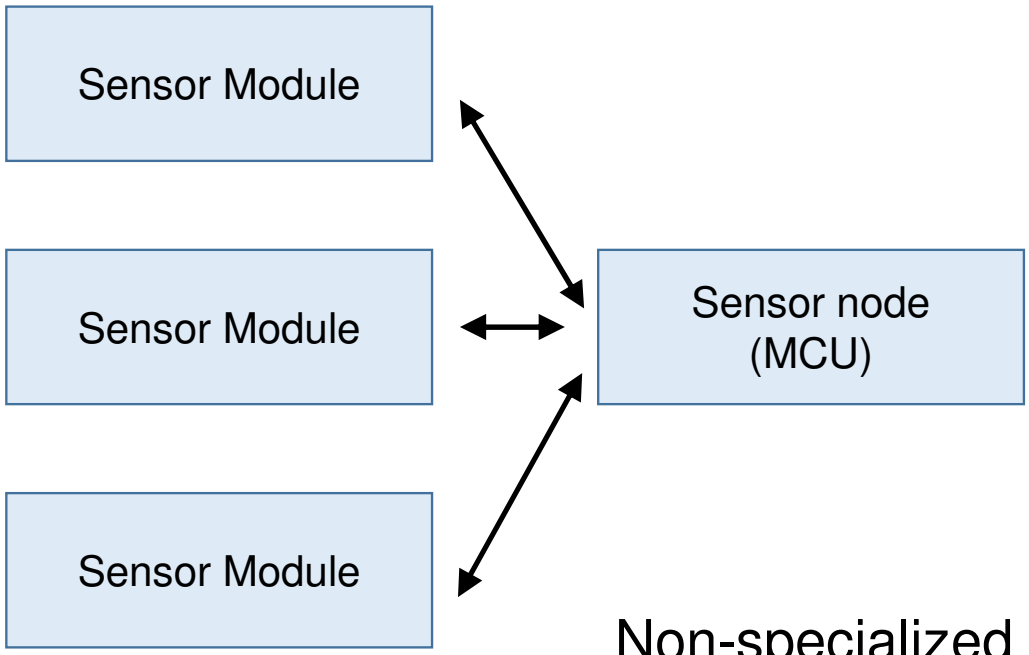
Common inter-chip digital communication protocols

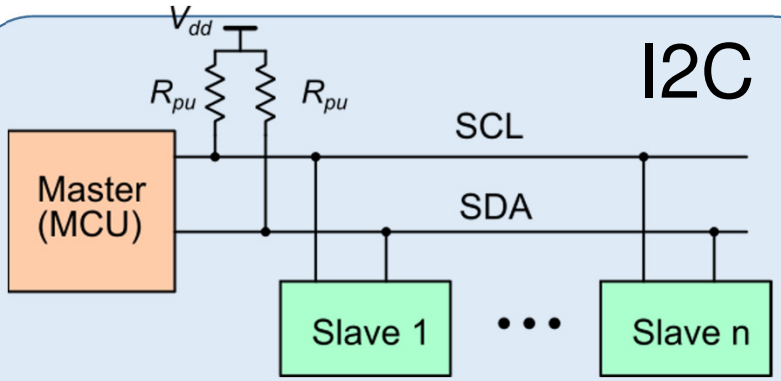


Non-specialized protocols

- **I2C** (Inter Integrated Circuit)
  2-wires (SDA, SCL)
  support addressing: can be used to build small busses
  require pull-up resistors
  standard 400 kbit/s (up to 5 Mbit/s)
- **SPI** (serial peripheral interface)
  Simpler and faster than I2C
  Up to 100 Mbit/s
  Does not support addressing
  4-wires (3 wire versions exist).
- **RS232 (UART)**
  Asynchronous, needs matched clocks
  2-wires full-duplex
  Slow: 115200 bit/s max
  Does not support addressing

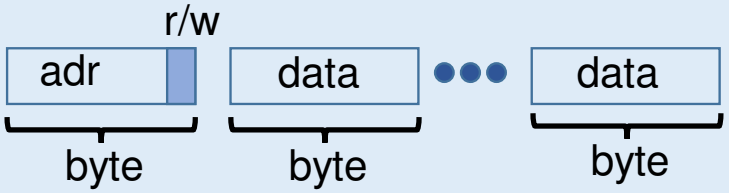
I2C, SPI, UART

I2C

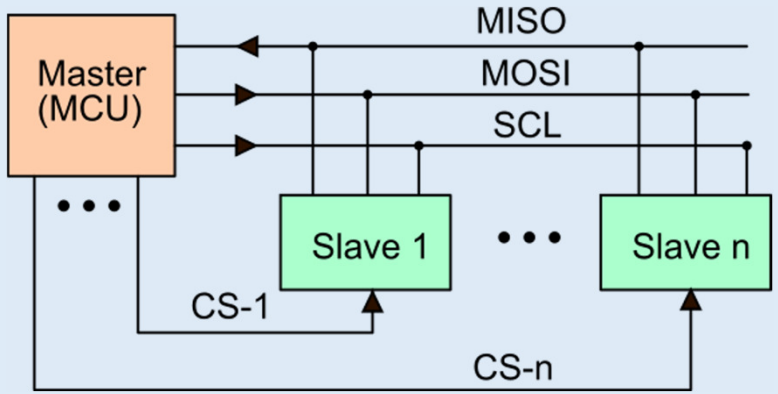


Pull-up resistors are necessary. Their value is chosen as a trade-off between power and speed

I2C Base: 7-bit addres (up to 128 slaves)

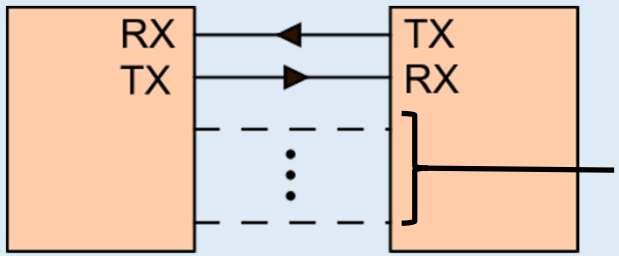


SPI



One CS line for each slave

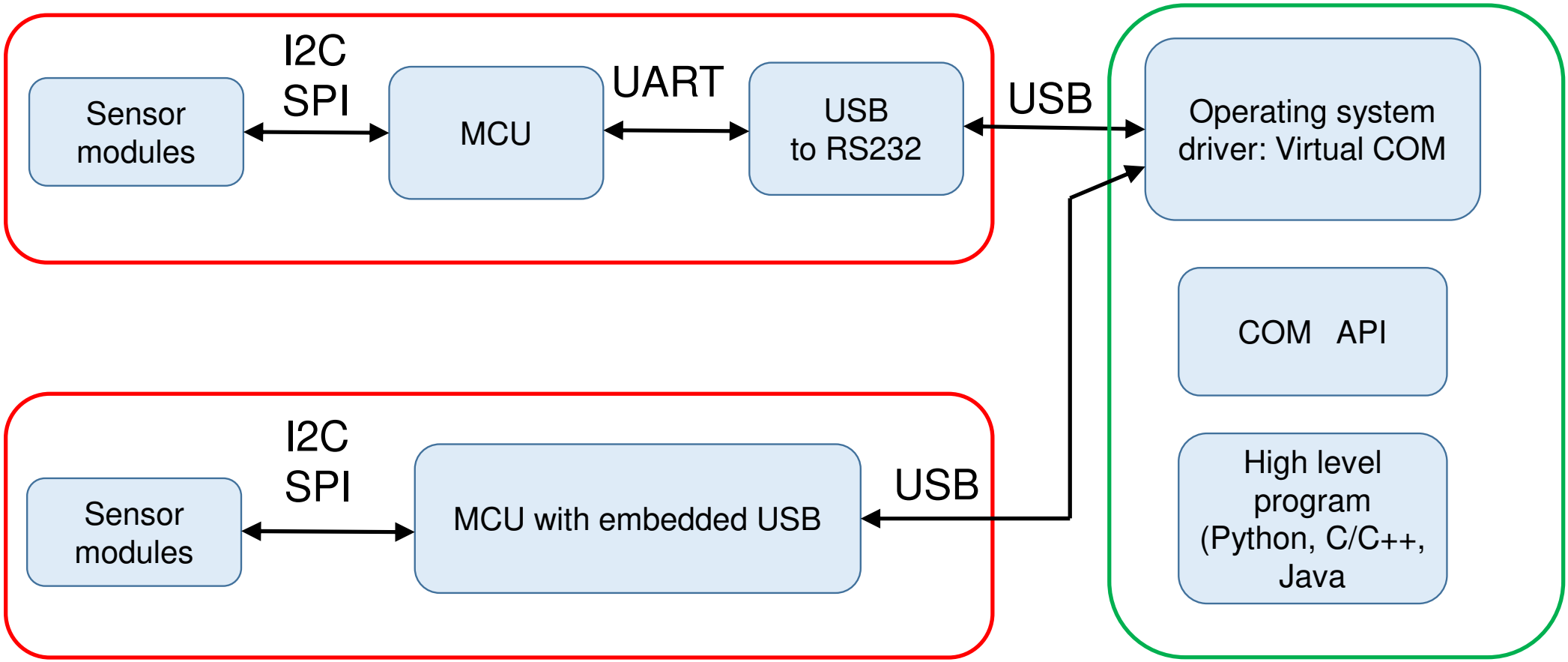
UART



Optional lines for handshaking

Only connection between two devices

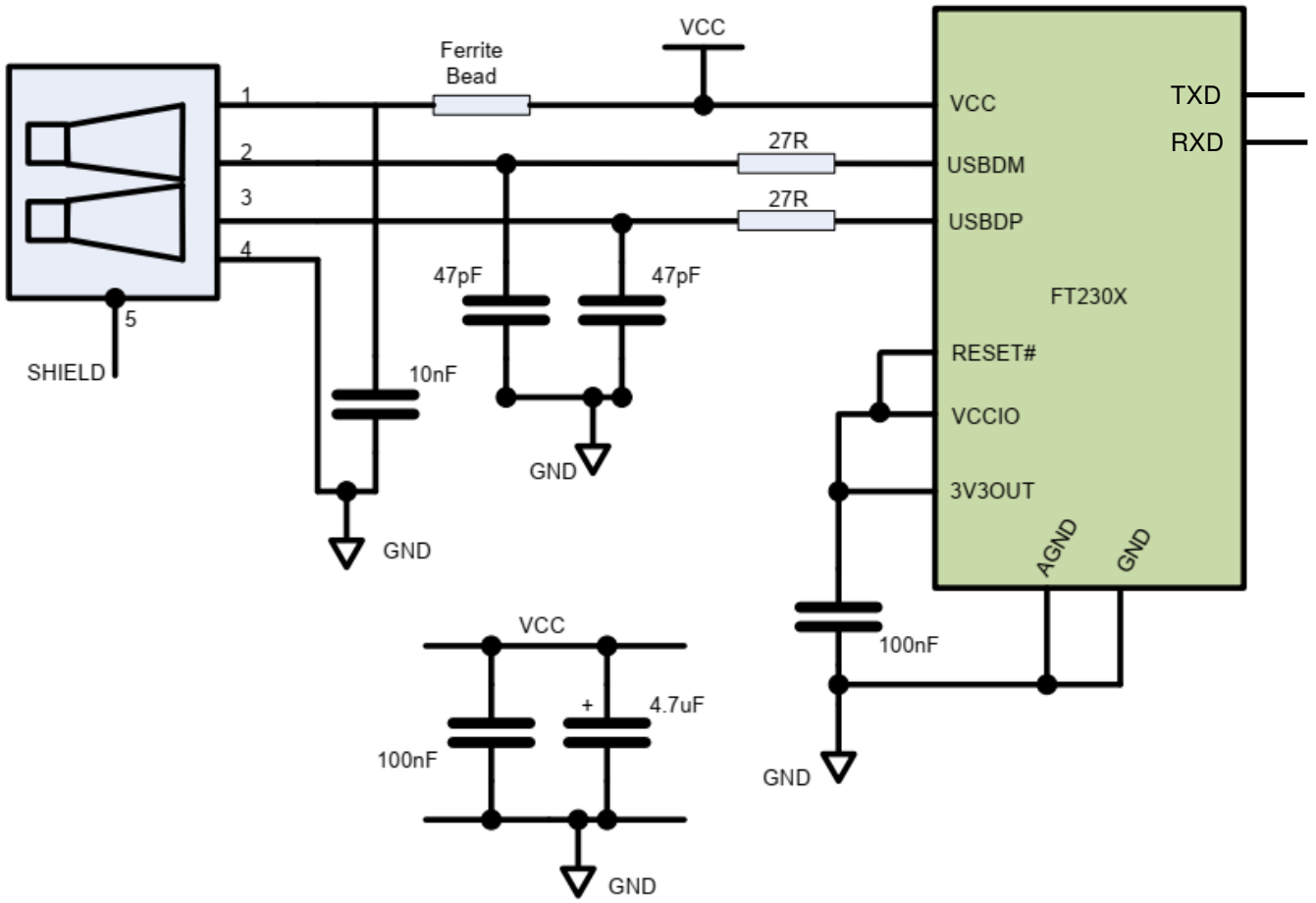
Possible **wired** links between a sensor node (MCU) and a high-level computer



Example of USB to RS232 converter

FTDI Chip

FT230X



Example of MCU with SPI/I2C and RS232

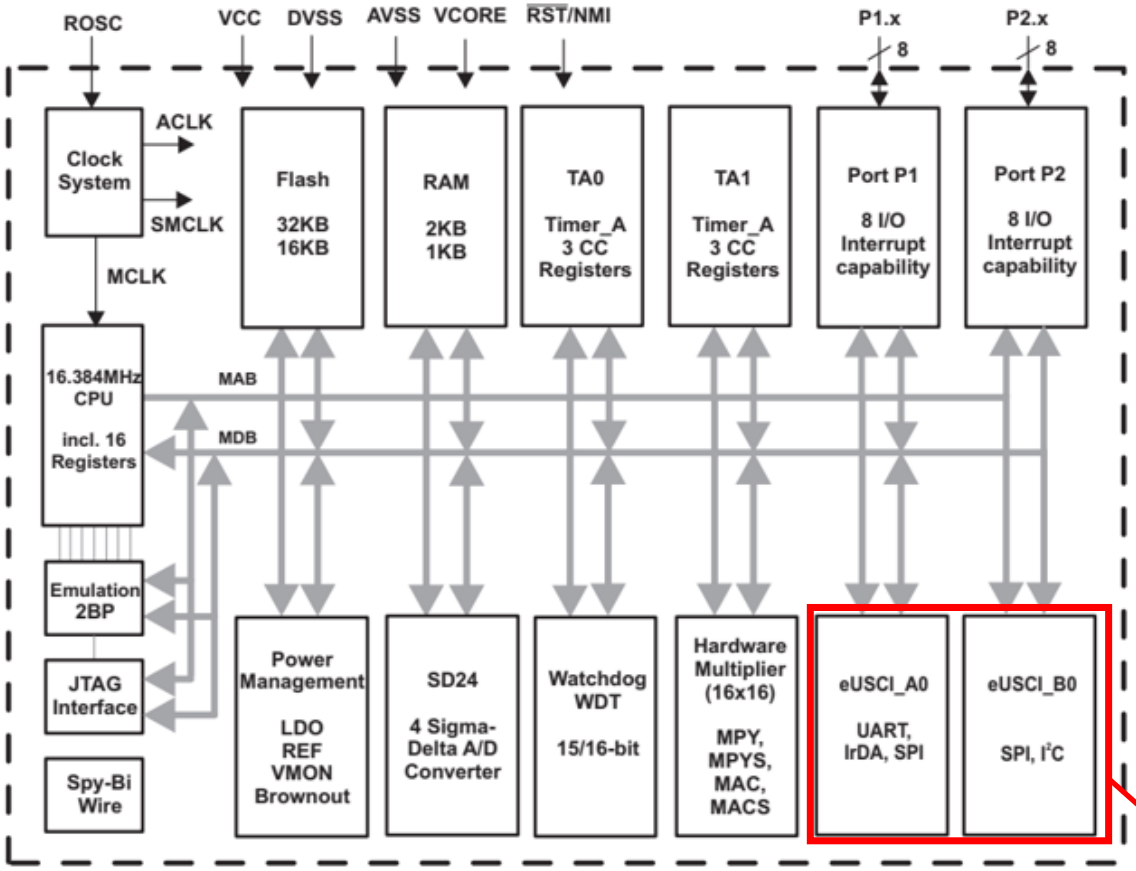
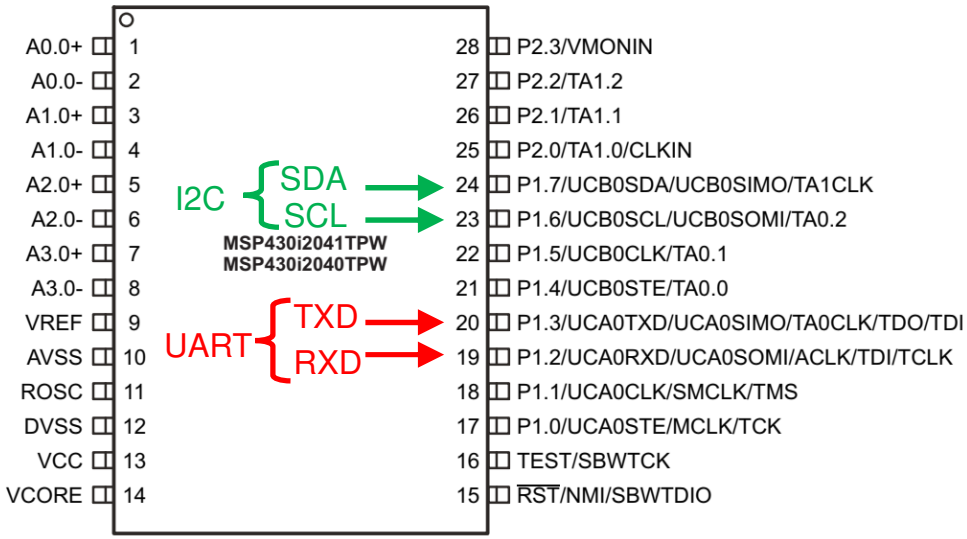


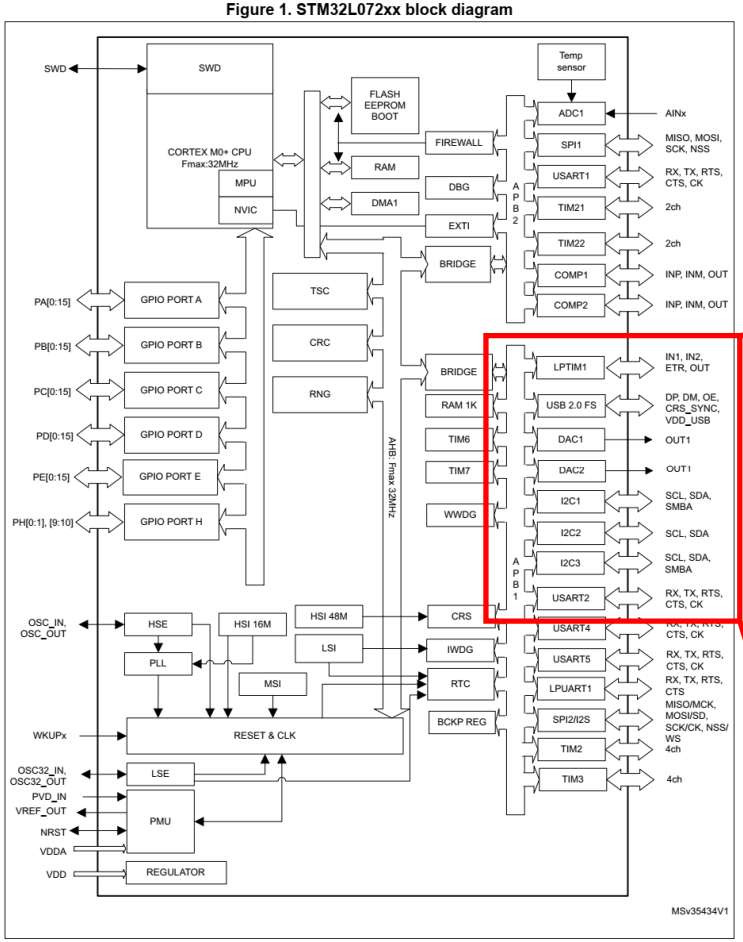
Figure 1-1. Functional Block Diagram - RHB Package - MSP430i204x

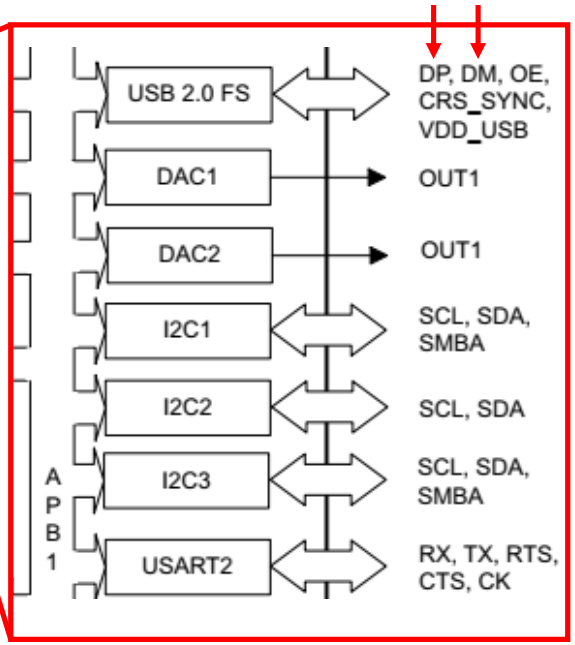


Versatile peripherals that can be configured to implement different communication protocols

Example of MCU with embedded USB

Figure 1. STM32L072xx block diagram



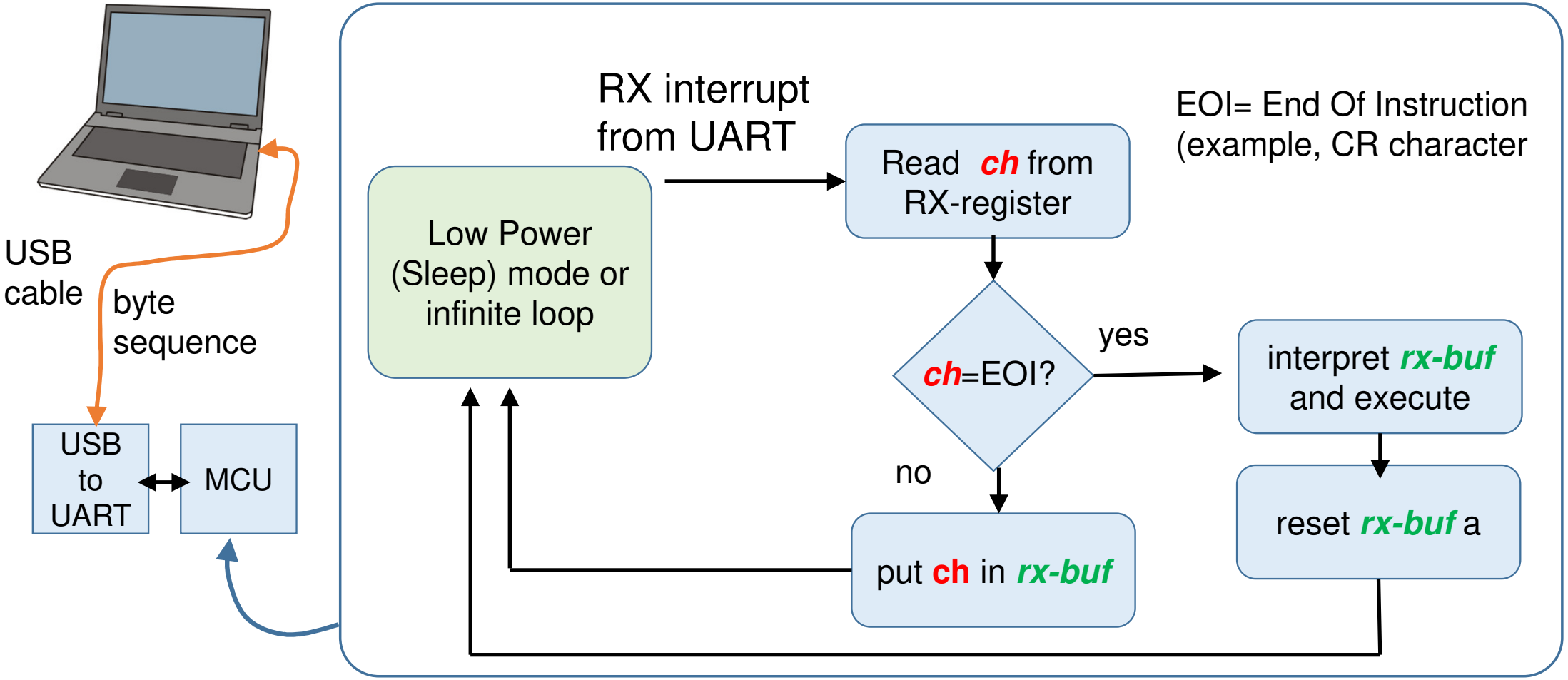


life.augmented

STM32L072x8 STM32L072xB
STM32L072xZ

Ultra-low-power 32-bit MCU Arm®-based Cortex®-M0+, up to 192KB Flash, 20KB SRAM, 6KB EEPROM, USB, ADC, DACs

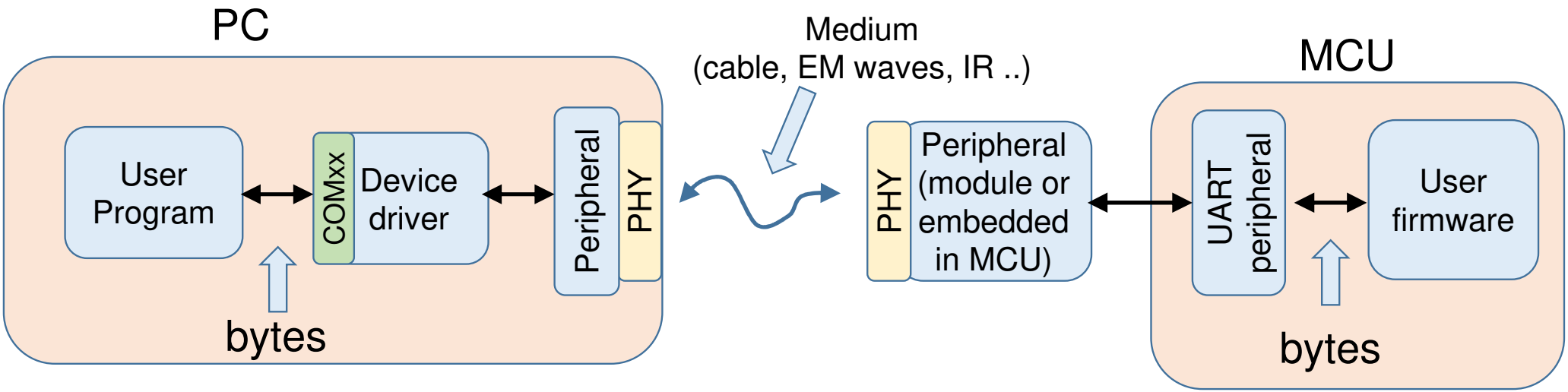
Simple slave program (user firmware) on the MCU



On the PC side: Windows OS - VCP (Virtual COM port)

- The COM port was originally the way a real RS232 port was accessible from programs in IBM PCs running the MSDOS.
- The RS232 defined logic levels that nowadays no more convenient (i.e. -12 V, +12 V). The communication protocol (UART Universal Asynchronous Receiver-Transmitter) has survived with voltage levels compatible with current integrated circuit supply voltages (e.g. 3.3 V, 5 V).
- A **Virtual COM Port** is a software object that simulate the presence of a physical RS232 port, and allows communication with a device that supports data transmission and reception as stream of bytes

Virtual COM port



User program:

1. opens the COM port, specifying the baud rate (e.g. 9600 b/s. This must be the same as the target device)
2. Write and/or read from the port sequence of bytes.
3. Closes the port at the end of the communication.

Device driver:
When the device is installed for the first time, one or more virtual COM ports are created.
For example:

COM8

Equivalent of COM ports in non-Windows operating systems

The advantage of Windows VCP is that this standard is used for a large variety of communication devices, not limited to USB but including also Bluetooth (BT). Many program can be adapted to switch from USB and BT by simply changing the number of the virtual COM.

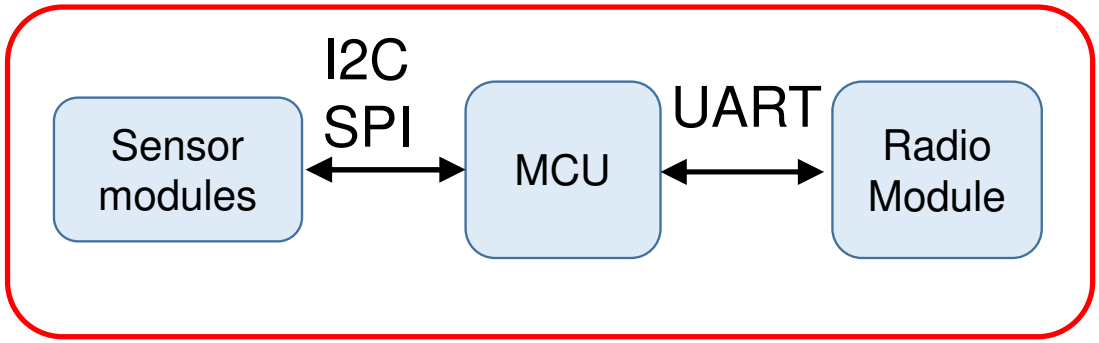
In Linux OS, COM ports are called "tty". Popular USB to UART conversion modules (e.g. FTDI modules) comes with a driver that installs a tty port that can be opened and used as a VCP. There is not an equivalent VCP for Bluetooth and special libraries has to be used with a syntax for sending bytes that is somewhat different from VCP one

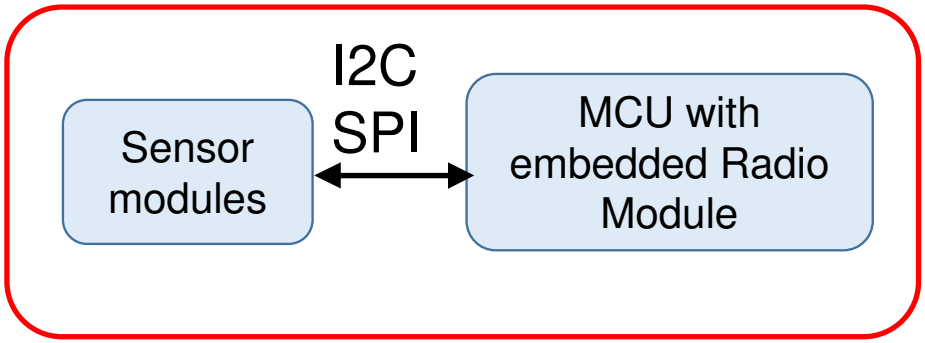
In Android systems, special Java classes have to be used to open a connection over a BT or USB link. To enable connection from the USB connector of a smartphone to a slave module, the USB must be OTG (on-the-go) enabled.

Possible wiress links between a sensor node (MCU) and a high-level computer

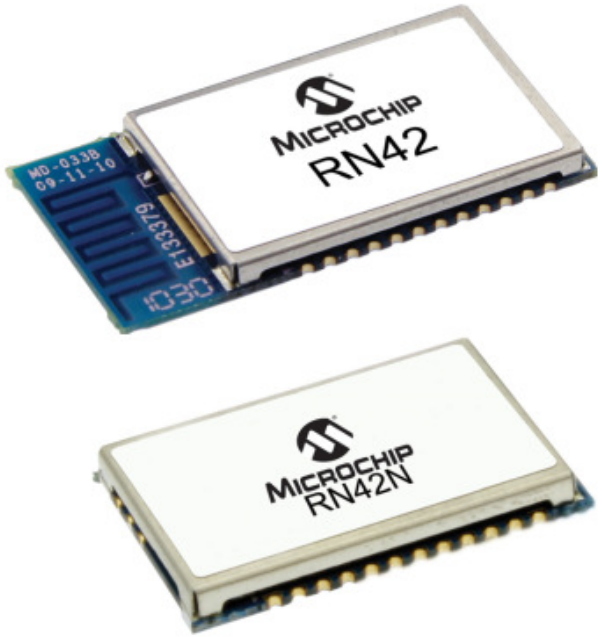
Protocols: Bluetooth
WiFi
} Embedded in most fixed or mobile computers

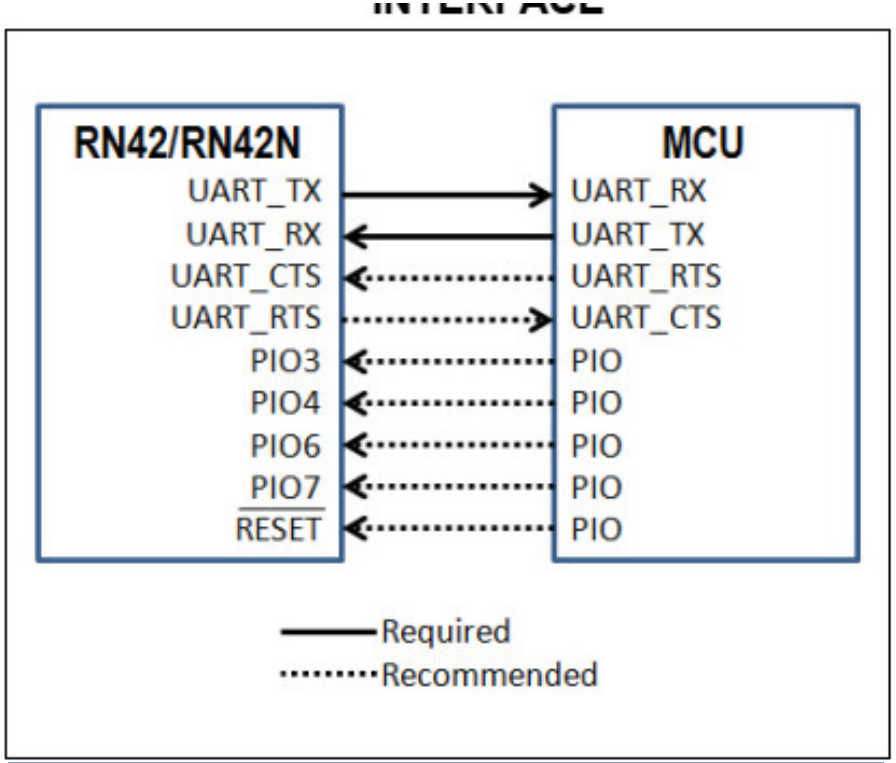
Zigbebee (IEEE 802.15.4)
LoRaWAN
} Need a Bridge to a wireless or wired (USB, Ethernet) standard





Example of Bluetooth module





Connection of the module to an MCU

Stable and intermittent mode in a wireless link

Stable link: allows the sensor node and the data collector (e.g. the high-level computing machine) to initiate a data transfer at any time and in any direction (uplink or downlink).
Examples: BT or Zigbee modules in "Cable Replacement Mode"
WiFi connection.

Intermittent link: the sensor and the data collector can talk each other only in short time intervals.

The stable link is more adequate when the data collector must send data to the sensor node in frequent and random fashion.
Unfortunately, this mode is extremely inefficient in terms of power

Power consumption in Wireless Modules

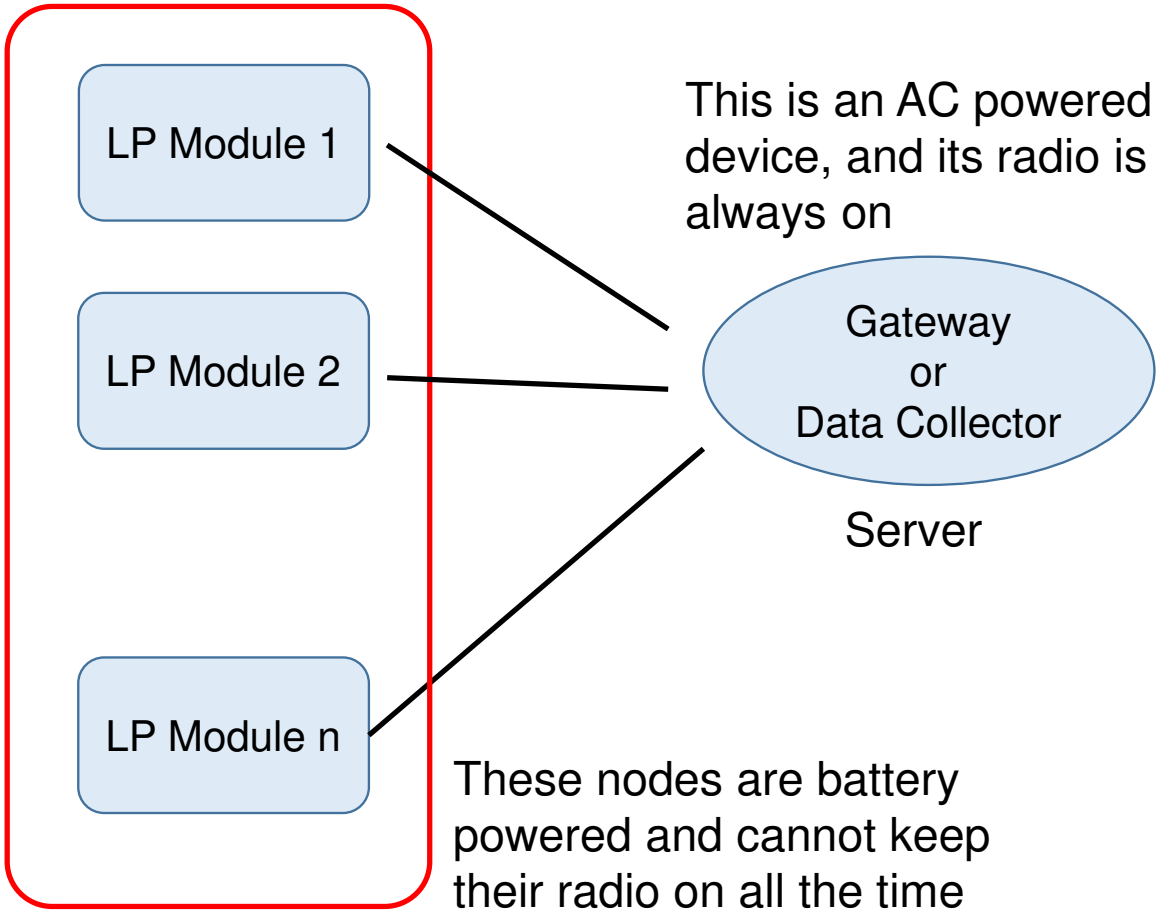
Important: in low power wireless modules, keeping the receiver on requires as much the power as transmitting data.

In a stable link, the sensor node must keep his receiver on, even when it is not communicating, since the data collector may send data / command at any time. Keeping the receiver on imply a power consumption generally in the range 10-40 mA for commercially available modules.

For battery operated modules, a steady power consumption of mA is often not acceptable

Different methods for intermittent link have been developed. Their details depend on the type of wireless protocol.

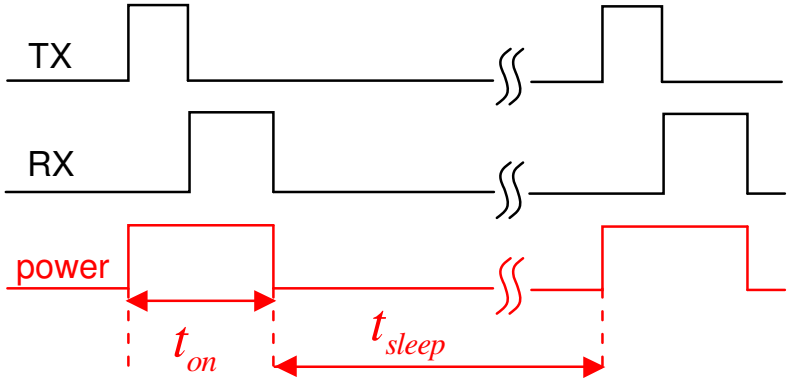
Intermittent link mode: a possible uplink-based approach



This is an AC powered device, and its radio is always on

These nodes are battery powered and cannot keep their radio on all the time

Used by LoRaWAN

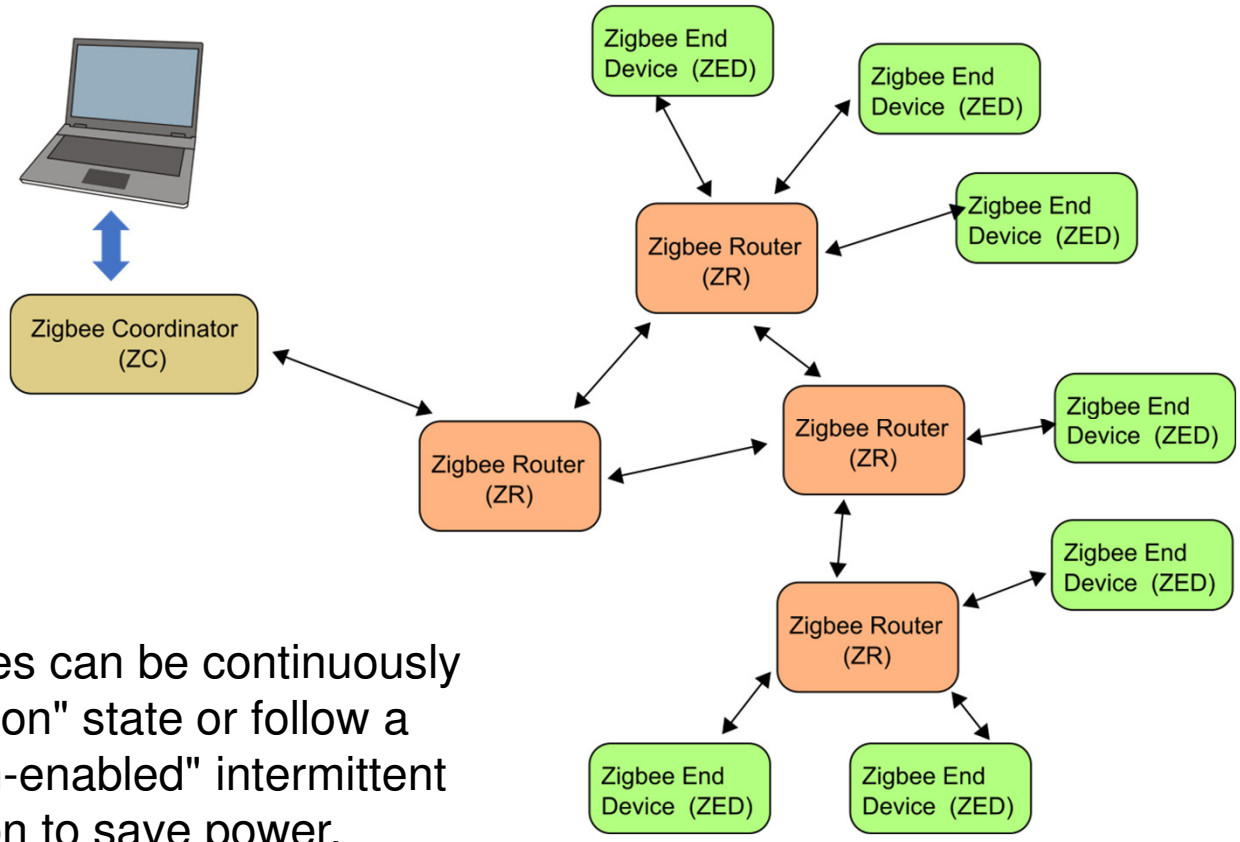


A sensor node that needs to communicate with the server, turns the radio-on and transmit a brief message ("uplink") to the gateway

Then, keeps the receiver on for a short period. Waiting for the server answer. If no answer is received, the radio is turned off.

$$duty-cycle = \frac{t_{on}}{t_{on} + t_{sleep}}$$

A MESH Wireless Sensor Network based on ZigBee



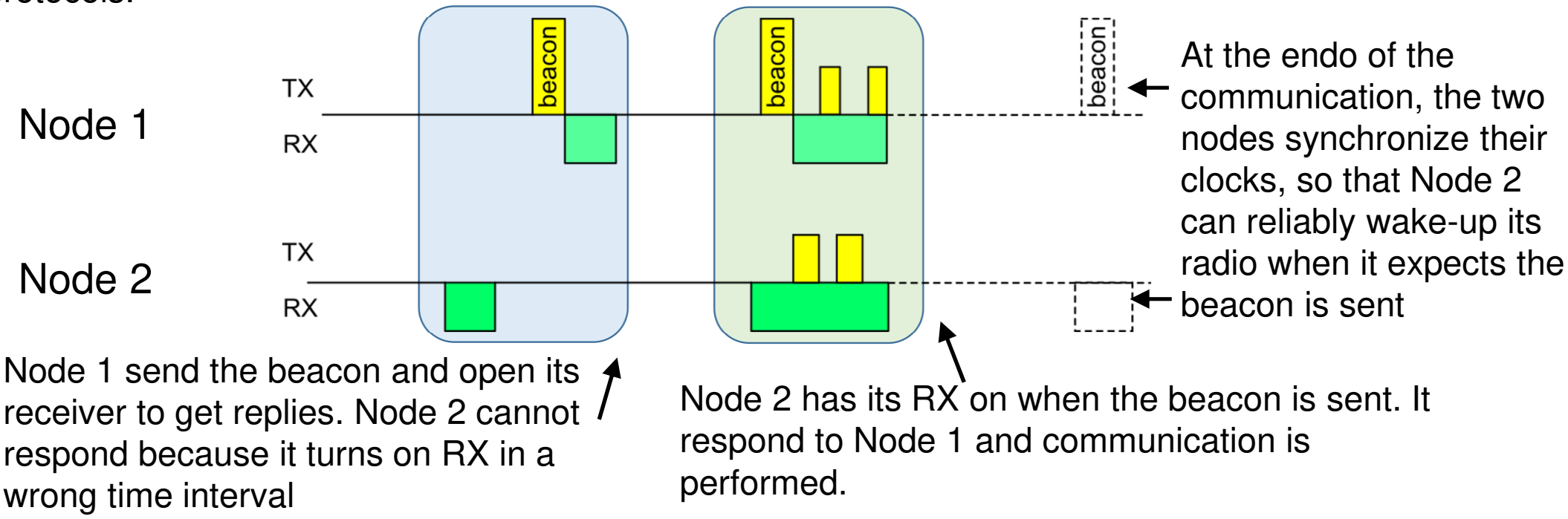
ZEDs are battery powered and perform sensing operations.
They communicate with a routing node (ZR) periodically. A ZR node (parent) can communicate with several ZEDs

ZR nodes can be continuously kept in "on" state or follow a "beacon-enabled" intermittent operation to save power.

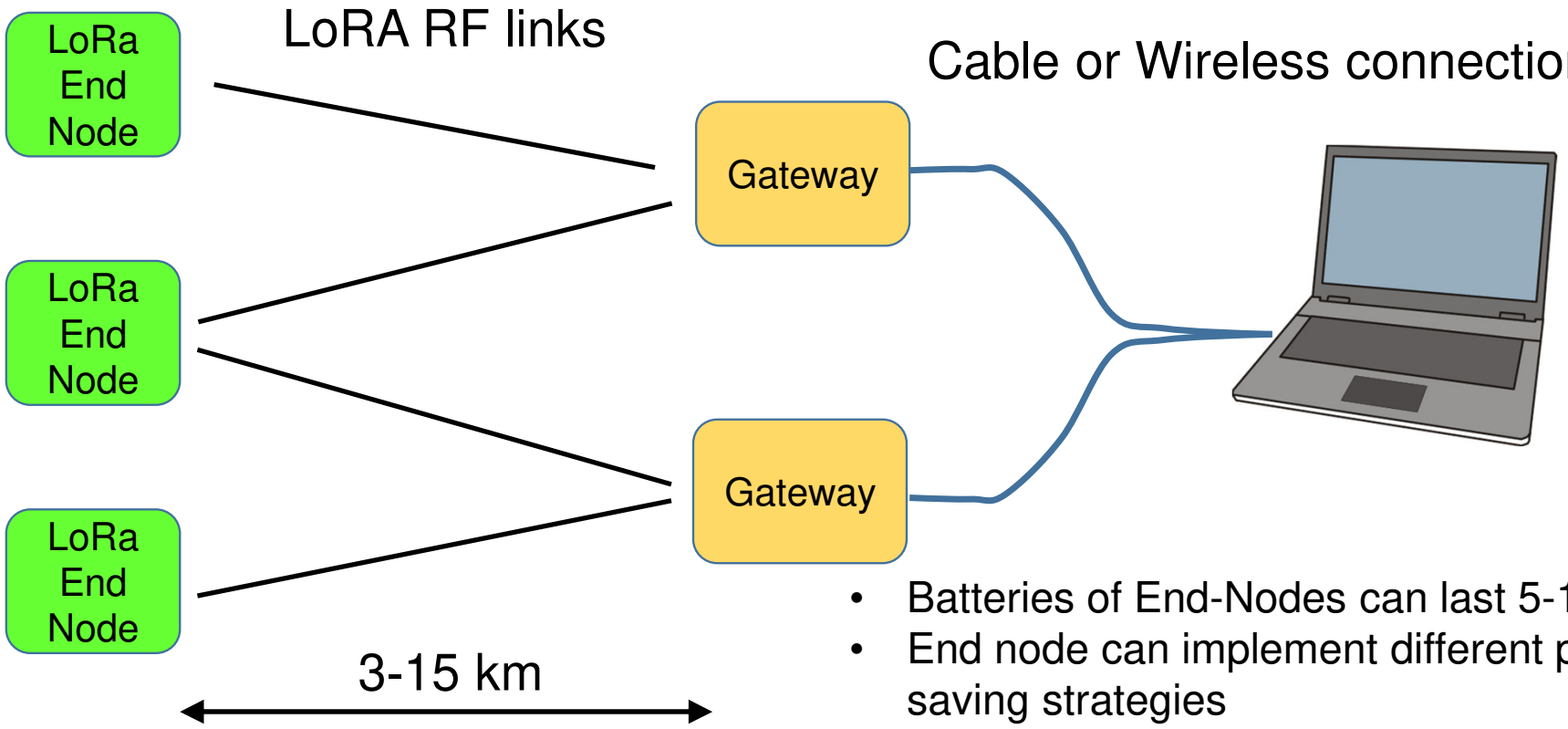
Frequencies (Europe): 868 MHz and **2.4 GHz**

Beacon-enabled intermittent mode

Beacons are special transmissions used to indicate that a wireless node is alive and ready to communicate. If the receiver gets the beacon, then communication can start between the two nodes. Beacons are used also in low power, WiFi intermittent links and in many other wireless network protocols.



The LoRaWan Star Sensor Network



- Batteries of End-Nodes can last 5-10 years
- End node can implement different power saving strategies
- Non-GPS localization is possible through comparison of data received by different gateways

Frequencies 867-869 MHz (Europe)

Examples of commercial products: LoraWan