VERSION UPDATE		
NR.	EDITION	CONFIRMED
1	V 2.1	2003-04-17

Data Sheet

MMI 4832

Integrated Circuit for Position Measuring Systems Interfacing

Table of contents

MAZeT GmbH Sales
Göschwitzer Straße 32
07745 JENA / GERMANY
Phone: +49 3641 2809-0
Fax: +49 3641 2809-12
E-Mail: sales@MAZeT.de
Url: http://www.MAZeT.de

Approvals	Date	MAZeT GmbH	
Compiled:	2002-04-30	Status: valid	
Checked:	2003-04-10		
Released:	2003-04-17	DOC. NR.: DB-99-031e	Page 1 of 45

1 Introduction

Integrated Circuit for Position Measuring Systems Interfacing via:

EnDat	*SSI*	*Square-wave incremental signals*
(Bidirectional interface for absolute encoders as defined by DR. J. HEIDENHAIN GmbH)	*(Unidirectional interface for absolute encoders)*	*(Incremental Z-track interface with 32-bit encoder counter)*

Features Outline

- MRS/address checking
- CRC checking
- Alarm bit checking
- Parity bit checking
- Gray-to-binary code conversion optionally selectable
- Edge evaluation for square-wave incremental signals

- 32-bit control register, 8-bit status register, 8-bit interrupt mask
- 8-/16-bit microcontroller interface (Intel or Motorola mode)
- Internal 18-bit timer
- System clock rate up to 33 MHz (EnDat: 24 MHz)

The MMI4832 is an integrated circuit to provide interfacing for absolute encoders, i.e. sensors with an EnDat® interface or a SSI port and/or encoders (of incremental mode type) alternatively. In EnDat® interface transfer mode, it operates in accordance with the transfer protocol as defined by the EnDat Interface® documentation V2.1 of DR. J. HEIDENHAIN.

1.1 “EnDat®“, “SSI“ transfer mode

“EnDat®” or “SSI” transfer mode means that the serial data from an encoder is received by a DATA_RC pin. In EnDat-Interface® mode, the data are sent via pin DATA_DV. For communication with a pickup sensor, a clock is provided via pin TCLK. The clock signals which are necessary to transfer any of the mode commands 1, 2, 3, 4, 5, 6 (compare with EnDat® documentation) are automatically generated (compare with section: 3).
For mode command 0 (telling the measuring system to send absolute position value(s)), a width must be defined for the expected position value via control register bit (24:29) – data word 2). The desired transfer rate can be specified via control register bit (17:8) (compare with section: 5.3.1). Transfer rates ifrom 100 KHz to 2 MHz are supported. 24 MHz is the recommended system rate setting.

Data transfer principle in EnDat® and SSI mode

Data transfer can be initiated in three different ways: by an external strobe (H/L-edge), by a timer strobe (H/L-edge) or by a software strobe. Figure 1-1 shows a principal data transfer diagram.

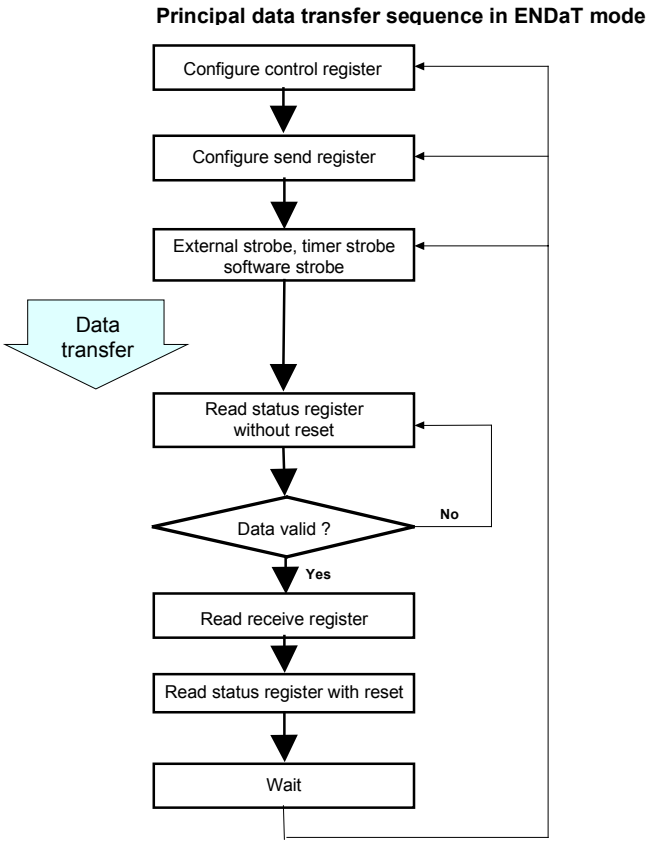

Figure 1-1: Principal data transfer diagram for EnDat® and SSI mode

1.2 "Incremental" transfer mode:

A direction discriminator uses the two 90°-square-wave signals (CHA, CHB) as input to generate pulses which are then further processed in a downline 32-bit counter. This may be accomplished with selectable single-, double- or quadruple-edge evaluation. CHC and CHD serve as index tracks. Optionally, the MMI4832 may also be operated in pulse direction mode. For explanations on how counting signals are generated and selected, you should refer to section 4.2. To reduce the influence of interferences, programmable digital input filters can be selectively connected to the four counter inputs CHA, CHB, CHC and CHD, the filter length being selectable in integer multiples of the system clock.

The counter has a downline strobe register for the counter content to be written into this register with a hardware strobe or a software strobe without any need to interrupt the counting process.

The counter is reset to zero if an index pulse is applied to any of the measuring system inputs (CHA*CHB*CHC *CHD) with a concurrent software enable signal (control register bit (3), control register bit (4)) or when the zero input is active (zero = low). /NIA is a low-active signal to serve as the reset output of a counter circuit which functions as a master, in order to allow synchronization of several measuring systems.

On selection of "Set-by-Reference" (control register bit (6)), an index pulse arriving at (CHA*CHB*CHC*CHD) will cause the counter to be loaded with the preset register value. If "Reset" and "Set-by-Reference" are active at the same time, the "Reset" function will prevail.

On selection of "Reference Strobe" (control register bit (1)), the latest counter values at the moment of an index pulse arriving at (CHA*CHB*CHC*CHD) will be loaded into the strobe register.

"Reference Compare" (control register bit (2)) allows the current counter reading to be compared with the value in the offset register. If the two values are found to be equal, status register bit (2) will be set.

On selection of "Add Offset", an offset is added to the incremental value. This function is enabled by bit 10 of the control register. The full offset value (receive register + offset) is read from the multifunction register address.

In "incremental" transfer mode, the status register keeps track of reset, reference strobe, strobe and interference events at the inputs of the measuring system.

1.3 Functions available in both transfer modes

Acting in conjunction with the 8-bit status register and the 8-bit interrupt mask, the control register supports easy recording and monitoring of the measured values.
The interrupt mask makes it possible to selectively trigger an interrupt request for messages from a measuring system.

The two inputs /IR6 and /IR7 are recorded by the status register. They allow specific application events (such as supply voltage availability, temperature, cable breakage, etc.) to be monitored.

In all three modes (EnDat, SSI, incremental), a timer for selection of a desired sampling repeat rate (sampling rate) is available (compare with section **5.6**). For a system clock rate of 24 MHz, the selected sampling rate may be in the range of 0.417μs to 10.9 ms. Alternatively, measured values can also be requested via the /STR input or a software strobe.

1.4 Block diagram

Figure 1-2 shows a block diagram of the MMI4832:

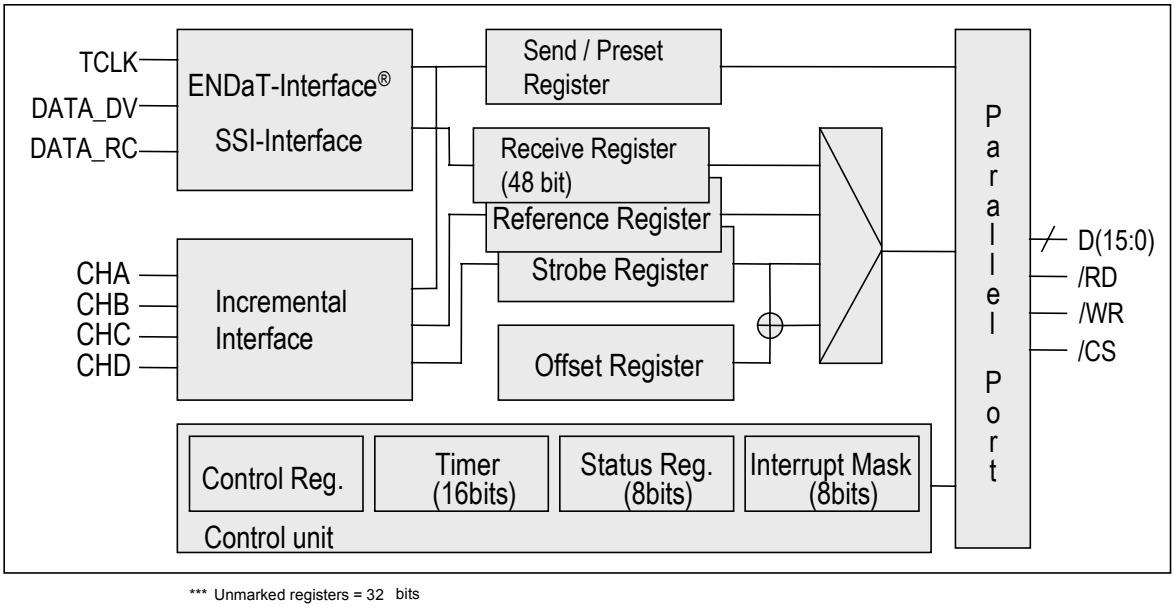

Figure 1-2: MMI4832 block diagram

2 Parallel Interface

The MMI4832 has a multiplexed data/address bus. With pin CMD=1, the desired operating mode (8-/16-bit port, autoincrement) and the resource address need to be written into the internal address register. Once this has happened, data access cycles (writing/reading of a resource e.g. the control register) may follow with CMD = 0.

Address register: CMD = 1 read/write

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	M16	AINC (auto increment)	Address A4	Address A3	Address A2	Address A1	Address A0

The number of cycles required to access a given resource depends both on the selected port width (8 or 16 bits, where 16-bit port width is set with M16 = 1) and the resource's width. If multiple access cycles are required for a given resource, writing in parallel-port mode ("Intel") must begin with the least significant and in port mode ("Motorola") with the most significant data word/byte.

With the L/H-edge (/WR) of each last access cycle to a resource, the access date will be written into this resource. In "Motorola" port mode, the two control signals /WR, /RD acquire a new meaning:
/WR -> R_/W: switching between read/write
/RD -> /ADS: address or data strobe respectively.

For reading cycles, one should consider that the current content of the resource being accessed is already loaded into the output register on writing into the address register. Accordingly, in order to obtain a value of latest validity, it is necessary to enter the resource's base address into the address register again before reading this resource for the next time.

Table 1 shows how resources are addressed:

Register		Base address A(4:2)	Access cycles 8-/16-bit port	Comment
Send/Preset RG	W/R	0	4/2	32 bits
Receive RG	RO	1	≤ 4/2	Data word length </= 32 bits Compare with control RG
Receive RG	RO	2	2/1	Data word lengths > 32 bits
Multifunction RG	W/R	3	4/2	32 bits
Control RG	W/R	4	4/2	32 bits
Status RG	RO	5	1/1	8 bits
Software strobe	WO	5	1/1	---- (function, no resource)
Interrupt mask	W/R	6	1/1	8 bits
Timer	WO	7	2/1	16 bits
Status RG	RO	7	1/1	8 bits

Table 1: Adressation of resources

When working in autoincrement mode (AINC=1), the base address (A(4:2)) of a resource needs to be entered into the address register and /WR or /RD be repeated for a number of “n“ times respectively. The byte/word address [A(1:0)] is automatically incremented after each access operation. On full completion of an access cycle, the address pointer will again refer to the base address, i.e. A(1:0)=0/dec. An incomplete access cycle may at any time be terminated by “Read status register with reset” [A(4:2)=5/dec] or by writing into the address register (CMD = 1). This will also reset the address pointer A(1:0). With autoincrement mode turned off (AINC=0), the byte/word addresses A(1:0) must be set by the user.

3 EnDat® -Interface

3.1 Transfer formats (EnDat®)

3.1.1 Transfer or position values

Mode command: 0 *Tells measuring system to send absolute position value(s)*

Interrupted clock:

Datenformat																		
	M M M / / /							S	A	L		M	C	C	C	C	C	
	M2 M1 M0							T	L	S	....	S	R	R	R	R	R	
				M2	M1	M0		A	A	B		B	C	C	C	C	C	

Uninterrupted clock:

New receive cycle:

Datenformat																	
	CRCC	CRCC	CRCC	CRCC	CRCC			STAA	ALSB	....	MSB	CRC	CRC	CRC	CRC	CRC	

3.1.2 Transfer of parameters

Mode command: 1 Selects memory range

Datenformat																			
	M2	/	M0	..	C7	C	..	D15	D	..	D0		STA	C7	C	..	D15	D	..	D0	CRC	CRC	CRC	CRC	CRC

Mode command: 3 Tells measuring system to receive parameters
Mode command: 4 Tells measuring system to send parameters
Mode command: 5 Tells measuring system to receive reset

Datenformat																			
	M2	/	M0	..	A7	A	..	D15	D	..	D0		STA	A7	A	..	D15	D	..	D0	CRC	CRC	CRC	CRC	CRC

3.1.3 Start-up diagnosis (test in idle state)

For start-up diagnosis, three steps are required:

a) *Mode command: 6 Tells measuring system to receive test command (with port address to be queried)*

Datenformat																			
	M2	/	M0	..	A7	P	..	D15	D	..	D0		STA	A7	P	..	D15	D	..	D0	CRC	CRC	CRC	CRC	CRC

b) *Mode command: 2 Tells measuring system to send test values*

Datenformat																	
	M2	/	M0	..	STAA	AD	..	D39	D	D0	CRC	CRC	CRC	CRC	CRC		

c) *Mode command: 6 Resets port addresses on instructing the measuring system to receive test command (with address XX000000/b)*

Datenformat																			
	M2	/	M0	..	A7	P	..	D15	D	..	D0		STA	A7	P	..	D15	D	..	D0	CRC	CRC	CRC	CRC	CRC

Legend:

Send
Receive

3.2 Registers

3.2.1 Send register

Base address: 0/h, R/W, 30 bits, reset value: 0000 0000 /h

The send register contains a mode command with related addresses and data:

"Byte 4" Bit (29:24)	Byte 3 Bit (23:16)	Byte 2 Bit (15:8)	Byte 1 (Bit (7:0)
Mode command D (5:0)	Selects memory/ addresses	Data high	Date low

Its task is to perform sequencing control of clock transfer processes to an encoder. The content of this register is automatically transferred to the PSC (compare with 3.2.2).
A new value is only strobed into the send register if the writing cycle was completed and the transfer cycle to/from the encoder was completed (compare with 3.2.2). This prevents interruption of a running transfer cycle to or from a measured-value transmitter.

The various mode commands for measuring system control need to be defined in bits (29:24).

Bit(29:24):
- 07/h Tells measuring system to send position values (Mode command 0)
- 0E/h Selects memory range (Mode command 1)
- 1C/h Tells measuring system to receive parameters (Mode command 3)
- 23/h Tells measuring system to send parameters (Mode command 4)
- 2A/h Tells measuring system to receive reset (Mode command 5)
- 15/h Tells measuring system to send test values (Mode command 2)
- 31/h Tells measuring system to receive test command (Mode command 6)

Byte (23:16) is available for storing related MRS codes, addresses and port addresses of the desired resources of a measuring system.

Bit(23:16):
- Mode command 1: MRS code C(7:0)
- Mode commands 3,4,5: Addresses A(7:0)
- Mode command 6: Port addresses P(7:0)

Bits(15:0) are reserved for the data to be sent to the measuring system (mode command 3 telling the measuring system to receive parameters).

In "incremental" transfer mode, the send register functions like a 32-bit preset register (compare with sections: 4.3.2, 4.3.3).

3.2.2 Parallel-to-serial converter (PSC)

Has no base address of its own, 30 bits, reset value: 0 /h

With a bit length of 30 bits, the parallel-to-serial converter generates a serial data stream that is transferred via the DATA_DV pin to a measured-value transducer. Data can only be written into the converter via the send register (indirectly).
A new value from the send register is only allowed to enter the PSC if the corresponding transfer cycle to or from the measured-value transducer was completed (compare with 3.2).

"Byte 4" Bit (29:24)	Byte 3 Bit (23:16)	Byte 2 Bit (15:8)	Byte 1 (Bit (7:0)
Mode command D (5:0)	Selects memory/ addresses	Data high	Data low

3.2.3 Seriell-to-parallel converter (SPC)

Has no base address of its own, 48 bits, reset value: 0 /h

The serial-to-parallel converter has 48 bits. It receives a stream of serial data from an absolute encoder via pin DATA_RC. Data recording begins when a start bit that was sent by the encoder has been identified and the alarm bit status been inquired, provided that an alarm bit is transferred as part of a selected mode command.

3.2.3.1 Receive mode 1

Measuring system is told to send absolute position values M(2:0) = 0/h

The number of required transfer clocks is determined by the "data word length" as defined in control register bit(29:24). The maximum allowed transfer length is 48 bits. Sending begins with the LSB. Depending on the data word length which was set in the control register, the MSB position will shift more or less in this register. Once the MSB has been received, the CRC hardware(*) will be enabled to receive a CRC code.

3.2.3.2 Receive mode 2

Selects memory range	*M(2:0) = 1/h*
Tells measuring system to receive parameters	*M(2:0) = 3/h*
Tells measuring system to send paramters	*M(2:0) = 4/h*
Tells measuring system to receive reset	*M(2:0) = 5/h*
Tells measuring system to receive test command	*M(2:0) = 6/h*
Tells measuring system to send test data	*M(2:0) = 7/h*

The number of required transfer clocks is automatically generated. In contrast to receive mode 1, always eight MRS or address bits and sixteen parameter bits are received in this mode. Sending begins with the MSB. Once the LSB was received, the CRC hardware is enabled to receive a CRC code.

3.2.3.3 Receive mode 3

Tells measuring system to send test values *M(2:0) = 2/h*

The number of required transfer clocks is automatically generated. The test data word to be received has a width of forty bits. Sending begins with the MSB. Once the LSB has been received, the CRC hardware is enabled to receive a CRC code.

3.2.3.4 CRC

In each of the three receive modes described before, the CRC bits are compared with those CRC bits which have been generated from the data stream that was received. On completion of a CRC test, the serial-to-parallel converter is again operational.

3.2.3.5 Data loading into the receive register

A value received from the SPC is loaded into the receive register on completion of a serial-to-parallel conversion process and completion of a CRC check, unless an error bit (compare with status register) was set. A new and valid will only overwrite a previous value if the status bit 0 was reset (read status register with reset).
(compare with section: **5.1.1.1**)

* CRC test hardware according to EnDat® definition V2.1 Appendix A4 of DR.J.HEIDENHAIN GmbH

4 Incremental Interface

4.1 Digital input filters

Programmable digital filters are provided for the four input signals CHA, CHD, CHC, CHD to allow interferences to be removed. For a signal to be identified as a useful signal, it must have a certain minimum duration. If a signal fails to last for this minimum prescribed time, it will be detected as an interference.

Filter programming, i.e. definition of the minimum required duration for a signal to qualify as “useful signal“ is made via the control register bits (15:11).

Preselection of value [WT_{usable}] for a desired minimum useful signal duration [T_{usable}]

$[WT_{usable}] = (T_{nutz} / 2*T_{sys}) - 1$ or

$[WT_{usable}] = (f_{sys} / 2*f_{usable}) - 1$

T_{usable} - is the minimum duration of a useful signal
f_{usable} - is the maximum frequency of a useful signal
f_{sys} - is the system clock rate
T_{sys} - is the period of a system clock

Example:
In order to detect only signals with a frequency ≤ 312.5 kHz as useful input signals if a system clock of 20 MHz was set, 1F/h must be written into the control register bits (15:11) for WTusable.

Following a reset, the filters are turned off as a standard measure. On defining a value of 0/dec for the control register bits (15:11), the filter will also be turned off.

4.2 Edge evaluation

Selections for counting pulse generation can be made through bits (9:8) of the control register. Selection code assignments are as shown in Table 2 below. Reset will select pulse direction mode.

CHA	CHB	Pulse Dir Bit(9:8)=0	Single Bit(9:8)=1	Double Bit(9:8)=2	Quadruple Bit(9:8)=1
↓	H				VZ
L	↓			VZ	VZ
↑	L	VZ			VZ
H	↑		VZ	VZ	VZ
H	↓		RZ	RZ	RZ
↓	L				RZ
L	↑			RZ	RZ
↑	H	RZ			RZ

Table 2: Generation of counting pulses

VZ: Counter increments RZ: Counter decrements

4.3 Counter

Base address: 1/h, R/W, 32 bits, reset value: 0 /h

Reading: via strobe register (base address: 1/h)
Writing: via preset register (base address: 0/h ->1/h)

The reversible counter is loadable. It counts the counting pulses which are generated by the discriminator.

4.3.1 Reset

Resetting (zeoring) fulfils three functions:

a) If the "Reset" bit (control register bit (3)) is set, the counter will be reset once with CHA*CHB*CHC*CHD. This will set status bit (3) "Reset". If "Repeat" (control register bit 4) is unset, no more resets will follow with a next index pulse CHA*CHB*CHC*CHD (only via bypass path conditions of preset register=0/h and "Set-by-Ref" (control register bit 6) – also compare with section: 4.3.2, 4.3.3).

b) If bits "Repeat" and "Reset" are set, the counter will be reset to 0/h with each CHA*CHB*CHC*CHD.

c) An external /NULL signal will set the counter to 0/h. It is intended for measuring systems with master-slave structures.

4.3.2 Set-by-Reference

If function "Set-by-Ref" is enabled (control register bit (6) = 1), the preset register value will be loaded into the counter on a CHA*CHB*CHC*CHD.
Note: If "Reset" and "Set-by-Ref" are enabled at the same time, "Reset" will prevail.

4.3.3 Counter loading via parallel port

Writing into the counter can not be accomplished directly from the parallel port. On writing to the counter's base address (1/h), a given value is loaded from the preset register into the counter.

4.3.4 Reference compare

On selection of this function (control register bit (2)) the current counter reading (value) will be permanently compared with the value that is contained in the multifunction register. If both values are found to be equal, status register bit (2) will be set.

4.3.5 Add offset

An offset is added to the incremental value. This function is enabled via control register bit (10).

The full offset value (receive register + offset) is read from the address of the multifunction register.

4.4 Preset register

Basa address: 0/h, R/W, 32 bits, reset value: 0 /h

A value which is temporarily stored in the preset register can be loaded into the counter by writing to the counter's base address (1/h) (compare with section: 4.3.2, 4.3.3).

In "EnDat®" transfer mode, the preset register functions like a 30-bit send register (compare with section: 3.2).

4.5 Strobe register

Base address: 1/h, *RO, 32 bits, reset value: 0 /h control register bit (16) = 0

Current counter values can be loaded into the strobe register via hardware strobe or software strobe:

4.5.1 Hardware strobe

If strobe is enabled (control register bit (0) = 1) and status register bit (0)=0) reset, the counter value will be loaded into the strobe register with a falling edge at the external strobe input /STR or at the timer (internal signal derived from /SRB). This will set status register bit (0)=1.

4.5.2 Software strobe

On "Writing" to the base address 5/h, a counter value will be loaded into the strobe register. However, *no* data is written from the parallel port into the MMI4832!
The software strobe is independent of the control register bit (0). It is being reflected in the status register.

4.5.3 Signal transit time when working with digital input filters

In the case of a hardware strobe, the internal strobe pulse is delayed by a certain amount of transit time depending on what input filter option has been enabled. This option has been incorporated, in order to ensure a precisely timed (and, hence, precisely positioned) loading of the counter information. In the case of a software strobe, the internal strobe pulse is derived from the falling edge of a /WR writing signal without any correlated delay.

4.6 Reference register

Base address: 1/h,* RO, 32 bits, reset value: 0 /h control register bit (16) = 1

The reference register supports the *Reference Strobe* function*:*

If “Reference Strobe” (control register bit (1)) is enabled, the current counter values will be loaded into the reference register on occurrence of an index pulse (CHA*CHB*CHC*CHD).

* Strobe and reference register are read via the receive register address!

5 General-Purpose Registers

5.1 Receive register

Base address: 1/h, RO, </= 48 bits, reset value: 0 /h

5.1.1 “EnDat®“, “SSI“ transfer mode

5.1.1.1 Data loading into receive register

There are two ways to load received data into the receive register.

5.1.1.1.1 Hardware strobe

Control register bit (0) = 1:
A value received from the SPC and found to be faultless is strobed into the receive register on completion of the transfer cycle. After this value has been read, the status register bit (0) needs to be reset to allow a new value to be loaded into the receive register.

Control register bit (0) = 0:
No transfer can be triggered by a hardware strobe or a timer strobe. Accordingly, no data are loaded into the receive register.

5.1.1.1.2 Software strobe

A software strobe (writing to base address 5/h) will trigger a transfer start independently of the state of control register bit (0).It will also load a previously received valid value (no error message) into the receive register.
After this value has been read, the status register bit (0) needs to be reset, in order to allow a new value to be loaded into the receive register!

5.1.1.2 Mapping of transfer packages in receive register

The following tables show what position the various transfer packages will occupy in the receive register.

Measuring system is told to send absolute position values M(2:0) = 0/h

Byte 6 Bit (47:40)	Byte 5 Bit (39:32)	Byte 4 Bit (31:24)	Byte 3 Bit (23:16)	Byte 2 Bit (15:8)	Byte 1 Bit (7:0)
M S B*	. . .	. . .	. . .	. . .	L S B

* The MSB position will shift depending on the length of a position value.

Memory range is selected *M(2:0) = 1/h*
Measuring system is told to receive parameters *M(2:0) = 3/h*
Measuring system is told to send parameters *M(2:0) = 4/h*
Measuring system is told to receive reset *M(2:0) = 5/h*
Measuring system is told to receive test command *M(2:0) = 6/h*
Measuring system is told to send test data *M(2:0) = 7/h*

Bit (47:40)	Byte 5 Bit (39:32)	Byte 4 Bit (31:24)	Byte 3 Bit (23:16)	Byte 2 Bit (15:8)	Byte 1 Bit (7:0)
---	---	---	D (15:8)	D (7:0)	A (7:0) A7...P0

Measuring system is told to send test values *M(2:0) = 2/h*

Byte 6 Bit(47:40)	Byte 5 Bit(39:32)	Byte 4 Bit(31:24)	Byte 3 Bit(23:16)	Byte 2 Bit(15:8)	Byte 1 Bit(7:0)
---	D (39:32)	D (31:24)	D (23:16)	D (15:8)	D (7:0)

5.1.2 "Incremental" transfer mode

Both the strobe register and the reference register are read via the receive register (compare with section: 4.5, 4.6).

5.2 Multifunction register (MFR)

Base address: 3/h, RO, R/W, 32 bits, reset value: 0 /h

5.2.1 "EnDat®", "SSI" transfer mode

The multifunction register contains specific IC test values.

5.2.2 "Incremental" transfer mode

The multifunction register (R/W) supports "Reference Compare" and "Add Offset". These functions are only available in incremental transfer mode (compare with section: 4.3.4, 4.3.5).

5.3 Control register

Base address: 4/h, R/W, 32 bits, reset value: 0000 0000 /h *

The control register allows you to set the various required transfer modes and functions:

5.3.1 "EnDat®", "SSI" transfer mode

With bit (7) = 0 the MMI4832 will work in "EnDat®" transfer mode. In this case, the control register will contain all required or available setups and functions for EnDat® mode:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	**0**	Gray2 binary	Parity even /odd	Parity check	Error check off	Uninter rupted clock	Reserved	Hardware strobe allowed
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Byte 2	Division factor	Division factor	Division factor	Division factor	Division factor	Division factor	Division factor	Division factor
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Byte 3	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Division factor	Division factor
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Byte 4	Disable 500ms	EnDat interface® / SSI	Data word length 2	Data word length 2	Data word length 2	Data word length 2	Data word length 2	Data word length 2

The following functions may be called up (bit x = 1):

Bit 0: Measured value loading allowed: On selection of timer or external strobe input /STR, a new value (found to represent no error signal) will only be loaded into the receive register if bit 0 = 1 and status register bit (0) = 0. A software strobe will, independently of the state of bit 0, cause the current (found to be faultless) value to be loaded into the receive register if status register bit (0) = 0.

Bit 2: As part of mode command "0" absolute position values are transferred with uninterrupted clock rhythm. This function is intended for fast measured value recording, e.g. where surface profiles have to be scanned (only "EnDat-Interface®" mode).
If control register bits (0)=1 while this function runs, the receive register will automatically be updated. Only a single strobe is necessary to trigger a transfer!

Bit 3: No check for MRS, address or CRC errors is performed and no error message writtten into the status register (only "EnDat-Interface®" mode).
A received value is loaded into the receive register.

Bit (5:4): In "SSI" mode: parity check is allowed
(bit 5= 0: even, bit 5= 1: odd)

Bit 6: Gray-to-binary conversion of serial data stream (only for "SSI")

Bit 7: Switches between modes:
"Incremental" (bit 7 = 1) or
"EnDat-Interface®" and "SSI" (bit 7 = 0)

Bit (17:8): Selects clock rate for measured-value transmitter *

Bit (29:24): Selects data word length 2 (in accordance with connected transmitter/encoder)**:
Allows defining the length of a data word to be received in both modes
"SSI" and "EnDat- Interface®" (receives position values with random programming).

Bit (31:30): Bit 31 = 0/1: Skips the turn-on procedure (500ms). The internal engine for encoder clock control will change to "EnDat-Interface®" mode if bit 30 = 0, and to "SSI" if bit30 = 1. This function is performed once on setting of bit 31.

* *For the desired division factor, a value of* $\frac{1}{2} * \frac{f_{clk}}{f_{tclk}}$ *needs to be set.*

(Expl: D(17:8) = 8/dec delivers a transmitter clock rate (TCLK) of 1 Mhz for fsys = 16 MHz)

** *For* "EnDat-Interface®" *mode, the desired data word length must be specified.*
(Expl.: D(29:24) = 011000/b for transferring a 24-bit data word).

For "SSI" mode, the desired data word length + 1 must be defined.
(Expl.: D(29:24) = 011001/b for transferring a 24-bit data word).

5.3.2 "Incremental" transfer mode

In "Incremental" transfer mode (bit(7)=1), a number of mode selections are required or can be made.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	**1**	Set-by-reference	Counter stop	Repeat Reset	Reset	Reference Compare	Reference strobe allowed	Hardware strobe allowed
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Byte 2	Filter	Filter	Filter	Filter	Filter	Add Offset	IPF1	IPF0
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Byte 3	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Strobe/ reference
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Byte 4	Reserved	Reserved	Data word length 2	Data word length 2	Data word length 2	Data word length 2	Data word length 2	Data word length 2

The following functions can be selected (bit x = 1):

Bit 0: Measured-value loading allowed.
On selection of timer or external strobe input /STR, a new value will only be loaded into the receive register if bit 0 = 1 and status register bit (0) = 0. A software strobe will, independently of the state of bit 0, cause the current (found to be faultless) value to be loaded into the receive register if status register bit (0) = 0.

Bit 1: Reference strobe is allowed (with CHA*CHB*CHC*CHD - edge-sensitive).

Bit 2: Reference compare (if value is found to be identical with that of the offset register, status register bit (2) will be set – edge-sensitive) *

Bit 3: One-time counter reset with CHA*CHB*CHC*CHD - edge-sensitive.

Bit 4: With bit 3 = 1: Repeated reset is allowed with each CHA*CHB*CHC*CHD - edge-sensitive.

Bit 5: Counter inputs are disabled.

Bit 6: Set-by-Reference enabled (with CHA*CHB*CHC*CHD) – edge-sensitive.

Bit (9:8): Edge evaluation for increment signals at CHA and CHB *

Bit 10: Adds an offset value to the strobe register value as part of reading

Bit (15:11): Filter setting **

Bit (29:24): Sets data word length 2 = counter width = 20/h ensures proper counter reading via receive register.

Note: If "Reset" and "Set-by-Ref" are simultaneously enabled, "Reset" will prevail.

** Following a reset, pulse direction mode is set as the default version. Count events are 0/1 transition edges at CHA. CHB=0 is defined as forward direction. .*

In edge evaluation mode, single, double and quadruple evaluation can be programmed via control register bit (9:8). Counting pulses are generated as shown in Table 2.

*** In order to allow the suppression of interference signals at the signal inputs CHA, CHB, CHD, CHD, easily programmable digital filters have been implemented. If a signal fails to reach a certain minimum required duration, it will be detected as an interference.*
Filter programming, i.e. definition of the minimum duration for a signal to be recognized as a useful signal, is accomplished with the help of control register bit (15:11). Given a system clock rate of 20 MHz, pulses lasting from 1.5 µs to 200 ns can be defined for suppression range.
Formula for setting the [WTnutz] value for a desired minimum useful signal duration [Tnutz]

$$[WTnutz] = (Tnutz / Tsys) - 1$$
or
$$[WTnutz] = (fsys / 2*fnutz) - 1$$

Tnutz - Mininimum required duration for a useful signal Tsys - Period of a system clock
fnutz - Maximum rate of a useful signal fsys - System clock rate

Example:
In order to detect only signals with a frequency ≤312.5 kHz as useful input signals when a system clock of 20 MHz is set, 1F/h must be written into the control register bits (15:11) for WTnutz.
Following the MMI4832 reset, all filters are turned off by default. Defining 0/dec value for control register (15:11) will turn the filters off.

While "EnDat-Interface®" and "*SSI*" modes are active, the encoder component will work with the standard quadruple edge evaluation. This makes it possible to switch to incremental mode without loss of data on completion of a transfer cycle of absolute position values. All reserved bits must be filled with '0'.

5.4 Status register

Base address: 5/h, RO (reading with reset), 8 bits, reset value: 0/h
Base address: 7/h, RO (reading without reset)

The status register informs about the various events that have occurred during a recording procedure of measured-values (D(x)=1).
Its content is different for "EnDat-Interface®" and incremental transfer mode.

5.4.1 "EnDat®", "SSI" transfer mode

The status register informs about events regarding the EnDat® measuring system and free interrupt inputs:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EnDat-Interface® /SSI	/IR7	/IR6	SSI	Parity error	CRC error	Address MRS error	Alarm	PV valid

Bit(0): Parallel value is valid if enabled in control register or on a software strobe
Bit(1): Alarm bit was set by measuring system if enabled in control register (only in "EnDat-Interface®")
Bit(2): Error on transferring MRS or address codes (memory select) to/from the measuring system if enabled in control register (only in EnDat-Interface®)
-> Acknowledged code is not identical with transmitted code
-> Acknowledged address is not identical with transmitted address
Bit(3): CRC error: error on testing a CRC code that was received from the measuring system if enabled in control register (only in "EnDat-Interface®")
Bit(4): Parity check if enabled in control register (only for SSI)

Bit(5): IC works in “SSI” transfer mode
Bit(6): Detects H/L-edge at input pin /IR6
Bit(7): Detects low level at input pin /IR7

The "Parallel value valid" bit is set at the end of a transfer process. However, if the timer or external strobe input /STR is involved, it will only be set – and, hence a new value (which was found to be faultless) only be loaded into the receive register – if bit (0) of the control register is “1“ and the status register had been reset to zero before. Independently of the state of control register bit (0), a software strobe will cause the current value (which was found to be faultless) to be loaded into the receive register if the status register bit (0) had been reset to zero before. Bits (1:4) in set (error!) state will in any case prevent the setting of the "Parallel value valid" bit.

5.4.2 “Incremental“ transfer mode

The status register informs about incremental measuring-system events and the free interrupt inputs:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Incremental	/IR7	/IR6	Spike, track errors	Repeat reset	Reset	Reference compare	Reference strobe	Strobe

Bit(0): Measured value is valid if enabled in control register or on software strobe
Bit(1): Performs reference strobe (with CHA*CHB*CHC*CHD – edge-sensitive)
Bit(2): Performs reference compare (if value found to be equal to the value of the offset register – edge-sensitive)
Bit(3): Performs reset (one-time counter reset (with CHA*CHB*CHC*CHD - edge-sensitive)
Bit(4): Performs repeat (with CHA*CHB*CHC*CHD – edge-sensitive)
Bit(5): Spike or track errors at CHA, CHB, CHC, CHD
Bit(6): Detects H/L-edge at input pin /IR6
Bit(7): Detects low level at input pin /IR7

If the timer or external strobe input /STR is used, status bit (0) will only be set and a new value only be loaded into the receive register if bit (0) of the control register is “1“ and the status register had been reset earlier.

Regardless of the state of control register bit (0), a software strobe will cause the current value (which was found to be faultless) to be loaded into the receive register, provided that bit (0) of the status register had been reset earlier.

5.4.3 /IR6, IR7 in all operating modes (“EnDat®“, “SSI“, “Incremental“)

Bits (6,7) inform about events at the two inputs /IR6 (H/L-edge-sensitive) and /IR7 (low active). These inputs are available for specific customer applications, for example, to monitor supply voltages, temperatures, to detect cable breakage, etc. On reading of the status register (base address: 5/h), its content is reset to 00/h.

5.5 Interrupt mask

Base address: 6/h, R/W, 8 bits, reset value: 0/h

The interrupt mask is intended to allow masking of the various sources that may cause an interrupt for the MMI4832. Each bit contained in the status register is able to create an interrupt. The bit assignments of the interrupt mask is identical with that of the status register. If a bit is set, the corresponding interrupt will be enabled (D(x)=1).

5.6 Timer

Base address: 7/h, WO, 16 bits, reset value: 0/h

The MMI4832 has its own internal timer. It is cpable of creating a programmable output frequency at pin /SRB. This frequency can be used as a repeat rate (sampling rate) for the measured values. The following table shows some selected sample frequencies.

Bit(15:0) /hex	Bit(15:0) /dec	Sampling rate
0	0	Stop
2	2	0.416 µs (2.4 MHz)
12C	300	50 µs (20 KHz)
1770	6000	1000 µs (1 KHz)
FFFF	65535	10.9 ms (91.7 Hz)

5.6.1 Transfer start in "EnDat®", "SSI" mode

A H/L-edge of /SRB will trigger the transfer clock for the encoder if control register bit (0) is enabled. In addition, the hardware strobe input /STR must be disabled (/STR = 1), because /STR and internal timer output are “or”-ed.

5.6.2 Setting a desired sampling rate

To set a desired sampling rate, you need to define a value [WTTF] = ¼ (f_{clk} / f_{Timer} -2)) in the timer preset register.
In other terms, the relationship between T_{timer} and T_{clk} = 4 * WTTF + 2.

6 Pin Description

Symbol	Type	MMI4832 (TQFP44)	Description
Function pins			
DATA_DV	O	40	Data drive (data sending to measured-value encoder)
D(0)	I/O	41	Data
D(1)	I/O	42	Data
D(2)	I/O	43	Data
D(3)	I/O	44	Data
D(4)	I/O	1	Data
D(5)	I/O	2	Data
D(6)	I/O	3	Data
D(7)	I/O	4	Data
D(8)	I/O	7	Data
D(9)	I/O	8	Data
D(10)	I/O	9	Data
D(11)	I/O	10	Data
D(12)	I/O	11	Data
D(13)	I/O	12	Data
D(14)	I/O	13	Data
D(15)	I/O	14	Data
/WR *R_/W*	I	15	Write request, low-active (Intel mode) *Switching between Read and Write (Motorola mode)*
/RD */ADS*	I	16	Read request, low-active *Address data strobe, low-active (Motorola mode)*
/CS	I	17	Chip select, low-active
DE	O	18	Drive enable (for external transceiver) *
CLK	I	20	System clock
DATA_RC	I	22	Data receive (data received from measured-value encoder)
/RES	I	23	Reset signal for MMI4832, low-active
/NULL	I	24	External counter reset (low-active)
CMD	I	25	Address select
MOT	I	26	Parallel port mode: MOT=0: “Intel”, MOT=1: “Motorola”
/IR6	I	27	Sensitive input for interrupt request, H/L-active
/IR7	I	28	Sensitive input for interrupt request, low-active
/STR_SYN	O	29	Test pin for MMI4832 manufacturer
NIA_TCLK	O	30	Transfer clock (“EnDat-Interface®” and *“SSI*) zero output (*incremental* mode)
/INT	O	31	Interrupt request (low/tristate)
/SRB	O	32	Timer output, low-active
/STR	I	35	Strobe input, H/L-active
CHA	I	36	Track A
CHB	I	37	Track B
CHC	I	38	Track C (index)
CHD	I	39	Track D (index)
System pins:			
V_{DD}	I	5, 19, 33	Operating voltage (5V)
V_{ss}	I	6, 21, 34	Earthground

Table 3: Pin assigments of MMI4832 circuit

Note:
Each pair of supply pins must be blocked with a 100nF ceramic multi-layer capacitor. We also recommend installing an electrolytic capacitor (220µF) near the MMI4832.
DE=1: IC sends data (via DATA_DV) to a measured-value encoder.

7 Package

Figure 7-1 shows the package of the MMI4832.

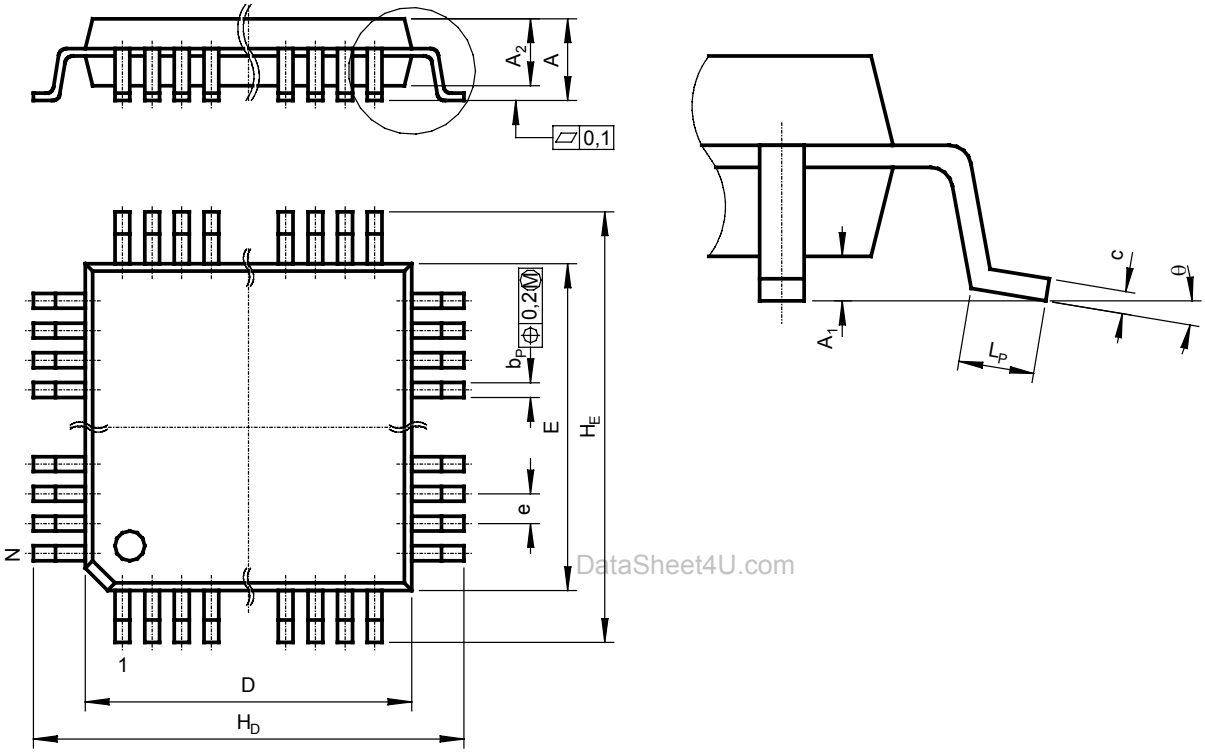

Figure 7-1: Designation of package: TQFP44

Table 4 contains all essential package dimensions:

TQFP-44					
Description	**Symbol**	**Min.**	**Nom.**	**Max.**	**Unit of measure**
Pin number	N	-	44	-	-
Pitch	e		0.80		mm
Overall height	A			1.60	mm
Pin width	b_P	0.30		0.45	mm
Pin length	L_P	0.45			mm
Side length, total	H_D	11.85		12.15	mm
Side length, total	H_E	11.85		12.15	mm
Package spacing	A_1	0.05		0.15	mm
Package thickness	A_2	1.35		1.45	mm
Pin thickness	c	0.09		0.20	mm
Side length, body	D	9.75		10.25	mm
Side length, body	E	9.75		10.25	mm
Pin inclination	θ	0		7	°
Ground	m			0.5	g
Package material		Plastic materials			
Pin materials		FeNi-alloy, electro-tin-platedt			
Pin shape		Z-shaped			

Table 4: Package dimensions

8 General Operating Conditions

Table 5 provides a summary of the MMI4832's general operating conditions:

Characteristic values (nominal: Ucc=5V)	**Symbol**	**Min.**	**Max.**	**Unit**	**Comment**
Supply voltage	Ucc	4.75	5.25	V	
Input voltage for H-level (CMOS)	UIH	0.7 * U_{CC}	U_{CC} + 0.5	V	
Input voltage for L-level (CMOS)	UIL	-0.5	0.3 * U_{CC}	V	
L-output voltage	UOL		0.4	V	IOL = 4mA
H-output voltage	UOH	2.4		V	IOH = 4mA
Operating temperature	Tamb	0	70	°C	
System clock rate	fclk	-	33	MHz	Incremental mode
System clock rate	fclk	Typically: 24		MHz	EnDat mode
ESD strength			2	kV	Human body model

Table 5: General conditions for MMI4832 operation

9 Time Relationships

- (general description of parallel port) -

Parallel port mode: “Intel”

Read cycle:

Symbol		Parameter	Min. [ns]	Max. [ns]
T1		Address-valid-to-RD-active	20	---
T2		RD-active-to-data-valid	---	$1/f_{clk}+20$
T3		Data-float-after-RD	---	35
T4		RD-pulse-width	$1/f_{clk}+20$	---
T5		Address-float-after-RD	0	---
T6		Recover	$1/f_{clk}$	---

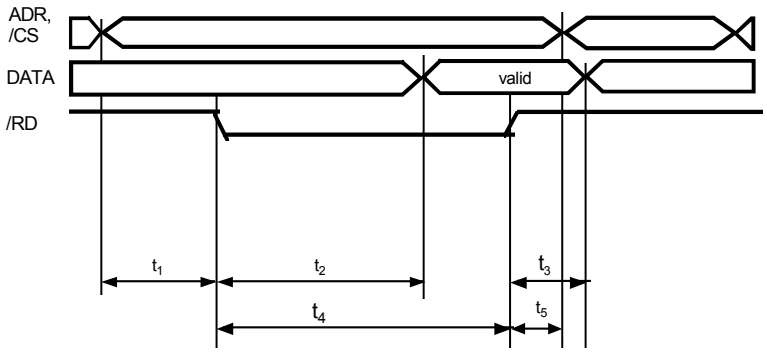

Figure 9-1: Signal diagram for PPM Intel read cycle

Write cycle:

Symbol		Parameter	Min. [ns]	Max. [ns]
T1		Address-valid-to-WR-active	20	---
T2		Data-valid-to-WR-transition	$1/f_{clk}+20$	---
T3		Data-hold-after-WR	0	---
T4		WR-pulse-width	$1/f_{clk}+20$	---
T5		Address-hold-after-WR	0	---
T6		Recover	$1/f_{clk}$	---

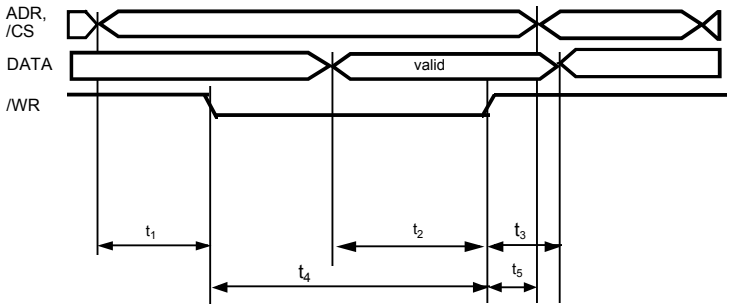

Figure 9-2: Signal diagram for PPM Intel write cycle

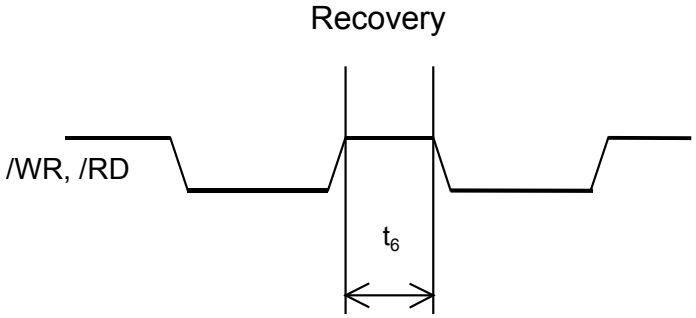

Figure 9-3: Signal diagram for PPM Intel recovery

Parallel port mode: “Motorola”

Read cycle:

Symbol		Parameter	Min. [ns]	Max. [ns]
t1		Address-valid-to-/ADS -active	20	---
t2		Data-valid-to-/ADS-transition	---	$1/f_{clk}+20$
t3		Data-hold-after-/ADS	---	35
t4		/ADS -pulse-width	$1/f_{clk}+20$	---
t5		Address-hold-after-/ADS	0	---
t6		Recover	$1/f_{clk}$	---
t7		R_/W-valid to /ADS asserted	0	---
t8		/ADS negated to R_/W-invalid	0	---

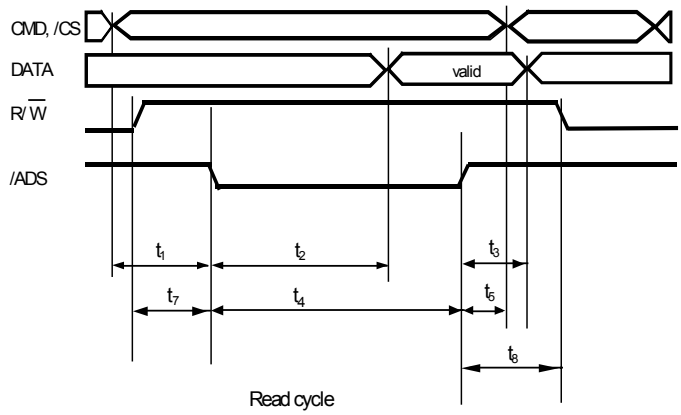

Figure 9-4: Signal diagram for PPM Motorola read cycle

Write cycle:

Symbol		Parameter	Min. [ns]	Max. [ns]
T1		Address-valid-to-/ADS-active	20	---
T2		Data-valid-to-/ADS-transition	$1/f_{clk}+20$	---
T3		Data-hold-after-/ADS	0	---
T4		/ADS -pulse-width	$1/f_{clk}+20$	---
T5		Address-hold-after-/ADS	0	---
T6		Recover	$1/f_{clk}$	---
T7		R_/W-valid to /ADS asserted	0	---
T8		/ADS negated to R_/W-invalid	0	---

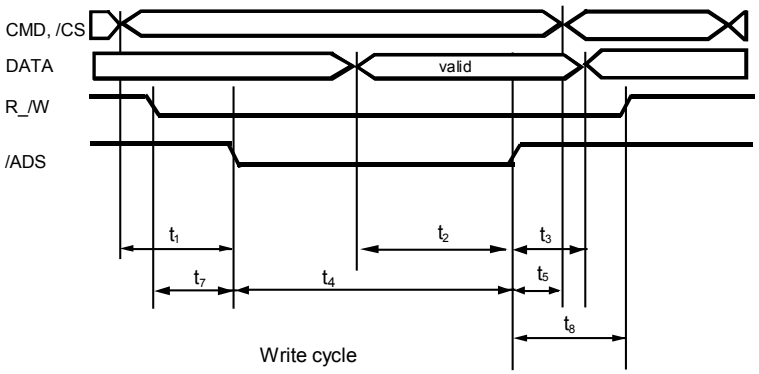

Figure 9-5: Signal diagram for PPM Motorola write cycle

Recovery:

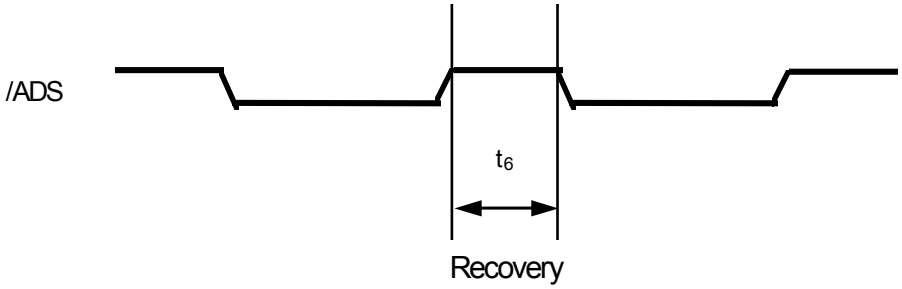

Figure 9-6: Signal diagram for PPM Motorola recovery

Appendix

MMI 4832

Table of Contents

1 µC Port Addressation

1.1 Intel, no autoincrement

A(4:2)	A(1:0)	Resource	16-bit write port	8-bit port	16-bit read port	8-bit port
		Parallel-serial converter	*Sets send register*		*Reads send register*	
0	0		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	1		Sets	D(7:15)	Reads	D(7:15)
	2		Sets D(16:31)	D(16:23)	Reads D(16:31)	D(16:23)
	3		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Receive register	*Loads serial-parallel converter value*		*Reads receive register, strobe RG*	
1	0		*into receive register*		Reads D(0:15)	D(0:7)
	1		---	---	Reads	D(7:15)
	2		---	---	Reads D(16:31)	D(16:23)
	3		---	---	Reads	D(24:31)
			1	1		
		Receive register			*Reads receive register, strobe RG*	
2	0		---	---	Reads D(32:47)	D(32:39)
	1		---	---	Reads	D(40:47)
					1 - 3 **	2 - 6 **
		Multifunction register (MFR)	*Sets MFR*		*Reads MFR*	
3	0		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	1		Sets	D(7:15)	Reads	D(7:15)
	2		Sets D(16:31)	D(16:23)	Reads D(16:31)	D(16:23)
	3		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Control register	*Sets control register*		*Reads control register*	
4	0		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	1		Sets	D(7:15)	Reads	D(7:15)
	2		Sets D(16:31)	D(16:23)	Reads D(16:31)	D(16:23)
	3		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Status register	*Software strobe*		*Reads status register with*	Reset
5	0		---		Reads D(0:7)	D(0:7)
			1	1	1	1
		Interrupt mask	*Sets interrupt mask*		*Reads iterrupt mask*	
6	0		Sets D(0:7)	D(0:7)	Reads D(0:7)	D(0:7)
			1	1	1	1
		Timer + status	*Sets timer register*		*Reads status register wihout*	Reset
7	0		Sets D(0:15)	D(0:7)	Reads D(0:7)	D(0:7)
	1		Sets	D(7:15)	---	---
	2		----	---	---	---
	3		----	---	---	---
			1	2	1	1

Table 6: Parallel port mode “Intel”: no autoincrement (MOT= low, AINC = low)

1.2 Intel, with autoincrement

A(4:2)	/WR-, /RD access	Resource	16-bit write port	8-bit port	16-bit read port	8-bit port
		Send register	*Sets send register*		*Reads send register*	
0	1		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	2		Sets D(16:31)	D(7:15)	Reads D(16:31)	D(7:15)
	3		Sets	D(16:23)	Reads	D(16:23)
	4		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Receive register	*Loads value of serial-parallel coverter*		*Reads receive register, strobe RG*	**
1	1		*into receive register*		Reads D(0:15)	D(0:7)
	2		---	---	Reads D(16:31)	D(7:15)
	3		---	---	Reads D(32:47)	D(16:23)
	4		---	---	Reads	D(24:31)
	5		---	---	Reads	D(32:39)
	6		---	---	Reads	D(40:47)
			1	1	1 - 3 **	2 - 6 **
2	---		---	---	---	---
		Multifunction register (MFR)	*Sets MFR*		*Reads MFR*	
3	1		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	2		Sets D(16:31)	D(7:15)	Reads D(16:31)	D(7:15)
	3		Sets	D(16:23)	Reads	D(16:23)
	4		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Control register	*Sets control register*		*Reads control register*	
4	1		Sets D(0:15)	D(0:7)	Reads D(0:15)	D(0:7)
	2		Sets D(16:31)	D(7:15)	Reads D(16:31)	D(7:15)
	3		Sets	D(16:23)	Reads	D(16:23)
	4		Sets	D(24:31)	Reads	D(24:31)
			2	4	2	4
		Status register	*Software strobe*		*Reads status register with*	Reset
5	1		---	---	Reads D(0:7)	D(0:7)
			1	1	1	1
		Interrupt mask	*Sets interrupt mask*		*Reads interrupt mask*	
6	1		Sets D(0:7)	D(0:7)	Reads D(0:7)	D(0:7)
			1	1	1	1
		Timer + status	*Sets timer register*		*Reads status register without*	Reset
7	1		Sets D(0:15)	D(0:7)	Lesen D(0:7)	D(0:7)
	2		Sets	D(7:15)	---	---
	3		----	---	---	---
	4		----	---	---	---
			1	2	1	1

Table 7: Parallel port mode “Intel”: autoincrement (MOT= low, AINC = high)

** *The number of required access cycles is determined by the width that has been preset for the data word to be received (control register (bit 24:29)).*

1.3 Motorola, no autoincrement

A(4:2)	A(1:0)	Resource	16-bit write port	8-bit port	16-bit read port	8-bit port
		Send register	*Sets send register*		*Reads send register*	
0	0		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	1		Sets	D(16:23)	Reads	D(16:23)
	2		Sets D(0:15)	D(7:15)	Reads D(0:15)	D(7:15)
	3		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
		Receive register	*Loads value of serial-parallel converter*		*Reads receive register, strobe RG*	
1	0		*into receive register*		Reads D(32:47)	D(40:47)
	1		---	---	Reads	D(32:39)
	2		---	---	Reads D(16:31)	D(24:31)
	3		---	---	Reads	D(16:23)
			1	1		
		Receive register			*Reads receive register, strobe RG*	***
2	0		---	---	Reads D(0:15)	D(7:15)
	1		---	---	Reads	D(0:7)
					1 - 3 **	2 - 6 **
		Multifunction register (MFR)	*Sets MFR*		*Reads MFR*	
3	0		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	1		Sets	D(16:23)	Reads	D(16:23)
	2		Sets D(0:15)	D(7:15)	Reads D(0:15)	D(7:15)
	3		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
		Control register	*Sets control register*		*Reads control register*	
4	0		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	1		Sets	D(16:23)	Reads	D(16:23)
	2		Sets D(0:15)	D(7:15)	Reads D(0:15)	D(7:15)
	3		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
		Status register	*Software strobe*		*Reads status register with*	Reset
5	0		---		Reads D(0:7)	D(0:7)
			1	1	1	1
		Interrupt mask	*Sets interrupt mask*		*Reads interrupt mask*	
6	0		Sets D(0:7)	D(0:7)	Reads D(0:7)	D(0:7)
			1	1	1	1
		Timer + status	*Sets timer register*		*Reads status register without*	Reset
7	0		Sets D(0:15)	D(7:15)	Reads D(0:7)	D(0:7)
	1		Sets	D(0:7)	---	---
	2		----	---	---	---
	3		----	---	---	---
			1	2	1	1

Table 8: Parallel-port mode “Motorola”: no autoincrement (MOT = high, AINC = low)

Reads receive register by the example of a 24-bit long data word:

A(4:2)	A(1:0)	Resource	16-bit write port	8-bit port	16-bit read port	8-bit port
		Receive register	*Loads value of serial-parallel converter*		*Reads receive register, strobe RG*	
1	1		*into receive register*		Reads D(16:23)	D(16:23)
	2		---	---	Reads D(0:15)	D(7:15)
	3		---	---	Reads	D(0:7)
			1	1	2	3

1.4 Motorola, with autoincrement

A(4:2)	Access	Resource	16-bit write port	8-bit port	16-bit read port	8-bit port
		Send register	*Sets send register*		*Reads send register*	
0	1		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	2		Sets D(0:15)	D(16:23)	Reads D(0:15)	D(16:23)
	3		Sets	D(7:15)	Reads	D(7:15)
	4		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
		Receive register	*Loads value of serial-parallel converter*		*Reads receive register, strobe RG*	
1	1		*into receive register*		Reads D(32:47)	D(40:47)
	2		---	---	Reads D(16:31)	D(32:39)
	3		---	---	Reads D(0:15)	D(24:31)
	4		---	---	Reads	D(16:23)
	5		---	---	Reads	D(7:15)
	6		---	---	Reads	D(0:7)
			1	1	1 - 3 **	2 - 6 **
2	---	---	---	---	---	---
		Multifunction register (MFR)	*Sets MFR*		*Reads MFR*	
3	1		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	2		Sets D(0:15)	D(16:23)	Reads D(0:15)	D(16:23)
	3		Sets	D(7:15)	Reads	D(7:15)
	4		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
		Control register	*Sets control register*		*Reads control register*	
4	1		Sets D(16:31)	D(24:31)	Reads D(16:31)	D(24:31)
	2		Sets D(0:15)	D(16:23)	Reads D(0:15)	D(16:23)
	3		Sets	D(7:15)	Reads	D(7:15)
	4		Sets	D(0:7)	Reads	D(0:7)
			2	4	2	4
5		Status register	*Software strobe*		*Reads status register with*	Reset
	1		---	---	Reads D(0:7)	D(0:7)
			1	1	1	1
6		Interrupt mask	*Sets interrupt mask*		*Reads interrupt mask*	
	1		Sets D(0:7)	D(0:7)	Reads D(0:7)	D(0:7)
			1	1	1	1
7		Timer + status	*Sets timer register*		*Reads status register without*	Reset
	1		Sets D(0:15)	D(7:15)	Reads D(0:7)	D(0:7)
	2		Sets	D(0:7)	---	---
	3		----	---	---	---
	4		----	---	---	---
			1	2	1	1

Table 9: Parallel-port mode “Motorola”: autoincrement (MOT= high, AINC = high)

** *The number of required access cycles is determined by the width that has been preset for the data word to be received (control register (bit 24:29)).*
Working in "Motorola" or "Autoincrement" mode (MOT=1 and AINC=1 resp.), the MMI4832 will at first read or write the most significant byte or word, whereas the least significant byte or word is read or written after "n" access cycles have been completed (depending on word length).

Example: 24-bit long data word being received

A(4:2)	Access	Resource Receive register	16-bit write port *Loads value of serial-parallel converter*	8-bit port	16-bit read port *Reads receive register, strobe RG*	8-bit port
1	1		*into receive register*		Reads D(16:23)	D(16:23)
	2		---	---	Reads D(0:15)	D(7:15)
	3		---	---	Reads	D(0:7)
			1	1	2	3

2 Application Examples

2.1 ISA

Figure 2-1 shows a block diagram for an MMI4832 applied as an interface for measuring systems which are equipped with EnDat-Interface® and ISA bus.

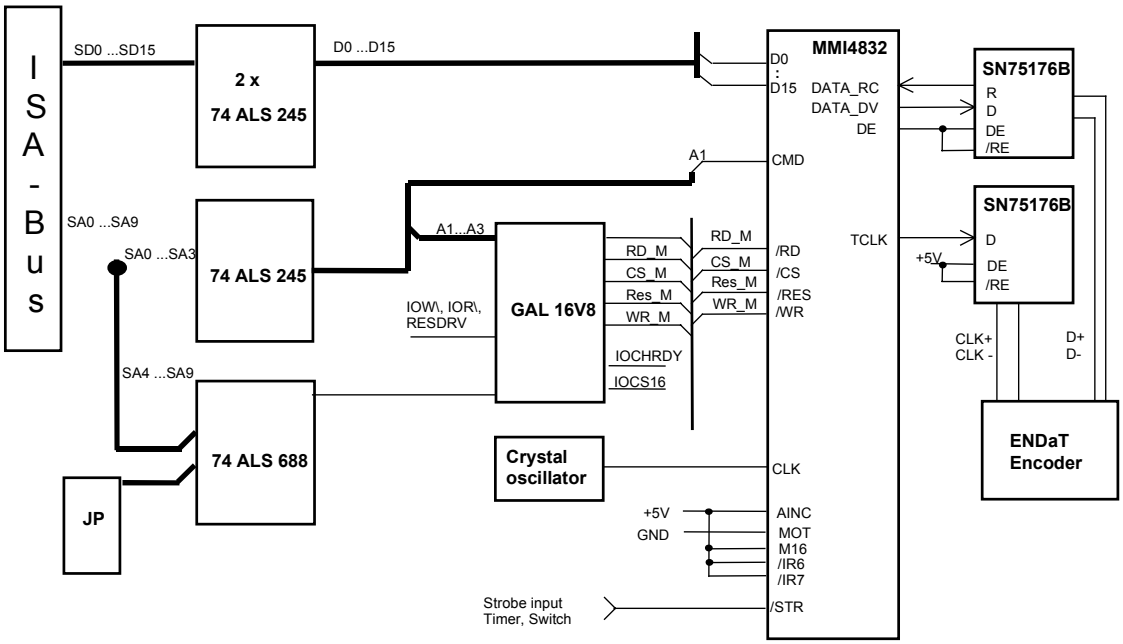

Figure 2-1: Application example: MMI4832 with EnDat-Interface®, measuring system and ISA bus

2.2 Transceivers

Figure 2-2 shows a recommended transceiver wiring interconnections diagram.

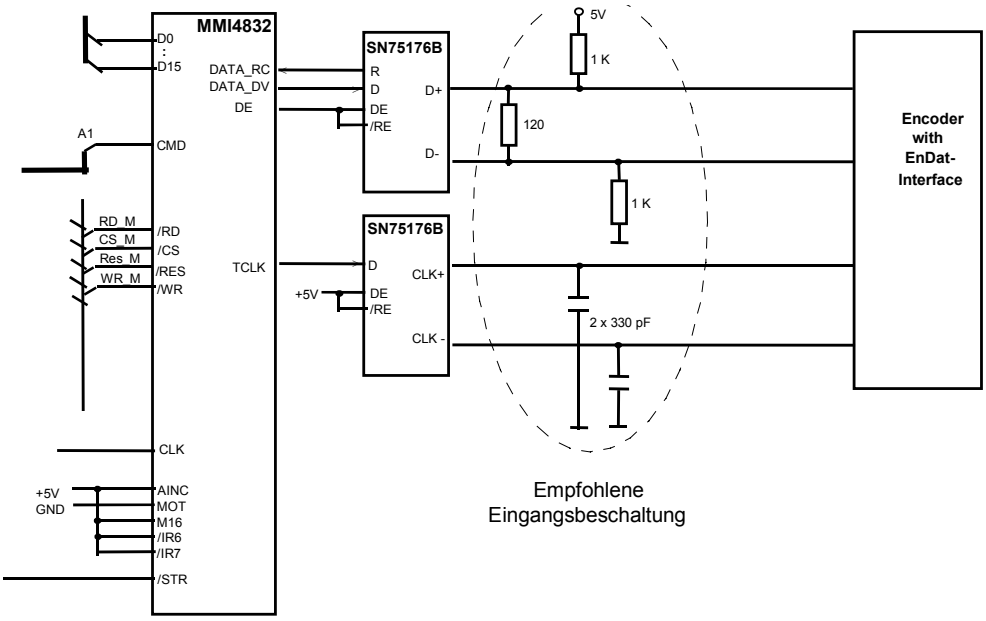

Figure 2-2: Wiring interconnections for transceivers

3 Programming Examples

3.1 Reading of parameter “Data word length of encoder“ (EnDat mode)

File Name: ROC413PL.C

```
#include <stdio.h>
#include <conio.h>

int krg_rg_l;
int krg_rg_h;
int sen_rg_l;
int sen_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;

/* Status clearing */

outpw (0x302,0x74);
sta_rg=inpw (0x300);
```

/* Control RG */

```
outpw (0x302,0x70);
outpw (0x300,0x3201);
outpw (0x300,0x0000);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);
```

/* Send RG – Selection of encoder memory range */

```
outpw (0x302,0x60);
outpw (0x300,0x0000);
outpw (0x300,0x0E00);

outpw (0x302,0x60);

sen_rg_l=inpw (0x300);
sen_rg_h=inpw (0x300);

printf ("SEN :  %4x", sen_rg_h);
printf (" %4x\n", sen_rg_l);
```


/* Software strobe (start of 1st transfer) */

```
outpw (0x302,0x74);
outpw (0x300,0x0000);

for (i=0; i<20000; i+=1)
printf ("wait\r");
```

/* Status clearing */

```
outpw (0x302,0x74);
sta_rg=inpw (0x300);
```

/* Send RG – Selection of parameter address (inquiry about encoder bit length) */

```
outpw (0x302,0x60);
outpw (0x300,0x0000);
outpw (0x300,0x2300);

outpw (0x302,0x60);

sen_rg_l=inpw (0x300);
sen_rg_h=inpw (0x300);

printf ("SEN :  %4x", sen_rg_h);
printf (" %4x\n", sen_rg_l);
```

```
/* Software strobe (start of 2nd transfer) */

outpw (0x302,0x74);
outpw (0x300,0x0000);

for (i=0; i<20000; i+=1)
printf ("wait\r");

/* Reading of receive RG (received value is encoder bit length) */

  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);

  outpw (0x302,0x74);
  sta_rg=inpw (0x300);

  printf ("STA: %4x  ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x\r", empf_rg_l);

}
```

3.2 Position value transfer with 13-bit encoder ROC413 (EnDat mode)

File Name: ROC413_M.C


```
#include <stdio.h>
#include <conio.h>

int krg_rg_l;
int krg_rg_h;
int sen_rg_l;
int sen_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;

/* Status clearing */

outpw (0x302,0x74);
sta_rg=inpw (0x300);
```

```
/* Control RG */

outpw (0x302,0x70);
outpw (0x300,0x3201);
outpw (0x300,0x0d00);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);

/* Send RG */

outpw (0x302,0x60);
outpw (0x300,0x0000);
outpw (0x300,0x0700);

outpw (0x302,0x60);

sen_rg_l=inpw (0x300);
sen_rg_h=inpw (0x300);

printf ("SEN :  %4x", sen_rg_h);
printf (" %4x\n", sen_rg_l);


/* Timer start */

outpw (0x302,0x7c);
outpw (0x300,0x800);


/* Receive RG */

while (!kbhit())
   {
  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);

  outpw (0x302,0x74);
  sta_rg=inpw (0x300);

  outpw (0x302,0x70);
  krg_rg_l=inpw (0x300);
  krg_rg_h=inpw (0x300);

  printf ("KRG :  %4x", krg_rg_h);
  printf (" %4x ", krg_rg_l);
  printf ("STA: %4x  ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x\r", empf_rg_l);
  }
}
```

3.3 Position value transfer with 25-bit encoder EQN1325 (EnDat mode)

File Name: EQN1325M.C

```
#include <stdio.h>
#include <conio.h>

int krg_rg_l;
int krg_rg_h;
int sen_rg_l;
int sen_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;

/* Status clearing */

outpw (0x302,0x74);
sta_rg=inpw (0x300);

/* Control RG */

outpw (0x302,0x70);
outpw (0x300,0x3201);
outpw (0x300,0x1900);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);

/* Send RG */

outpw (0x302,0x60);
outpw (0x300,0x0000);
outpw (0x300,0x0700);

outpw (0x302,0x60);

sen_rg_l=inpw (0x300);
sen_rg_h=inpw (0x300);

printf ("SEN :  %4x", sen_rg_h);
printf (" %4x\n", sen_rg_l);
```

```
/* Timer start */

outpw (0x302,0x7c);
outpw (0x300,0x800);


/* Receive RG */

while (!kbhit())
   {
  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);

  outpw (0x302,0x74);
  sta_rg=inpw (0x300);

  outpw (0x302,0x70);
  krg_rg_l=inpw (0x300);
  krg_rg_h=inpw (0x300);

  printf ("KRG :  %4x", krg_rg_h);
  printf (" %4x ", krg_rg_l);
  printf ("STA: %4x  ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x\r", empf_rg_l);
  }
}
```


3.4 Incremental measured-value transfer with hardware strobe

File Name: HW_STR.C

```
#include <stdio.h>
#include <conio.h>

int krg_rg_l;
int krg_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;


/* Clearing of status RG */

outpw (0x302,0x74);
sta_rg=inpw (0x300);
```

```
/* Control RG*/

outpw (0x302,0x70);
outpw (0x300,0x0381);
outpw (0x300,0x2000);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);


/* Timer start */

outpw (0x302,0x7C);
outpw (0x300,0x0800);


/* Receive RG */

while (!kbhit())
  {
  /* Reading of receive RG (CT content is read via receive RG) */
  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);
  outpw (0x302,0x74);
  sta_rg=inpw (0x300);

  printf ("STA: %4x  ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x\r", empf_rg_l);
  }
}
```


3.5 Incremental measured-value transfer with reference strobe

File Name: REF_STR.C

```
#include <stdio.h>
#include <conio.h>

int krg_rg_l;
int krg_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;
```

/* Control RG */

```
outpw (0x302,0x70);
outpw (0x300,0x0382);
outpw (0x300,0x2001);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);
```

/* Receive RG */

```
while (!kbhit())
  {
  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);
  outpw (0x302,0x7c);
  sta_rg=inpw (0x300);

  printf ("STA: %4x  ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x\r", empf_rg_l);
  }
}
```


3.6 Incremental measured-value transfer with add offset

File Name: ADD_OFF.C

```
#include <stdio.h>
#include <conio.h>

int pre_rg_l;
int pre_rg_h;
int ref_rg_l;
int ref_rg_h;
int krg_rg_l;
int krg_rg_h;
int empf_rg_l;
int empf_rg_h;
int sta_rg;

main ()
{
int i;
```

```
/* Clearing of status RG */

outpw (0x302,0x74);
sta_rg=inpw (0x300);


/* Clearing old counter value (counter+receive RG) */

outpw (0x302,0x60);
outpw (0x300,0x0000);
outpw (0x300,0x0000);

outpw (0x302,0x64);
outpw (0x300,0x0000);

outpw (0x302,0x74);
outpw (0x300,0x0000);


/* Control RG*/
/* Add-offset bit set for ref. RG to be readable */
/* -> otherwise test bits contained under this address (e.g. eclk Automat) */

outpw (0x302,0x70);
outpw (0x300,0x0584);
outpw (0x300,0x2000);

outpw (0x302,0x70);
krg_rg_l=inpw (0x300);
krg_rg_h=inpw (0x300);

printf ("KRG :  %4x", krg_rg_h);
printf (" %4x\n", krg_rg_l);


/* Offset RG */

outpw (0x302,0x6C);
outpw (0x300,0x4444);
outpw (0x300,0x3333);

outpw (0x302,0x6C);
ref_rg_l=inpw (0x300);
ref_rg_h=inpw (0x300);

printf ("OFF :  %4x", ref_rg_h);
printf (" %4x\n", ref_rg_l);
```

```
/* Preset RG */

outpw (0x302,0x60);
outpw (0x300,0x3333);
outpw (0x300,0xcccc);

outpw (0x302,0x60);
pre_rg_l=inpw (0x300);
pre_rg_h=inpw (0x300);

printf ("PRE :  %4x", pre_rg_h);
printf (" %4x\n", pre_rg_l);


/* Loading Preset into counter */

outpw (0x302,0x64);
outpw (0x300,0x0000);


/* Receive RG */

while (!kbhit())
  {

  /* SW strobe counter -> STR-RG */
  outpw (0x302,0x74);
  outpw (0x300,0x0000);

  /* Reading receive RG (CT content is read via receive-RG) */
  outpw (0x302,0x64);
  empf_rg_l=inpw (0x300);
  empf_rg_h=inpw (0x300);
  outpw (0x302,0x6C);
  ref_rg_l=inpw (0x300);
  ref_rg_h=inpw (0x300);
  outpw (0x302,0x7C);
  sta_rg=inpw (0x300);

  printf ("STA: %4x   ", sta_rg);
  printf ("EMPF:  %4x", empf_rg_h);
  printf (" %4x      ", empf_rg_l);
  printf ("EMPF + OFF:  %4x", ref_rg_h);
  printf (" %4x\r", ref_rg_l);
  }
}
```

3.7 Measured-value transfer with SSI

```
#include <stdio.h>
#include <conio.h>

int krg_rg_l1;
int krg_rg_h1;
int sen_rg_l1;
int sen_rg_h1;
int empf_rg1;
int sta_rg1;
int int_rg;

main ()
{
int i;


/* Clearing of status RG */
outpw (0x302,0x74);  /* Kanal1 */
sta_rg1=inpw (0x300);


/* Control RG channel1 */

outpw (0x302,0x70);
outpw (0x300,0x3201);
outpw (0x300,0x4d00);

outpw (0x302,0x70);
krg_rg_l1=inpw (0x300);
krg_rg_h1=inpw (0x300);

printf ("KRG1 :  %4x", krg_rg_h1);
printf (" %4x\n", krg_rg_l1);


/* Timer start */
outpw (0x302,0x7c);
outpw (0x300,0x1000);


/* Receive RG */

while (!kbhit())
   {
  outpw (0x302,0x64);
  empf_rg1=inpw (0x300);

  outpw (0x302,0x74);
  sta_rg1=inpw (0x300);

  outpw (0x302,0x70);
  krg_rg_l1=inpw (0x300);
  krg_rg_h1=inpw (0x300);
```

```
    printf ("KRG1: %4x", krg_rg_h1);
    printf (" %4x ", krg_rg_l1);
    printf ("STA1: %2x", sta_rg1);
    printf (" Empf1: %4x \r", empf_rg1);
    }
}
```

3.8 Setting of EnDat transfer rate

Table 10 shows the control register (17:8) values to be selected for recommended EnDat transfer rates:

Bit(17:8) /hex	Bit(17:8) /dec	Sampling rate
0	0	Inactive
3	3	2 MHz
6	6	1 MHz
3C	60	200 KHz
78	120	100 KHz

Table 10 : EnDat transfer rates

4 FAQ‘s

For further information and answers to frequently asked questions (FAQ’s), you are referred to MAZeT’s Websites at:

http://www.mazet.de

We will also be glad to send you supporting material or provide answers on special request.

For further information, pleas feel free to contact:

MAZeT GmbH
Sales office:
Dr. Winfried Mahler
Göschwitzer Straße 32
07745 JENA
GERMANY
Phone: +49 3641 2809-0
Fax: +49 3641 2809-12
E-Mail: mahler@MAZeT.de
Url: http://www.MAZeT.de