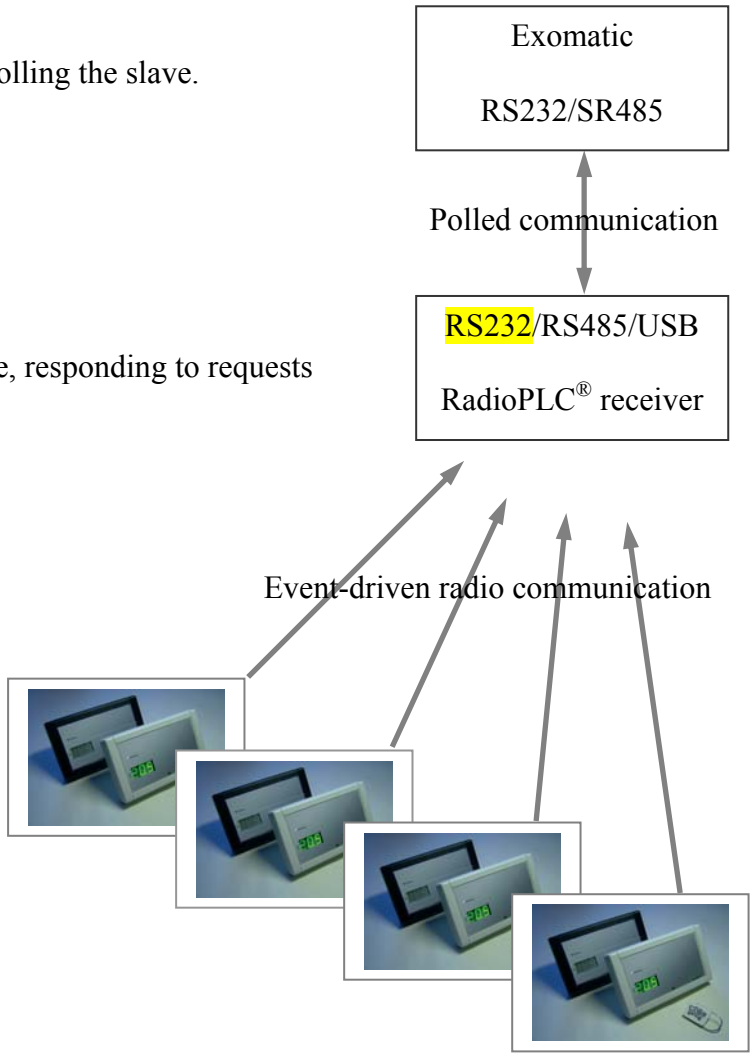
RadioPLC® versus Exomatic ModBus communication.

Outline of communication implementation to enable wireless collection of data from the RadioPLC® system. The primarily selected way of communication is noted with yellow markings. Communication may be asynchronously point-to-point via RS232 or multidrop with multiple receivers via RS485.

Exomatic DUC = ModBus master, polling the slave.

RadioPLC® receiver = ModBus slave, responding to requests

RadioPLC® transmitting sensors with unique RF-ID and a local identity of 1-60.



The ModBus communication might be possible to run simultaneously with the RadioPLC® terminal mode, thus simplifying test, setup and field service.

ModBus message layout (RTU-mode = Binary)

The start and end of messages are embedded by a 3,5 character timeout.
Most significant (MSB), high order bytes (HO), comes first (to the left below)
Least significant, (LSB), low order bytes (LO), comes next (to the right below)

Overall message layout

Slave no	Function	Data	CRC
1 byte	1 byte	X bytes	2 bytes

Slave number: 1 - 247(dec) are valid values in multipoint RS485 connections
0 = used for broadcast write to multiple units
1 = Used slave number in RS232 point-to-point connections

Function code:
1 = Read coil status
2 = Read input status
3 = Read holding registers
4 = Read input registers
5 = Force single coil
6 = Preset single register
15 = Force multiple coils
16 = Preset multiple registers

Data: As described below

CRC: Standard ModBus CRC (Cyclic Redundancy Check)

Specific query from the Exomatic; E.g. ask for some or all the 60(dec) 3C(hex) temperature nodes.

Slave no	Function	Data	CRC
1 byte	1 byte	4 bytes	2 bytes
		First addr Number	
01	04	0000 003C	

Proposed addressing scheme;
0000 – The receiver itself
0001 – First temperature node, local ID 1
0002 – Second temperature node
Etc.
003C – Last possible temperature node, nr 60(dec)

Later on, higher addresses (64-127 and 128-191 etc.) may be used for readout of supplementary information such as device errors, radio connectivity, timeouts, etc.

Answer from the RadioPLC®

In case of bad CRC, or timeout, no answer.

In case of other errors;

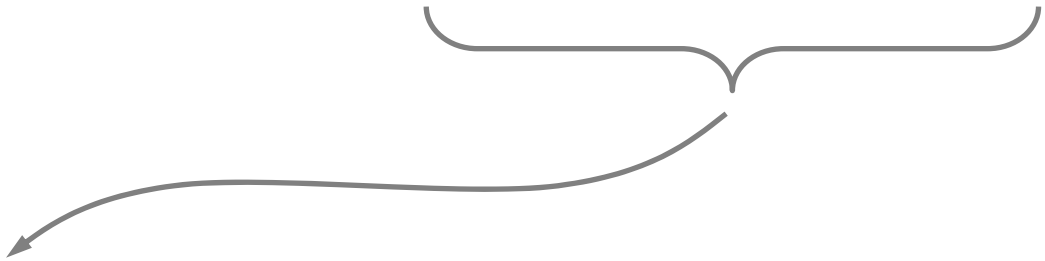
Slave no	Err. response	Data len	Error code	CRC
1 byte	1 byte	1 byte	1 byte	2 bytes
01	84(hex)	1	XX	

Error codes;

1 "Illegal function"
2 "Illegal data address"
3 "Illegal data value"
4 "Slave device failure"
5 "Acknowledge"
6 "Slave device busy"
8 "Memory parity error"
10 "Gateway path unavailable"
11 "Gateway target device failed to respond"

The RadioPLC® receiver answers immediately with current and valid readings, this is a must for the ModBus implementation. The answer is as follows when all conditions above are met;

Slave no	Function	Data len	1:st data	2:nd data	Etc.	Last data	CRC
1 byte	1 byte	1 byte	2 bytes	2 bytes		2 bytes	2 bytes
01	04	02-120(d)					



The returned data is HO, LO.
The data is a signed binary integer.
The integer represents 1/10 degrees temperature, e.g. 26,1 °C is sent as 261(dec) or 01 05 (hex). E.g. a negative temperature of -12,3 °C is sent as -123(dec) or FF 85 (hex).

Low level data format;

1 startbit
8 data bits
no parity (set by processor hardware)
1 stopbit (set by processor hardware)

Baud rate: 9600 baud (It is also possible to use 19200 or higher, but for simplicity and quick development we propose 9600 as a start)