

## Specification E2 Interface

### Version 2.0

	Name	Date
Created:	T. Pflügl	02.07.2004
Checked:	Hr.Froschauer	27.07.2004
Released:	Schwarz Roswitha	27.07.2004
Reason for change:	Supplements in section 2.3.1	

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	Overview .....	3
1.2	Agreements.....	3
<b>2</b>	<b>CHARACTERISTICS.....</b>	<b>3</b>
2.1	Layer 1: Physical Layer .....	3
2.2	Layer 2: Data Link Layer.....	5
2.2.1	Bit Transfer .....	5
2.2.2	Data Transfer Formats.....	5
2.2.3	Control Byte .....	6
2.2.4	PEC Packet Error Code (Checksum).....	6
2.3	Layer 3: Network Layer (Protocol) .....	7
2.3.1	Read Byte from Slave .....	7
2.3.1.1	Sensor Type (Group).....	8
2.3.1.2	Sensor Type (Subgroup) .....	8
2.3.1.3	Available Measured Variables.....	8
2.3.1.4	Interrogate Status Byte.....	8
2.3.1.5	Read out Measured Variables.....	8
<b>3</b>	<b>SUPPLEMENT: .....</b>	<b>9</b>
3.1	Read Command .....	9

## 1 Introduction

### 1.1 Overview

The E2 interface was specified by E+E Elektronik in 2004 and represents a subset of the E2 bus protocol. The E2 bus is based on similar principles as the I<sup>2</sup>C Bus<sup>1</sup> or the SMBus<sup>1</sup> introduced by Phillips Semiconductors in 1982. The most significant differences to the I<sup>2</sup>C Bus are the slower transmission rate, the slightly divergent addressing mechanism and the error detection (checksum). The E2 interface is used for the digital, bi-directional data transmission between a master module (e.g.: climate controller, microcontroller...) and a slave module (commonly an E+E transmitter). The data transmission takes place synchronous and serial, whereby the master is responsible for the generation of the clock pulse. The slave cannot send any data independently.

### 1.2 Agreements

The technical specifications in this document are to be regarded as recommendations and are applicable for all E+E transmitters with E2 interface, insofar as nothing to the contrary is specified in their data sheets.

#### Terms:

Master: Module with E2 interface that initiates a communication and can drive a clock signal.

Slave: Module with E2 interface that cannot drive a clock signal.

## 2 Characteristics

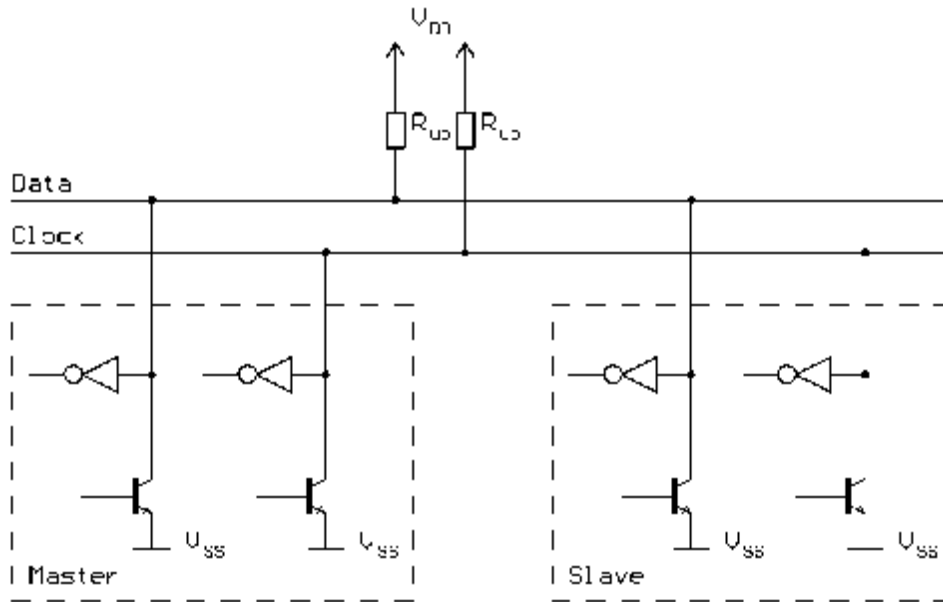
In the following, the E2 interface will be specified in accordance with the ISO-OSI Layer Model up to Layer 3. In addition, it should also be stated, that due to the great similarity with the SM Bus or the I<sup>2</sup>C Bus, the E2 interface is hardware-compatible with popular microcontrollers with interfaces of that type. E2-slaves can therefore be connected directly to the corresponding pins of such controllers.

### 2.1 Layer 1: Physical Layer

The E2 interface consists of two active lines (Clock and Data) as well as a grounding line as reference potential. The Clock and Data lines are connected with the operating voltage over pull-up resistors. In the idle state, both lines are at High-Level (positive logic). The respective inputs and outputs of the modules are designed as Open Drain (or Open Collector), and thus can be connected directly as „wired AND“.

---

<sup>1</sup> All brands, names, product names and logos listed are registered trade marks or brands of their respective owners, that we hereby explicitly acknowledge.



**Fig. 1: Principle joining together of Master- and Slave Module**

### Parameters

Symbol	Parameter	Minimum	Maximum	Unit	Remark
$V_{DD}$	Operating voltage			V	See module description
$V_{IH}$	Input High Level	$0.8 V_{DD}$	$V_{DD} + 0.3$	V	
$V_{IL}$	Input Low Level	$V_{SS} - 0.3$	$0.2 * V_{DD}$ or 0.8	V	Smallest value is valid
$V_{OL}$	Output Low Level		0.7	V	$I_{in} = 0.5 \text{ mA}$
$C_{max}$	Line capacity to ground		1	nF	at $R_{up} = 22 \text{ k}\Omega$ see remark 1
$f_{CLK}$	Clock frequency (data rate)	500	5000	Hz	See remark 1
$t_{CLKH}$	Clock-High time	100		$\mu\text{s}$	
$t_{CLKL}$	Clock-Low time	100		$\mu\text{s}$	
$R_{up}$	Pull-up resistor	1	100	$\text{k}\Omega$	See remark 1

For further parameters not specified here, the specifications in the data sheets of the modules used are applicable.

Remark 1: The maximum data rate that can be achieved is dependent on the combination of the line capacity and the pull-up resistors.

## 2.2 Layer 2: Data Link Layer

### 2.2.1 Bit Transfer

The data transmission fundamentally takes place serial by bit and synchronous, whereby the clock line is used as synchronization line, and always controlled from the master. With the exception of the start- and stop conditions, a change of the level on the data line is always only permissible during a LOW-phase of the clock line. At the beginning of every communication there is a start condition. This is realized by a negative edge on the data line, with simultaneous sustained high-level on the clock line. After a delay of at least 4 $\mu$ s the clock line is drawn to low-level, and the first data bit (MSB) can be placed on the data line. The data transfer takes place during the high-phase of the clock line. After all data bits (incl. ACK/ NACK) have been sent, the communication is terminated with a stop condition. One terms a stop condition as a positive edge on the data line with simultaneous high-level on the clock line.

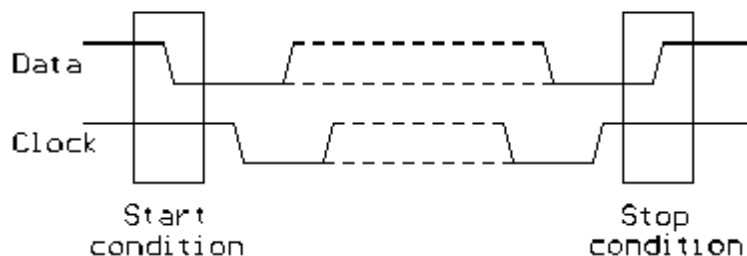


Fig. 2: Bit Transfer

### Clock Low Extension

The slave has the possibility of holding the clock line at low level for up to 25ms after every transmitted data bit, in order to gain a little time for internal operations. The total transmission time for a complete byte may not however exceed 35ms.

### 2.2.2 Data Transfer Formats

The data are transmitted from the transmitter in bytes, and confirmed by the receiver individually (for each byte) with ACK (data line = Low) or NACK (data line = High) as ninth bit. The first data byte (control byte) is always sent from the master to the slave. Thereby the first seven bits represent the „Address“ of the slave module. The eighth bit (R/W) specifies the direction of the data transfer (R/W=0 data from master to slave; R/W=1 data from slave to master).

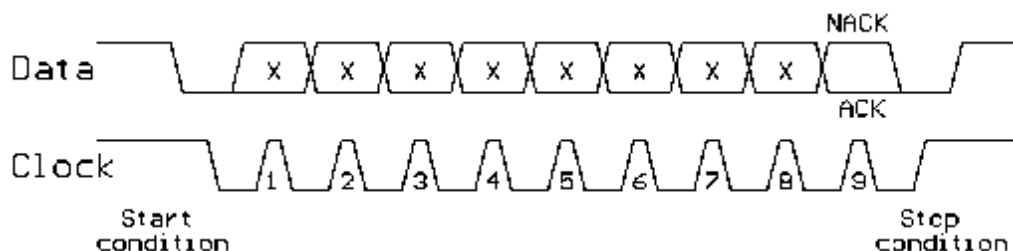


Fig. 3: Byte Transfer

### 2.2.3 Control Byte

The control byte of the E2 interface is used (in contrast to the E2 bus protocol) merely for the differentiation of various command modes (which are defined in the so-called Main Command) and the data flow direction (R/W). A differentiation of several slave modules by means of a different device address is not provided for with the E2 interface. The device address for all slaves is always 0, but for reasons of compatibility must also be transmitted.

The control byte is comprised as follows:

Control Byte							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main Command				Device Address			R/W

**Fig. 4: Control Byte Structure**

Bit	Meaning
0 (LSB)	R/W
1	Device Address low bit (always 0)
2	Device Address (always 0)
3	Device Address high bit (always 0)
4	Main Command low bit
5	Main Command
6	Main Command
7 (MSB)	Main Command high bit

### 2.2.4 PEC Packet Error Code (Checksum)

For the detection of transmission errors a checksum byte is transmitted as the last data byte of every transmission. The checksum byte corresponds with the low byte of the sum (unsigned char) of all transmitted bytes.

Read: Checksum byte = (Control byte + Data byte) MOD 0x100

Write: Checksum byte = (Control byte + Address byte + Data byte) MOD 0x100

## 2.3 Layer 3: Network Layer (Protocol)



In this section, all defined commands of the E2 interface are described. For the data format of the transmitted data bytes, refer to the data sheets of the modules used.

### 2.3.1 Read Byte from Slave

This command is supported by all modules with E2 interface and is used for the reading out of individual data bytes.

Command structure:

Start	Control byte									Data byte									Checksum									Stop
	7	6	5	4	3	2	1	0	ACK	7	6	5	4	3	2	1	0	ACK	7	6	5	4	3	2	1	0	ACK	
Start								1	A	x	x	x	x	x	x	x	x	A	x	x	x	x	x	x	x	x	NA	Stop

 From Master to Slave  
 From Slave to Master

**Attention:** For the Read command, the bit 0 of the control byte is always 1 (High). The Read command is terminated by a **NACK** and a stop condition of the master.

For Read Byte from Slave, the Main Commands are organized in such a form, that important data- and information bytes (all measured values, status byte, sensor type...) can be read out with one single command.

The following Main Commands are defined:

Control byte	Bit number				Type of command
	7	6	5	4	
0x11	0	0	0	1	Output sensor type (group)
0x21	0	0	1	0	Output sensor type (subgroup)
0x31	0	0	1	1	Read out available measured variables
0x41	0	1	0	0	Reserve
0x51	0	1	0	1	Reserve
0x61	0	1	1	0	Reserve
0x71	0	1	1	1	Status byte
0x81	1	0	0	0	Measured variable 1 low byte
0x91	1	0	0	1	Measured variable 1 high byte
0xA1	1	0	1	0	Measured variable 2 low byte
0xB1	1	0	1	1	Measured variable 2 high byte
0xC1	1	1	0	0	Measured variable 3 low byte
0xD1	1	1	0	1	Measured variable 3 high byte
0xE1	1	1	1	0	Measured variable 4 low byte
0xF1	1	1	1	1	Measured variable 4 high byte

### 2.3.1.1 Sensor Type (Group)

Identifies the current sensor type.

Example: Group = 3 hex for EE03.  
Group = 7 hex for EE07.

### 2.3.1.2 Sensor Type (Subgroup)

Identifies the sub-type of the current sensor (upper 4 bits) and the method of the output (lower 4 bits).

Example: Subgroup = 19 hex for EE<sub>xx</sub>-1 in FT9 execution (E2 interface).  
Subgroup = 46 hex for EE<sub>xx</sub>-4 in FT6 execution (4-20mA).

For further information see the module description.

### 2.3.1.3 Available Measured Variables

This field specifies which measured variables can be acquired with the current sensor. The following assignment to the individual bits is thereby used:

Available Measured Variables							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	T-passive	Reserved	CO <sub>2</sub>	Flow	Temperature	Humidity

1...supported  
0...not supported

### 2.3.1.4 Interrogate Status Byte

Through the interrogation of the status byte, the start of a measurement (in the slave) can be forced. After the waiting of the module-specific measuring time, the new measured values can then be read out. The status byte provides information about the validity of the last measurement. The assignment of the bits to the measured variables is equivalent to the assignment in the measured variables byte.

1...Measured value faulty (see module description)  
0... Measured value in order

### 2.3.1.5 Read out Measured Variables

When reading out a 16 bit measured variable, it is expedient to read out the low byte first and then the associated high byte. This method of procedure ensures, that two associated bytes are always read out (when reading the low byte, the high byte is „captured“ in the slave). For the measured values data format used, refer to the data sheets of the corresponding modules.



### 3 Supplement:

Principle structure of a communication over E2 interface (Slave).

#### 3.1 Read Command

