full separation, smaller difference can cause interference. This also applies to repeaters. Make sure to increase hop count if you want repeaters to work. By default, telegram can pass only one radio hop.

For relay control it will be enough to have less sending devices because the communication is mostly one-way.