

Comandos HART suportados pelos modelos SMAR LD301, FY301 e TT301

Comandos Universais (0x00 a 0x13)

Estes comandos **devem ser suportados** por todos dispositivos HART. A seguir, cada comando é listado com os bytes de dados (**body**) de sua resposta ou requisição, conforme o caso:

- **Comando "00" – Read Unique Identifier (Identificação do Dispositivo)** -> **body** = FE + **Manufacturer ID** (master/slave bits + código fabricante) + **Device Type** + **Required Preambles** + **HART Revision** + **Device Revision** + **Software Revision** + **Hardware Revision** + **Device Flags** + **Device ID**. Em outras palavras, o dispositivo retorna 1 byte fixo 0xFE, seguido pelo código do fabricante com bits de identificação de mestre/escravo, código do tipo de dispositivo, número de preâmbulos requerido, revisões (do protocolo HART, do dispositivo/transmissor, do software e do hardware), flags de status do dispositivo (normalmente 0x00) e o **ID único de 3 bytes** do dispositivo.
- **Comando "01" – Read Primary Variable (PV)** -> **body** = **PV Unit Code** (1 byte) + **PV Value** (4 bytes float). Retorna o valor da variável primária medido pelo transmissor e o código da unidade de medida conforme a tabela HART. *Exemplo:* para um transmissor de temperatura, a unidade pode ser o código de °C e os próximos 4 bytes representam a temperatura medida em formato ponto flutuante IEEE-754 de 32 bits.
- **Comando "02" – Read Current and Percent of Range** -> **body** = **Loop Current** (4 bytes float, em mA) + **Percentage of Range** (4 bytes float, %). O dispositivo retorna a corrente de saída em mA e a porcentagem do span configurado correspondente a essa corrente.
- **Comando "03" – Read Dynamic Variables and PV Current** -> **body** = **Loop Current** (4 bytes float) + **PV Unit** (1 byte) + **PV Value** (4 bytes) + **SV Unit** (1 byte) + **SV Value** (4 bytes) + **TV Unit** (1 byte) + **TV Value** (4 bytes) + **QV Unit** (1 byte) + **QV Value** (4 bytes). Este comando retorna a corrente de malha e até quatro variáveis dinâmicas (Primária, Secundária, Terciária, Quaternária) com seus valores e códigos de unidade. *Nota:* Em dispositivos HART5, é comum somente PV e corrente estarem disponíveis; variáveis adicionais podem retornar valores inválidos (NaN) se não se aplicam. Cada variável (exceto a corrente) é precedida por um byte indicando sua unidade de medida.
- **Comando "04"** – (Não utilizado na especificação padrão HART). – **Sem dados.** (Os códigos 0x04 e 0x05 não correspondem a comandos universais definidos e não são suportados nos dispositivos; são reservados.)
- **Comando "05"** – (Não utilizado na especificação padrão HART). – **Sem dados.**
- **Comando "06" – Write Polling Address** -> **body (requisição)** = **Polling Address** (1 byte). Define o endereço de polling (endereço curto) do dispositivo. A resposta ecoa o endereço atribuído.

- **Comando "07" – Read Loop Configuration** -> **body** = **Polling Address** (1 byte) + **Current Mode Flag** (1 byte). Retorna o endereço de polling atualmente configurado e indica se a saída de corrente 4–20mA está habilitada (1) ou travada em 4mA (0, modo multidrop).
- **Comando "08" – Read Dynamic Variable Class** -> **body** = **Variable Class Code** (1 byte). Retorna um código que identifica a classificação da variável dinâmica principal do dispositivo (por exemplo, 0x41 para pressão em um transmissor de pressão). No caso do TT301 (temperatura), esse código corresponde a *temperature* (temperatura).
- **Comando "09" – Read Device Variables and Status (HART6/7)** -> **body** = **Extended Device Status** (1 byte) + até 4 conjuntos de {**Device Variable Code** (1 byte) + **Classification** (1 byte) + **Unit** (1 byte) + **Value** (4 bytes float) + **Status** (1 byte)}. Esse comando, introduzido em HART6+, permite ler até quatro variáveis do dispositivo de uma só vez, cada qual acompanhada de seu código de variável, classificação, unidade, valor e status. *Nota:* O TT301 é HART5, portanto normalmente não suporta o comando 9 – em seu lugar utiliza-se o comando 33 para ler variáveis múltiplas.
- **Comando "0A" (10) – (Não utilizado na especificação padrão HART.) – Sem dados.**
- **Comando "0B" (11) – Read Unique ID by Tag** -> **body (requisição)** = **Tag** (6 bytes, ASCII compactado). A resposta deste comando é igual à do comando 0 (Unique ID), retornando os mesmos campos de identificação do dispositivo, **desde que** o tag enviado na requisição coincida com o tag do dispositivo.
- **Comando "0C" (12) – Read Message** -> **body** = **Message** (24 bytes, texto ASCII compactado em 6 bits cada caractere). Retorna a mensagem de 32 caracteres configurada no transmissor (por exemplo, via comando 17), codificada em formato packed ASCII (24 bytes).
- **Comando "0D" (13) – Read Tag, Descriptor, Date** -> **body** = **Tag** (6 bytes, ASCII 6-bit) + **Descriptor** (12 bytes, ASCII 6-bit) + **Date** (3 bytes). Retorna o tag curto (até 8 caracteres, em formato compactado de 6 bits – ocupando 6 bytes) do transmissor, o descriptor (descrição, 16 caracteres compactados em 12 bytes) e a data de atualização (3 bytes: ano, mês, dia).
- **Comando "0E" (14) – Read Primary Variable Sensor Information** -> **body** = **Sensor Serial Number** (3 bytes) + **Sensor Units Code** (1 byte) + **Sensor Upper Range Limit** (4 bytes float) + **Sensor Lower Range Limit** (4 bytes float) + **Sensor Minimum Span** (4 bytes float). Retorna informações sobre o elemento sensor conectado ao transmissor, incluindo o número de série do sensor, a unidade de engenharia dos limites, e os valores de limite superior, limite inferior e span mínimo do sensor primário.
- **Comando "0F" (15) – Read Primary Variable Output Information (Device Information)** -> **body** = **PV Alarm Selection Code** (1 byte) + **PV Transfer Function** (1 byte) + **PV Units** (1 byte) + **PV Upper Range Value** (4 bytes float) + **PV Lower Range Value** (4 bytes float) + **PV Damping Time** (4 bytes float, seg) + **Write Protect Status** (1 byte: 0 = sem proteção de escrita, 1 = protegido) + **Manufacturer ID** (1 byte) + **Analog Channel Flag** (1 byte). Este comando retorna diversos dados de configuração do transmissor: o tipo de alarme de corrente (ex. upscale/downscale), a função de transferência do PV (linear ou raiz quadrada), a unidade, os valores de fundo de escala (URV) e zero (LRV) do PV, o valor de damping (atraso) aplicado, se o dispositivo está com proteção de escrita ativa, o código do fabricante e um flag indicando se há sinal analógico (ADC) conectado ao canal do PV.

- **Comando "10" (16) – Read Final Assembly Number** -> **body = Assembly Number** (3 bytes). Retorna o número de montagem/fabricação do conjunto eletrônico do transmissor (número que identifica o hardware/eletrônica do dispositivo).
- **Comando "11" (17) – Write Message** -> **body (requisição) = Message** (24 bytes, texto ASCII compactado). Escreve uma mensagem de até 32 caracteres no dispositivo (armazenada para consulta via comando 12). A resposta normal ecoa os 24 bytes escritos (igual aos dados enviados).
- **Comando "12" (18) – Write Tag, Descriptor, Date** -> **body (requisição) = Tag** (6 bytes) + **Descriptor** (12 bytes) + **Date** (3 bytes). Permite configurar o tag (6 bytes ASCII compactado), descriptor (12 bytes ASCII compactado) e data (3 bytes – normalmente codificados como dia, mês e ano) no transmissor. A resposta normal ecoa os dados escritos (mesma estrutura enviada).
- **Comando "13" (19) – Write Final Assembly Number** -> **body (requisição) = Final Assembly Number** (3 bytes). Grava/atualiza o número de montagem final do dispositivo (normalmente não alterado em campo). A resposta ecoa os 3 bytes enviados.

Comandos Common Practice Selecionados (0x19, 0x21, 0x26, 0x28, 0x29, 0x2A, 0x2D, 0x2E, 0x50)

Os comandos a seguir fazem parte do conjunto *Common Practice* do HART (podem ou não ser suportados dependendo do fabricante/modelo). Nos dispositivos SMAR TT301/LD301/FY301, os seguintes comandos são suportados para funcionalidades adicionais:

- **Comando "19" (25) – (Desconhecido/Reservado)**. *Não foi encontrada documentação indicando o uso do comando 0x19 nesses dispositivos.* (Esse código está fora dos universais padrão; possivelmente não é implementado pelo TT301 ou é reservado sem função definida.)
- **Comando "21" (33 decimal) – Read Device Variables (Leitura de Variáveis do Dispositivo)** -> **body** = sequência de {**Unit Code** + **Variable Value**} para cada variável solicitada. Esse comando permite ler múltiplas variáveis do dispositivo em uma única resposta. A mensagem de requisição deve conter a quantidade de variáveis e os códigos de índice das variáveis desejadas (até 5 variáveis podem ser lidas de uma vez em HART5). A resposta traz, para cada variável solicitada, o código de unidade seguido do valor em ponto flutuante de 32 bits. *Exemplo:* O TT301, que é HART5, utiliza o comando 33 para ler variáveis como PV, temperatura ambiente, etc., retornando cada valor precedido por seu código de unidade. (No HART6/7, o equivalente seria o comando 9, mas no HART5 o comando 33 realiza essa função.)
- **Comando "26" (38) – Reset Configuration Changed Flag (Reiniciar Flag de Configuração Alterada)** -> **body** = (*nenhum dado*). Este comando informa ao dispositivo para limpar/resetar o indicador de “configuração modificada” em seu status. Basicamente, ele reseta o bit 6 do primeiro byte de status do dispositivo, que é setado quando alguma configuração foi alterada. A resposta não contém dados além dos bytes de status indicando sucesso.
- **Comando "28" (40) – Enter/Exit Fixed Current Mode (Modo de Corrente Fixa)** -> **body (requisição) = Fixed Current Value** (4 bytes float, em mA). Coloca o transmissor em modo de corrente fixa (quando um valor não-zero é enviado) ou retorna ao modo normal de controle de corrente (quando o valor 0.0 mA é enviado). O dispositivo irá fixar a saída de corrente no valor

especificado (dentro da faixa 4–20 mA) até receber comando para sair. A resposta **body** ecoa o **Actual Fixed Current Level** alcançado (4 bytes float) – geralmente igual ao valor solicitado, limitado aos limites de 4–20 mA.

- **Comando "29" (41) – Perform Self-Test (Executar Autoteste)** -> **body** = *(nenhum dado)*. Faz o dispositivo executar um autoteste interno. Não há dados na requisição; o dispositivo normalmente responde apenas com os bytes de status (e possivelmente entra em modo de teste). Este comando aciona rotinas de diagnóstico no transmissor (verificação de memória, sensores, etc.). A resposta não contém dados além do status (que indicará sucesso ou erro do autoteste).
- **Comando "2A" (42) – Perform Device Reset (Master Reset)** -> **body** = *(nenhum dado)*. Reinicia completamente o dispositivo, como se fosse desligado e ligado novamente. **Atenção:** após enviar este comando, o dispositivo HART reinicializa e não responderá a comandos durante alguns segundos até completar o boot. Não há dados de resposta além do acknowledgment padrão (que pode nem chegar devido ao reset). Este é um comando Common Practice padronizado.
- **Comando "2D" (45) – Trim DAC Zero (Calibrar 4 mA)** -> **body (requisição)** = **Measured Loop Current at 4 mA** (4 bytes float, em mA). Este comando ajusta o ponto de 4 mA do circuito de corrente do transmissor. O usuário deve medir externamente a corrente real quando o transmissor está solicitando 4 mA e fornecer esse valor medido na requisição. O dispositivo então calibra seu DAC de saída para corrigir qualquer desvio. **Resposta: Actual Loop Current** (4 bytes float) medido internamente após o ajuste – tipicamente próximo de 4.000 mA, indicando o resultado após o trim.
- **Comando "2E" (46) – Trim DAC Gain (Calibrar 20 mA)** -> **body (requisição)** = **Measured Loop Current at 20 mA** (4 bytes float). Semelhante ao comando 45, porém ajusta o ponto de 20 mA (ganho) da saída de corrente. Deve-se aplicar uma condição de saída 20 mA, medir a corrente real e fornecer o valor medido. **Resposta: Actual Loop Current** (4 bytes float) após o ajuste de ganho – esperado próximo de 20.000 mA.
- **Comando "50" (0x50 hex, 80 dec) – Read Dynamic Variable Assignments** -> **body** = **PV Variable Code** (1 byte) + **SV Code** (1 byte) + **TV Code** (1 byte) + **QV Code** (1 byte). Este comando (introduzido em revisões mais recentes do HART) retorna quais variáveis do dispositivo estão mapeadas para PV, SV, TV e Quarta Variável. Cada byte é um código de variável do dispositivo indicando, por exemplo, que variável interna foi atribuída como PV, qual como SV, etc. Se alguma posição não for usada pelo dispositivo, o código 250 (0xFA) pode ser retornado nessa posição indicando “não disponível”. *Exemplo:* Em um dispositivo que só tem PV, os campos de SV, TV, QV podem vir com 250 (indicando ausência dessas variáveis). *(Observação: O TT301 sendo HART5 provavelmente não suporta esse comando, mas foi listado para completude, pois aparece em ferramentas modernas de configuração.)*

Comandos Específicos do Fabricante (Device-Specific)

Os comandos a seguir estão na faixa 0x80–0xFF e são **específicos da SMAR** para funcionalidades particulares dos modelos LD301/FY301/TT301. Esses comandos normalmente são detalhados nas DDs (Device Descriptions) ou manuais do fabricante, e permitem acesso a configurações avançadas, calibrações de sensor, e recursos internos do dispositivo. *Não encontramos documentação pública*

detalhando o formato exato de cada um destes comandos; o entendimento usual é baseado na funcionalidade implícita de cada código:

- **Comando "82" (0x82 hex / 130 dec)** – (*Smar Specific Function*). Usado para alguma leitura/escrita específica do transmissor (por exemplo, pode ler configurações estendidas do sensor ou variáveis internas – não documentado publicamente).
- **Comando "84" (0x84 / 132 dec)** – (*Smar Specific Function*). Similar ao acima: função proprietária da SMAR (possivelmente relacionada a configuração de sensor secundário, não há detalhes disponíveis).
- **Comando "87" (0x87 / 135 dec)** – (*Smar Specific Function*). Comando específico – possivelmente ligado aos recursos de controle PID no TT301 (por exemplo, poderia ler ou ajustar parâmetros do controlador PID interno, como setpoint ou ganho – especulativo, já que não há referência oficial).
- **Comando "88" (0x88 / 136 dec)** – (*Smar Specific Function*). Comando específico do fabricante (função exata não documentada abertamente).
- **Comando "8A" (0x8A / 138 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada publicamente).
- **Comando "8C" (0x8C / 140 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada publicamente).
- **Comando "98" (0x98 / 152 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "A2" (0xA2 / 162 dec)** – (*Smar Specific Function*). Possivelmente relacionado a calibração ou ajuste de sensor (dedução, sem documentação).
- **Comando "A4" (0xA4 / 164 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "A6" (0xA6 / 166 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "A8" (0xA8 / 168 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "AD" (0xAD / 173 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "B9" (0xB9 / 185 dec)** – (*Smar Specific Function*). Função específica SMAR – *nota:* pelo contexto de outros dispositivos, comandos próximos a esta faixa podem estar relacionados a travar/destravar configuração (write lock) ou operações de manutenção. Sem documentação confirmada.
- **Comando "BB" (0xBB / 187 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).

- **Comando "C6" (0xC6 / 198 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada).
- **Comando "DF" (0xDF / 223 dec)** – (*Smar Specific Function*). Função específica SMAR (não documentada). Por estar próximo do valor máximo (0xFF), pode estar relacionado a alguma operação especial ou modo de fábrica dos dispositivos SMAR.

Observação Final: Os comandos *device-specific* acima são implementações particulares da SMAR – a descrição exata de cada um reside nos arquivos de descrição do dispositivo (DD/EDD) da SMAR. Em geral, eles contemplam funcionalidades como seleção de tipo de sensor (termopar, RTD, etc.), leituras de valores internos (temperatura ambiente, tensão de sensor), ajuste de parâmetros do controlador PID interno (no caso do TT301 com função de controle), travamento de escrita, execução de diagnósticos específicos, etc. Como essas informações não estão disponíveis em catálogos abertos, a implementação exata deve ser obtida junto à documentação técnica da SMAR ou analisando a comunicação via ferramentas como Pactware/Wireshark, que parece ter sido a abordagem do usuário na identificação destes comandos.

Referências: As definições dos comandos universais e common practice foram baseadas nas especificações do protocolo HART e em manuais técnicos. Detalhes adicionais e exemplos de resposta foram obtidos de documentação da HART Communication Foundation e de análises de comunicação HART. Como esperado, os comandos específicos do fabricante não são detalhados em fontes públicas, sendo necessário consultar a SMAR para sua descrição completa.
