AN11E Application Note Transducer Electronic Data Sheet (TEDS) to IEEE 1451.4 Basics of Data Storage

1. Introduction

Transducer Electronic Data Sheet (TEDS) is a standardized method of storing sensor identification, calibration, correction data, and manufacturer-related information. TEDS formats are defined in the IEEE 1451 series of standards describing smart transducers as a set of open, common, network-independent communication interfaces for connecting transducers to microprocessors, instrumentation systems, and control/field networks.

One of the key elements of the IEEE 1451 standards is the definition of TEDS for a transducer. TEDS in its most useful form is implemented as a memory device, typically EEPROM, as part of the transducer. It contains the information needed by a measurement instrument or control system to interface with a transducer. Other forms, like a “virtual TEDS” are not discussed here.

Most TEDS applications use a 1-Wire® EEPROM memory device which is manufactured by Maxim Integrated. As the name suggests the 1-Wire protocol allows read and write communication via a single wire and a ground terminal. This feature makes it particularly suitable for the integration into the analog signal conditioning electronics of IEPE transducers. Part 4 of IEEE 1451 deals with this particular subject.

Communication with the TEDS memory device inside the sensor is performed via the single-ended signal cable. The memory device is supplied by a logic voltage between 3 and 5 V with negative polarity referred to ground. This negative voltage is modulated with the standardized 1-Wire communication protocol. To separate the positive IEPE sensor output from the negative TEDS data two diodes are provided. Figure 1 shows the principle.

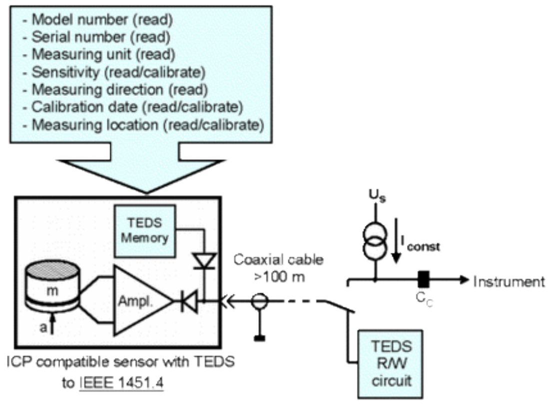

Figure 1: Basic concept of TEDS in an IEPE sensor

The purpose of this document is to provide you with the necessary information how TEDS data is stored inside the sensor and how it can be decoded on the instrument side. Due to the small size of the used memory device the data needs to be stored in a packed format.

This document shows application examples with the memory device **DS2430A** which is obsolete but commonly used in legacy TEDS sensors and the **DS2431** that is under production and currently used for Metra sensors.

2. Basic TEDS

Basic TEDS is a 64 bit portion of the memory containing the most essential information to identify the sensor. It includes the following data:

- Manufacturer code
- Type (coded or clear text)
- Version number / letter
- Serial number

It is obvious that this data needs to be programmed only once in the manufacturing process and should not be manipulated later.

The following table shows how basic TEDS data is coded.

Entry	No. of bits	Data type	Meaning	Example
Man_ID	14	binary	manufacturer code (17 to16381) *	3Dh (61)
Model	15	binary	type number (0 to 32767) **	46h (70)
Ver_Let	5	Chr5	version letter (0 = space, 1 = A, 2 = B... , . / _ @)	1h (A)
Ver_No	6	binary	version number (0 to 63)	2h (2)
Ser_No	24	binary	serial number (0 to 16777215)	202h (0514)

Manfred Weber
Metra Mess- und Frequenztechnik in Radebeul e.K.
Meißner Str. 58 P.O. Box 01 01 13
D-01445 Radebeul D-01435 Radebeul
Tel. +49-351-836 21 91 Fax +49-351-836 29 40
Internet: www.MMF.de
Email: Info@MMF.de

* Manufacturer codes are issued by IEEE. Metra uses code 61. The public listing file including all codes can be downloaded from the IEEE web site. At the last editing of this document the list was found at http://standards-oui.ieee.org/manid/manid.txt

** A reference to the actual type name can be provided by the manufacturer as a text file **.xdl* This can be useful if the type name and type numbering according to the TEDS scheme do not match. Metra uses this possibility and provides the name file under this link: https://mmf.de/download/Model003D.xdl

3. IEEE Templates

Templates describe how the actual sensor data is arranged in the memory. There are templates for different sensor types.
IEEE 1451.4 defines a "Template Description Language" (TDL). The industry has adopted a number of standard templates.
Data in the template memory section can be overwritten. This will typically be the task of calibration labs. In addition there are some bytes reserved for the description of measuring points which may also be written by the user.

Most relevant for sensors with IEPE interface are the following templates:

- Template 25 for accelerometers and force transducers
- Template 27 for microphones with built-in pre-amplifier

IEEE Template 25 for Accelerometers and Force Transducers

We will describe this template here in detail because it is applied in Metra products.
Template 25 contains data relevant for accelerometers like sensitivity, high pass frequency, resonance frequency and quality, sensor weight, axis orientation, calibration frequency and date. A template can have several switch bits for different sub-versions.
Metra uses the template 25 without transfer function as default TEDS.
In Template Description Language Template 25 is specified as follows:

```
TEMPLATE 0,8,25,"Accelerometer and Force Transducer"
  //The first 0 in the Template field indicates IEEE defined template
  //The 8 is the number of bits to read from the sensor to get the template ID
  //the 25 is the decimal value of this template ID.
TDL_VERSION_NUMBER 2  //Version 2 refers to the final IEEE 1451.4 version 1.0 TDL specification
ABSTRACT IEEE 1451.4 Default Accelerometer and Force Transducer Template
SPACING

//Physical Base Units: (ratio, radian, steradian, meter, kg, sec, Amp, kelvin, mole, candela, scal-
ing, offset)
PHYSICAL_UNIT "V/(m/s²)",   (0,0,0,1,1,-1,-1,0,0,0,1,0) // Volts per Acceleration: V/(m/s²) = m*kg/(s*A)
PHYSICAL_UNIT "V/N",        (0,0,0,1,0,-1,-1,0,0,0,1,0) // Volts per Newton: V/(mokg/s²) = m/(s*A)
PHYSICAL_UNIT "Hz",         (0,0,0,0,0,-1,0,0,0,0,1,0)          // Frequency: Hertz = 1/second
PHYSICAL_UNIT "degrees",    (0,1,0,0,0,0,0,0,0,0,0.0174533,0)    // degrees = 0.0174533 radians
PHYSICAL_UNIT "°C",         (0,0,0,0,0,0,0,1,0,0,1,-273.15)       // Celsius is (kelvin - 273.15 K)
PHYSICAL_UNIT "%/decade",   (6,0,0,0,0,-1,0,0,0,0,0.01,0)  // Flatness over frequency: 0.01 / (log10(Hz/Hz))
PHYSICAL_UNIT "%/°C",       (0,0,0,0,0,0,0,-1,0,0,0.01,0)  // Temperature drift: %/°C = 0.01 x (1/Kelvin)
PHYSICAL_UNIT "N/m",        (0,0,0,0,1,-2, 0,0,0,0,1,0)    // Newton/m equals (mokg/s²)/m = kg/s²
PHYSICAL_UNIT "g",          (0, 0,0, 0,1, 0, 0,0,0,0,0.001,0)     // gram = 0.001 kg
PHYSICAL_UNIT "days",       (0,0,0,0,0,1,0,0,0,0,86400,0)  // Time: Days = 86400 seconds

SELECTCASE "Transducer Type", ID, 1

  CASE "Accelerometer", 0
    SELECTCASE "Extended Functionality (Programmable Sensitivity)", ID, 1
      CASE "No Extended Functionality", 0
         UGID "I25-0-0-0", "Accelerometer"
        %Sens@Ref, "Sensitivity @ reference condition", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/m/s²)"
        %TF_HP_S, "High pass cut-off frequency (F hp)", CAL, 8, ConRelRes, 0.005, 0.03, "rp", "Hz"
      ENDCASE
      CASE "Extended Functionality (Programmable Sensitivity)", 1
         UGID "I25-0-1-0", "Accelerometer, programmable Sensitivity"
        %passive[Initialize], "Initialize not needed", ID, 1, UNINT, "", "" = 0
        %passive[CtrlFunctionMask], "Control Function Mask",ID, 4, BitBin, "", "" = "11"
        %passive[ReadWrite], "Write only", ID, 2, UNINT, "", "" = 3
        %passive[FunctionType], "Passive control type", ID, 2, UNINT, "","" = 0//checkmark
        %passive[Function], "Passive mode", USR, 10, BitBin, "","" =  "xx,00"
        %sens[Initialize], "Initialize not needed", ID, 1, UNINT, "", "" = 0
        %sens[CtrlFunctionMask], "Control Function Mask",ID, 4, BitBin, "", "" = "11"
        %sens[ReadWrite], "Write only", ID, 2, UNINT, "", "" = 3
        %sens[FunctionType], "Sensitivity control type", ID, 2, UNINT, "", "" = 1//One Exactly
        %sens[Function], %Sens@Ref["10"], USR, 4, BitBin, "", "" = "10" //High sensitivity
```

```
        %sens[Function], %Sens@Ref["01"], USR, 4, BitBin, "", "" = "01" //Low sensitivity
        %defaultFR, "Default setting", ID, 2, UNINT, "", ""
        %Passive, "Supports multiplexer mode", ID, 1, UNINT, "", ""
        %Sens@Ref["01"], "Low sensitivity @ Fref", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/(m/s²)"
        %Sens@Ref["10"], "High sensitivity @ Fref", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/(m/s²)"
        %TF_HP_S["01"], "Low sensitivity high pass cut-off frequency", CAL, 8, ConRelRes, 0.005,
0.03, "rp", "Hz"
        %TF_HP_S["10"], "High sensitivity high pass cut-off frequency", CAL, 8, ConRelRes, 0.005,
0.03, "rp", "Hz"
      ENDCASE
    ENDSELECT
  ENDCASE

  CASE "Force Transducer", 1
    SELECTCASE "Extended Functionality (Programmable sensitivity)", ID, 1
      CASE "No Extended Functionality", 0
        UGID "I25-1-0-0", "Force Transducer"
        %Sens@Ref, "Sensitivity @ reference condition", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/N"
        %TF_HP_S, "High pass cut-off frequency (F hp)", CAL, 8, ConRelRes, 0.005, 0.03, "rp", "Hz"
        %Stiffness, "Stiffness of transducer", CAL, 6, ConRelRes, 1E6, 0.10, "rp", "N/m"
        %Mass_below, "Mass below gage", CAL, 6, ConRelRes, 0.1, 0.1, "rp", "g"
      ENDCASE
      CASE "Extended Functionality (Programmable sensitivity)", 1
        UGID "I25-1-1-0", "Force Transducer, programmable sensitivity"
        %passive[Initialize], "Initialize not needed", ID, 1, UNINT, "", "" = 0
        %passive[CtrlFunctionMask], "Control Function Mask",ID, 4, BitBin, "", "" = "11"
        %passive[ReadWrite], "Write only", ID, 2, UNINT, "", "" = 3
        %passive[FunctionType], "Passive control type", ID, 2, UNINT, "","" = 0//checkmark
        %passive[Function], "Passive mode", USR, 10, BitBin, "","" =  "xx,00"

        %sens[Initialize], "Initialize not needed", ID, 1, UNINT, "", "" = 0
        %sens[CtrlFunctionMask], "Control Function Mask",ID, 4, BitBin, "", "" = "11"
        %sens[ReadWrite], "Write only", ID, 2, UNINT, "", "" = 3
        %sens[FunctionType], "Sensitivity control type", ID, 2, UNINT, "", "" = 1//One Exactly
        %sens[Function], %Sens@Ref["10"], USR, 4, BitBin, "", "" = "10" //High sensitivity
        %sens[Function], %Sens@Ref["01"], USR, 4, BitBin, "", "" = "01" //Low sensitivity

        %defaultFR, "Default setting", ID, 2, UNINT, "",""
        %Passive, "Supports multiplexer mode", ID, 1, UNINT, "", ""

        %Sens@Ref["01"], "Low sensitivity @ Fref", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/N"
        %Sens@Ref["10"], "High sensitivity @ Fref", CAL, 16, ConRelRes, 5E-7, 0.00015, "rp", "V/N"
        %TF_HP_S["01"], "Low sensitivity high pass cut-off frequency", CAL, 8, ConRelRes, 0.005,
0.03, "rp", "Hz"
        %TF_HP_S["10"], "High sensitivity high pass cut-off frequency", CAL, 8, ConRelRes, 0.005,
0.03, "rp", "Hz"
        %Stiffness, "Stiffness of transducer", CAL, 6, ConRelRes, 1E6, 0.10, "rp", "N/m"
        %Mass_below, "Mass below gage", CAL, 6, ConRelRes, 0.1, 0.1, "rp", "g"
        %PhaseCorrection, "Phase correction @ reference condition", CAL, 6, CONRES, -3.2, 0.1, "rp",
"degrees"
      ENDCASE
    ENDSELECT
  ENDCASE

ENDSELECT

ENUMERATE DirectionEnum,"x","y","z"
%Direction, "Sensitivity direction (x,y,z)",  CAL, 2, DirectionEnum, "e", ""
%Weight, "Transducer weight", CAL, 6, CONRELRES, 0.1, 0.1, "rp", "g"

ENUMERATE ElecSigTypeEnum,"Voltage Sensor","Current Sensor","Resistance Sensor","Bridge
Sensor","LVDT Sensor","Potentiometric Voltage Divider Sensor","Pulse Sensor","Voltage
Actuator","Current Actuator","Pulse Actuator"
%ElecSigType, "Transducer Electrical Signal Type", ID, 0, ElecSigTypeEnum,"e","" = "Voltage Sensor"

ENUMERATE MapMethEnum, "Linear","Inverse m/(x+b)","Inverse (b+m/x)", "Inverse 1/(b+m/x)","Thermocou-
ple","Thermistor","RTD","Bridge"
%MapMeth, "Mapping Method", ID, 0, MapMethEnum,"e","" = "Linear"

ENUMERATE ACDCCouplingEnum,"DC","AC"
%ACDCCoupling, "AC or DC Coupling", ID, 0, ACDCCouplingEnum,"e","" = "AC"
```

```
ENUMERATE SignEnum,"Positive","Negative"
%Sign, "Polarity (Sign)", CAL, 1, SignEnum, "e", ""

SELECTCASE "Transfer Function", ID, 1
  CASE "No Transfer Function Specified", 0
  ENDCASE
  CASE "Transfer Function Specified", 1
    %TF_SP, "Low pass cut-off frequency (F lp)", CAL, 7, ConRelRes, 10, 0.05, "rp", "Hz"
    %TF_KPr, "Resonance frequency (F res)", CAL, 9, ConRelRes, 100, 0.01, "rp", "Hz"
    %TF_KPq, "Quality factor @ F res (Q)", CAL, 9, ConRelRes, 0.4, 0.01,  "rp",""
    %TF_SL, "Amplitude slope (a)", CAL, 7, ConRes, -6.3, 0.1, "0.0", "%/decade"
    %TempCoef, "Temperature coefficient (b)", CAL, 6, ConRes, -0.8, 0.025,"0.000","%/°C"
  ENDCASE
ENDSELECT

%Reffreq, "Reference frequency (F ref)", CAL, 8, ConRelRes, 0.35, 0.0175, "0p","Hz"
%RefTemp, "Reference temperature (T ref)", CAL, 5, ConRes, 15, 0.5, "0.0","°C"
%CalDate, "Calibration Date", CAL, 16, DATE,"d-mmm-yyyy",""
%CalInitials, "Calibration Initials", CAL, 15, CHR5,"s",""
%CalPeriod, "Calibration Period (Days)", CAL, 12, UNINT, "0", "days"

%MeasID, "Measurement location ID", USR, 11, UNINT,"0", ""

ENDTEMPLATE
```

The table below shows how the individual parameters are coded. TEDS uses different data types to pack the data as dense as possible. Several switch bits can be used to modify the template. Metra disables, for example, the transfer function by default.

Entry	Bit no.	Type	Meaning	Example
Chksum	8	binary	check sum, two's complement of the sum of all bytes including the basic TEDS, except the byte of the checksum itself. The sum of all bytes, including the checksum, is zero.	4Fh
Sel_Desc	2	binary	selector of descriptor bits; 0 = TEDS to IEEE 1451.4	0h
Templ_No	8	binary	template no.; 8 Bits	19h (No.25)
Case_a/F	1	binary	case bit sensor type: 0 = accelerometer; 1 = force transducer	0h
Case_Gain	1	binary	case bit gain: 0 = fixed gain; 1 = programmable gain	0h
Sens	16	ConRelRes	sensitivity B_{ua} in V/ms^{-2} $B_{ua}=5\cdot10^{-7}\cdot1{,}0003^{Sens}$ $Sens=\frac{1}{\log(1{,}0003)}\cdot\log\left(\frac{B_{ua}}{5\cdot10^{-7}}\right)=7676{,}4\cdot\log\left(\frac{B_{ua}}{5\cdot10^{-7}}\right)$	6752h (B_{ua} = 1,395 mV/ms^{-2})
HP*	8	ConRelRes	high pass frequency in Hz $f_{hp}=5\cdot10^{-3}\cdot1{,}06^{HP}$ $HP=\frac{1}{\log(1{,}06)}\cdot\log\left(\frac{f_{hp}}{5\cdot10^{-3}}\right)=39{,}5\cdot\log\left(\frac{f_{hp}}{0{,}005}\right)$	46h (f_{hp} = 0,295 Hz)
Dir	2	Enumeration	0 = X; 1 = Y; 2 = Z; 3 = not specified	3h (not specified)
Weight*	6	ConRelRes	weight of sensor in grams $m=0{,}1\cdot1{,}2^{Weight}$ $Weight=\frac{1}{\log(1{,}2)}\cdot\log\left(\frac{m}{0{,}1}\right)=12{,}6\cdot\log\left(\frac{m}{0{,}1}\right)$	20h (m = 34 g)
Polarity	1	Enumeration	0 = positive; 1 = negative	0h (positive)

Entry	Bit no.	Type	Meaning	Example
CaseTransf	1	binary	case bit for transfer function: 1 = specified; 0 = none	1h
LP*	7	ConRelRes	low pass frequency in Hz $f_{lp}=10\cdot 1{,}1^{LP}$ $LP=\frac{1}{\log(1{,}1)}\cdot\log(\frac{f_{lp}}{10})=24{,}16\cdot\log(\frac{f_{lp}}{10})$	7Fh* (not specified)
Res_Freq_x*	9	ConRelRes	resonance frequency in Hz $f_{res}=100\cdot 1{,}02^{\text{Res\_Freq}}$ $\text{Res\_Freq}=\frac{1}{\log(1{,}02)}\cdot\log(\frac{f_{res}}{100})=116{,}3\cdot\log(\frac{f_{res}}{100})$	118h (f_{res} = 25589 Hz)
Q_Res_x*	9	ConRelRes	resonance quality $Q=0{,}4\cdot 1{,}02^{\text{Q\_Res}}$ $\text{Q\_Res}=\frac{1}{\log(1{,}02)}\cdot\log(\frac{Q}{0{,}4})=116{,}3\cdot\log(\frac{Q}{0{,}4})$	A3h (Q = 10,1)
Slope*	7	ConRes	amplitude slope in %/decade $S=-6{,}3+0{,}1\cdot Slope$ $Slope=(S+6{,}3)\cdot 10$	53h (S = 2 %/decade)
Temp_Co*	6	ConRes	temperature coefficient in %/K $TK(B_{ua})=-0{,}8+0{,}025\cdot\text{Temp\_Co}$ $\text{Temp\_Co}=(TK(B_{ua})+0{,}8)\cdot 40$	1Dh ($TK(B_{ua})$ = -0,075 %/K)
Ref_Freq*	8	ConRelRes	calibration frequency in Hz $f_{ref}=0{,}35\cdot 1{,}035^{\text{Ref\_Freq}}$ $\text{Ref\_Freq}=\frac{1}{\log(1{,}035)}\cdot\log(\frac{f_{ref}}{0{,}35})=66{,}9\cdot\log(\frac{f_{ref}}{0{,}35})$	9Eh (f_{ref} = 80,3 Hz)
Ref_Temp*	5	ConRes	calibration temperature in °C $T_{ref}=15+0{,}5\cdot\text{Ref\_Temp}$ $\text{Ref\_Temp}=(T_{ref}-15)\cdot 2$	10h (T_{ref} = 23 °C)
Cal_Date	16	DATE	calibration date in days since 01/01/1998 65535 = not specified	EF2h (23/06/08)
Cal_Init	15	Chr5	calibration initials, 3 characters, starting in LSB (0 = space, 1 = A , 2 = B... , . / _ @)	AB2h (BUR)
Cal_Per	12	UNINT	calibration period in days (integer) 1 – 4094; 4095 = not specified	16Dh (365 days)
Meas_ID	11	UNINT	measurement point ID no. (integer) 1 – 2046; 2047 = not specified	2h (2)
Sel	2	binary	selector bits for remaining memory (normally 3)	3h
Ext_End_Sel	1	binary	extended end selector, 1 = remaining bits used as user text data	1h
User	94	7 bit ASCII	user text data	“abcdefghijklm”

* All bits set in ConRelRes and ConRes format mean “not defined”.

4. TEDS Data Storage in the EEPROM DS2430A

Most legacy TEDS transducers include the discontinued DS2430A memory device featuring a 256 bit EEPROM and a 64 bit one-time programmable memory called application register.
The one-time programmable application register holds the Basic TEDS. The following table shows the arrangement of the data. The bit values under the identifiers show the data used as example in chapter 2.

7	6	5	4	3	2	1	0	Bit no.	Example
Man_ID_7 0	Man_ID_6 0	Man_ID_5 1	Man_ID_4 1	Man_ID_3 1	Man_ID_2 1	Man_ID_1 0	Man_ID_0 1	00	3Dh
Model_1 1	Model_0 0	Man_ID_13 0	Man_ID_12 0	Man_ID_11 0	Man_ID_10 0	Man_ID_9 0	Man_ID_8 0	08	80h
Model_9 0	Model_8 0	Model_7 0	Model_6 1	Model_5 0	Model_4 0	Model_3 0	Model_2 1	16	11h
Ver_Let_2 0	Ver_Let_1 0	Ver_Let_0 1	Model_14 0	Model_13 0	Model_12 0	Model_11 0	Model_10 0	24	20h
Ver_No_5 0	Ver_No_4 0	Ver_No_3 0	Ver_No_2 0	Ver_No_1 1	Ver_No_0 0	Ver_Let_4 0	Ver_Let_3 0	32	08h
Ser_No_7 0	Ser_No_6 0	Ser_No_5 0	Ser_No_4 0	Ser_No_3 0	Ser_No_2 0	Ser_No_1 1	Ser_No_0 0	40	02h
Ser_No_15 0	Ser_No_14 0	Ser_No_13 0	Ser_No_12 0	Ser_No_11 0	Ser_No_10 0	Ser_No_9 1	Ser_No_8 0	48	02h
Ser_No_23 0	Ser_No_22 0	Ser_No_21 0	Ser_No_20 0	Ser_No_19 0	Ser_No_18 0	Ser_No_17 0	Ser_No_16 0	52	00h

In EEPROM the template data is stored as shown below. The bit values under the identifiers show the example data of template no. 25 from chapter 3. The transfer function is disabled by bit *CaseTransf*. Therefore the data of the transfer function is not included. The first byte of the EEPROM holds the check sum calculated from the basic TEDS and all bytes after the checksum.

MSB							LSB	Bit no.	Example
Chksum_7	Chksum_6	Chksum_5	Chksum_4	Chksum_3	Chksum_2	Chksum_1	Chksum_0	00	
1	0	0	0	1	0	0	1		89h
Templ_No_5	Templ_No_4	Templ_No_3	Templ_No_2	Templ_No_1	Templ_No_0	Sel_Desc_1	Sel_Desc_0	08	
0	1	1	0	0	1	0	0		64h
Sens_3	Sens_2	Sens_1	Sens_0	Case_Gain	Case_a/F	Templ_No_7	Templ_No_6	16	
0	0	1	0	0	0	0	0		20h
Sens_11	Sens_10	Sens_9	Sens_8	Sens_7	Sens_6	Sens_5	Sens_4	24	
0	1	1	1	0	1	0	1		75h
HP_3	HP_2	HP_1	HP_0	Sens_15	Sens_14	Sens_13	Sens_12	32	
0	1	1	0	0	1	1	0		66h
Weight_1	Weight_0	Dir_1	Dir_0	HP_7	HP_6	HP_5	HP_4	40	
0	0	1	1	0	1	0	0		34h
Ref_Freq_1	Ref_Freq_0	Case_Transf	Polarity	Weight_5	Weight_4	Weight_3	Weight_2	48	
1	0	0	0	1	0	0	0		88h
Ref_Temp_1	Ref_Temp_0	Ref_Freq_7	Ref_Freq_6	Ref_Freq_5	Ref_Freq_4	Ref_Freq_3	Ref_Freq_2	56	
0	0	1	0	0	1	1	1		27h
Cal_Date_4	Cal_Date_3	Cal_Date_2	Cal_Date_1	Cal_Date_0	Ref_Temp_4	Ref_Temp_3	Ref_Temp_2	64	
1	0	0	1	0	1	0	0		94h
Cal_Date_12	Cal_Date_11	Cal_Date_10	Cal_Date_9	Cal_Date_8	Cal_Date_7	Cal_Date_6	Cal_Date_5	72	
0	1	1	1	0	1	1	1		77h
Cal_Init_4	Cal_Init_3	Cal_Init_2	Cal_Init_1	Cal_Init_0	Cal_Date_15	Cal_Date_14	Cal_Date_13	80	
0	0	0	1	0	0	0	0		10h
Cal_Init_12	Cal_Init_11	Cal_Init_10	Cal_Init_9	Cal_Init_8	Cal_Init_7	Cal_Init_6	Cal_Init_5	88	
0	1	0	1	0	1	0	1		55h
Cal_Per_5	Cal_Per_4	Cal_Per_3	Cal_Per_2	Cal_Per_1	Cal_Per_0	Cal_Init_14	Cal_Init_13	96	
1	0	1	1	0	1	1	0		B6h
Meas_ID_1	Meas_ID_0	Cal_Per_11	Cal_Per_10	Cal_Per_9	Cal_Per_8	Cal_Per_7	Cal_Per_6	104	
1	0	0	0	0	1	0	1		85h
Meas_ID_9	Meas_ID_8	Meas_ID_7	Meas_ID_6	Meas_ID_5	Meas_ID_4	Meas_ID_3	Meas_ID_2	112	
0	0	0	0	0	0	0	0		00h
User_3	User_2	User_1	User_0	Ext_End_Sel	Sel_1	Sel_0	Meas_ID_10	120	
1	1	0	0	1	1	1	0		AEh
User_11	User_10	User_9	User_8	User_7	User_6	User_5	User_4	128	
1	1	0	0	1	1	1	1		CFh
User_19	User_18	User_17	User_16	User_15	User_14	User_13	User_12	136	
1	1	1	0	0	0	1	1		E3h
User_27	User_26	User_25	User_24	User_23	User_22	User_21	User_20	144	
1	1	1	0	1	1	1	1		EFh
User_35	User_34	User_33	User_32	User_31	User_30	User_29	User_28	152	
1	1	1	1	0	1	1	0		F6h
User_43	User_42	User_41	User_40	User_39	User_38	User_37	User_36	160	
0	0	1	1	1	0	1	0		3Ah
User_51	User_50	User_49	User_48	User_47	User_46	User_45	User_44	168	
0	1	1	1	1	1	0	1		7Dh
User_59	User_58	User_57	User_56	User_55	User_54	User_53	User_52	176	
0	0	1	0	1	1	1	0		2Eh
User_67	User_66	User_65	User_64	User_63	User_62	User_61	User_60	184	
1	0	0	0	1	1	1	1		8Fh
User_75	User_74	User_73	User_72	User_71	User_70	User_69	User_68	192	
1	1	0	0	0	0	1	1		C3h
User_83	User_82	User_81	User_80	User_79	User_78	User_77	User_76	200	
1	1	0	1	1	1	1	1		DFh
User_91	User_90	User_89	User_88	User_87	User_86	User_85	User_84	208	
1	1	1	0	1	1	1	0		EEh
User_99	User_98	User_97	User_96	User_95	User_94	User_93	User_92	216	
0	0	1	1	0	1	1	0		36h
User_107	User_106	User_105	User_104	User_103	User_102	User_101	User_100	224	
0	1	1	1	1	0	1	1		7Bh
User_115	User_114	User_113	User_112	User_111	User_110	User_109	User_108	232	
1	0	1	0	1	1	0	1		ADh
User_123	User_122	User_121	User_120	User_118	User_118	User_117	User_116	240	
0	1	0	0	1	1	1	0		4Eh
User_131	User_130	User_129	User_128	User_127	User_126	User_125	User_124	248	
0	0	0	0	0	0	1	1		03h

5. TEDS Data Storage in the EEPROM DS2431

For sensors made in 2020 or later Metra uses the 1024 bit EEPROM DS2431 with four blocks of 256 bit each. This memory device does not have a one-time programmable application register. Due to this fact the basic TEDS data must be stored also in the erasable EEPROM section. Thus it will not be protected anymore against manipulation.
Each of the four blocks contains a check sum in the first byte. The checksum is calculated from the remaining 31 bytes of the block. The additional memory space compared to the DS2431A can be filled with user text.
The following table shows the arrangement of data in the DS2431.

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Block 1	C	B	B	B	B	B	B	B	B	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Block 2	C	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Block 3	C	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Block 4	C	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

C: check sum (1 byte)
B: Basic TEDS (8 bytes)
T: TEDS template and user data (remaining bytes)

6. Detecting the Memory Device

Each 1-Wire device has a 64 bit lasered ROM section containing unique ID data. The data is arranged as follows:

8 bit CRC code	48 bit serial number	8 bit family code
MSB		LSB

The family code in the lower 8 bits is used to identify the memory device so that the software can decode the data accordingly.
DS2430A 14h
DS2431 2Dh

More details regarding 1-Wire communication can be found in the corresponding data sheets of Maxim Integrated
https://www.maximintegrated.com/en/products/interface/one-wire.html

Manfred Weber
Metra Mess- und Frequenztechnik in Radebeul e.K.
Meißner Str. 58
D-01445 Radebeul
Tel. +49-351-836 21 91
P.O. Box 01 01 13
D-01435 Radebeul
Fax +49-351-836 29 40

Internet: www.MMF.de
Email: Info@MMF.de