ModbusTCP

SunSpec Tutorial Part I

Michael Schoeffler
mschoeffler.de

SunSpec Tutorial Series

SunSpec

Consortium and specification that aim at increasing the interoperability within the solar and distributed energy industry.

This tutorial series addresses only a fraction of SunSpec!

Look for more information on sunspec.org

Content		Example use case
Part I	*ModbusTCP* – *learn about a technology base of the SunSpec specification*	
Part II	*SunSpec Information Model* – *get to know which data can be exchanged with your system*	
Part III	*SunSpec Hands-on* – *use an ESP8266 or ESP32 to retrieve data from your system*	*The idea is to make use of SunSpec for a typical rooftop photovoltaic system*

What is the SunSpec Specification?

- Some years ago, communication with solar devices was done with proprietary protocols
- The SunSpec specification enables to access data on solar devices in a standardized way
  - Software applications can communicate with solar devices from different manufacturers by using the same mechanisms and protocol (“Sunspec”)
  - Already supported by a wide range of manufacturers
- The SunSpec specification builds up on Modbus
- (…more on Sunspec in Part II of this series)

What is Modbus?

- Modbus is a communication protocol
- Originally published by Modicon in 1979
  - Still widely used, especially in the industrial domain
- Modbus is governed by the Modbus Organization
  - Specification is freely available at www.modbus.org
- Main purpose: store and retrieve data from devices
- As of today, Modbus is the most important protocol in order to interchange data based on SunSpec specification

- This tutorial covers only parts of Modbus that are relevant for SunSpec
  - There is much more to discover about Modbus!

Modbus Overview

- Modbus describes a “full communication stack”
  - Therefore, Modbus provides multiple specifications and documents

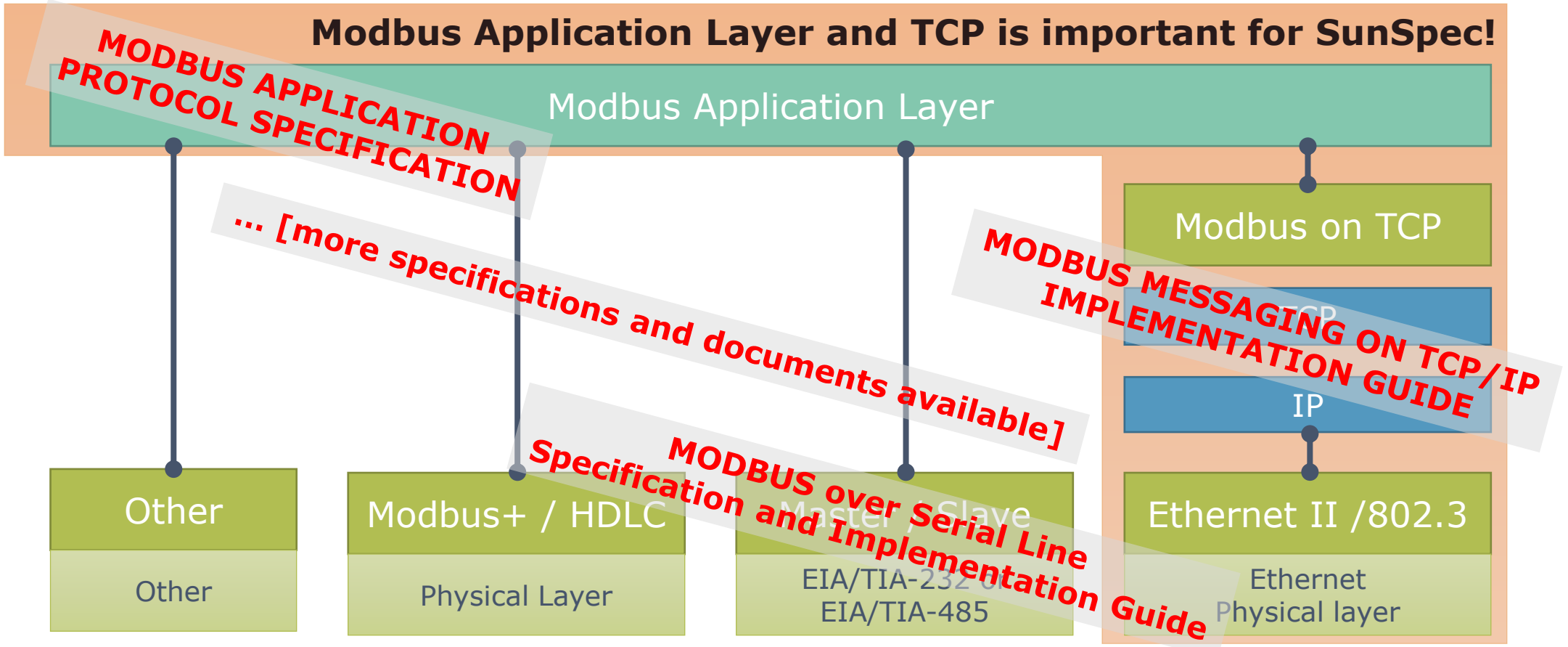


Illustration based on Figure 1, Modbus Application Protocol V1.1b3, modbus.org, 2012

Modbus main communication principle

- On application level, Modbus follows a client–server model

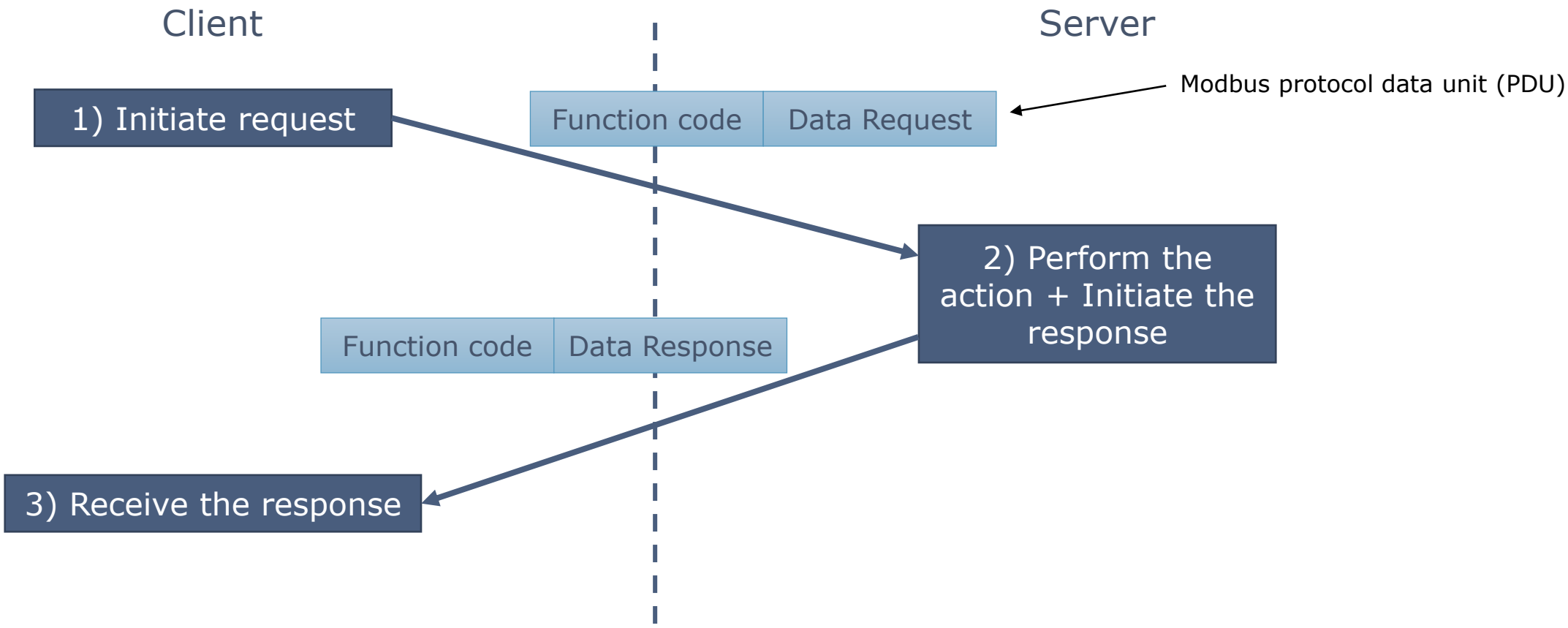


Illustration based on Figure 4, Modbus Application Protocol V1.1b3, modbus.org, 2012

Client, server, master, slave etc. confusion

- When working with Modbus, the words client, server, master, slave etc. are sometimes used in “a confusing way”
- Modbus follows on application layer a client—server model
  - On a serial line layer, Modbus follows a master—slave model (technical reasons)
  - On TCP/IP layer, Modbus follows again a client—server model

> **MODBUS over serial line specification and implementation guide V1.02:**
> “MODBUS application layer messaging protocol, positioned at level 7 of the OSI model, provides client/server communication between devices connected on buses or networks.
> On MODBUS serial line the client role is provided by the Master of the serial bus and the Slaves nodes act as servers.”

- Many libraries support multiple Modbus implementations (TCP/IP, RTU, …)
  - Often, master—slave wording is also used on programming level
    - Example: “a class of name ‘slave’ represents server logic of Modbus TCP implementation”
    -

```
class ModbusTCPSlave
{
public:
  ModbusTCPSlave(void);
#ifdef MB_ETHERNET
    void begin();
    void begin(uint8_t ip[4],uint8_t gate
```

Other example from “ESP8266_Industrial_ModbusTCP_V2” library

Client, server, master, slave etc. confusion

- Press release 09 July 2020: “Modbus Organization Replaces Master-Slave with Client-Server”

PRESS RELEASE

Modbus Organization Replaces Master-Slave with Client-Server

July 9, 2020 — The Modbus Organization Board of Trustees announces it is expunging all occurrences of inappropriate language of the query and response paradigm of Modbus communications. With recent events, all companies and individuals with the ability to make positive changes should do so. As the holder of the standard for an industrial network with one of the largest installed bases in the world, it is important to complete the replacement of the current terminology with more thoughtful references. All instances of "master-slave" in the organization's literature and on its website will be removed. Current electronic content is being updated as of this announcement.

Board Chair Todd Snide commented, "The time is overdue for these words to be removed. The terms are unnecessary and insensitive to the people who experience racism in this country and around the world. We urge our member companies and all those who use the Modbus protocol to take similar action to rid their documentation and communications of these terms."

The organization is using "client-server" to describe Modbus communications, characterized by communication between client device (s), which initiates communication and makes requests of server device(s), which process requests and return an appropriate response (or error message).

Who’s the server? / Who’s the client?

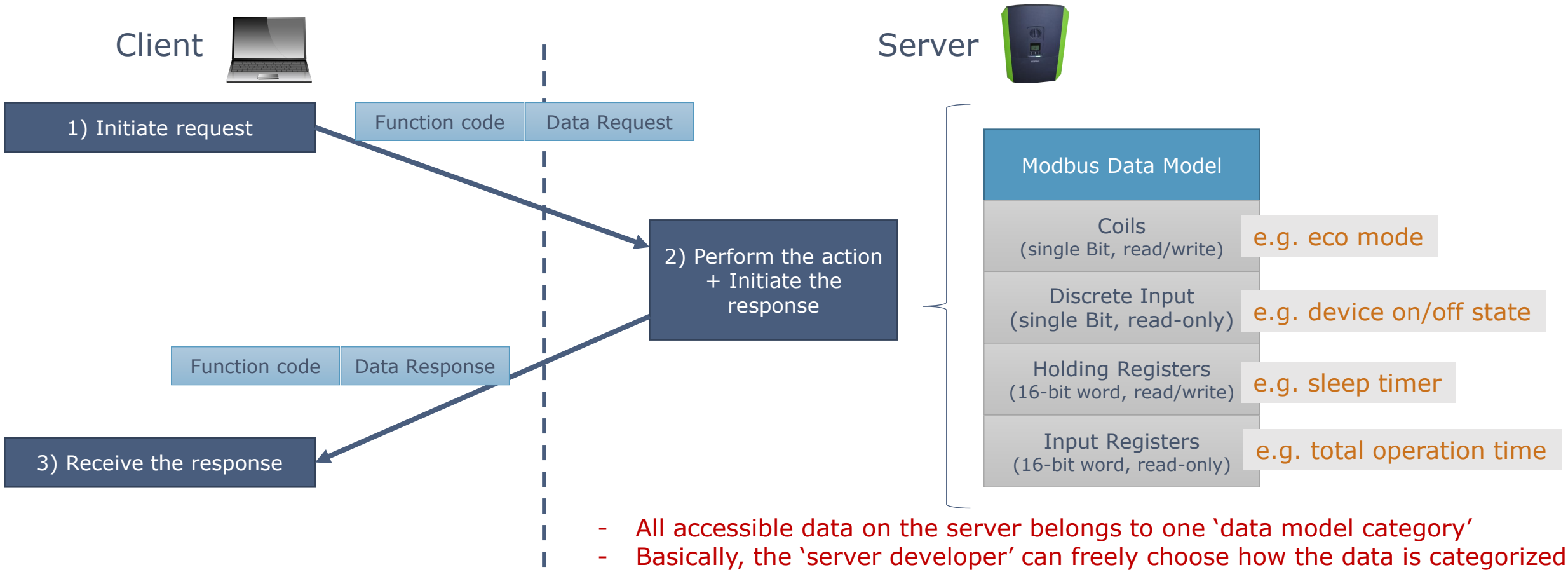
- Typical setup:
  - …client (or master) is a computer that wants to read or modify data
    - e.g. a notebook requesting data from a solar inverter device

  - …server (or slave) is a device that holds data to be read or to be modified
    - e.g. a solar inverter device

  - Other variants are also possible!
    - E.g. in the industrial domain, programmable logic controllers (PLCs) are sometimes clients and servers at the same time

Modbus main communication principle

- Modbus data model (1/2)



- All accessible data on the server belongs to one 'data model category'
- Basically, the 'server developer' can freely choose how the data is categorized!

Illustration based on Figure 4, Modbus Application Protocol V1.1b3, modbus.org, 2012

Modbus main communication principle

- Modbus data model (2/2)

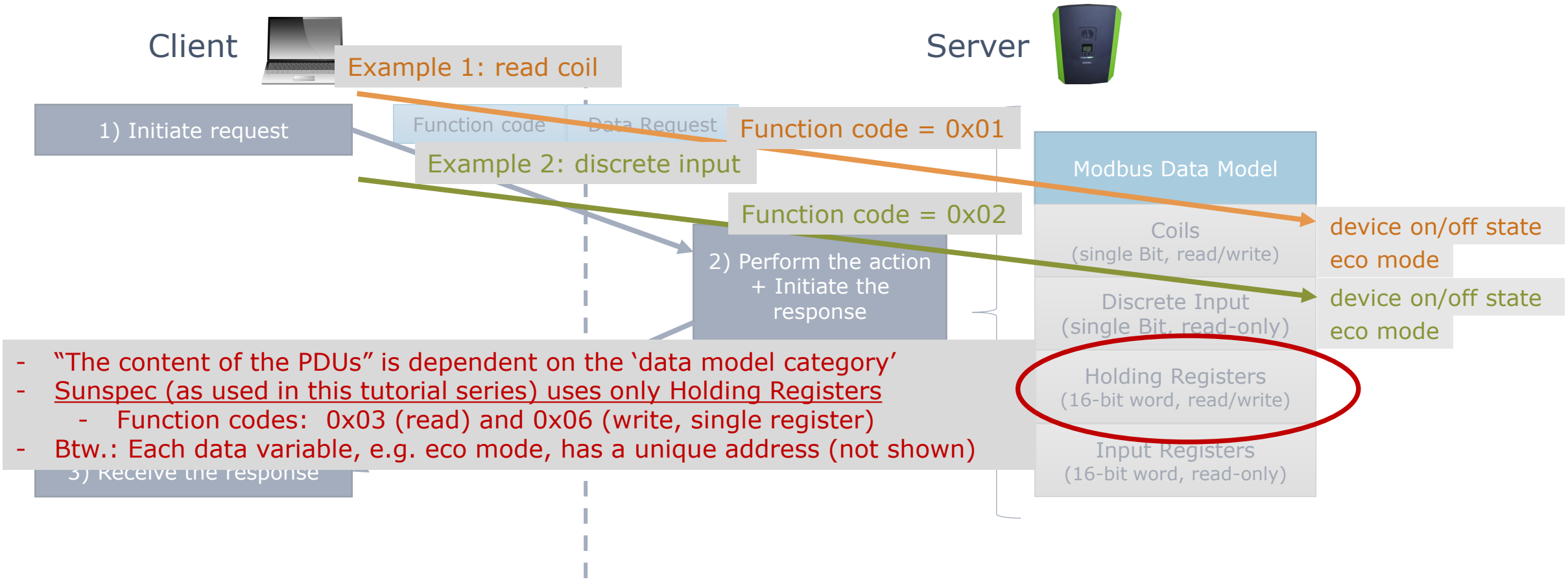
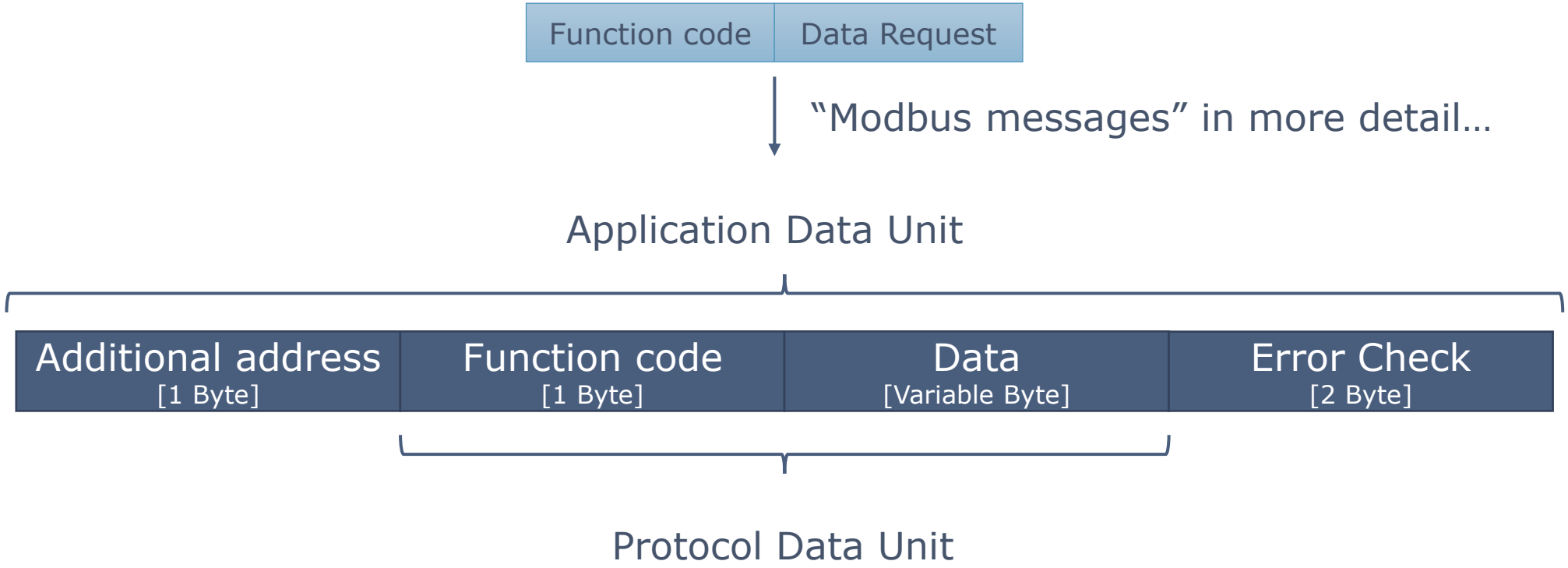


Illustration based on Figure 4, Modbus Application Protocol V1.1b3, modbus.org, 2012

Modbus main communication principle

- “Modbus messages”



When working with SunSpec, it is sufficient to know that you need a device address, function code and data address to access data via Modbus!

Additional remarks

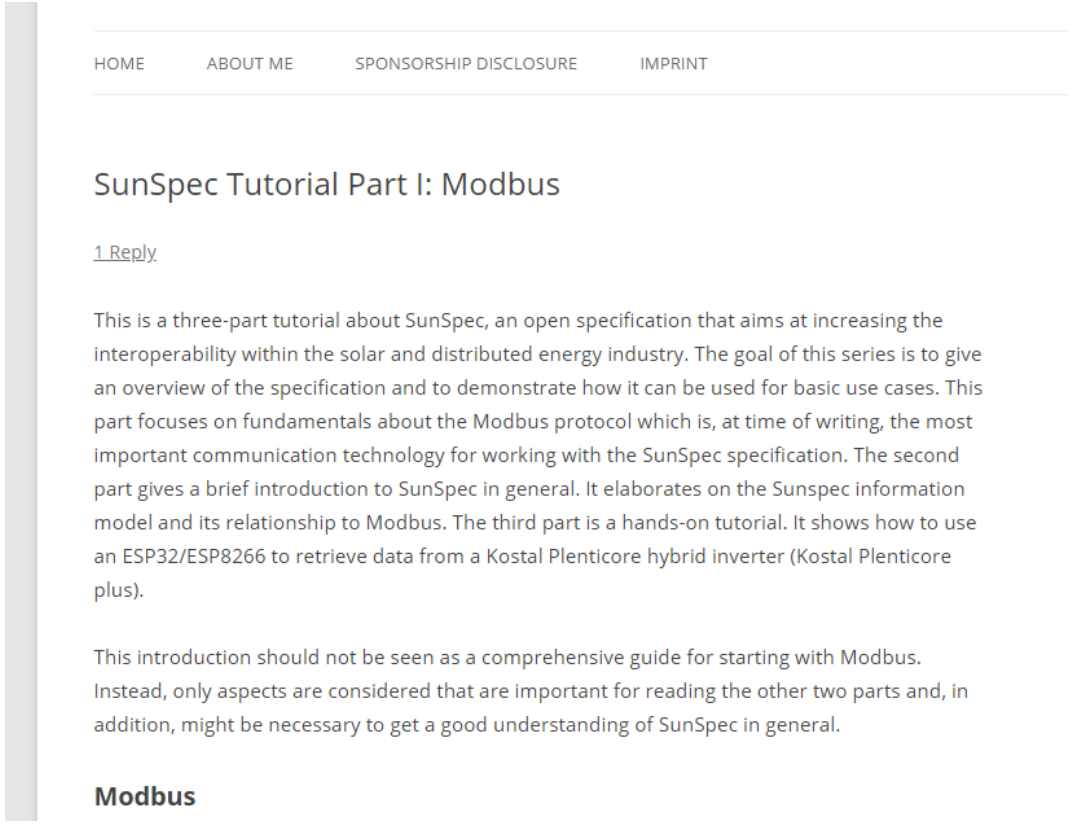
- “Modbus on TCP” is mainly relevant for SunSpec
  - builds up on TCP/IP
- Each device with SunSpec support must be connected to a network to work with the device
  - ModbusTCP Servers can be accessed by an IP address + a TCP port
  - The IP address can often be set on the device or is set by your router (“DHCP”)
  - TCP Port 502 is reserved for the Modbus Protocol and, therefore, typically used by device manufacturers for the default device settings

Wrap up

- SunSpec: “Consortium and specification that aim at increasing the interoperability within the solar and distributed energy industry”
  - Enables to access data on solar devices in a standardized way
- Modbus TCP is a technological basis for SunSpec
- Devices with SunSpec support (e.g. solar inverters, battery storages, etc.) come with a Modbus server
- When working with ‘SunSpec-enabled’ devices, you should have a minimal understanding of…
  - ModbusTCP,
  - Device addressing,
  - Function codes,
  - Data layer model and
  - data addressing.

Add-on: Written Article

- Written article available on my personal website: https://mschoeffler.de
  - https://www.mschoeffler.de/2019/11/02/sunspec-tutorial-part-i-modbus/



HOME ABOUT ME SPONSORSHIP DISCLOSURE IMPRINT

SunSpec Tutorial Part I: Modbus

1 Reply

This is a three-part tutorial about SunSpec, an open specification that aims at increasing the interoperability within the solar and distributed energy industry. The goal of this series is to give an overview of the specification and to demonstrate how it can be used for basic use cases. This part focuses on fundamentals about the Modbus protocol which is, at time of writing, the most important communication technology for working with the SunSpec specification. The second part gives a brief introduction to SunSpec in general. It elaborates on the Sunspec information model and its relationship to Modbus. The third part is a hands-on tutorial. It shows how to use an ESP32/ESP8266 to retrieve data from a Kostal Plenticore hybrid inverter (Kostal Plenticore plus).

This introduction should not be seen as a comprehensive guide for starting with Modbus. Instead, only aspects are considered that are important for reading the other two parts and, in addition, might be necessary to get a good understanding of SunSpec in general.

Modbus