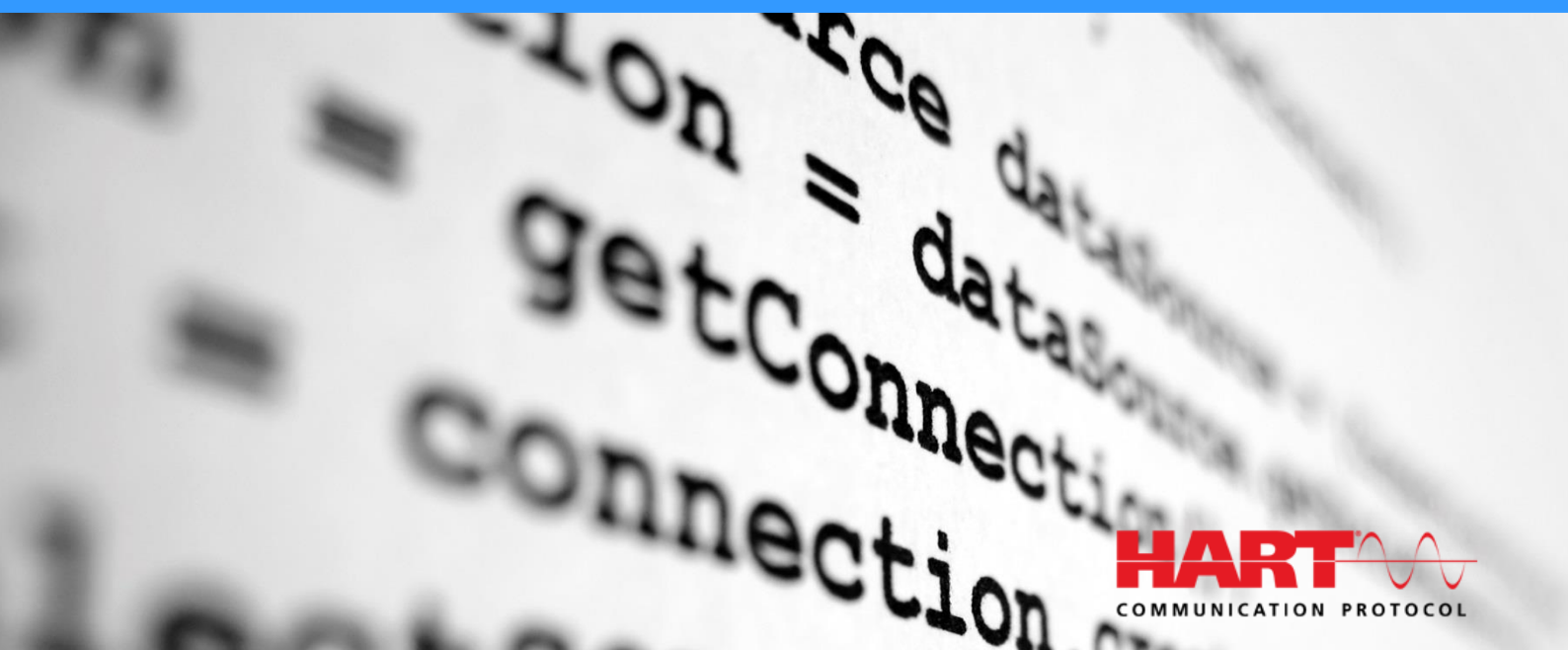


## MESCO HART<sup>®</sup> Slave Stack 7.6



## Datasheet for HART<sup>®</sup> Slave Stack





## Table of content

Table of content .....	2
1. Overview .....	3
2. Stack Highlights .....	3
3. Delivery Content.....	3
4. Technical Data .....	4
4.1. System Requirements.....	4
4.2. HART Slave Stack 7.6 .....	4
5. Stack interfaces / API - Overview.....	7
6. Stack Architecture.....	8



## 1. Overview

The HART Slave Stack is a quick and easy solution to add HART 7 capabilities to your field device. It includes all necessary features to achieve a HART 7 compliant device. This includes all universal commands and a large number of common practice commands. The HART Slave Stack from MESCO is backwards compatible to HART 5 and HART 6 masters.

## 2. Stack Highlights

- Pre-certified stack passes HCF conformance tests
- Full description and source code
- Reduced development time & development risk
- Small & efficient footprint: runs on almost any controller (8/16/32 bit)
- Supports various integration scenarios, with or without operating system (RTOS)
- Supports external HART Modem
- Easy parameterization (device specific)
- Well defined hardware interface for platform customization
- API for device specific HART commands
- API for command 48 data (device specific)
- API for non-volatile data
- Prepared for actuator

## 3. Delivery Content

- Source code
- Integration manual
- Application Programmable Interface
- Support
- Transmitter demo application
- Actuator demo application

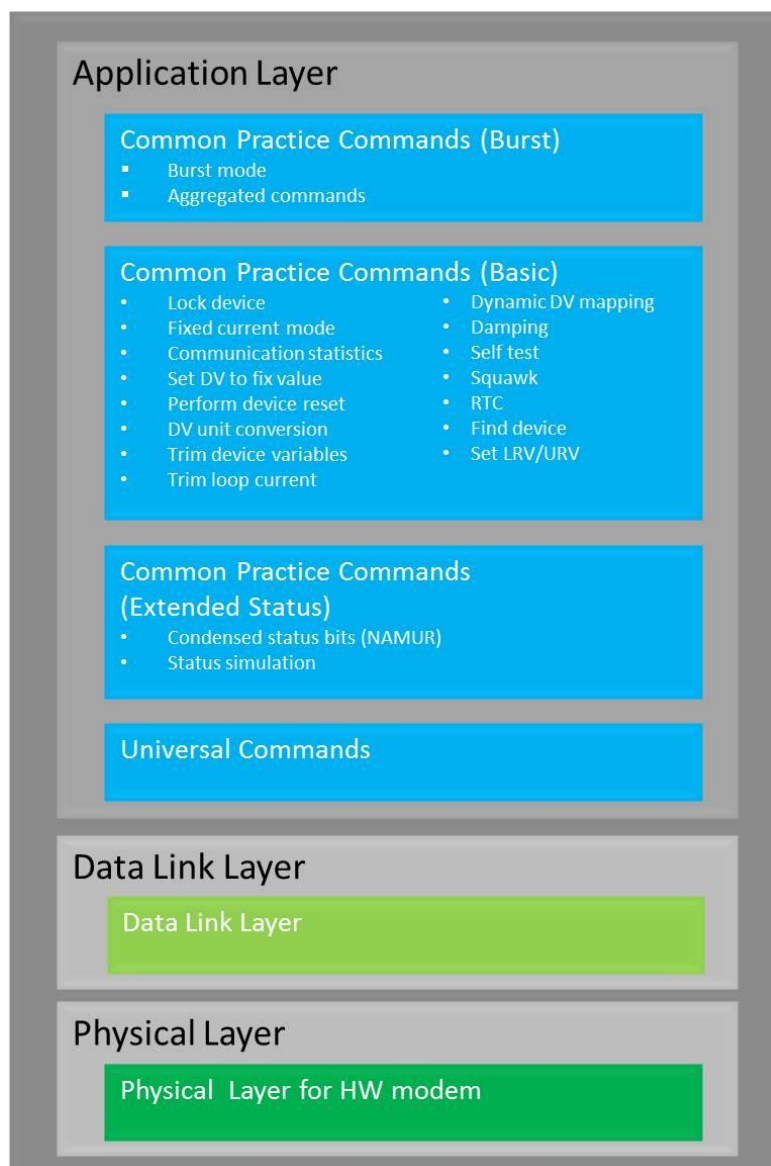


## 4. Technical Data

### 4.1. System Requirements

<b>Data memory (RAM)</b>	1KB – 2KB (depending on enabled functionality)
<b>Program memory (FLASH)</b>	20KB – 25KB (depending on enabled functionality)
<b>Microcontroller</b>	8-bit/16-bit/32-bit controller $\geq$ 3MHz
<b>Hardware peripherals</b>	1 timer used as system time counter 1 UART with parity generator 2 GPIOs

### 4.2. HART Slave Stack 7.6





## The MESCO HART Slave Stack includes following features:

- **Universal commands**

- 0 Read Unique Identifier
- 1 Read Primary Variable
- 2 Read Loop Current And Percent Of Range
- 3 Read Dynamic Variables And Loop Current
- 6 Write Polling Address
- 7 Read Loop Configuration
- 8 Read Dynamic Variable Classifications
- 9 Read Device Variables with Status
- 11 Read Unique Identifier Associated With Tag
- 12 Read Message
- 13 Read Tag, Descriptor, Date
- 14 Read Primary Variable Transducer Information
- 15 Read Device Information
- 16 Read Final Assembly Number
- 17 Write Message
- 18 Write Tag, Descriptor, Date
- 19 Write Final Assembly Number
- 20 Read Long Tag
- 21 Read Unique Identifier Associated With Long Tag
- 22 Write Long Tag
- 38 Reset Configuration Changed Flag
- 48 Read Additional Device Status

- **Common practice commands**

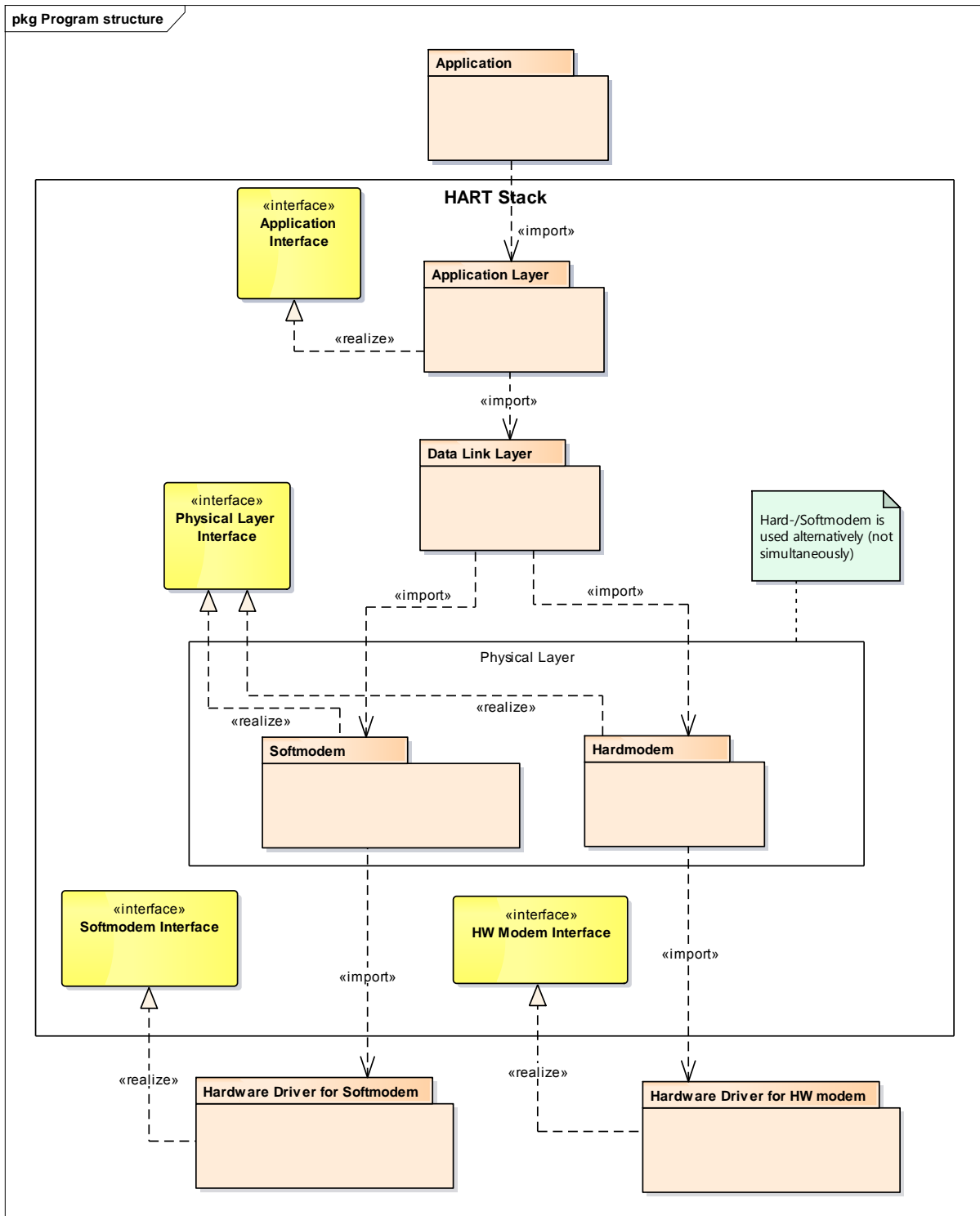
- 33 Read Device Variables
- 34 Write Primary Variable Damping Value
- 35 Write Primary Variable Range Values
- 36 Set Primary Variable Upper Range Value
- 37 Set Primary Variable Lower Range Value
- 40 Enter/Exit Fixed Current Mode
- 41 Perform Self Test
- 42 Perform Device Reset
- 43 Set Primary Variable Zero
- 44 Write Primary Variable Units
- 45 Trim Loop Current Zero
- 46 Trim Loop Current Gain
- 47 Write Primary Variable Transfer Function
- 49 Write Primary Variable Transducer Serial Number
- 50 Read Dynamic Variable Assignments



- 51 Write Dynamic Variable Assignments
- 52 Set Device Variable Zero
- 53 Write Device Variable Units
- 54 Read Device Variable Information
- 55 Write Device Variable Damping Value
- 56 Write Device Variable Transducer Serial No.
- 59 Write Number Of Response Preambles
- 71 Lock Device
- 72 Squawk
- 73 Find Device
- 76 Read Lock Device State
- 78 Read Aggregated Commands
- 79 Write Device Variable
- 80 Read Device Variable Trim Points
- 81 Read Device Variable Trim Guidelines
- 82 Write Device Variable Trim Points
- 83 Reset Device Variable Trim
- 89 Set Real-Time Clock
- 90 Read Real-Time Clock
- 95 Read Device Communications Statistics
- 100 Write Primary Variable Alarm Code
- 103 Write Burst Period
- 104 Write Burst Trigger
- 105 Read Burst Mode Configuration
- 106 Flush Delayed Responses
- 107 Write Burst Device Variables
- 108 Write Burst Mode Command Number
- 109 Burst Mode Control
- 512 Read Country Code
- 513 Write Country Code
- 516 Read Device Location
- 517 Write Device Location
- 518 Read Location Description
- 519 Write Location Description
- 520 Read Process Unit Tag
- 521 Write Process Unit Tag
- 523 Read Condensed Status Mapping Array
- 524 Write Condensed Status Mapping
- 525 Reset Condensed Status Map
- 526 Write Status Simulation Mode
- 527 Simulate Status Bit

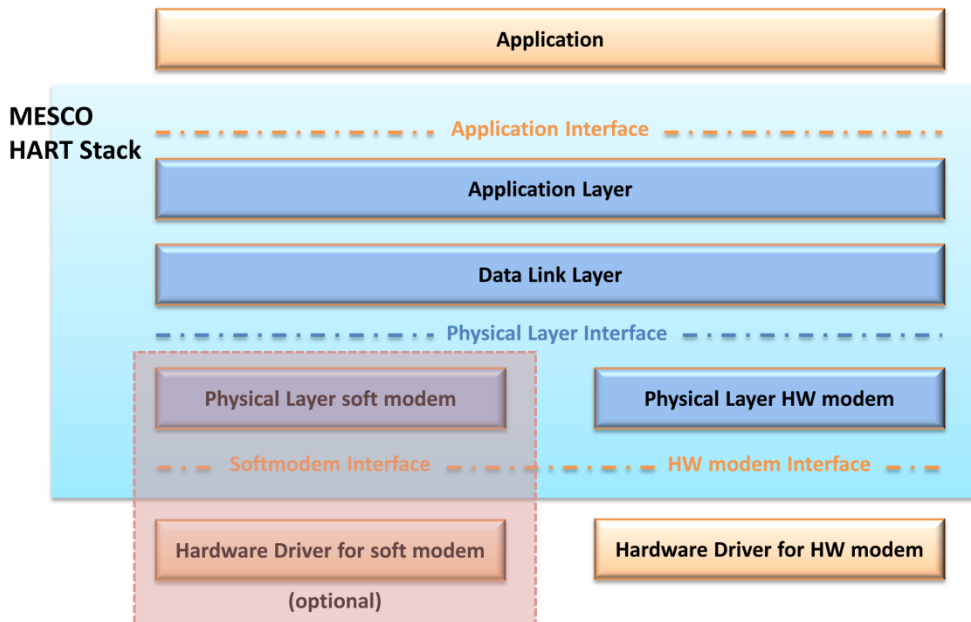


## 5. Stack interfaces / API - Overview





## 6. Stack Architecture



**Figure 1: Stack layer model**

The layers correspond to the layers of the OSI model:

- Application Layer (OSI layer 7)
- Data Link Layer (OSI layer 2)
- Physical Layer (OSI layer 1)

The Application is the system software that integrates the HART Slave Stack. This term is used throughout this document.

The Application Layer is the stack's top layer. The Application Layer implements control routines as well as the stack's command handlers and related high-level functionality, whereas the Data Link Layer is the connection between Application Layer and the Physical Layer. The generic part of MESCO HART Slave Stack consists of the Application Layer and the Data Link Layer.

The Physical Layer is responsible for writing the HART messages to the physical hardware and reading response messages from it. The HART standard defines various Physical Layers. MESCO provides implementation only for the FSK physical layer. The following two options are available:

- The HW modem variant works with an external FSK modem chip. If this option is used, the HW modem interface has to be served by the user.





- The soft modem variant implements a FSK modulator and demodulator in software using an ADC and a DAC. If this variant is used the soft modem interface has to be served by the user.

The hardware drivers are hardware specific and not part of the generic stack. MESCO provides support and engineering services for porting the existing sample-drivers to new hardware platforms.