Real-time Ethernet Deterministic BUS

What is REDBUS?

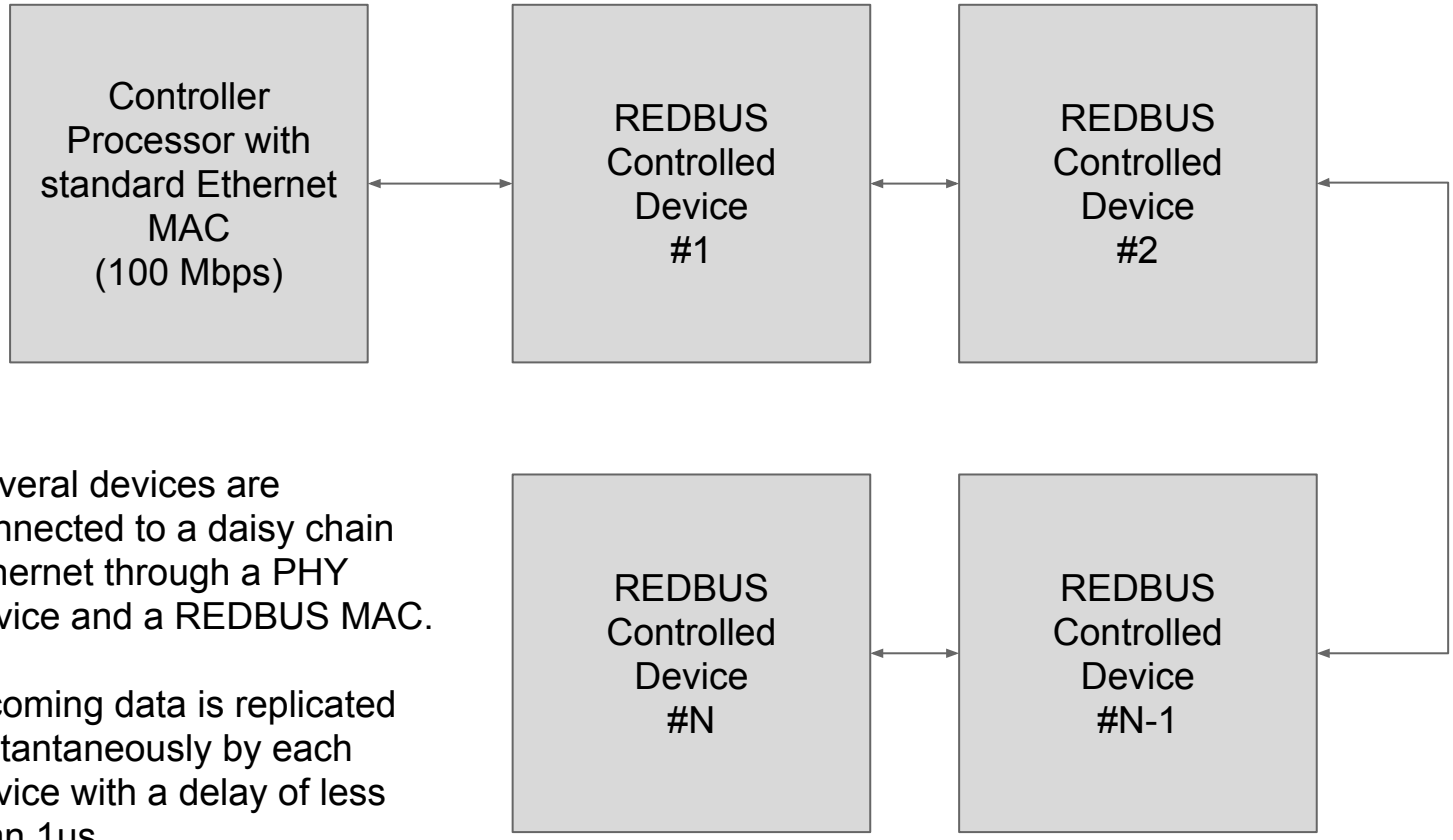
REDBUS is an Ethernet-based BUS with a patented topology.

It provides a fast connection between multiple devices without SWITCH or HUB and requires only one PHY device and one transformer per board.

Features

- High Speed, real-time. The 100 Mbps deterministic communications make it ideal for time-critical applications. It's based on IEEE 802.3 standard Ethernet frames.
- Robust and reliable. It provides high noise immunity and low EMI, up to 100 meters.
- Cost-effective. It uses only one Ethernet PHY device per node and does not require Ethernet switches. Different kinds of information can be transported in just two pairs of small cables, simplifying wiring.
- Versatile. The interfacing and integration simplicity with processors makes it suitable for use in a large variety of contexts. REDBUS is FPGA-based. It can be optionally combined with your logic on the same device.
- Compact. The REDBUS IP has a low logic density, starting from just 250 Logic Elements.

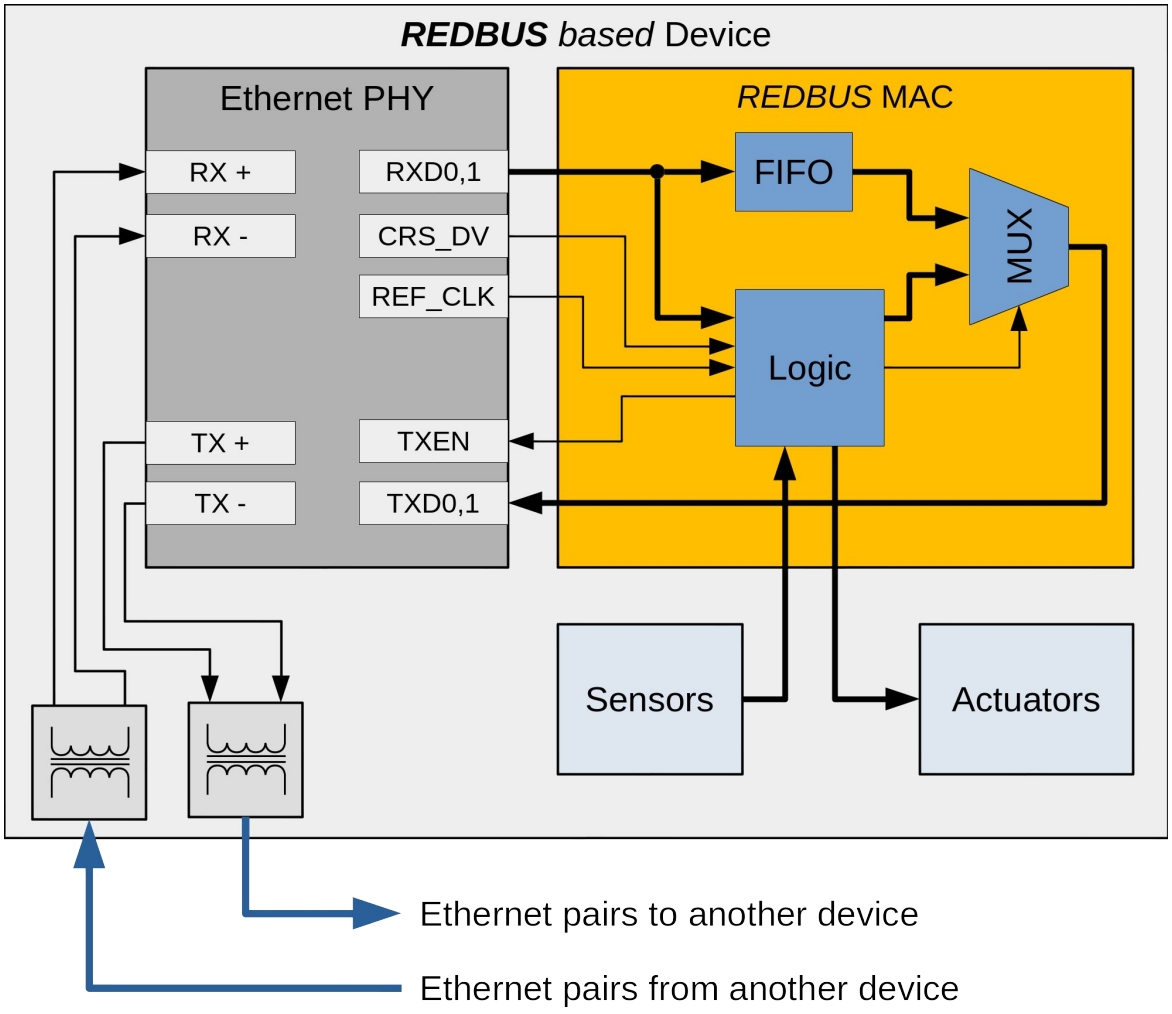
What is REDBUS?



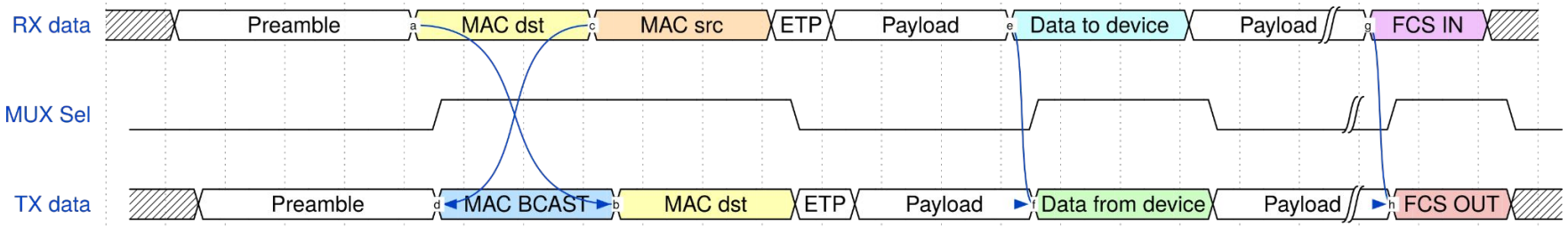
Several devices are connected to a daisy chain Ethernet through a PHY device and a REDBUS MAC.

Incoming data is replicated instantaneously by each device with a delay of less than 1µs.

What is REDBUS?

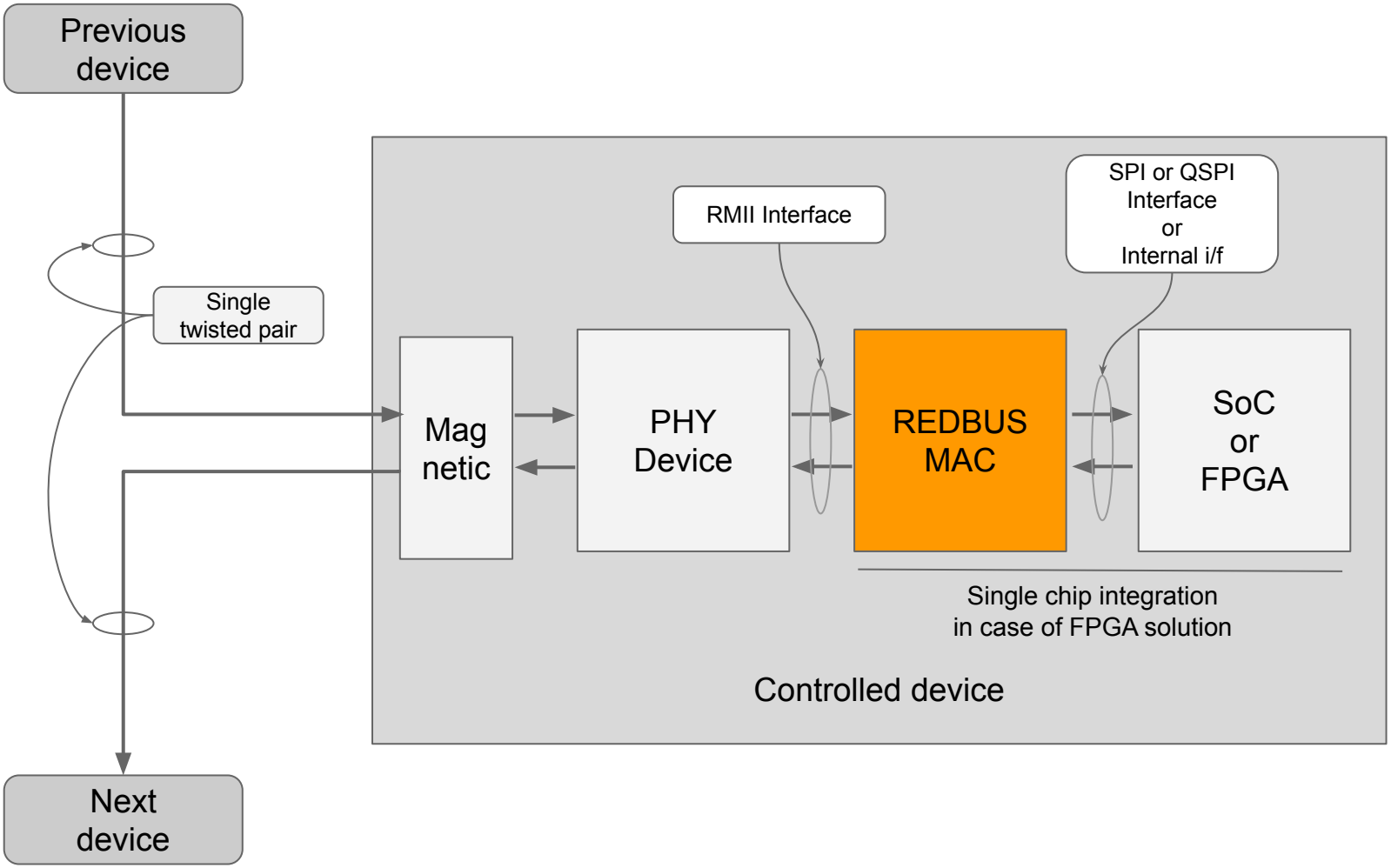


What is REDBUS?

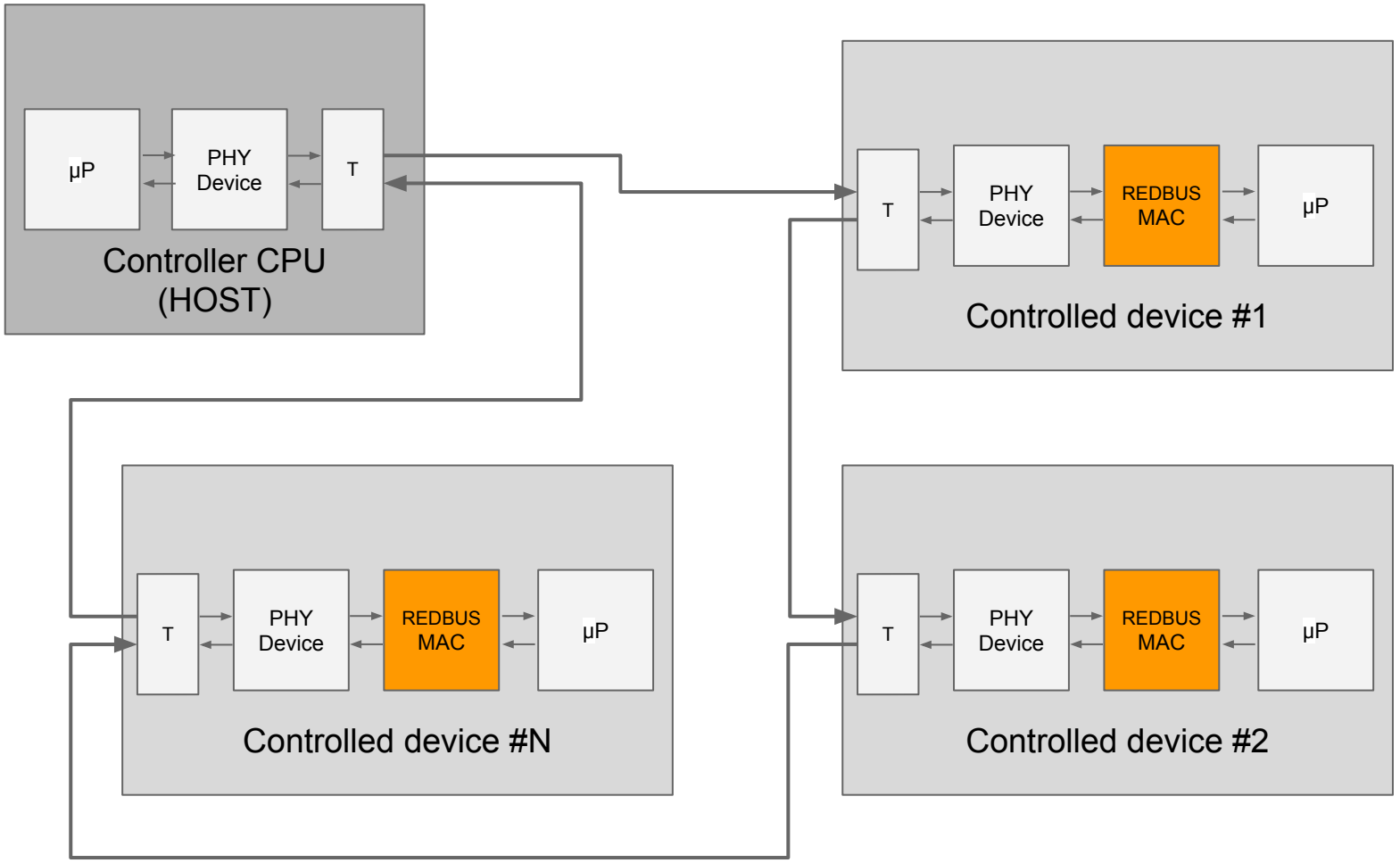


The REDBUS MAC repeats the Ethernet data received through the RMII interface from the PHY device.
The source and destination MAC address are modified so the controller unit that originates the Ethernet frame will accept it back when returning from the last device.
Data to a target is substituted with the response data.
Finally, the FCS is recomputed to validate the entire Ethernet frame.

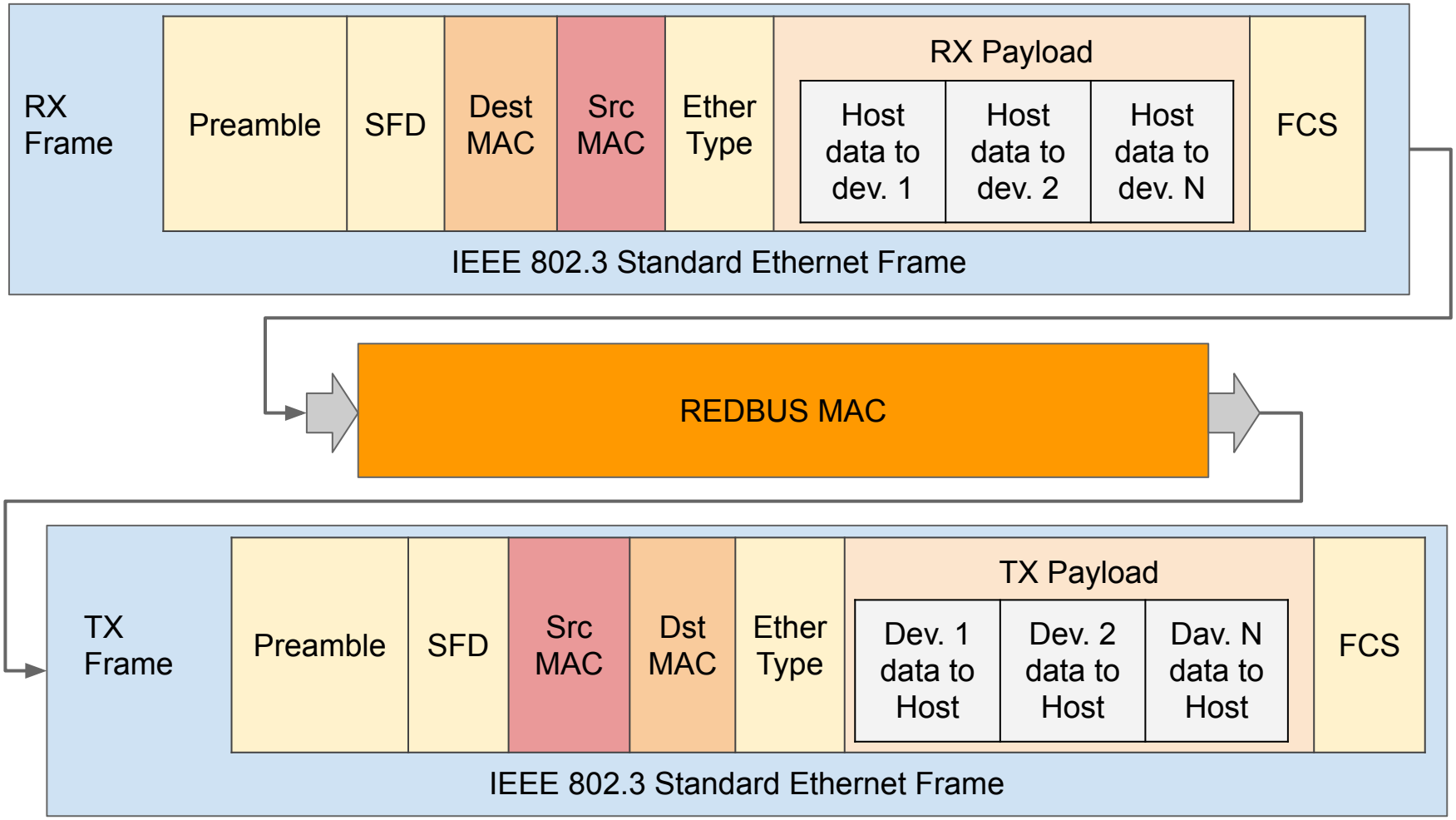
Devices block diagram



REDBUS topology



Transport Protocol



Transport Protocol

The payload in the Ethernet frame contains data for several devices. First, the controller (host) initiates the transmission.

The first device receives the RX frame and immediately replicates it on its TX frame. The following device will receive it on its RX frame.

Each addressed device takes the data coming from HOST and replaces it with its data in the TX Payload.

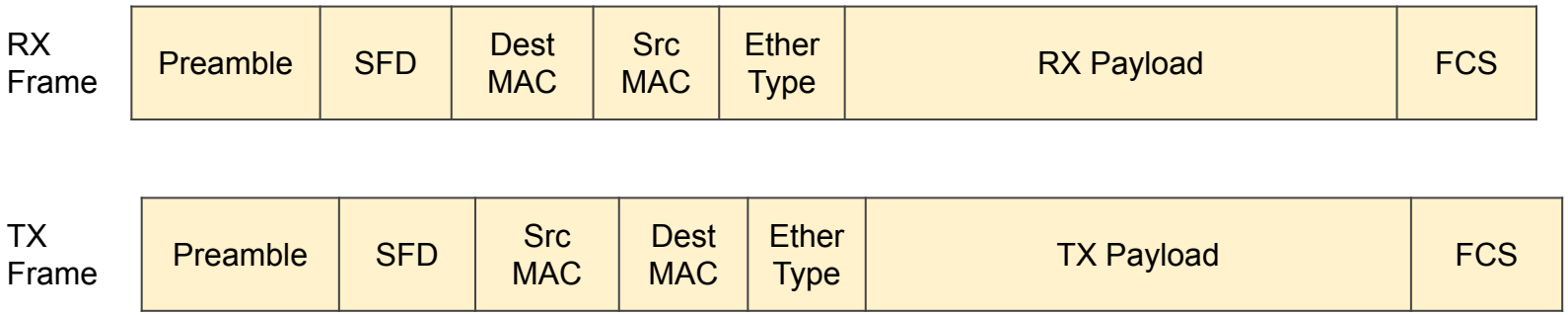
RX Frame	Preamble	SFD	Dest MAC	Src MAC	Ether Type	RX Payload	FCS

TX Frame	Preamble	SFD	Src MAC	Dest MAC	Ether Type	TX Payload	FCS

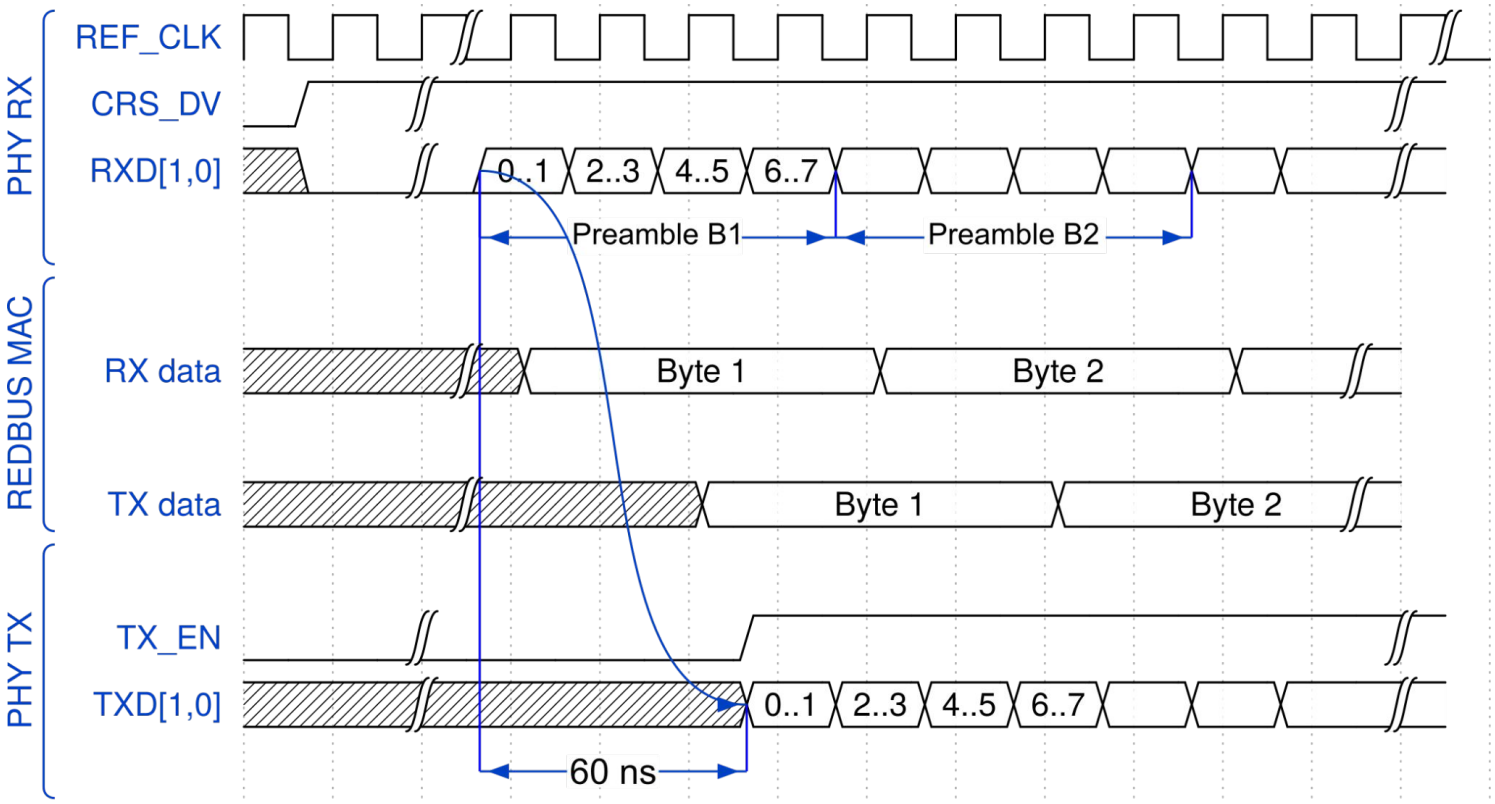
Propagation time

The TX Frame starts as soon as the first bit of the received preamble comes into REDBUS MAC. The PHY device also introduces a fixed delay for a total delay of about **960 ns**.

The total propagation delay depends on the number of connected devices in the chain plus a small amount caused by the cable length.

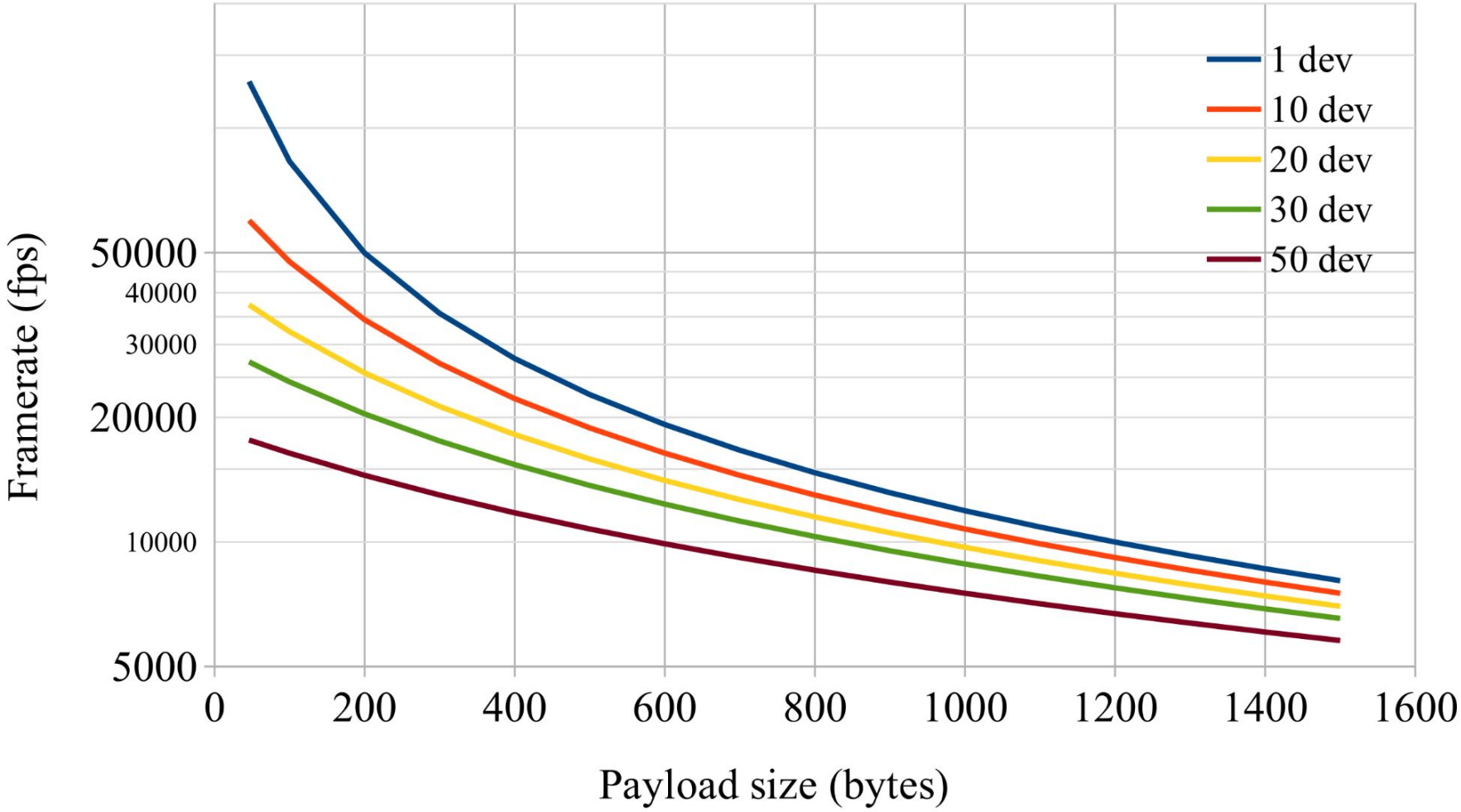


Propagation time

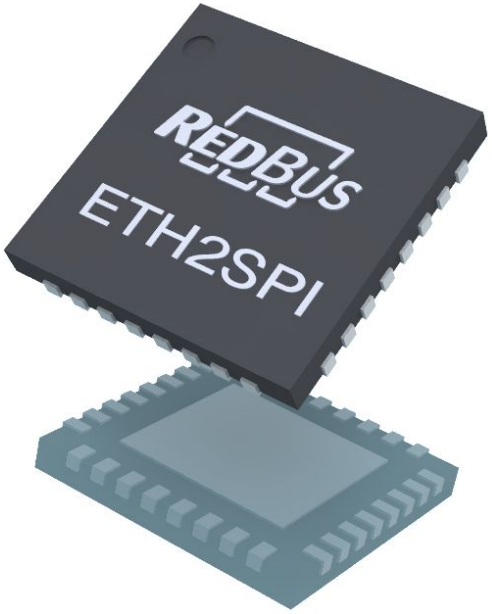


The REDBUS MAC repeats the incoming data after 60 ns.

Framerate vs payload and number of devices



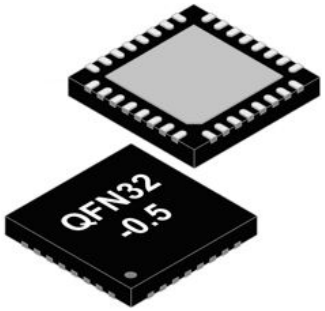
REDBUS Devices

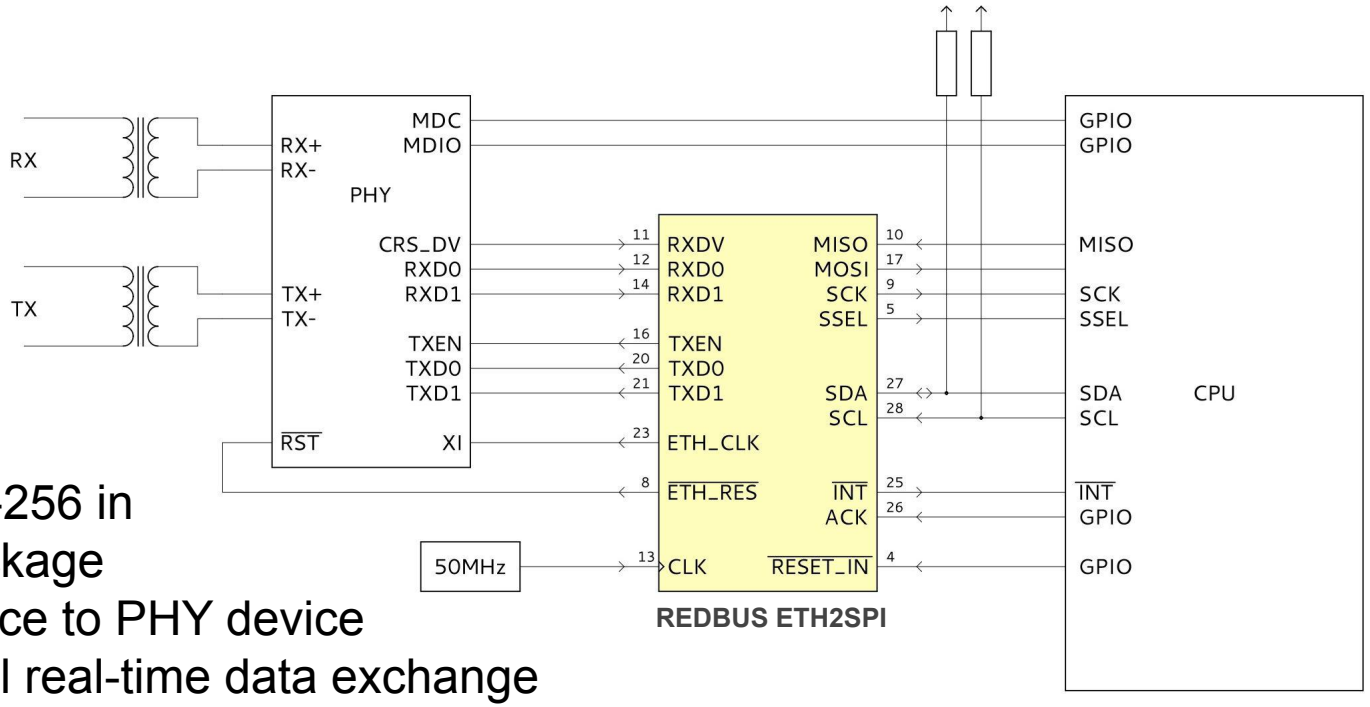


Real-time Ethernet-to-SPI* Converter

*(Serial Peripheral Interface, also called Synchronous Serial Interface)

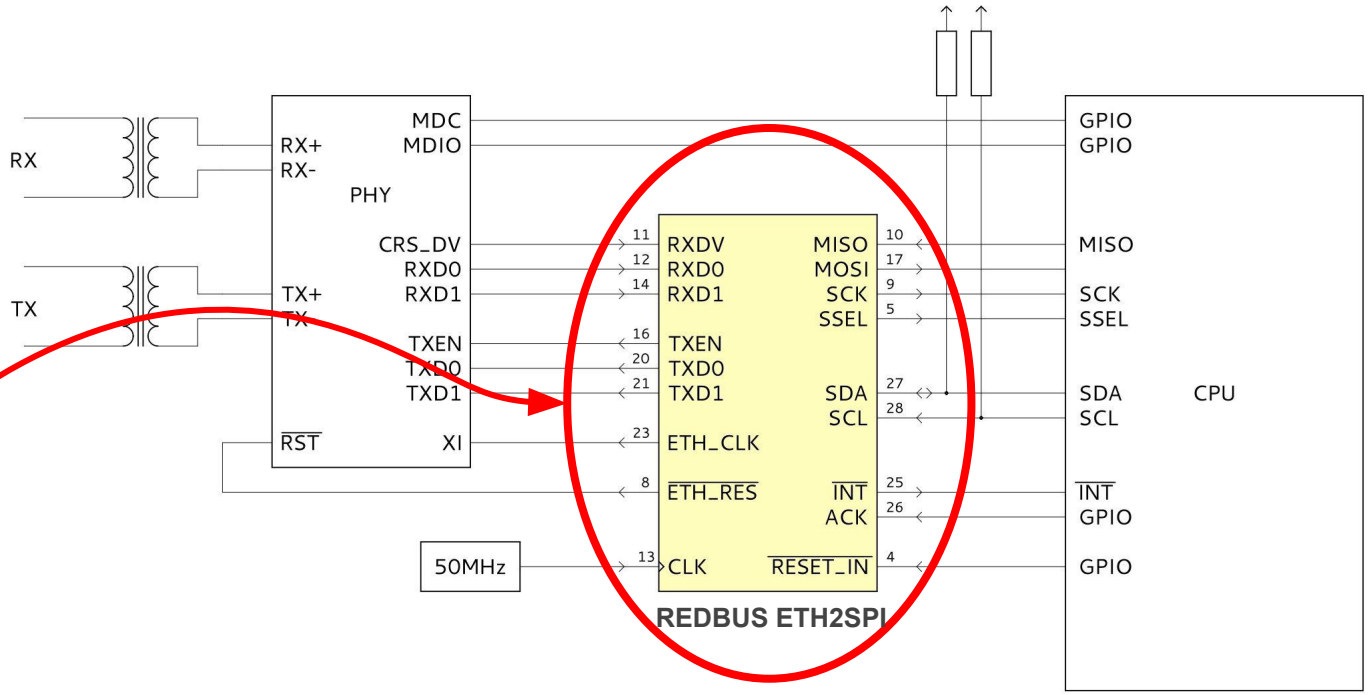
REDBUS Ethernet to SPI converter





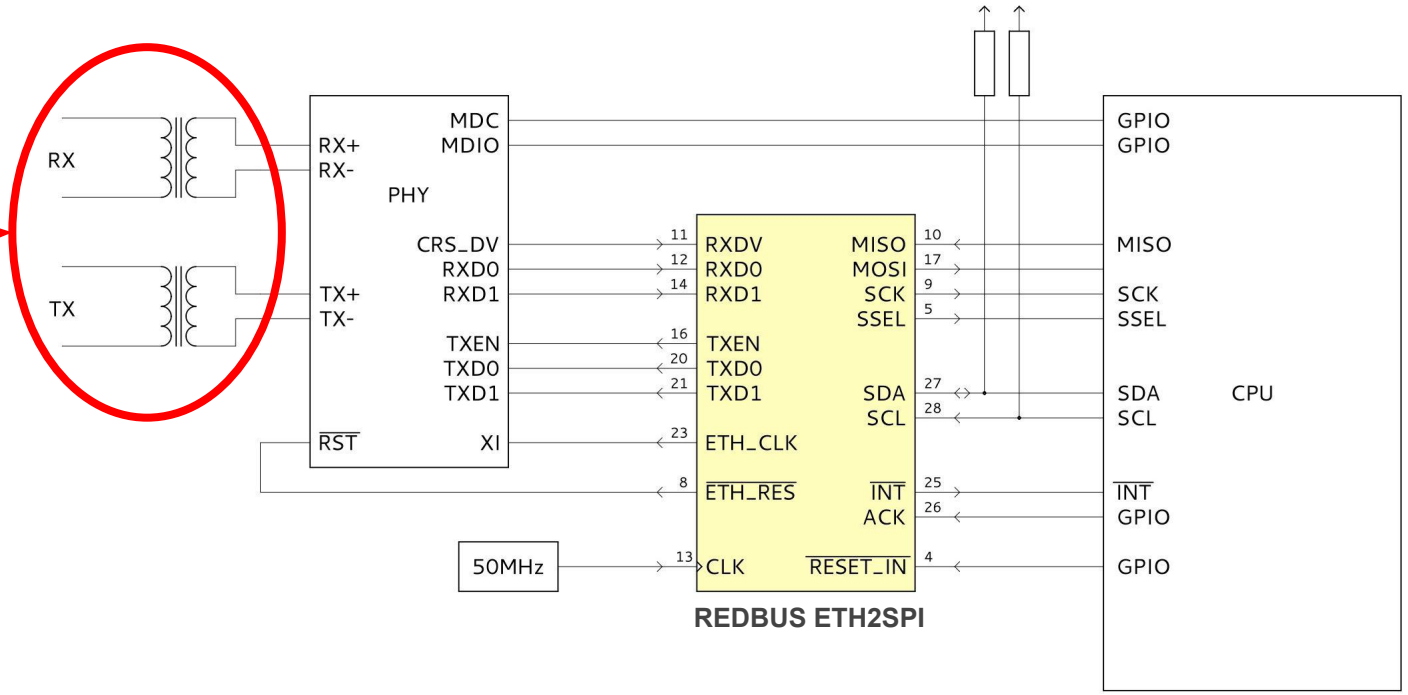
- Ultra small
- Based on a MACHXO2-256 in QFN32 Package
- RMII interface to PHY device
- Bidirectional real-time data exchange
- I2C interface to set device address
- Up to 8192 device addressing
- Can exchange up to 44 bytes per ethernet frame per slave
- Starting from 3.125 MHz SPI clock

REDBUS Ethernet to SPI converter



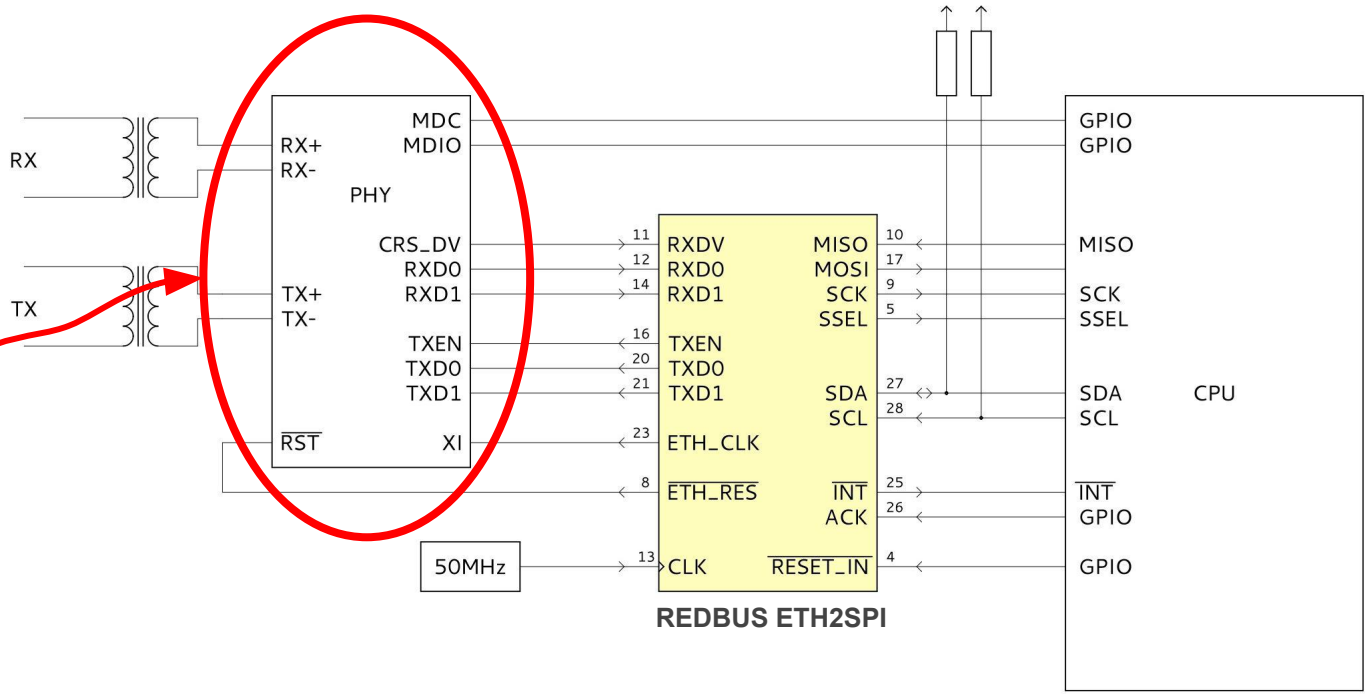
Real-time Ethernet REDBUS MAC on a Lattice Semiconductor tm MachXO2-256

REDBUS Ethernet to SPI converter



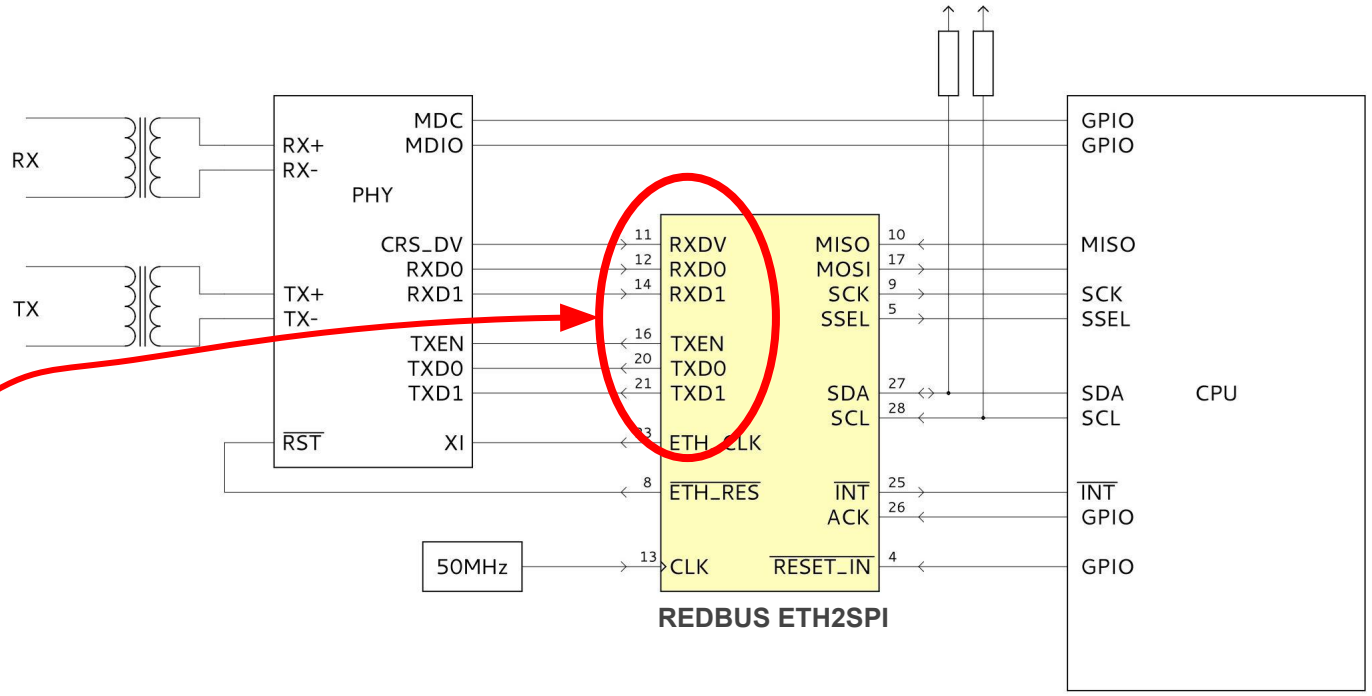
Ethernet RX/TX signals through UTP wires

REDBUS Ethernet to SPI converter



Standard Ethernet PHY device

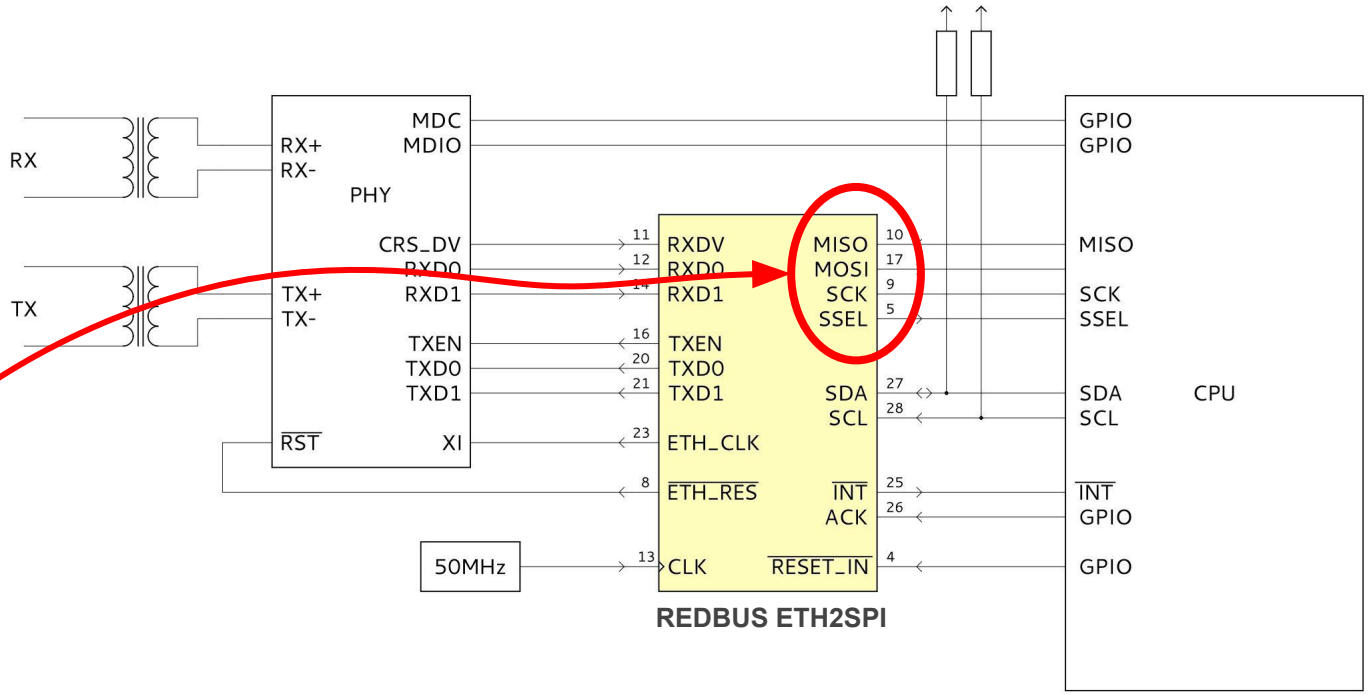
REDBUS Ethernet to SPI converter



RMII Interface,

Data clocked @ 50 MHz

REDBUS Ethernet to SPI converter

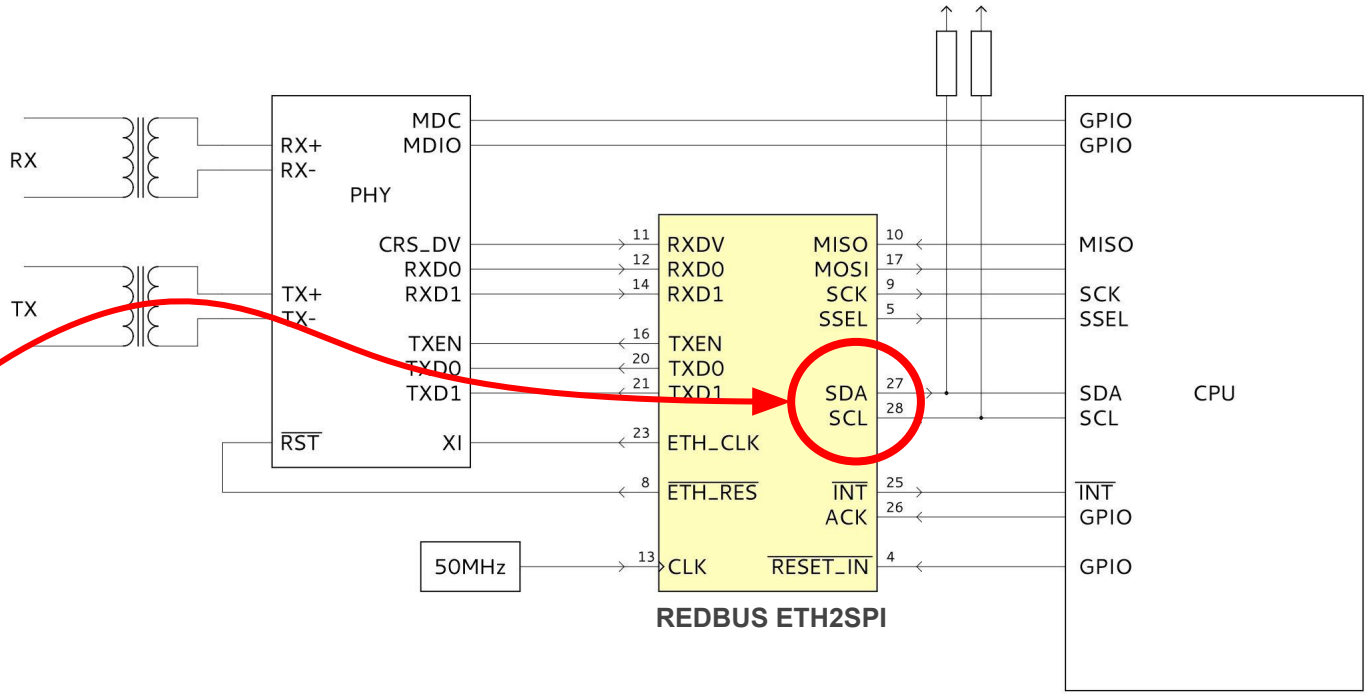


SSI: Synchronous Serial Interface

Data from/to Ethernet is clocked @ 3.125 MHz*

* devices are available on three versions of SPI clock frequency: 3.125, 6.25 or 12.5 MHz

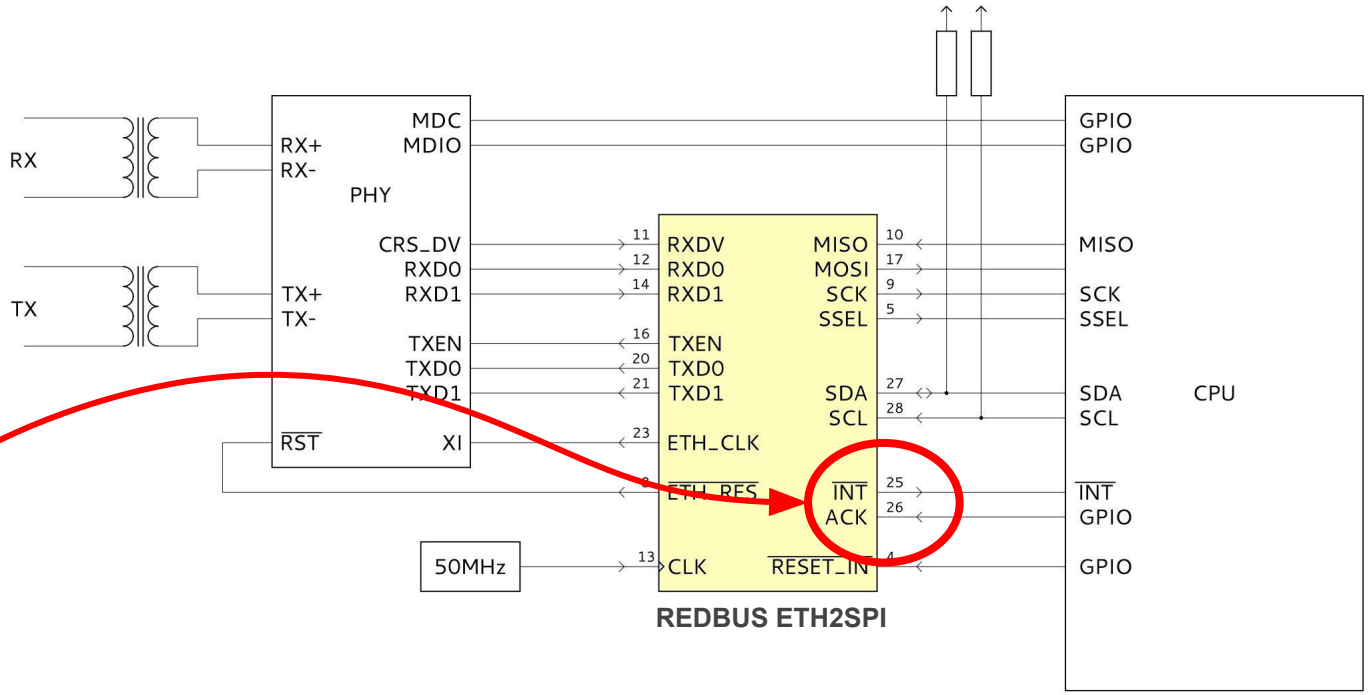
REDBUS Ethernet to SPI converter



I2C access to registers:

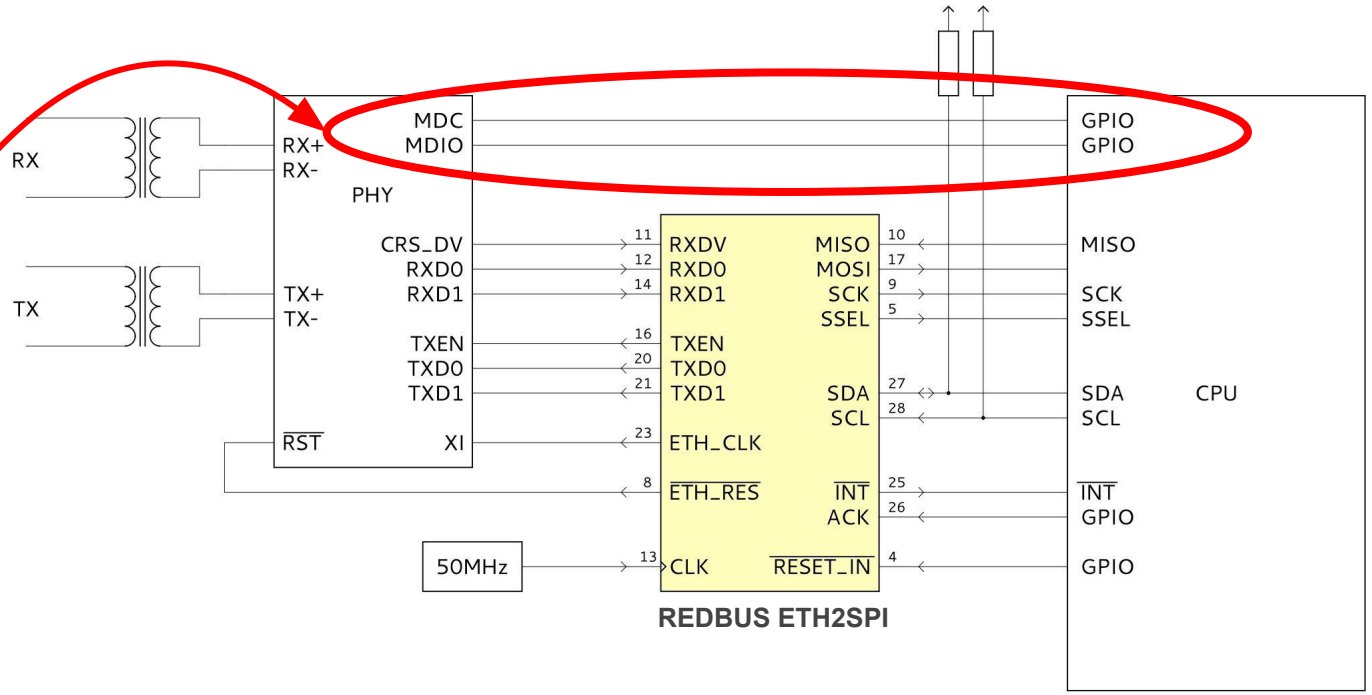
Address, Unique ID

REDBUS Ethernet to SPI converter



uP interrupt generated on every Ethernet Frame

REDBUS Ethernet to SPI converter



The PHY MDIO interface is controlled by CPU, for PHY configuration and link management

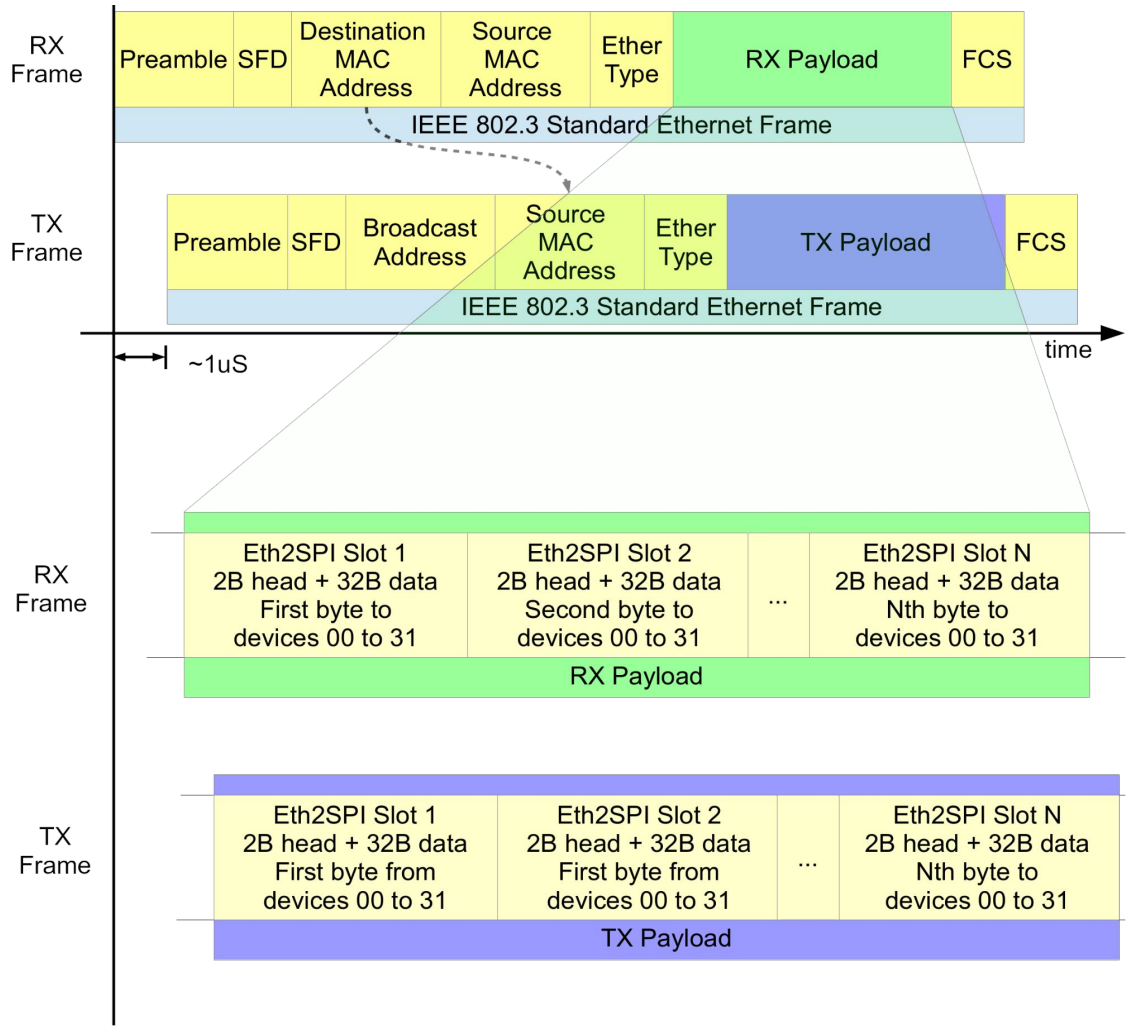
REDBUS Ethernet to SPI converter

Each Ethernet frame can transport data for a plurality of device's groups. Depending on IP-Core configuration, each group may address 8, 16 or 32 devices.

Number of SUB-Address (#bit)	SPI Clock Frequency	Max bytes per device inside a single Ethernet frame	Addressable devices
8 (#3)	12.5 MHz	150	2032
16 (#4)	6.25 MHz	83	4064
32 (#5)	3.125 MHz	44	8128

For example, the 3.125 MHz version can address 254 groups of 32 devices, for a total of 8128 devices. Each group has a unique 8 bit address (1-254), each device in a group has a 5 bit sub-address ranging from 0 to 31.

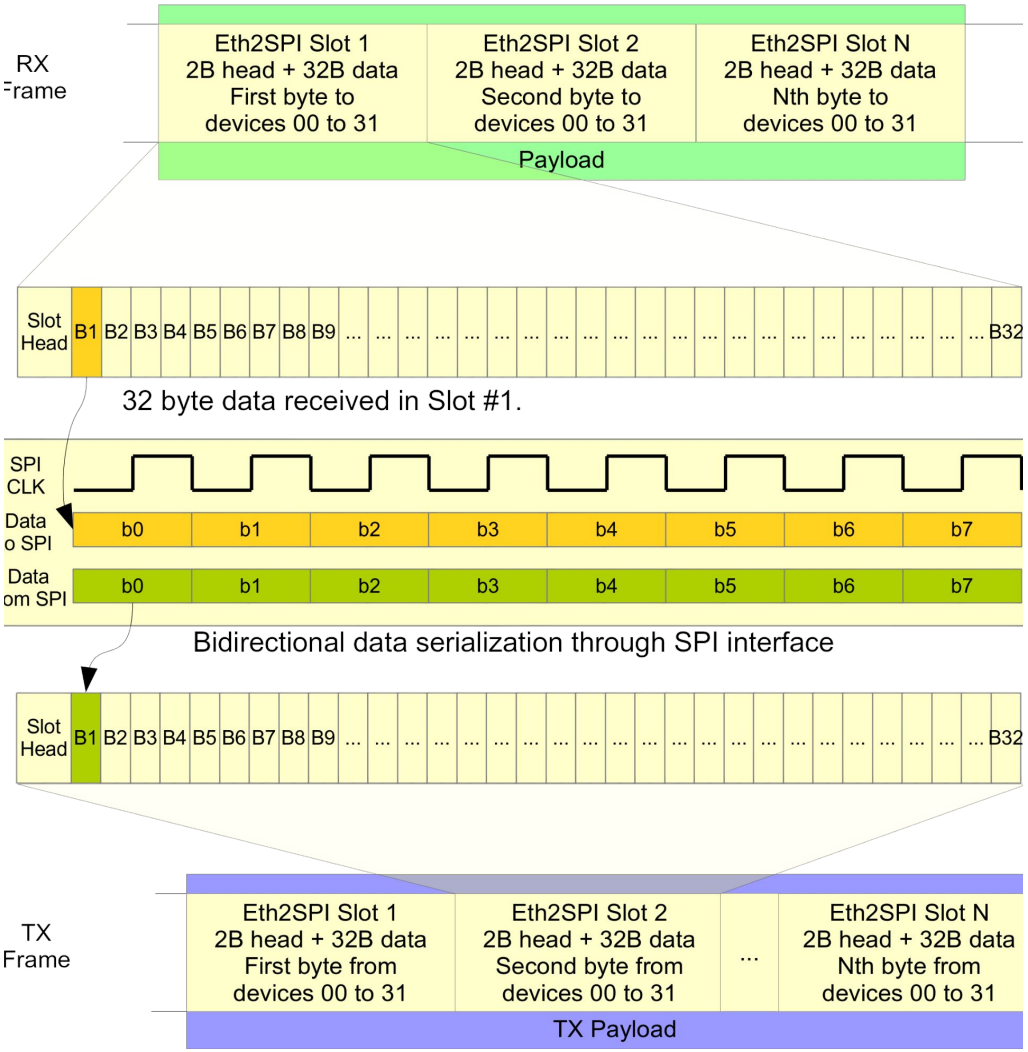
Ethernet to SPI frame



This Example shows the frame of Ethernet-to-SPI devices with an output clock of 3.125 MHz.

The Ethernet bitrate of 100 Mbps is divided by 32 that results in a SPI bitrate of 3.125 Mbps.

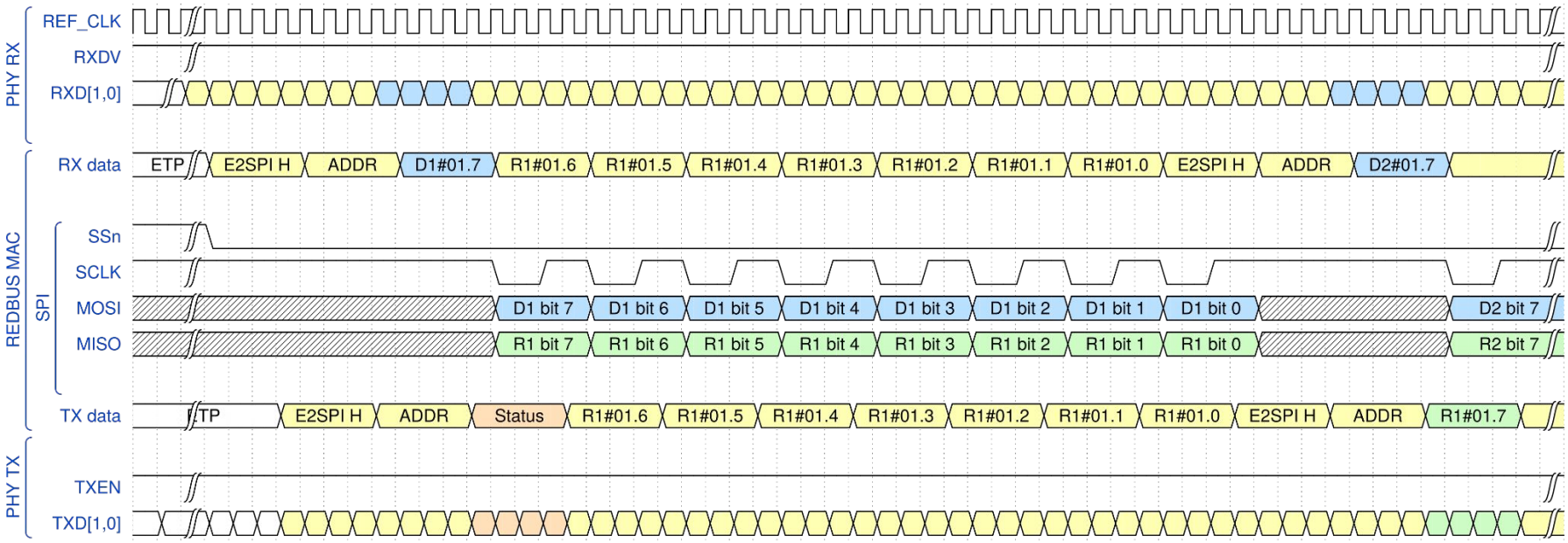
Ethernet to SPI frame



Data is organized in slots of 32 bytes. A slot is addressed to a group of 32 devices. Each slot contains one byte for each device in the addressed group.

Each device exchanges one byte of a slot through SPI. As each device introduces about one µs delay, to improve performance the first byte in the slot is assigned to the last device in the group, and the last byte in the slot is assigned to the first device in the group.

Ethernet to SPI frame



This graphic shows data exchange on a REDBUS MAC. The RMII REF_CLK signal is used as a time reference. In this example, serial data is clocked at 12.5 MHz; each slot data is 8 bytes wide.

Measures

The total latency is 32.59 µs.

Latency measurement on a system based on a main controller and 32 REDBUS based devices.

The yellow track is connected to the RMII_TXEN signal of the main controller (host). It is active when the CPU starts Ethernet transmission.

The cyan track is connected to the RMII_CRSDV signal of the main controller. It is active when the Ethernet frame is received back by the PHY device on the host.

Measures

The latency on first device is 760 ns.

Latency measurement on the first REDBUS based device.

The first magenta track is connected to the RMII_CRSDV signal on the first REDBUS device.
The second magenta track is connected to the SPI_SCK signal on the first REDBUS device. The data is serialized in and out on the rising edge of this clock.

The third and fourth magenta tracks are connected to CRSDV and SCK signals on the second REDBUS device.

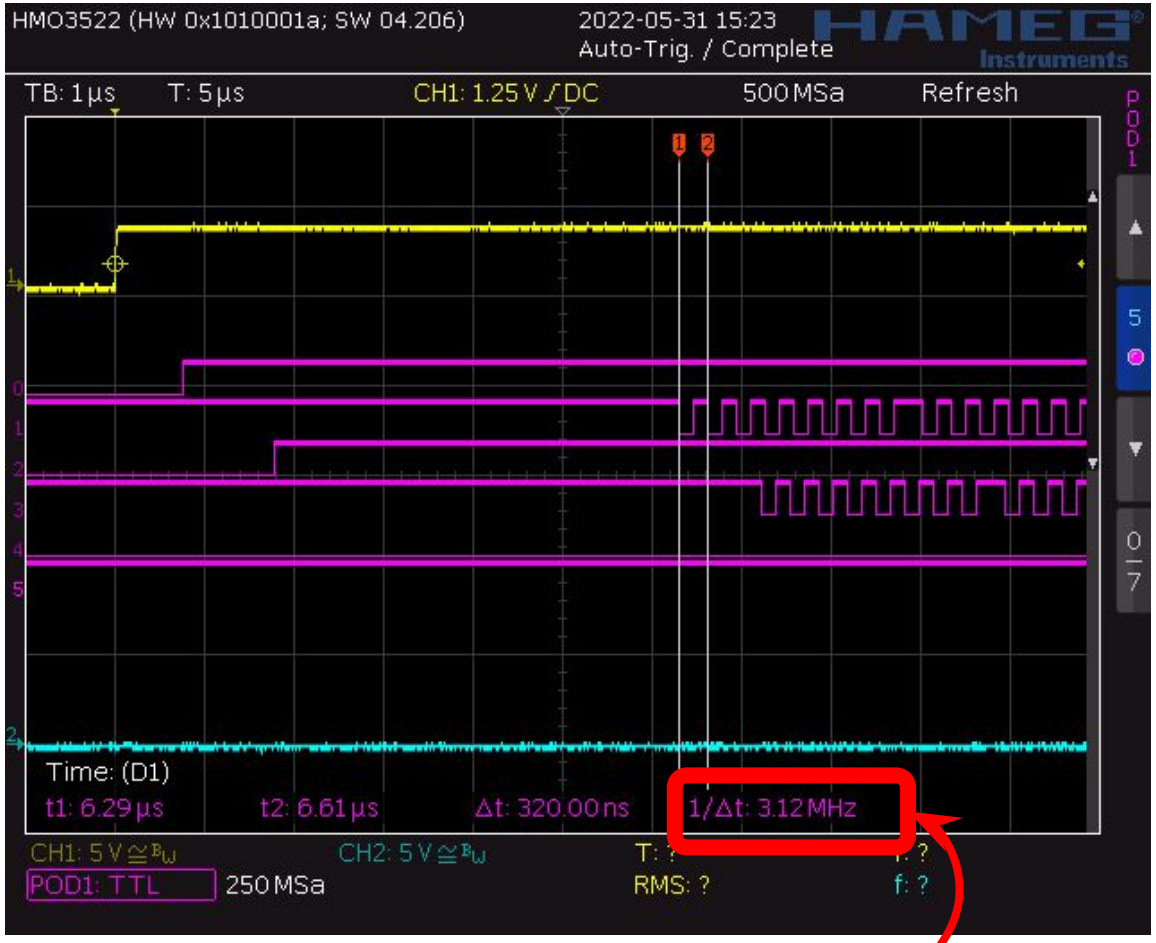
The last two magenta tracks are connected to CRSDV and SCK signals of last device.

Measures

Latency measurement on the last (32th) REDBUS based device.

The latency on 32th device is 31.5 µs.

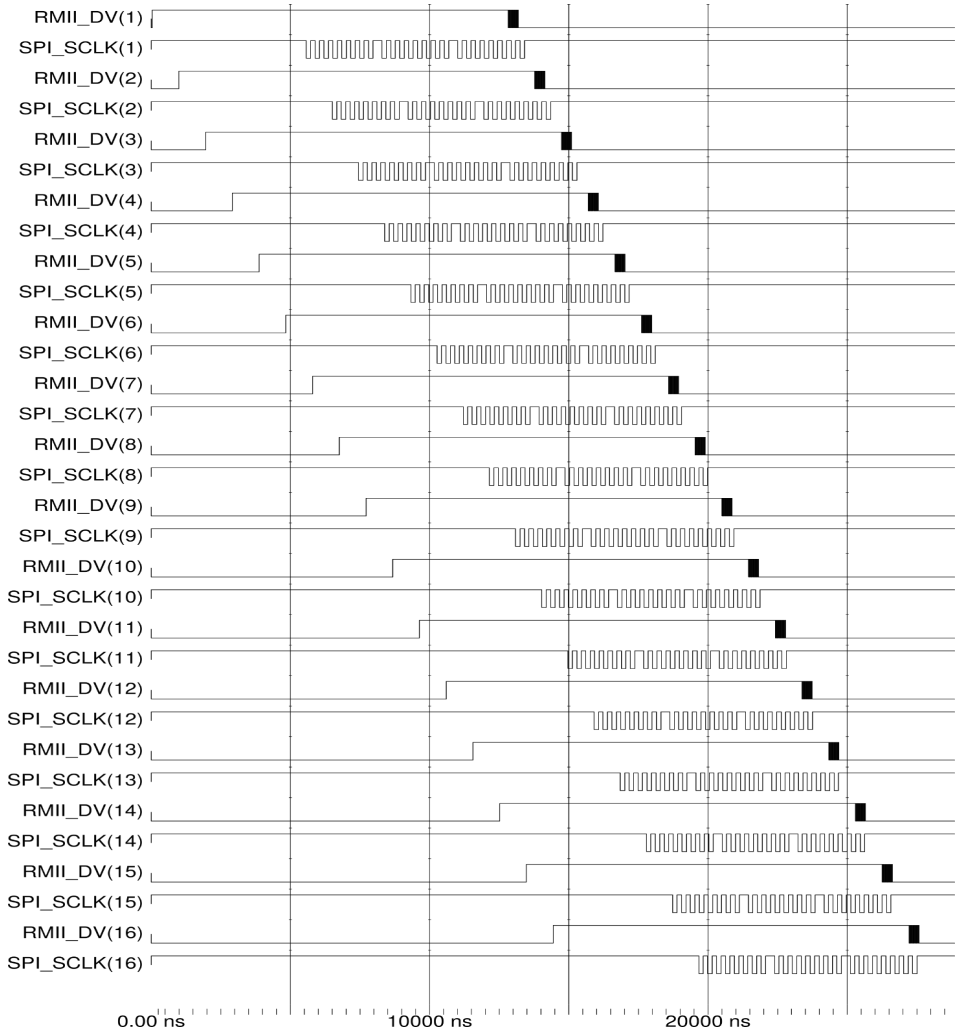
Measures



Zoom in on the SPI_SCK signal and frequency measurement on the first device.

The SPI_SCK frequency is 3.125 MHz.

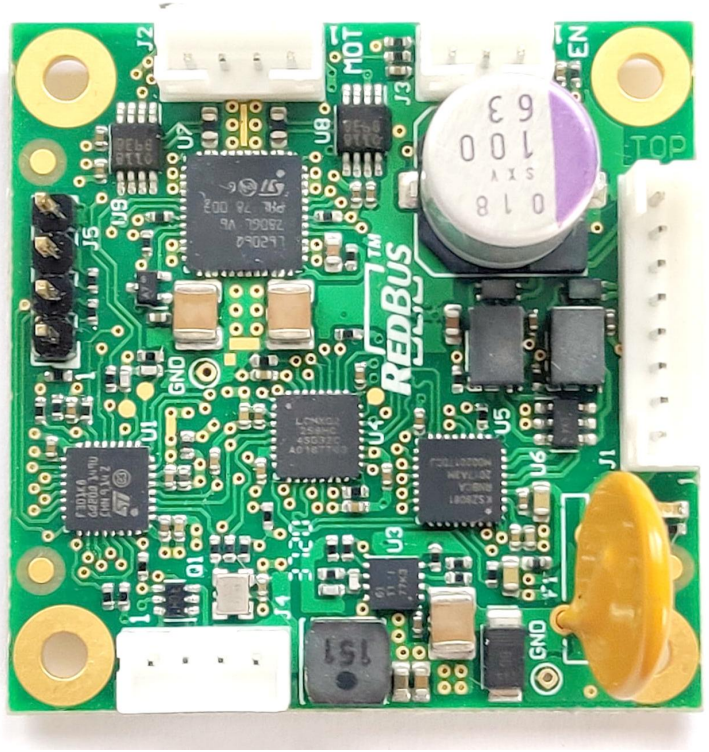
Measures



RMII_CRSDV and SPI_SCK signals on a system composed by 16 devices.

The REDBUS MACs are configured to divide by 16 the serial rate on the SPI interface. The resulting SPI_SCK clock has a frequency of 6.25 MHz.

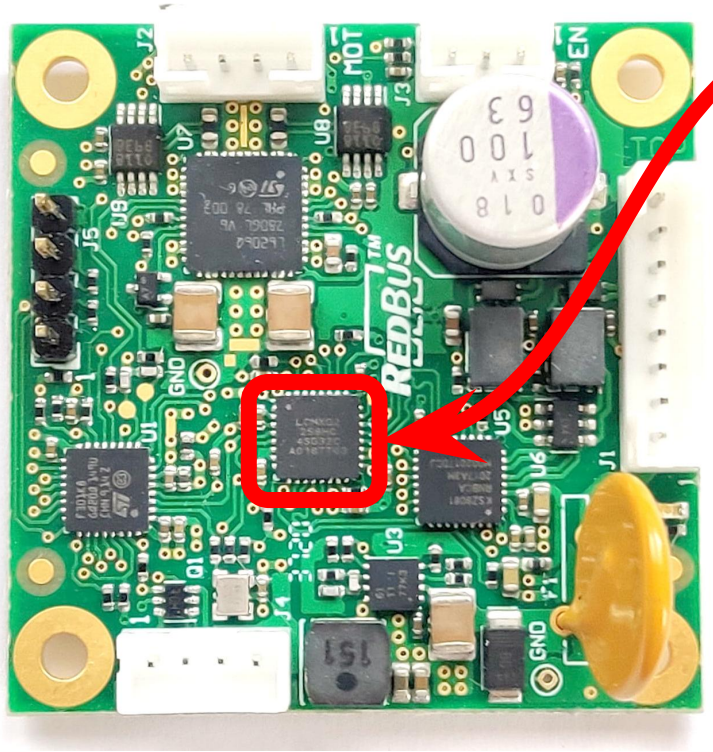
Reference design boards



REDBUS

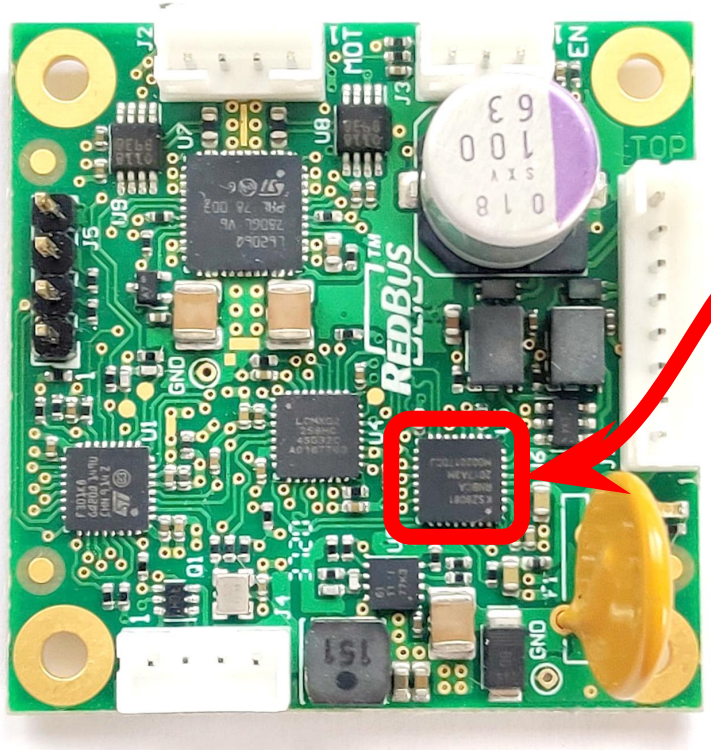
Motor Control reference design board

Reference design boards



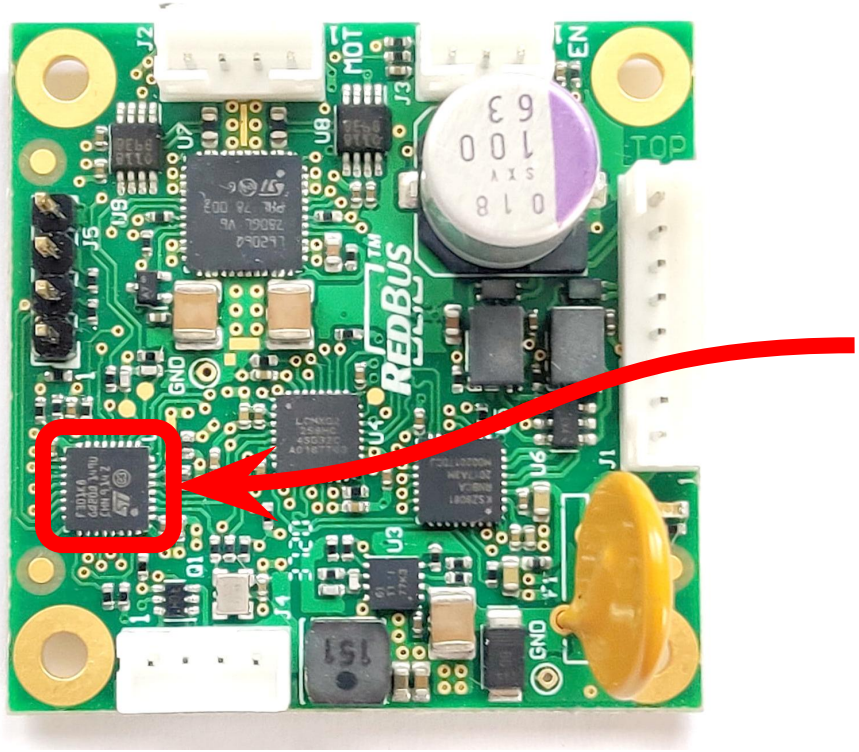
REDBUS Ethernet to SPI MAC

Reference design boards



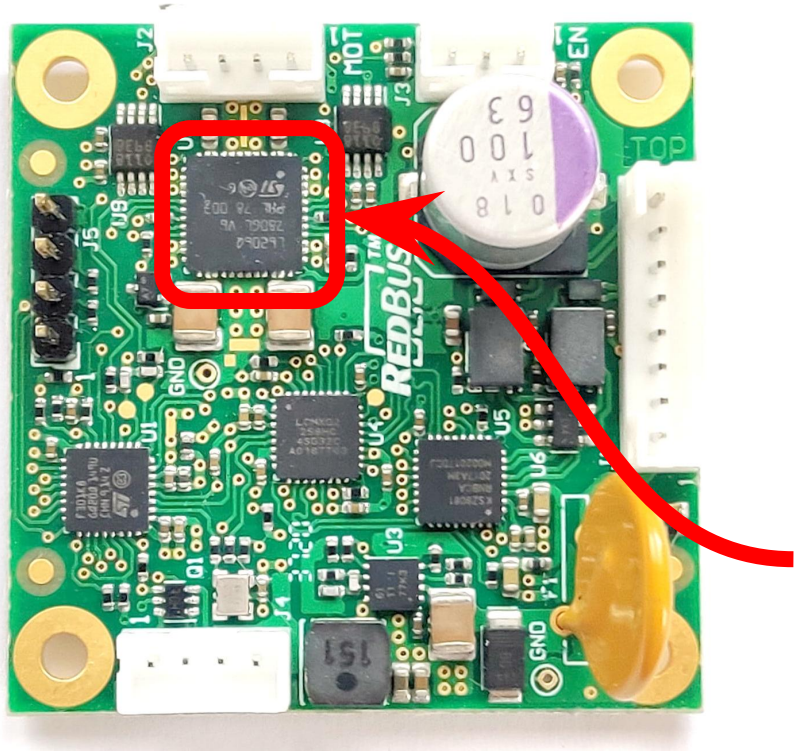
Ethernet PHY device

Reference design boards



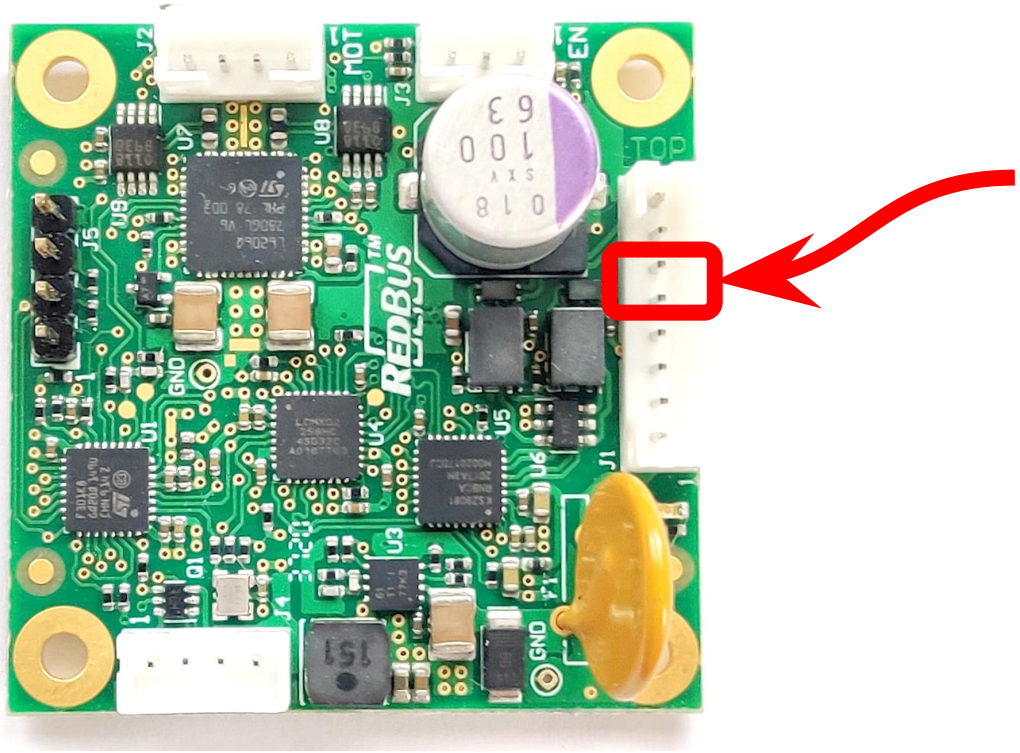
Microcontroller
(STM32F301)

Reference design boards



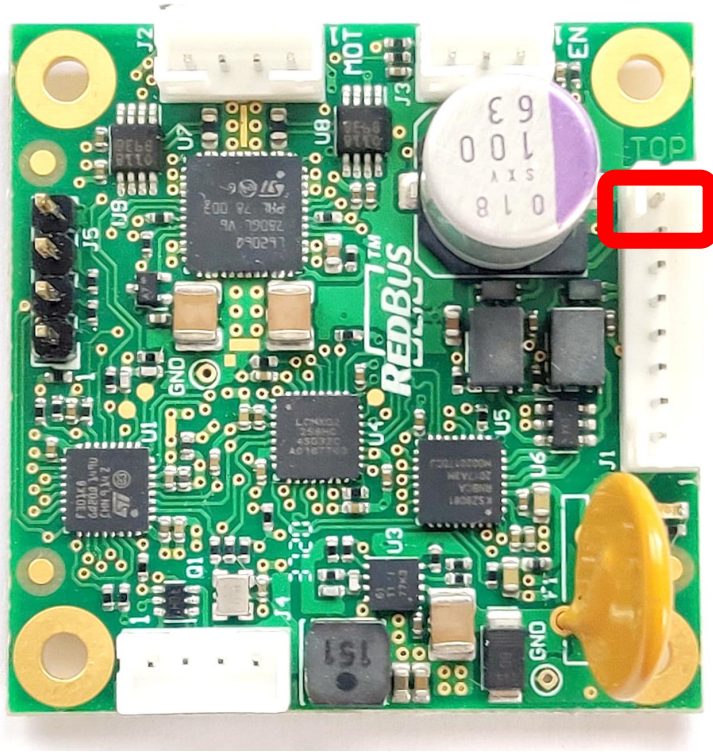
Motor Driver,
4 H-bridges MOSFETS

Reference design boards



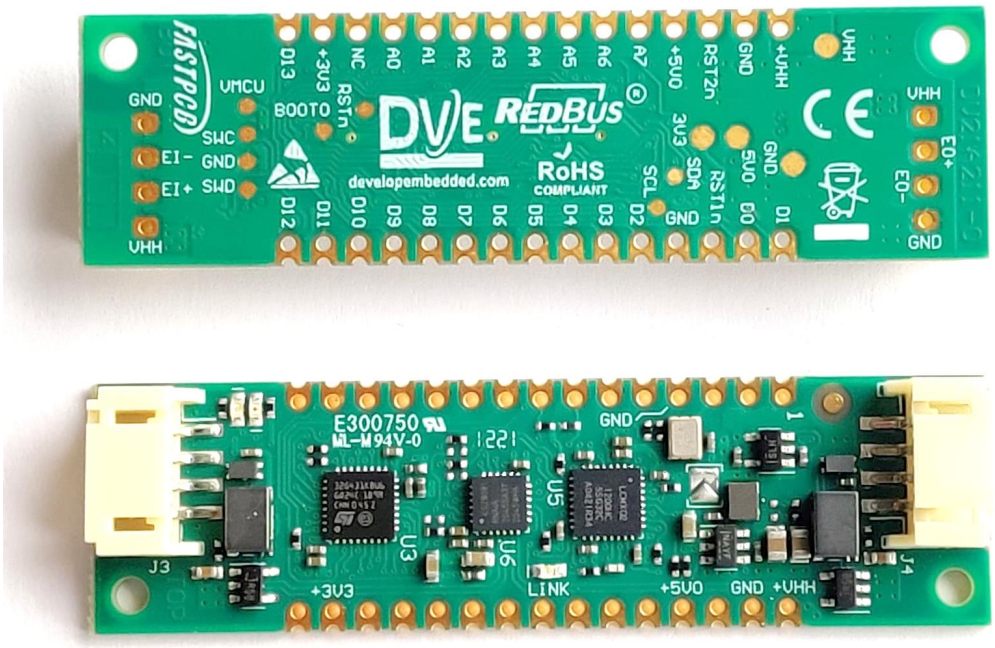
Ethernet IN

Reference design boards



Ethernet OUT

Reference design boards



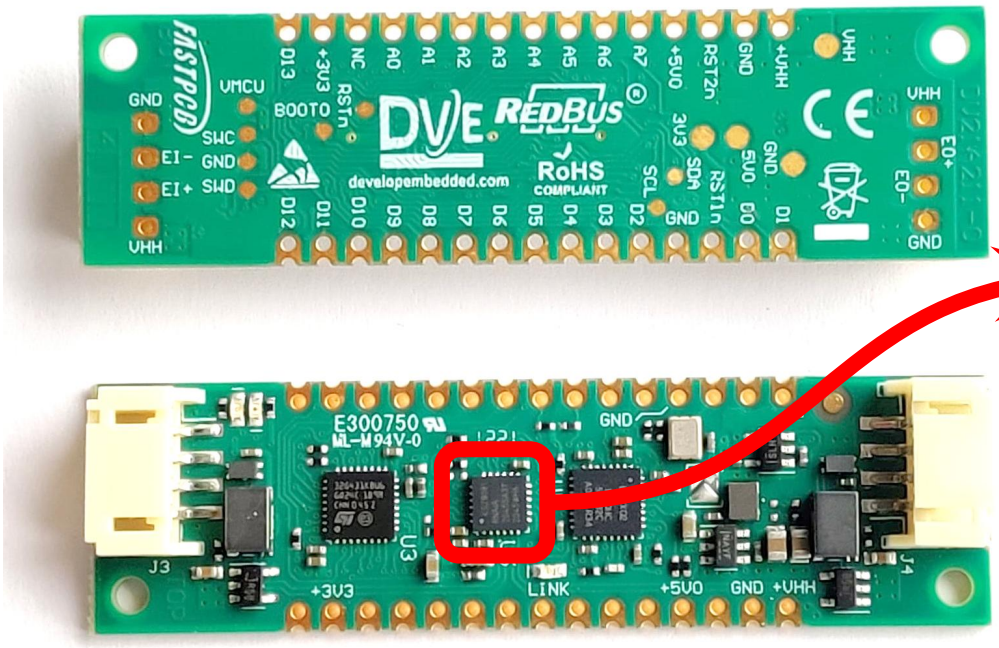
REDBUS

General Purpose I/O board

It provides: GPIO, PWM, Timers, ADC, DAC, UART and other common interfaces.

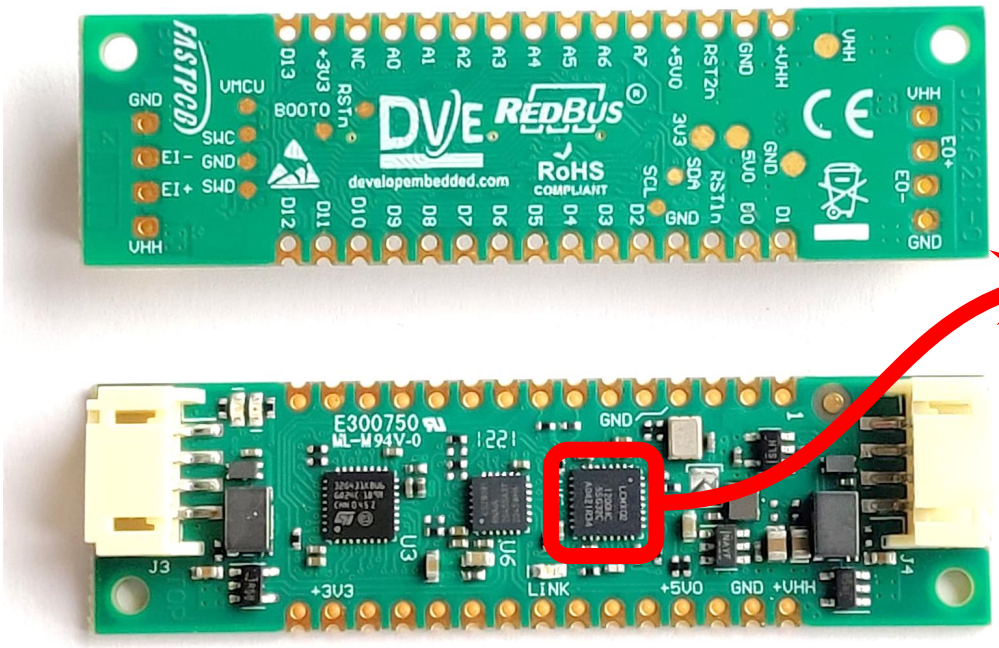
It's based on STM32G431 processor.

Reference design boards



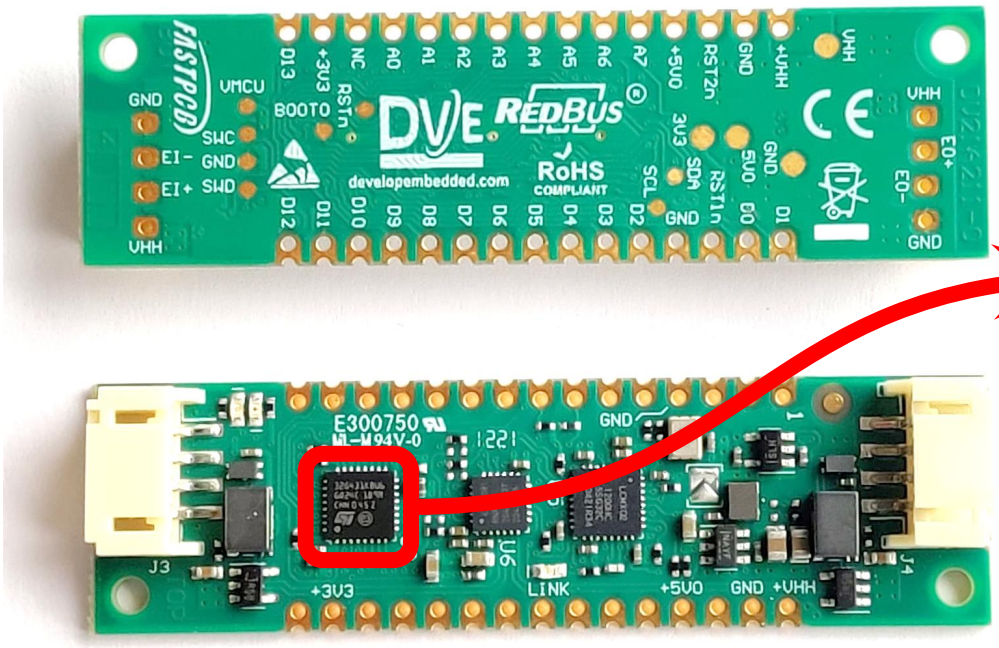
Ethernet PHY device

Reference design boards



REDBUS MAC

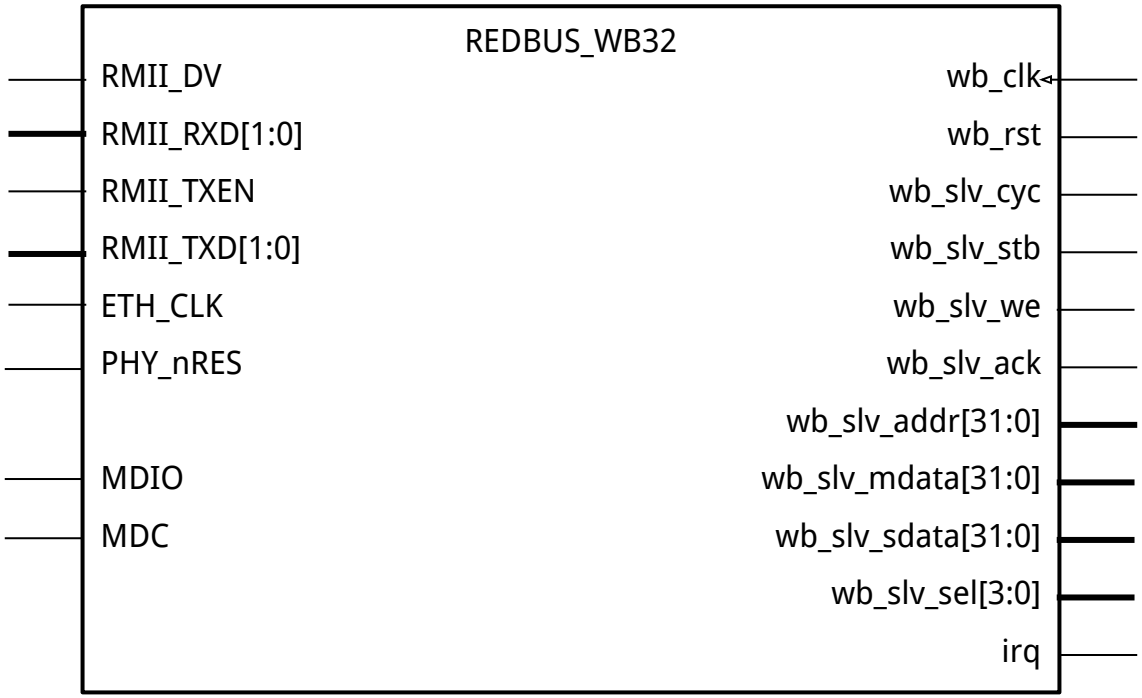
Reference design boards



Microcontroller (STM32G431)

REDBUS IP-Core for FPGA or CPLD

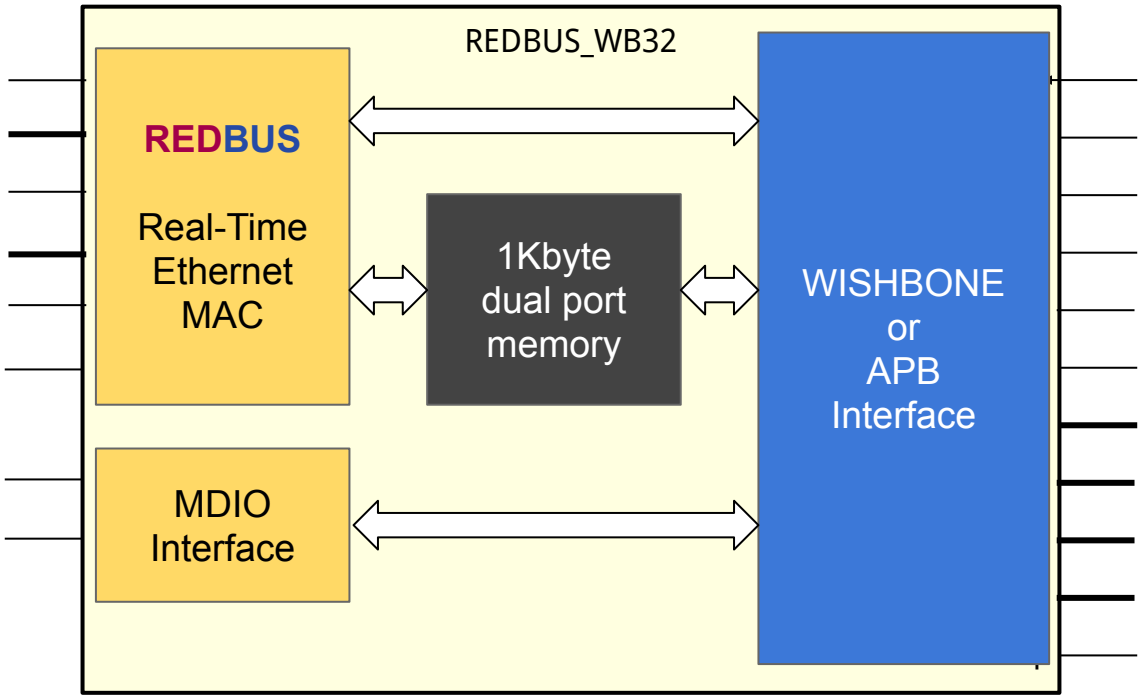
REDBUS MAC is available also as a black-box RTL IP-Core, for CPLD or FPGA devices.



REDBUS IP-Core for FPGA or CPLD

It provides real-time data exchange between Ethernet and logic fabric through a 1KByte dual-port memory.

IP-Core architecture:



REDBUS Patents

- Italian patent number **0001423488**.
- European patent number **EP2930896**.

Contacts

EMail: info@developembedded.com

www.developembedded.com

Thanks for your attention