TDA9951

CEC/I^{2}C-bus translator

Rev. 01 — 7 August 2008

Product data sheet

1. General description

The TDA9951 is a single-chip Consumer Electronics Control (CEC) to I^{2}C-bus translator dedicated to the control and interfacing Consumer Electronics products. The built-in processor simplifies Consumer Electronics (CE) product CPU design by managing a range of interfacing and control functions including the CEC protocol, timings and interrupts. Designed as an I^{2}C-bus slave device the TDA9951 enables control of these features to any I^{2}C-bus master device such as the CE host processor.

This data sheet describes the I^{2}C-bus interface, together with the control and management features of the TDA9951. The TDA9951 is an enhancement of the TDA9950.

2. Features

- CEC support:
  - Receive and transmit messages using compliant signal free time handling
  - Up to 16 bytes message length
  - Multiple logical addresses
  - Comprehensive arbitration and collision handling
- Dedicated processor control of CEC-line and I^{2}C-bus interface utilizing embedded software
- I^{2}C-bus interface to host communication in Standard mode (100 kbit/s) and Fast mode (400 kbit/s)
- Automatic Idle mode reduces power consumption if no messages are on CEC-line and I^{2}C-bus plus
- Managed Standby and Wake-up power modes
- Active LOW reset input and on-chip Power-On Reset (POR) enables operation without external reset components
- Reset counter and reset glitch circuitry prevents false and incomplete resets
- Programmable on-chip retry counter
- Controls specific Vacuum Fluorescent Display (VFD) devices and/or up to four LEDs
- Decode up to 10 panel switches
- Decode infrared protocol RC5, RC5 enlarged and RC6 Mode 0
- Battery operation detection maintains clock and calendar in low power mode
- Provides real-time clock features including time-of-day alarm and periodic timer
- V_{DD} operating range 3.0 V to 3.6 V
- 5 V tolerant input/output pins
- On-chip oscillator for a 12 MHz crystal
- Schmitt trigger port inputs

founded by Philips

3. Applications

- All devices using an HDMI connector
- YCbCr or RGB high-speed video digitizer
- Projector, plasma and LCD TV
- Rear-projection TV
- High-end TV
- Home-theater amplifier
- DVD recorder

4. Quick reference data

Table 1. Quick reference data

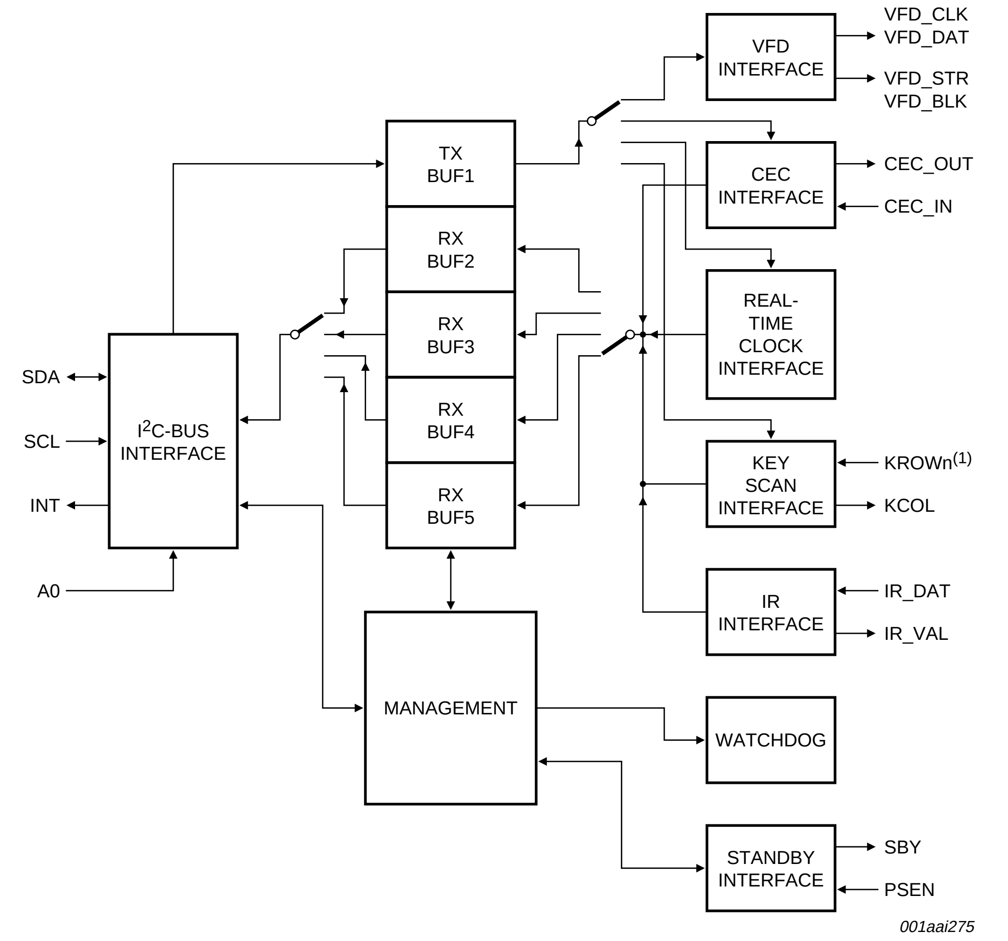
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	supply voltage		2.4	3.0	3.6	V
T_{amb}	ambient temperature		–40	-	+85	°C
P_{tot}	total power dissipation	based on package heat transfer, not device power consumption	-	-	1.5	W
I^{2}C-bus: pins SDA and SCL; 5 V tolerant						
f_{clk}	clock frequency	Standard mode	-	-	100	kHz
		Fast mode	-	-	400	kHz

5. Ordering information

Table 2. Ordering information

Type number	Package		
	Name	Description	Version
TDA9951TT	TSSOP28	plastic thin shrink small outline package; 28 leads; body width 4.4 mm	SOT361-1

6. Block diagram



(1) Where n is 0, 1, 2, 3 and 4.

Fig 1. Block diagram

7. Pinning information

7.1 Pinning

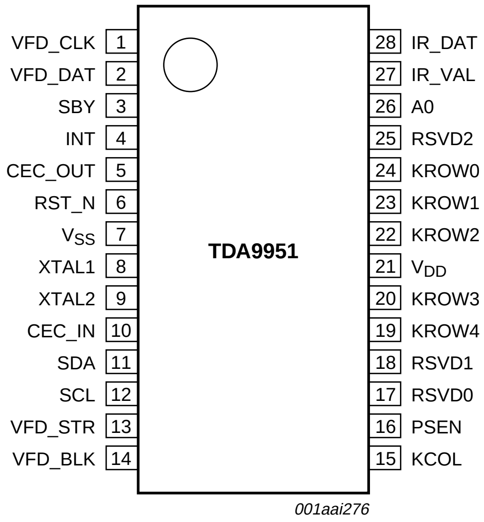


Fig 2. Pin configuration

7.2 Pin description

Table 3. Pin description

Symbol	Pin	Type[1]	Description
VFD_CLK	1	O	pulsed HIGH to clock output of the VFD display driver
VFD_DAT	2	O	data output to the VFD display driver
SBY	3	O	power control output to the host processor. Operating mode = LOW-level Standby mode = HIGH-level
INT	4	O	interrupt line to the host processor. Indicates data is available for reading. The polarity of operation is configured using the common configuration register; default is INT_POL bit = 1, active-HIGH
CEC_OUT	5	O	output to the CEC interface[2]
RST_N	6	I	external reset input. Holding this input LOW resets the TDA9951. It must be held LOW for a time after power-up when using an 18 MHz crystal[2].
V_{SS}	7	GND	ground; 0 V reference
XTAL1	8	I	input to the oscillator and internal clock generator circuits (18 MHz crystal)[2]
XTAL2	9	O	output from the oscillator amplifier
CEC_IN	10	I	input from the CEC interface[2]
SDA	11	I/O	I^{2}C-bus serial data input/output[2]
SCL	12	I	I^{2}C-bus serial clock input[2]

Table 3. Pin description *…continued*

Symbol	Pin	Type[1]	Description
VFD_STR	13	O	VFD strobe line. Pulsed HIGH latches clocked output at the VFD driver
VFD_BLK	14	O	VFD blank line. Default HIGH blanks output at the VFD display driver. The polarity is configurable using the I^{2}C-bus
KCOL	15	O	key matrix column output[2]

PSEN	16	I	power sense input detects power supply type.
			LOW-level = battery
			HIGH-level = main power

RSVD0	17	I	reserved pin; connect to ground
RSVD1	18	O	reserved pin
KROW4	19	I	key matrix row input[2]
KROW3	20	I	key matrix row input[2]
V_{DD}	21	P	power supply voltage
KROW2	22	I	key matrix row input[2]
KROW1	23	I	key matrix row input[2]
KROW0	24	I	key matrix row input[2]
RSVD2	25	O	reserved pin
A0	26	I	I^{2}C-bus slave address bit. This pin configures the least significant bit A0 of the I^{2}C-bus slave address; connect to: V_{DD} (HIGH) sets A0 to logic 1 (address 35h) V_{SS} (LOW) sets A0 to logic 0 (address 34h)
IR_VAL	27	O	pulse LOW for 65 ms after every valid IR frame is received, to indicate activity using an LED

IR_DAT	28	I	input from external infrared demodulator.
			LOW-level = active IR pulse,
			HIGH-level = no IR pulse (a space)

[1] P = power supply, I = input, O = output and I/O = Input and Output.

[2] See Figure 15 “Application diagram” on page 35.

8. Functional description

The TDA9951 controls the interface between the CEC-line and the I^{2}C-bus using its internal processor and embedded software.

8.1 Device addressing

The SDA and SCL pins are managed by the I^{2}C-bus peripheral which automatically communicates in Standard mode (100 kbit/s) or Fast mode (400 kbit/s) as an I^{2}C-bus slave.

The TDA9951 signals the host processor that data is ready by asserting the INT output. The polarity is configurable using the INT_POL bit in the CCONR register (see Table 14).

The seven-bit I^{2}C-bus slave address is hard-coded as 34h and can be changed to 35h by setting the pin A0, as shown in Table 3 and Table 4. This enables two TDA9951 to be connected to the same host using the addresses 34h and 35h. Alternatively, changing the address enables one TDA9951 to avoid address clashes with other I^{2}C-bus slaves.

Table 4. I^{2}C-bus slave address

Bit	7	6	5	4	3	2	1	0
Value	0	1	1	0	1	0	A0	$R/\overline{W}$

8.2 Configuring the TDA9951

The TDA9951 is controlled using a series of registers.

Table 5. I^{2}C Register configuration

Register	Description	Address	Read/Write	Reference
APR	Address Pointer Register	00h	W	Table 6
CSR	Common Status Register	00h	R	Table 7
CER	Common Error Register	01h	R	Table 8
CVR	Common Version Register	02h	R	Table 9
CCR	Common Control Register	03h	R/W	Table 10
ACKH	CEC Address ACK High register	04h	R/W	Table 11
ACKL	CEC Address ACK Low register	05h	R/W	Table 12
CCONR	Common Configuration Register	06h	R/W	Table 14
CDR	Common Data Registers	07h to 19h	R/W	Table 15

The first byte of any I^{2}C-bus write frame configures the address pointer register APR. This determines the TDA9951 register accessed by the next I^{2}C-bus read or write.

If for example, a read is carried out without first writing to the address pointer register, the register returned is the register that address pointer register was last set to. The address pointer auto-increments after each successful read or write for all address pointer values other than 00h. When the address pointer register is set 00h, the common status register is polled using successive reads without needing to reset the address pointer register each time.

When the address pointer register is set higher than 07h, this is treated as setting it to 07h. This is because all message data transfers must start from register 07h and continue by auto-incrementing in one contiguous transfer.

When the host writes to two or more non-contiguous registers, two separate write sequences are used with either a STOP/START sequence or repeated START between them.

Before a read takes place, the host must first write to the address pointer register (if required) and then, repeat the START condition or STOP/START sequence. Finally, it starts reading data bytes until the read sequence is complete.

8.3 Using the INT line

The TDA9951 is an I^{2}C-bus slave device and when data is ready to be read, it uses the INT output line to signal the host processor. The common configuration register INT_POL bit sets the operating polarity of the INT line. When the INT line is active, it matches the state of the INT_POL bit. The INT line state is updated in the common status register making it possible to poll the register instead of monitoring the INT line. This method is less efficient consequently, it is not recommended. The common status register INT indication is not affected by the common configuration register INT_POL bit.

8.4 Register descriptions

Table 6. APR - Address pointer register (address 00h) bit description

Bit	Symbol	Access	Value	Description
7 to 5	reserved	W	000	reserved
4 to 0	REG_PTR[4:0]	W	-	address pointer: Address of the register to be read/written in the next I^{2}C-bus communication.

Table 7. CSR - Common status register (address 00h) bit description

Bit	Symbol	Access	Value	Description
7	BUSY	R	0	default; requests accepted
			1	busy; cannot accept further requests
6	INT	R	0	default; INT interrupt output is inactive
			1	the INT interrupt output is active
5	ERR	R	0	default; no error
			1	an error occurred; cleared on reading the common error register
4	SBY	R	0	default; the SBY (standby) output pin is inactive LOW
			1	the SBY output pin is HIGH
3	PSEN		0	the PSEN input pin is LOW
			1	the PSEN input pin is HIGH
2 to 0			000	not used (must be set to 000)

Table 8. CER - Common error register (address 01h) bit description

Bit	Symbol	Access	Value	Description
7 to 0	CER[7:0]	R		This register contains details of the last error. Reading this register resets the ERR bit in the common status register
			00h	no error has occurred since reset
			01h	a watchdog reset has occurred
			02h	a long CEC message with no End Of Message (EOM) was detected
			03h	CEC input has overrun. No buffer was available to hold new data

This register enables the host processor to read the TDA9951 software version.

Table 9. CVR - Common version register (address 02h) bit description

Bit	Symbol	Access	Value	Description
7 to 4	CVR_MAJ[3:0]	R	-	major version
3 to 0	CVR_MIN[3:0]	R	-	minor version

Table 10. CCR - Common control register (address 03h) bit description

Bit	Symbol	Access	Value	Description
7	RESET	R/W	0	no action
		W	1	resets the TDA9951 and returns it to its power-up state. CEC transmissions are completed before the reset. Only performed when the TDA9951 is in Idle mode, restores all default values.
6	CEC	R/W	0	default; the CEC interface is disabled after completion or reception of a pending CEC transmission. Further messages on the CEC line and messages for transmission are no longer acknowledged or accepted.
			1	The CEC interface is enabled and acknowledges messages based on the contents of the CEC address ACK high and address ACK low registers.
5	RC5	R/W	0	default; ignores RC5 commands from the IRX interface
			1	accepts RC5 commands
4	RC6	R/W	0	default; ignores RC6 commands from the IRX interface
			1	accepts RC6 commands
3	KEY	R/W	0	default; key matrix is disabled
			1	key matrix is enabled
2	VFD	R/W	0	default; VFD display output is disabled
			1	VFD display output is enabled
1	not used	-	0	not used
0	SBY	R/W	1	enters Standby mode and sets the SBY output. This bit is then cleared automatically. The common status register SBY bit indicates the SBY output state

Table 11. ACKH - CEC address ACK high register (address 04h) bit description

Bit	Symbol	Access	Value	Description
7	reserved	R	0	reserved; must be set to 0
6 to 0	ACKH[6:0]	R/W		for each bit:
			0	messages are not acknowledged
			1	messages are acknowledged and forwarded to the host

Table 12. ACKL - CEC address ACK low register (address 05h) bit description

Bit	Symbol	Access	Value	Description
7 to 0	reserved	R/W		for each bit:
			0	messages are not acknowledged
			1	messages are acknowledged and forwarded to the host

Using ACKH and ACKL, each bit of ADDR[14:0] corresponds to a CEC logical address. CEC reserves ADDR[15] as a broadcast address. ADDR[14:0] is built-up from ACKH[6:0] and ACKL[7:0].

Table 13. ADDR[14:0] definition

Bit	7	6	5	4	3	2	1	0
ACKH	-	ADDR[14]	ADDR[13]	ADDR[12]	ADDR[11]	ADDR[10]	ADDR[9]	ADDR[8]
ACKL	ADDR[7]	ADDR[6]	ADDR[5]	ADDR[4]	ADDR[3]	ADDR[2]	ADDR[1]	ADDR[0]

Table 14. CCONR - Common configuration register (address 06h) bit description

Bit	Symbol	Access	Value	Description
7 to 5	not used	R	000	not used; set to 000
4	ENABLE_ ERROR	R/W		controls how the TDA9951 notifies the host processor of errors:
			0	default; errors are not reported using the TDA9951 Data.err service or common error register
			1	errors reported using the TDA9951 Data.err service or common error register
3	INT_POL	R/W		sets the polarity of the INT output when it is active:
			0	default; the I2C_INT output is active-LOW
			1	the I2C_INT output is active-HIGH
2 to 0	RETRY[2:0]	R/W		these bits set the CEC retry count used by the TDA9951. The maximum value is 5; values greater than 5 give 5 retries:
			0 to 4	valid retry count
			5	default; maximum valid retry count
			6 to 7	accepted as 5 retries

Communication between the TDA9951 and the host processor for the data registers is carried out using information frames transferred using the common data register subaddress range 07h to 19h. The common data registers CDR0 to CDR18 are described in detail in Section 8.5.

8.5 Data register protocol

Before a frame is read or written, the host processor must set the REG_PTR field in the address pointer register to the base data register address. Message transfers can only start from the first data register at address 07h. They must not start from higher addresses because message transfer must be in complete sequences and not in fragments.

Each frame consists of a byte count, service selector, followed by zero or more parameters as shown in Figure 3.

Register 07h	Register 08h	Register 09h	[...]	[...]
FrameByteCount	ServiceSelector	[Parameter]	[...]	[...]

001aai277

Fig 3. Frame format for the data register protocol

The FrameByteCount is the number of bytes in the frame (including the FrameByteCount itself).

The service is specified by the ServiceSelector (see Table 15). If an unused ServiceSelector is sent to the TDA9951, it responds with the confirm Bad.req service (see Table 17, Table 28, Table 32 and Table 36).

The remaining bytes of the frame can contain up to 17 parameters associated with the service. Services do not have optional parameters. The TDA9951 only accepts one outstanding request.

The data service provided using the common data registers comprises:

- Host to the TDA9951: requests.
  - A request is sent from the host to the TDA9951 or to a device on one of its interfaces.
- TDA9951 to the host: confirmations, indications and errors.
  - Confirmations are used to answer requests for status and flow control, to show if the request has been passed on to the intended recipient or if it has been accepted/rejected.
  - Indications are messages from the TDA9951 interface to the host. Generally relating to normal operation.
  - Errors are messages from a TDA9951 to the host. Generally relating to an error state.

Table 15 lists the data services and their ServiceSelector values. The contents of each service is described in detail in the following Protocol Layer sections.

Table 15. Data services

ServiceSelector	Host to TDA9951	TDA9951 to host	Message type
00h	CECData.req		request
01h		CECData.cnf	confirmation
02h	VFDData.req		request
03h		VFDData.cnf	confirmation
04h	RTCData.req		request
05h		RTCData.cnf	confirmation
06h	SBYData.req		request
07h		SBYData.cnf	confirmation
08h	KEYData.req		request
09h		KEYData.cnf	confirmation
81h		CECData.ind	CEC indication, no error

Table 15. Data services *…continued*

ServiceSelector	Host to TDA9951	TDA9951 to host	Message type
82h		CECData.err	no indication, error
83h		CECData.ier	indication and error
84h		RTCData.ind	RTC alarm indication, no error
85h		IRData.ind	IR command indication, no error
86h		KEYData.ind	KEY indication, no error

8.5.1 CECData.req service

This CECData.req request service is sent from the host to the TDA9951 instructing it to transmit an addressed or broadcast message. If the correct signal free time rules are met, transmission of the CEC message starts as soon as the complete message is received by the TDA9951. Table 16 shows the frame byte for the service.

Table 16. Frame Bytes for CECData.req service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h to 18h	
08h	ServiceSelector	00h	CECData.req
09h	AddressByte	-	source and destination logical addresses in the format: SSSS DDDD
0Ah to 18h	DataBytes	-	zero to fifteen bytes up to the FrameByteCount – 3 data length

8.5.2 CECData.cnf service

Using this service the TDA9951 informs the host of the success or failure of a CECData.req service. The frame bytes are shown in Table 17.

Table 17. Frame Bytes for CECData.cnf service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h	
08h	ServiceSelector	01h	CECData.cnf
09h	ResultCode		a value indicating the result of the transmission
		00h	success
		80h	CEC in off-state
		81h	Bad.req service
		82h	failed; unable to access CEC line
		83h	failed; arbitration error
		84h	failed; bit timing error
		85h	failed; destination address not acknowledged
		86h	failed; data byte not acknowledged

8.5.3 CECData.ind service

Using this service the TDA9951 transfers a CEC indication message to the host which was received from another remote device. The frame bytes are shown in Table 18.

Table 18. Frame Bytes for CECData.ind service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h to 12h	
08h	ServiceSelector	81h	CECData.ind
09h	AddressByte	-	source and destination logical addresses in the format: SSSS DDDD
0Ah to 18h	DataBytes	-	zero to fifteen bytes up to the FrameByteCount – 3 data length

8.5.4 CECData.err service

Using this service, the TDA9951 alerts the host to a CEC error condition. There are no parameters. The host should read the Common Error Register (CER) for details of the error. Only active when bit 4 of the Common Configuration Register (CCONR) is set to enable error indications. The frame bytes for the service are shown in Table 19.

Table 19. Frame Bytes for CECData.err service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	02h	
08h	ServiceSelector	82h	CECData.err

8.5.5 CECData.ier service

Using this service, the TDA9951 transfers a CEC message to the host which was received from another remote device. In addition, it alerts the host to a CEC error condition. The host should read the TDA9951 Common Error Register (CER) for details of the error. Only active when bit 4 of the CEC Common Configuration Register (CCONR) is set to enable error indications. The frame bytes are listed in Table 20.

Table 20. Frame Bytes for CECData.ier service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h to 13h	
08h	ServiceSelector	83h	CECData.ier
09h	AddressByte	-	source and destination logical addresses in the format: SSSS DDDD
0Ah to 19h	DataBytes	-	zero to sixteen bytes up to the FrameByteCount – 3 data length

8.6 Example communication sequences

8.6.1 Notes on writing the CEC common data registers

Common data registers should be written in one contiguous operation between a START and STOP condition. The write action starts from the first data register and includes all registers indicated by the contents of that data register. The length of the message is given by the byte in the first data register. This is at least three for the shortest message. Lower values than three indicate an invalid message.

Data registers ignore data in the following cases:

- When fewer data registers are written than the number indicated by the first data register. The partial message is ignored and a confirmation is not returned.
- When more data registers are written than the number indicated by the first data register. The message is processed once the message’s last data register is written but the extra bytes written are ignored.
- When the highest data register is written and more message bytes are indicated by the first data register. The message is processed once the highest data register is written but the extra bytes written are ignored.

8.6.2 Notes on reading the CEC common data registers

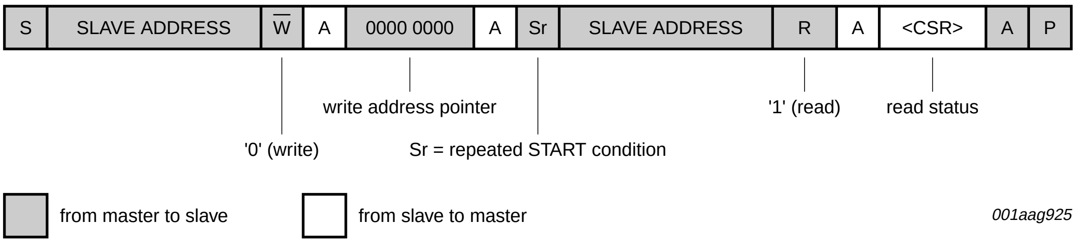
Data registers should be read in one contiguous operation, starting from the first data register up to the last register indicated by the Data register.

The data registers can only contain valid messages when the INT line and the INT bit in the TDA9951 status register are set.

Typical read situations:

- When data registers are read and the INT line is not set, the first data register contains 0 (no bytes to read). Any additional read sequences before a STOP condition return the value FFh.
- When the host writes to data registers and starts reading without first resetting the address pointer register, the read sequence commences from the first data register.
- When reading stops before all indicated data registers are read, the TDA9951 resets the INT line, ignores the message and the message is lost.
- When reading continues for more data registers than indicated by the first data register, the value FFh is read. The INT line is reset when the last valid data register for the message is read.

8.6.3 Communication use cases

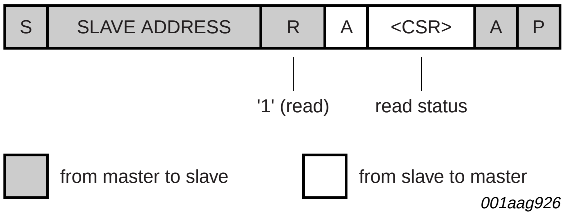


A = acknowledge (SDA = LOW).

S = START condition.

P = STOP condition.

Fig 4. Host reads TDA9951 status register after setting address pointer

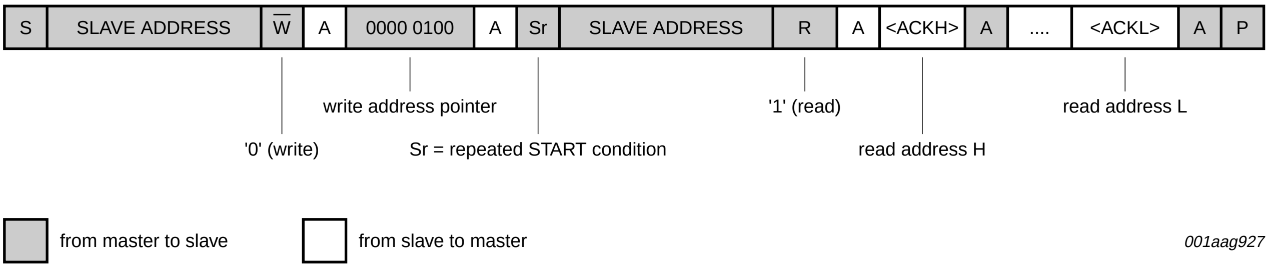


A = acknowledge (SDA = LOW).

S = START condition.

P = STOP condition.

Fig 5. Host reads TDA9951 status register without setting address pointer (pointer was at 0 already)



A = acknowledge (SDA = LOW).

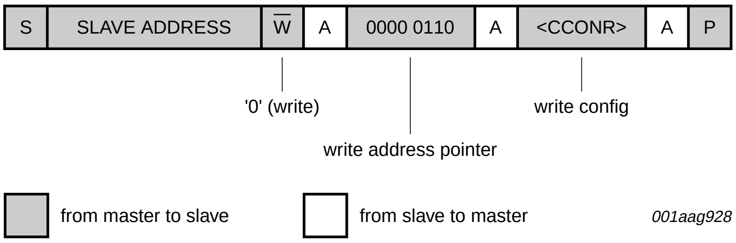
S = START condition.

P = STOP condition.

R = read.

$\overline{W}$ = write.

Fig 6. Host reads address ACK registers after setting address pointer



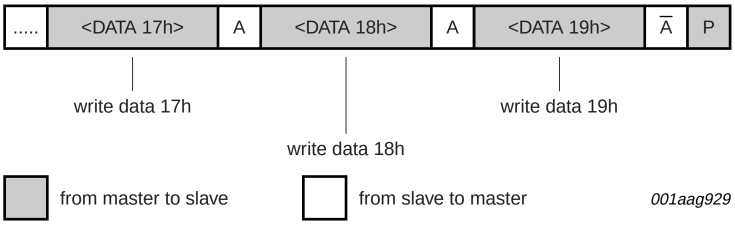
A = acknowledge (SDA = LOW).

S = START condition.

P = STOP condition.

$\overline{W}$ = write.

Fig 7. Host writes configuration register after setting address pointer



A = acknowledge (SDA = LOW).

$\overline{A}$ = not acknowledge (SDA = HIGH).

P = STOP condition.

Fig 8. Host writes last three data registers

8.7 I^{2}C command examples

8.7.1 Initialization

After a reset, configure the TDA9951 with its logical address or addresses (as required):

- **I2C_WRITE: 04h, 00h, 08h**

  Set address pointer to 04h (ACKH), set ACKH to 00h and set ACKL to 08h (example).

  The TDA9951 is now configured to acknowledge messages to logical address 3 (Tuner 1).

Remark: It is then mandatory to set the TDA9951 to the ON state as follows:

- **I2C_WRITE: 03h, 40h**

  Set address pointer to 03h (CCR) and set CCR to 40h.

  The TDA9951 is now enabled. Messages addressed to logical address Tuner 1 is acknowledged and forwarded to the host processor.

8.7.2 Sending CEC messages

Example: the host processor of playback device 1 wishes to send the message <TextView On> to TV:

- **I2C_WRITE: 00h; I2C_READ, I2C_READ, ...**

  Set address pointer to 00h (CSR) and read common status register. Repeat the read sequence until TDA9951 is no longer busy (bit 7 = 0).
- **I2C_WRITE: 07h, 04h, 00h, 40h, 0Dh**

  Set address pointer to 07h (Data Register 1) and write data registers. FrameByteCount = 4, ServiceSelector = CECData.req, AddressByte = DVD/TV and DataByte = <TextView On>.
- **Wait for INT line to be asserted**

  When TDA9951 has a response, it asserts the I2C_INT line (could also poll bit 6 of CSR).
- **I2C_WRITE: 07h; I2C_READ: 03h, 01h, 00h**

  Set address pointer to 07h (Data Register 1) and read data registers. FrameByteCount = 3, ServiceSelector = CECData.cnf and ResultCode = Success.

8.7.3 Receiving CEC messages

Example: TV sends the message <Give Physical Address> to Playback Device 1:

- **INT line is asserted**

  The TDA9951 at playback device 1 has acknowledged the message from TV and it is now available for reading by the playback device 1 host processor.
- **I2C_WRITE: 07h; I2C_READ: 04h, 81h, 04h, 83h**

  Set address pointer to 07h (Data Register 1) and read data registers. FrameByteCount = 4, ServiceSelector = CECData.ind, AddressByte = TV/DVD and DataByte = <Give Physical Address>.

8.8 Infrared receiver interface

8.8.1 Infrared hardware

The input port pin IR_DAT is active LOW and receives demodulated infrared data from an external demodulator device that strips the 36 kHz infrared carrier. The output port pin IR_VAL is driven LOW for 65 ms after a valid frame is received enabling it to drive an external LED activity indicator.

Wave form timing is achieved using the chip’s capture and compare unit based on the defined timing limits.

- In RC5 mode, timing tolerances are checked and frames containing any bits outside the limits are rejected.
- Start bits have wider timing tolerances than other bits to avoid false frame rejection.
- In RC6 Mode 0, timing tolerances are checked and frames containing any bits outside the limits are rejected.
- Dropout spikes of 60 µs or less are ignored during start bit or leader pulse detection.

8.8.2 Protocol discrimination

The three protocol variants are automatically selected by measuring the length of the first active pulse, as follows:

Table 21. Protocol discrimination

Protocol	Active pulse	Nominal time (µs)	Time range (µs)
RC5	second half of start bit S1	889	676 to 1306
RC5 Enlarged	second half of start bit S1 and first half of command bit /C6	1778	1352 to 2178
RC6 Mode 0	first part of leader pulse	2667	2179 to 3360

8.8.3 IRData.ind service

This service is used to transfer an infrared remote control commands to the host. The frame bytes for the service are shown in Table 22.

Remark: The SBYData.req request service configures some IRX commands to enter or leave Standby mode.

Table 22. Frame Bytes for IRData.ind service

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		05h	
08h	ServiceSelector		85h	IRData.ind
09h	CmdFlags			bit flags associated with the command:
		7 to 2	00h	unused; set to 0
		1		protocol type
			0	RC5
			1	RC6
		0	0 or 1	a copy of the received Toggle bit
0Ah	CmdAddress		-	command address byte
0Bh	CmdData		-	command data or command byte

8.9 Key matrix interface

8.9.1 Key matrix hardware

The Key Matrix Interface (KEY) is designed to manage up to 10 buttons. The key matrix interface uses five input lines (KROW0 to KROW4) and one output line (KCOL) to decode a matrix of up to ten normally open switch contacts. Contact de-bouncing is achieved by reading the input again after a nominal 20 ms period and discarding the event if the state changes in that time.

8.9.2 Key matrix decoding

The decoding algorithm is described in Application Note *AN10184 (Connecting a keyboard to the LPC9xx microcontroller, 14/09/2002)*.

Multiple simultaneous contact closure events are discarded.

8.9.3 Key matrix encoding

The KROW0 to KROW4 input lines (matrix rows) are connected to the chip port 0. The port produces an interrupt when certain port inputs match or deviate from a set pattern. This allows the unused pins of port 0 to be masked out of this process and used for other purposes.

The KCOL output line (matrix column) is used to identify which set of five switches the input contact closure belongs to.

Key events are encoded as follows. The matrix rows are numbered 0 to 4 and the matrix columns 0 to 1. A contact closure is defined as the intersection of a row and column in the matrix:

Table 23. Key matrix

Column	Row	Key ID	Column	Row	Key ID
0	0	1	1	0	6
0	1	2	1	1	7
0	2	3	1	2	8
0	3	4	1	3	9
0	4	5	1	4	10

8.9.4 KEYData.req service

Using this service the host requests the current matrix key switch states. The frame bytes for the service are shown in Table 24.

Table 24. Frame Bytes for KEYData.req service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	02h	
08h	ServiceSelector	08h	KEYData.req

8.9.5 KEYData.cnf service

Using this service, the TDA9951 informs the host of the matrix key switch states after a KEYData.req service request. The frame bytes are shown in Table 25.

Table 25. Frame Bytes for KEYData.cnf service

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		04h	
08h	ServiceSelector		09h	KeyData.cnf
09h	KeyStates1			bit flags showing the states of key switches 1 to 8:
		7		key 8
			0	open
			1	closed
		6		key 7
			0	open
			1	closed
		5		key 6
			0	open
			1	closed
		4		key 5
			0	open
			1	closed
		3		key 4
			0	open
			1	closed
		2		key 3
			0	open
			1	closed
		1		key 2
			0	open
			1	closed
		0		key 1
			0	open
			1	closed
0Ah	KeyStates2			bit flags showing the states of key switches 9 to 10
		7 to 2		not used
		1		key 10
			0	open
			1	closed
		0		key 9
			0	open
			1	closed

8.9.6 KEYData.ind service

Using this service, the TDA9951 transfers a key press or release command to the host. The frame bytes are listed in Table 26.

Table 26. Frame Bytes for KEYData.ind service

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		03h	
08h	ServiceSelector		86h	KEYData.ind
09h	KeyData			bit field associated with the key switch event:
		0 to 3	-	key code 1 to 10
		4 to 6	-	not used
		7	1	key open
			0	closed

8.10 Real time clock interface

8.10.1 Real time clock hardware

The Real Time Clock (RTC) interface communicates with a software clock-calendar running in the TDA9951 which uses the dedicated 23-bit RTC. The RTC timer is set to generate a 1 s interrupt.

When the power source changes from power supply to battery, the PSEN input goes LOW. All interface and alarm/timer activity is stopped. The chip switches to low-power mode but continues to count RTC timer ticks.

The host can read the clock-calendar on demand by using the RTCData.req request service. The current date and time is returned using the RTCData.cnf confirm service. The data registers are fixed during a read action ensuring a consistent time value is returned.

8.10.2 Clock-calendar

The 1 s event handler increments a long integer representing the number of seconds since the start of a time epoch. It also manages any active alarm and timers.

The date and time fields in the request and confirmation services are converted to and from the long integer using code adapted from the *“Maxim/Dallas Semiconductor application note 3721 Interfacing the DS1318 with an 8051 type Microcontroller, 9/12/2005”*.

8.10.3 Alarm

Only one alarm is active at a time with the clock-calendar and timers. The alarm is triggered when the real-time clock matches all date and time bytes set in an active alarm request. Setting an alarm replaces a previously set alarm.

Alarms are cancelled by clearing the alarm active bit in the RTCData.req ServiceFlags byte. The Date and time bytes in RTCData.req are ignored if the active bit is cleared. Alarms are not active after a power-up or reset.

An alarm sets Standby mode when its standby bit in the RTCData.req ServiceFlags byte is set. An alarm triggered in Standby mode exits and its standby bit is not set.

8.10.4 Timers

A maximum of two timers may be simultaneously active with the clock-calendar and alarm. Each timer may be either one-shot or periodic.

- One-shot timers are triggered when the real-time clock matches the current time plus the time and day bytes set by an active timer request.
- Periodic timers work in the same way but are triggered repeatedly at the specified interval.

The timer resolution is 1 s and allowed timer interval range is from 1 s up to 31 days, 23 hours, 59 minutes and 59 s. Setting timer 1 replaces the previous timer 1 setting but does not change timer 2 or a pending alarm. Timer 2 functions in the same way.

Timers are stopped by clearing its respective timer active bit in the RTCData.req ServiceFlags byte. Date and time bytes in RTCData.req are ignored when the active bit is clear. After power-up and reset, the timers are not active.

A timer sets Standby mode if its standby bit is set in the RTCData.req ServiceFlag byte. Timers triggered in Standby mode exits Standby mode when the respective standby bit is not set.

8.10.5 RTCData.req service

This service is used to read or set the clock date and time, to set an alarm or to set the timers. The frame bytes are listed in Table 27.

The ServiceType byte determines the action to be performed. When the clock is read, all other bytes are ignored and can be set to 00h. When a timer is set, the Month and Year bytes are ignored and can be set to 00h.

Table 27. Frame Bytes for RTCData.req service

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		0Ah	
08h	ServiceSelector		04h	RTCData.req
09h	Service Type		00h	read the clock date and time. All following bytes in this frame are not used, set to 00h
			01h	set the clock date and time. Uses all date and time bytes but the ServiveFlags byte is not used
			02h	set the alarm date and time. Uses all date and time bytes; standby operation is set by ServiceFlags
			03h	set timer 1 to current time plus Day, Hour, Minute and Second bytes. Month and Year bytes not used; standby and periodic operation are set by ServiceFlags
			04h	set timer 2 to current time plus Day, Hour, Minute and Second bytes. Month and Year bytes not used; standby and periodic operation are set by ServiceFlags

Table 27. Frame Bytes for RTCData.req service *…continued*

Register	Frame Byte	Bit	Value	Comments
0Ah	ServiceFlags	7 to 3		not used
		2		not used for alarm
			0	one-shot timer
			1	periodic timer
		1	0	normal
			1	standby alarm or timer
		0	0	stopped
			1	alarm or timer active
0Bh	Year		00h to 99h	Year 2000 to 2099
0Ch	Month		01h to 12h	Month 1 (January) to 12 (December)
0Dh	Day		01h to 31h	Day 1 to 31
0Eh	Hour		00h to 23h	Hour 0 to 23
0Fh	Minute		00h to 59h	Minute 0 to 59
10h	Second		00h to 59h	Second 0 to 59

8.10.6 RTCData.cnf service

Using this service the TDA9951 informs the host of the success or failure of a RTCData.req service and always returns the current date and time. The frame bytes are shown in Table 28.

Table 28. Frame Bytes for RTCData.cnf service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	0Ah	
08h	ServiceSelector	05h	RTCData.cnf
09h	ResultCode		A value indicating the result of the request
		00h	success, time and date are valid
		81h	Bad.req service
		90h	time and date cannot be read, they have not been set
		91h	failed; bad Year
		92h	failed; bad Month
		93h	failed; bad Day
		94h	failed; bad Hour
		95h	failed; bad Minute
		96h	failed; bad Second
0Ah	Year	00h to 99h	Year 2000 to 2099
0Bh	Month	01h to 12h	Month 1 (January) to 12 (December)
0Ch	Day	01h to 31h	Day 1 to 31
0Dh	Hour	00h to 23h	Hour 0 to 23

Table 28. Frame Bytes for RTCData.cnf service *…continued*

Register	Frame Byte	Value	Comments
0Eh	Minute	00h to 59h	Minute 5 to 59
0Fh	Second	00h to 59h	Second 0 to 59
10h	Weekday	01h to 07h	Day 1 (Sunday) to 7 (Saturday). This is always calculated and never set.

8.10.7 RTCData.ind service

This indication service is used to alert the host of an alarm or timer event. In doing so, it always returns the current date and time. The frame bytes are listed in Table 29.

No Indication is sent to the host in Standby mode.

Table 29. Frame Bytes for RTCData.ind service

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		0Bh	
08h	ServiceSelector		84h	RTCData.ind
09h	ServiceType		02h	alarm event
			03h	timer 1 event
			04h	timer 2 event
0Ah	ServiceFlags	7 to 3	-	not used
		2		not used for alarm
			0	one-shot timer
			1	periodic timer
		1	0	normal
			1	standby alarm or timer
		0	0	stopped
			1	alarm or timer active
0Bh	Year		00h to 99h	Year 2000 to 2099
0Ch	Month		01h to 12h	Month 1 (January) to 12 (December)
0Dh	Day		01h to 31h	Day 1 to 31
10h	Hour		00h to 23h	Hour 0 to 23
11h	Weekday		01h to 07h	Day 1 (Sunday) to 7 (Saturday). Always calculated, never set

8.11 Standby interface

8.11.1 Standby hardware

The SBY output is asserted to set the host or other CE device to lower-power operation. Due to the continuous interface monitoring required in this mode, the IC itself does not operate in low-power mode. However, PSEN input is driven LOW it operates in low-power mode). In Standby mode, no indication services are sent to the host.

Table 30. Standby mode changes

Current mode	Event	New mode	SBY pin
Undefined	Power-up	Standby On	HIGH
Standby On	CEC or IRX or KEY interface receives a configured wake-up command	Standby Off	LOW
	RTC interface raises a wake-up alarm		
Standby Off	Host sets SBY bit in CCR register	Standby On	HIGH
	CEC or IRX or KEY interface receives a configured Standby command		
	RTC interface raises a Standby alarm		
	PSEN input goes low		

8.11.2 SBYData.req service

Using this service the TDA9951 is instructed to enter or exit Standby mode when certain CEC opcodes, IR commands or key events occur. These three sources can have up to eight commands/events that can enter or leave Standby mode. A separate SBYData.req service must be sent from the host for each source.

In addition, each IR command source also requires the corresponding address byte and flags byte are sent as separate requests. A received IR command must match on all three parameters (command, address and flags bytes) to trigger the Standby mode state change.

The frame bytes are listed in Table 31.

Table 31. Frame Bytes for SBYData.req service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	13h	
08h	ServiceSelector	06h	SBYData.req
09h	CmdSource	01h	CEC; command codes are Opcodes 00h to FFh
		02h	KEY; command codes key IDs in 0
		03h	IRX; command codes are commands 00h to FFh
		04h	IRX; command codes are addresses 00h to FFh
		05h	IRX; command codes are flags 00h to FFh
0Ah	StandbyCmd1		for all command bytes:
		01h	unused
		01h to FFh	Valid
0Bh	StandbyCmd2	-	user defined
0Ch	StandbyCmd3	-	user defined
0Dh	StandbyCmd4	-	user defined
0Eh	StandbyCmd5	-	user defined
0Fh	StandbyCmd6	-	user defined
10h	StandbyCmd7	-	user defined
11h	StandbyCmd8	-	user defined
12h	WakeupCmd1	-	user defined

Table 31. Frame Bytes for SBYData.req service *…continued*

Register	Frame Byte	Value	Comments
13h	WakeupCmd2	-	user defined
14h	WakeupCmd3	-	user defined
15h	WakeupCmd4	-	user defined
16h	WakeupCmd5	-	user defined
17h	WakeupCmd6	-	user defined
18h	WakeupCmd7	-	user defined
19h	WakeupCmd8	-	user defined

8.11.3 SBYData.cnf service

Using this service the TDA9951 informs the host of the success or failure of the SBYData.req service. The frame bytes are shown in Table 32.

Table 32. Frame Bytes for SBYData.cnf service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h	
08h	ServiceSelector	07h	SBYData.cnf
09h	ResultCode		the result of the request
		00h	success
		81h	Bad.req service
		A0h	bad CmdSource

8.12 Vacuum fluorescent display interface

8.12.1 VFD hardware

The output lines VFD_DAT, VFD_CLK, VFD_STR and VFD_BLK use an industry-standard serial protocol common to several driver chip manufacturers, such as Allegro Microsystems for more details see the *“Data sheet 6810, DABiC-IV 10-bit serial-input, latched source driver, Allegro Microsystems, 26182.124E”*. The external driver chip latches the serial data and handles the high voltage I/O needed by VFD devices.

A timer is used to output data at a constant rate, one grid at a time, to support multiplexed displays. Each grid must be refreshed at least 30 times a second to avoid display flicker. Non-multiplexed displays are not refreshed.

8.12.2 VFD driver connection topologies

8.12.2.1 VFD driver device

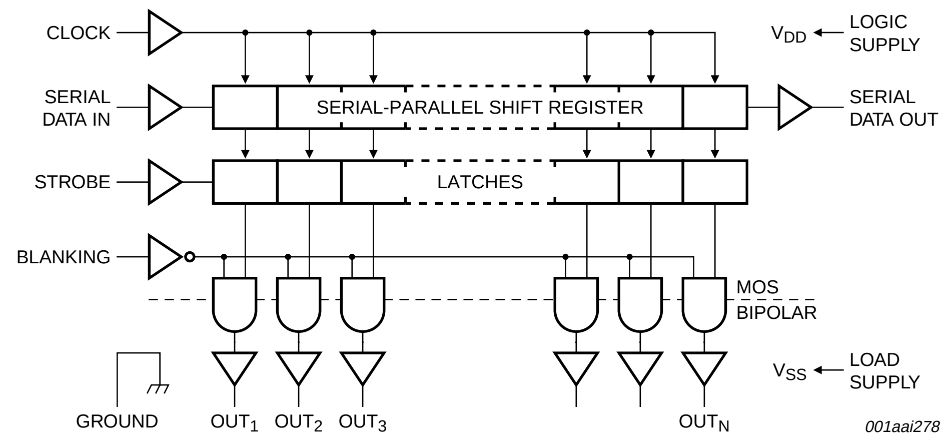
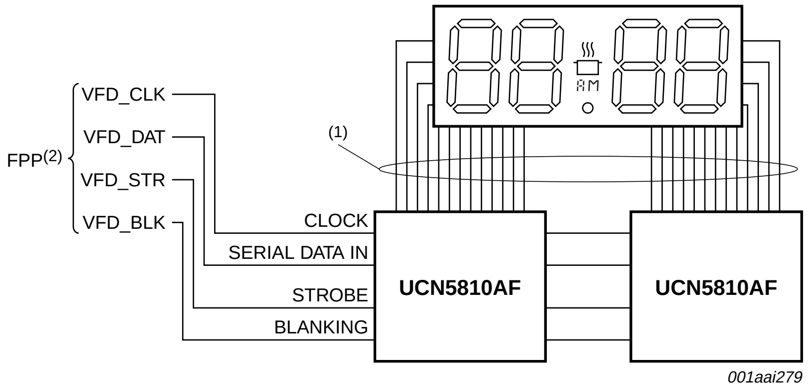


Fig 9. VFD driver device

The four-wire serial interface sends data to the external driver device as follows:

1. The TDA9951 sets its VFD_DAT line to the required level and toggles its clock VFD8CLK output for each bit. These are connected to the device serial data in and Clock lines.
2. Data clocked in on the serial data in is moved to the serial data out when the shift register is full to allow driver cascading. This feeds the next drivers serial data in.
3. When all bits have been clocked into all drivers, the TDA9951 toggles VFD_STR which in turn toggles the device strobe line.
4. The TDA9951 VFD_BLK line drives the device blanking line which turns all driver outputs on or off.

8.12.2.2 Static drive



(1) One latched driver line for every display anode; the grids are permanently powered.

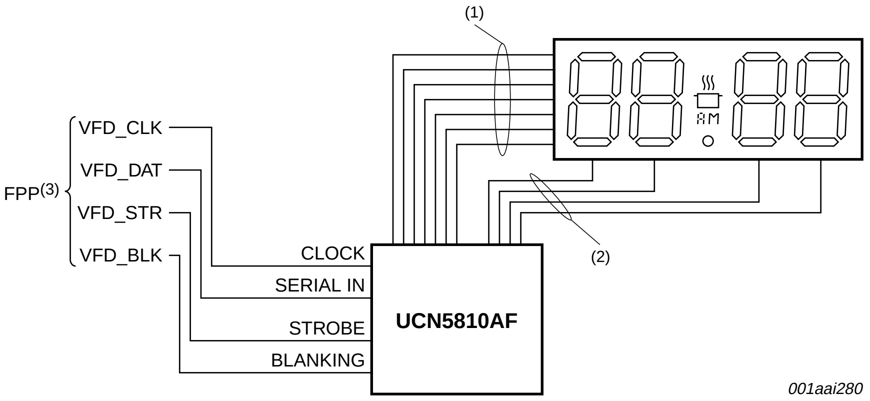
(2) The Front Panel Processor (FPP) is the TDA9951.

Fig 10. Static drive

The driver outputs are connected to each separate display segment in one-grid displays.

- All data bits on VFD_DAT for every display segment are clocked in by VFD_CLK.
- VFD_STR is pulsed
- The required display segments are now lit with without the need for further control actions.
  - Advantage: data is only clocked out to the display when the host changes the data.
  - Disadvantage: more driver devices and I/O lines are required for a given size.

8.12.2.3 Multiplexed drive



(1) Segment lines are connected to all corresponding anodes in a character position.

(2) Grid lines drive each character.

(3) The Front Panel Processor (FPP) is the TDA9951.

Fig 11. Multiplexed drive

A multiplexed scheme can use the same four wire interface for displays with multiple grids. In this case, only one character grid is enabled at a time together with the correct segment.

8.12.3 Host request bit mapping to clocked output bits

The SegmentByte value in the VFDData.req request form a bit array holding one or more grids of segment data.

8.12.3.1 Static drive

all VFDData.req SegmentByte bits are clocked out once as soon as the last request in a series is received, no extra grid bits are sent by the TDA9951 for a one grid static display:

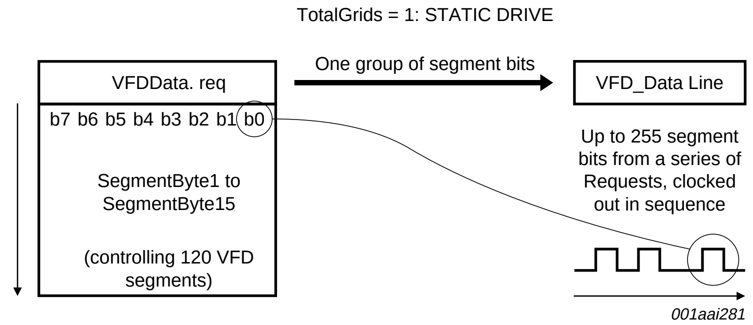

TotalGrids = 1: STATIC DRIVE
VFDData. req
b7 b6 b5 b4 b3 b2 b1 b0
SegmentByte1 to SegmentByte15
(controlling 120 VFD segments)
One group of segment bits
VFD_Data Line
Up to 255 segment bits from a series of Requests, clocked out in sequence
001aai281

Fig 12. Static drive

8.12.3.2 Multiplexed drive

Displays with more than one grid are multiplexed. The host manages the VFDData.req SegmentByte bits in groups of successive grids. Each of segment bits is padded to occupy the total number of SegmentByte locations, enabling easy specification of a subset of grids by SegmentByteOffset and FrameByteCount in the request. A large display may be driven in the same way by multiple requests.

The TDA9951 is responsible for clocking out the grid bits, either before or after the request's segment bits, based on the GridMuxMode. Only one grid is set, corresponding to the grid group that is clocked out from the request. The TDA9951 repeats this procedure indefinitely.

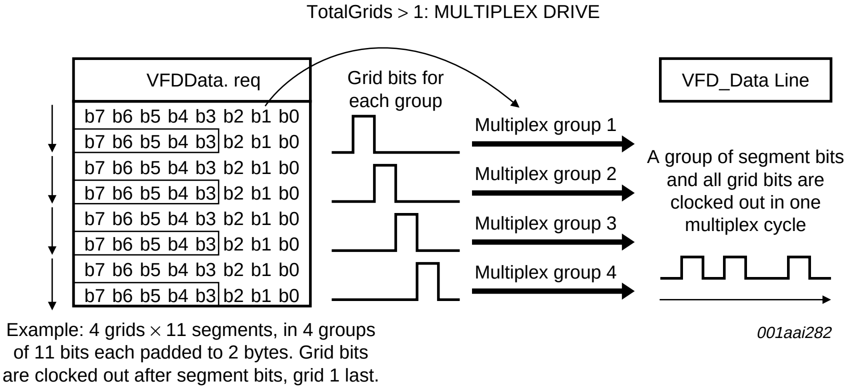
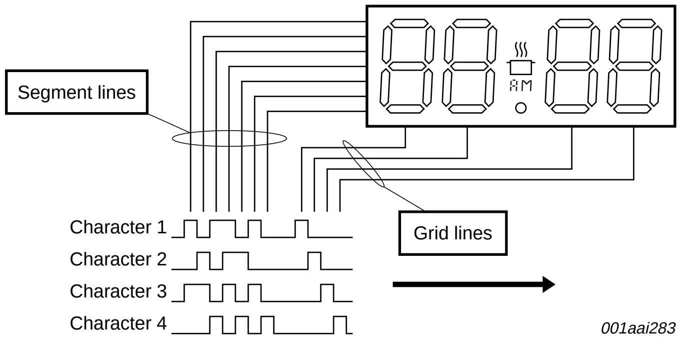


Fig 13. Grid bits clocked out after the segment bits



(1) TDA9951 automatically sets the grid lines before the segment lines with grid 1 as the last (GridMuxMode = 1).

Fig 14. Output of four multiplex cycles comprising four characters

8.12.4 Multiplex drive timing

Table 33 shows the maximum number of segments supported for each number of grids, the timer interrupt rate and number of requests required. It assumes 5 µs per bit for setting the data line, pulsing the clock line to send each grid and the segment bit to the external driver.

Table 33. Multiplex drive timing

Grids (character positions)	Recommended number of segments	Timer interrupt rate (Hz)	Grid update rate (31 Hz to 42 Hz)	Requests for maximum segments
1	255	N/A	N/A	3
2	248	80	40	5
3	157	125	41.66	4
4	121	160	40	5
5	95	200	40	4
6	74	250	41.66	4
7	73	250	35.71	5
8	54	320	40	4
9	53	320	35.55	5
10	40	400	40	4
11	39	400	36.36	4
12	28	500	41.66	4
13	27	500	38.36	4
14	26	500	38.36	4
15	17	625	41.66	3
16	16	625	39.06	3
17	15	625	36.76	3
18	14	625	34.72	3
19	13	625	32.89	3
20	12	625	31.25	3
21	7	800	38.09	2

8.12.5 Multiplex drive pitfalls

There are several potential pitfalls to avoid with a multiplexed drive:

- Multiplexing is too slow causing the display to flicker

  Solution: light each character grid at least 30 times a second
- If a one character grid is immediately enabled after the previous grid, ghosting may be seen where the drives for the previous character segment are enabled for a short time during the new character

  Solution: use an inter-character blank time at the start of each grid time slot by setting VFD_BLK for 10 μs to 50 μs.
- If multiplexing stops, display damage can occur

  Solution: assert VFD_BLK if a reset occurs or use an external watchdog protection circuit. See Noritake Itron *“Chip in glass Driver VFD application note”* for more detailed information.
- The contents of the VFD driver are undefined and may result in irrelevant data display during start-up

  Solution: Set VFD_BLK during the first multiplex cycle

8.12.6 VFDData.req service configuration

Using this service the host can send configuration data to the TDA9951. Only one configuration service is acted on after power-up because the TDA9951 does not support dynamic display changes. The frame bytes are shown in Table 34.

When the request is received, the VFD_BLK output is reset to a state based on the current blanking polarity.

Table 34. Frame Bytes for VFDData.req service - configuration

Register	Frame Byte	Value	Comments
07h	FrameByteCount	07h	
08h	ServiceSelector	02h	VFDData.req
09h	ServiceType	00h	configuration request
0Ah	TotalGrids	1 to 20	Total grids:
		1	VFD is statically driven
		> 1	VFD is multiplexed
0Bh	SegmentsPerGrid	1 to 255	segments or dots per grid. See Table 33 for the maximum SegmentsPerGrid value for each TotalGrids value
0Ch	GridMuxMode	0 to 3	control how the TDA9951 clocks out grid control bits for multiplexed displays (Ignored when TotalGrids = 1):
		0	grids clocked before segments, 1 first
		1	grids clocked before segments, 1 last
		2	grids clocked after segments, 1 first
		3	grids clocked after segments, 1 last
0Dh	BlankingPolarity	0	VFD_BLK line is active LOW
		1	VFD_BLK line is active HIGH (default)

8.12.7 VFDData.req service segment data

Using this service the host sends display segment data to the TDA9951. The frame bytes for this function are shown in Table 35.

Multiple requests are used to send segment data that is longer than one request. SegmentByteOffset bit 7 is set only in the last request. If this request is received while the VFD bit in the common control register is clear, the data is stored but the display is not driven until the VFD bit is set.

Table 35. Frame Bytes for VFDData.req service segment data

Register	Frame Byte	Bit	Value	Comments
07h	FrameByteCount		05h to 13h	1 segment byte (05h) to 15 bytes (13h)
08h	ServiceSelector		02h	VFDData.req
09h	ServiceType		01h	segment data request
0Ah	SegmentByteOffset	6 to 0	00h to 7Fh	byte offset applied to SegmentByte1 for displays that require multiple requests
				request 1; SegmentOffset = 0
				request 2; SegmentOffset = 15 etc.
				a request can also update part of a display, e.g. to set SegmentByte2, SegmentByte3:
				SegmentByteOffset = 1
				FrameByteCount = 6
		7	0 or 1	indicates the last request in a multi-request series
0Bh	SegmentByte1		00h to FFh	segment bits 1 to 8 clocked out in sequence start in with bit 1
0Ch	SegmentByte2			segment bits 9 to 16
0Dh	SegmentByte3			segment bits 17 to 24
0Eh	SegmentByte4			segment bits 25 to 32
0Fh	SegmentByte5			segment bits 33 to 40
10h	SegmentByte6			segment bits 41 to 48
11h	SegmentByte7			segment bits 49 to 56
12h	SegmentByte8			segment bits 57 to 64
13h	SegmentByte9			segment bits 65 to 72
14h	SegmentByte10			segment bits 73 to 80
15h	SegmentByte11			segment bits 81 to 88
16h	SegmentByte12			segment bits 89 to 96
17h	SegmentByte13			segment bits 97 to 104
18h	SegmentByte14			segment bits 105 to 112
19h	SegmentByte15			segment bits 113 to 120

8.12.8 VFDData.cnf service

Using this service the TDA9951 informs the host of the success or failure of a VFDData.req service. It is mainly used for flow control and ensures the host does not send new data before the previous request has completed. The frame bytes are shown in Table 36.

Table 36. Frame Bytes for VFDData.cnf service

Register	Frame Byte	Value	Comments
07h	FrameByteCount	03h	
08h	ServiceSelector	03h	VFDData.cnf
09h	ResultCode		A value indicating the result of the request
		00h	Success
		80h	VFD in Off state
		81h	bad.req service
		B0h	bad ServiceType
		B1h	bad TotalGrids
		B2h	bad SegmentsPerGrid
		B3h	SegmentsPerGrid is too large for TotalGrids
		B4h	bad GridMuxMode
		B5h	bad BlankingPolarity
		B6h	bad SegmentByteOffset
		B7h	segment service sent before Configuration service

9. Limiting values

Table 37. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).[1]

Symbol	Parameter	Conditions	Min	Max	Unit
$T_{amb(bias)}$	bias ambient temperature	operating	–55	+125	°C
T_{stg}	storage temperature		–65	+150	°C
I_{OH}	HIGH-level output current	all input/output pin	-	20	mA
I_{OL}	LOW-level output current	all input/output pin	-	20	mA
$I_{IO(tot)}$	total input/output current		-	100	mA
$V_{DD(xtal)}$	crystal supply voltage	on pins XTAL1, XTAL2	-	V_{DD} + 0.5	V
V_n	voltage on any other pin	except pins XTAL1, XTAL2, V_{DD}	-	3.5	V
P_{tot}	total power dissipation	based on package heat transfer, not device power consumption	-	1.5	W

[1] Parameters are valid for T_{amb} = –40 °C to +85 °C temperature range, unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

10. Static characteristics

Table 38. Static characteristics

V_{DD} = 2.4 V to 3.6 V; T_{amb} = –40 °C to +85 °C for industrial application; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ[1]	Max	Unit
I_{DD}	supply current	Operating mode; V_{DD} = 3.6 V; f_{osc} = 12 MHz	[2]	-	11	18	mA
		Idle mode; V_{DD} = 3.6 V; f_{osc} = 12 MHz	[2]	-	3.25	5	mA
		Power-down mode, voltage comparators powered down; V_{DD} = 3.6 V	[2]	-	55	80	µA
		Total Power-down mode; V_{DD} = 3.6 V	[3]	-	1	5	µA
dV/dt	rate of change of voltage	rise rate of V_{DD}		-	-	2	mV/µs
		fall rate of V_{DD}		-	-	50	mV/µs
V_{DD}	supply voltage			2.4	3.0	3.6	V
V_{RAM}	RAM keep-alive voltage			1.5	-	-	V
$V_{th(HL)}$	HIGH to LOW threshold voltage	except pins SCL, SDA		$0.22V_{DD}$	$0.4V_{DD}$	-	V
$V_{th(LH)}$	LOW to HIGH threshold voltage	except pins SCL, SDA		-	$0.6V_{DD}$	$0.7V_{DD}$	V
V_{IL}	LOW-level input voltage	SCL, SDA only		–0.5	-	$+0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage	SCL, SDA only		$0.7V_{DD}$	-	5.5	V
V_{hys}	hysteresis voltage	port 1		-	$0.2V_{DD}$	-	V
V_{OL}	LOW-level output voltage	I_{OL} = 20 mA; V_{DD} = 2.4 V to 3.6 V all ports, all modes except high-Z	[4]	-	0.6	1.0	V
		I_{OL} = 3.2 mA; V_{DD} = 2.4 V to 3.6 V all ports, all modes except high-Z		-	0.2	0.3	V
V_{OH}	HIGH-level output voltage	I_{OH} = –20 µA; V_{DD} = 2.4 V to 3.6 V; all ports, quasi-bidirectional mode		V_{DD} – 0.3	V_{DD} – 0.2	-	V
		I_{OH} = –3.2 mA; V_{DD} = 2.4 V to 3.6 V; all ports, push-pull mode		V_{DD} – 0.7	V_{DD} – 0.4	-	V
		I_{OH} = –10 mA; V_{DD} = 3.6 V; all ports, push-pull mode		-	3.2	-	V
$V_{DD(xtal)}$	crystal supply voltage	on pins XTAL1, XTAL2; with respect to V_{SS}		–0.5	-	+4.0	V
V_n	voltage on any other pin	except pins XTAL1, XTAL2, V_{DD}; with respect to V_{SS}	[5]	–0.5	-	+5.5	V
C_i	input capacitance		[6]	-	-	15	pF
I_{IL}	LOW-level input current	logical 0; V_I = 0.4 V	[7]	-	-	–80	µA
I_{LI}	input leakage current	V_I = V_{IL}, V_{IH} or $V_{th(HL)}$	[8]	-	-	±10	µA
$I_{T(HL)}$	HIGH to LOW transition current	all ports; V_I = 1.5 V at V_{DD} = 3.6 V	[9]	–30	-	–450	µA

Table 38. Static characteristics *…continued*

V_{DD} = 2.4 V to 3.6 V; T_{amb} = −40 °C to +85 °C for industrial application; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ[1]	Max	Unit
$R_{pu(int)}$	internal pull-up resistance	pin RST_N		10	-	30	kΩ
V_{bo}	brownout voltage	2.4 V < V_{DD} < 3.6 V; with BOE = 1, BOPD = 0	[10]	2.40	-	2.70	V
$V_{ref(bg)}$	band gap reference voltage			1.11	1.23	1.34	V
TC_{bg}	band gap temperature coefficient			-	10	20	10^{-6}/°C
T_{amb}	ambient temperature			−40	-	+85	°C

[1] Typical ratings are not guaranteed. The values listed are at room temperature and V_{DD} = 3.3 V.

[2] The I_{DD} in Operating mode, Idle mode and Power-down mode specifications are measured using an external clock with the following functions disabled: comparators, real-time clock and watchdog timer.

[3] The I_{DD} total Power-down mode specification is measured using an external clock with the following functions disabled: comparators, real-time clock, brownout detect and watchdog timer.

[4] See Section 9 “Limiting values” on page 32 for steady state (non-transient) limits on I_{OL} or I_{OH}. If I_{OL}/I_{OH} exceeds the test condition, V_{OL}/V_{OH} may exceed the related specification.

[5] This specification can be applied to pins which have an A/D input or analog comparator input functions and when the pin is not used for those analog functions. When the pin is used as an analog input pin, the maximum voltage on the pin must be limited to 4.0 V with respect to V_{SS}.

[6] Pin capacitance is characterized but not tested.

[7] Measured with port in quasi-bidirectional mode.

[8] Measured with port in high-impedance mode.

[9] Port pins source a transition current when used in quasi-bidirectional mode and externally driven from HIGH to LOW. This current is highest when V_I is approximately 2 V.

[10] BOE is brownout enabled and BOPD is the state of brownout detection (0 = active).

11. Dynamic characteristics

Table 39. Dynamic characteristics (12 MHz)

V_{DD} = 2.4 V to 3.6 V; T_{amb} = −40 °C to +85 °C for industrial applications; f_{osc} = 12 MHz (crystal); unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Glitch filter						
t_{gr}	glitch rejection time	pin RST_N	-	-	50	ns
		any pin except RST_N	-	-	15	ns
t_{sa}	signal acceptance time	pin RST_N	125	-	-	ns
		any pin except RST_N	50	-	-	ns
I²C-bus: pins SDA and SCL; 5 V tolerant						
f_{clk}	clock frequency	Standard mode	-	-	100	kHz
		Fast mode	-	-	400	kHz

12. Application information

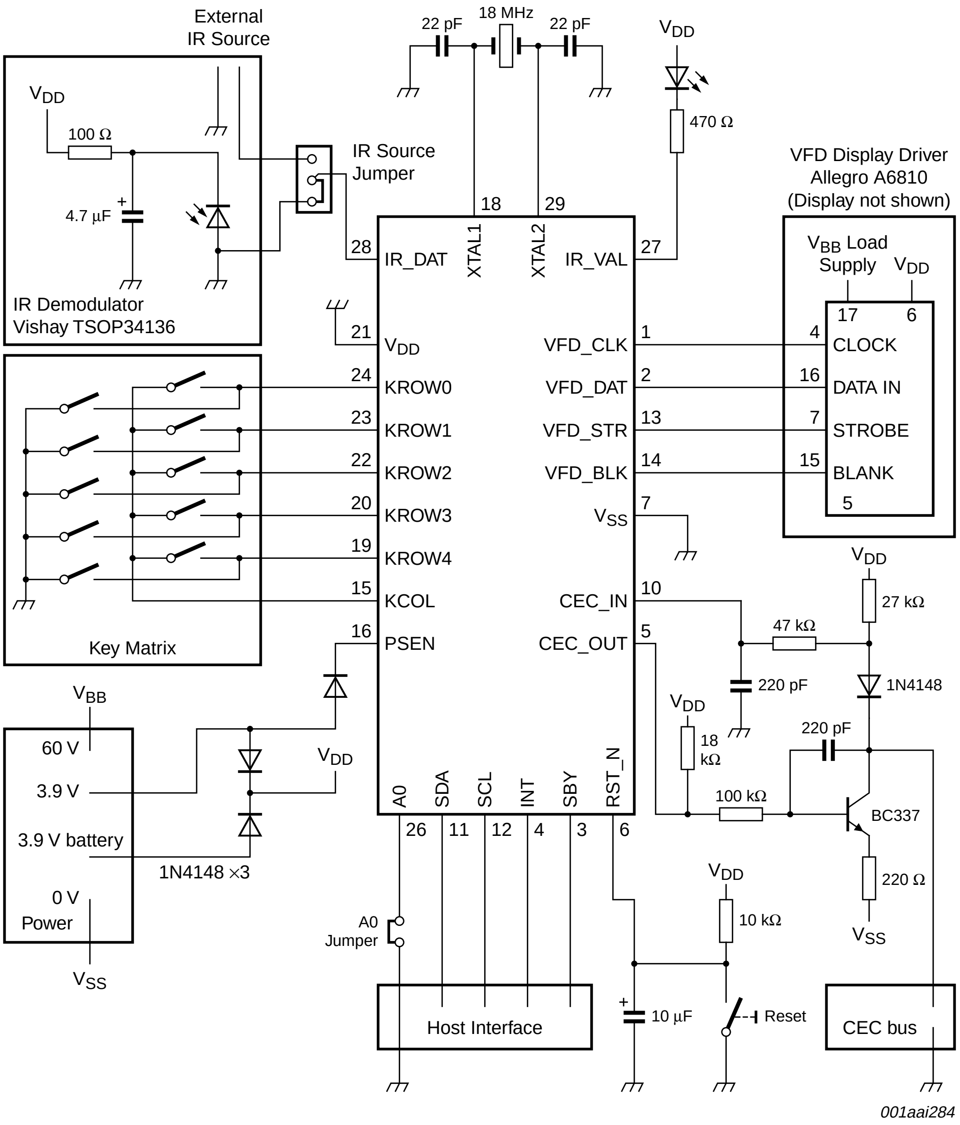
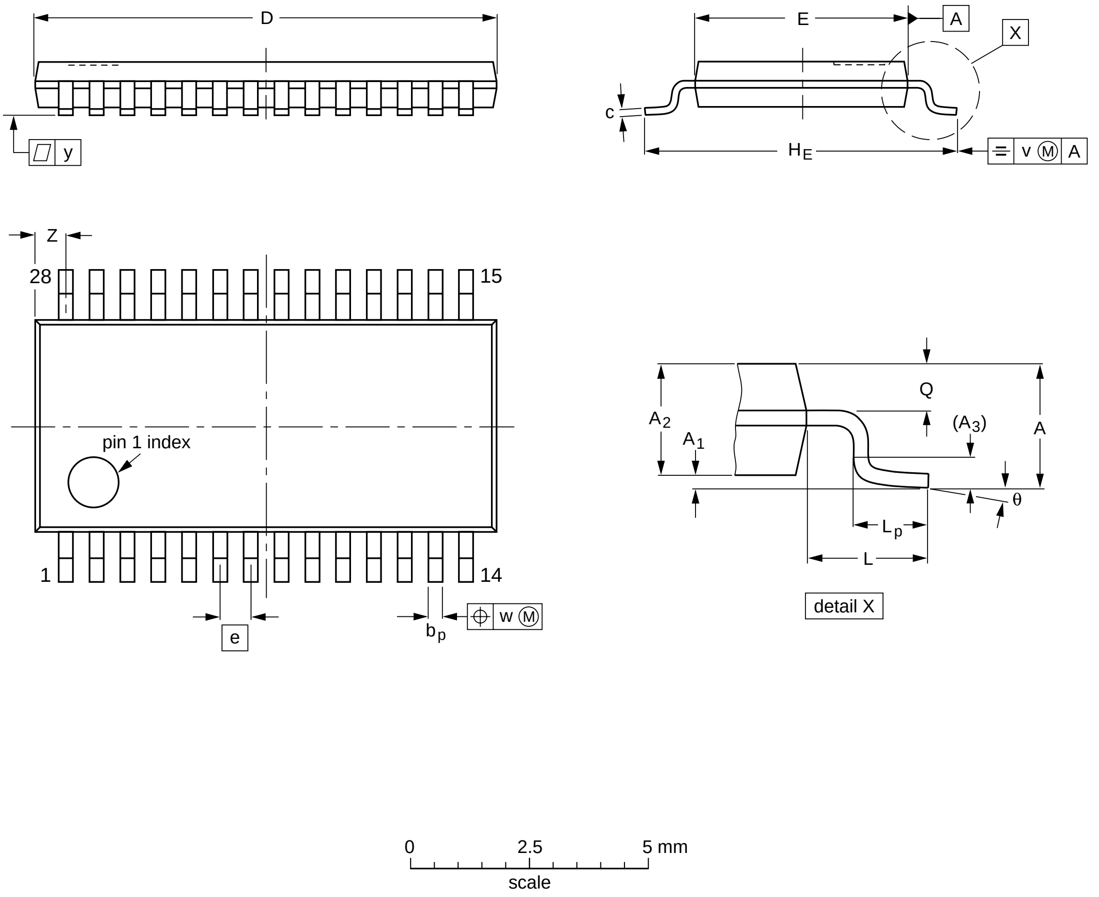


Fig 15. Application diagram

13. Package outline

TSSOP28: plastic thin shrink small outline package; 28 leads; body width 4.4 mm **SOT361-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	9.8 9.6	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.8 0.5	8° 0°

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT361-1		MO-153				~~99-12-27~~ 03-02-19

Fig 16. Package outline SOT361-1 (TSSOP28)

14. Abbreviations

Table 40. Abbreviations

Acronym	Description
CE	Consumer Electronics
CEC	Consumer Electronics Control
DVD	Digital Versatile Disc
EOM	End Of Message
FPP	Front Panel Processor
HDMI	High-Definition Multimedia Interface
IR	InfraRed
RGB	Red Green Blue
RTC	Real Time Clock
TV	TeleVision
VFD	Vacuum Fluorescent Display
YCbCr	Y = Luminance, Cb = Chroma blue, Cr = Chroma red

15. Revision history

Table 41. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
TDA9951_1	20080807	Product data sheet	-	-

16. Legal information

17. Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term ‘short data sheet’ is explained in section “Definitions”.

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

17.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

17.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

17.3 Licenses

ICs with RC5 functionality
Purchase of an NXP Semiconductors IC with RC5 functionality does not convey an implied license under any trade secret, copyright, know-how or patent right to use this IC in any RC5 application. A license under applicable rights of Koninklijke Philips Electronics N.V. needs to be obtained via Philips Intellectual Property and Standards (www.ip.philips.com), e-mail: info.licensing@philips.com.

ICs with RC6 functionality
Purchase of an NXP Semiconductors IC with RC6 functionality does not convey an implied license under any trade secret, copyright, know-how or patent right to use this IC in any RC6 application. A license under applicable rights of Koninklijke Philips Electronics N.V. needs to be obtained via Philips Intellectual Property and Standards (www.ip.philips.com), e-mail: info.licensing@philips.com.

17.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I^{2}C-bus — logo is a trademark of NXP B.V.

18. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

19. Tables

Table 1. Quick reference data 2
Table 2. Ordering information 2
Table 3. Pin description 4
Table 4. I2C-bus slave address. .6
Table 5. I2C Register configuration 6
Table 6. APR - Address pointer register (address 00h) bit description 7
Table 7. CSR - Common status register (address 00h) bit description 7
Table 8. CER - Common error register (address 01h) bit description 7
Table 9. CVR - Common version register (address 02h) bit description 8
Table 10. CCR - Common control register (address 03h) bit description 8
Table 11. ACKH - CEC address ACK high register (address 04h) bit description 8
Table 12. ACKL - CEC address ACK low register (address 05h) bit description 9
Table 13. ADDR[14:0] definition 9
Table 14. CCONR - Common configuration register (address 06h) bit description 9
Table 15. Data services 10
Table 16. Frame Bytes for CECData.req service 11
Table 17. Frame Bytes for CECData.cnf service 11
Table 18. Frame Bytes for CECData.ind service 12
Table 19. Frame Bytes for CECData.err service 12
Table 20. Frame Bytes for CECData.ier service 12
Table 21. Protocol discrimination 17
Table 22. Frame Bytes for IRData.ind service 17
Table 23. Key matrix 18
Table 24. Frame Bytes for KEYData.req service 18
Table 25. Frame Bytes for KEYData.cnf service 19
Table 26. Frame Bytes for KEYData.ind service 20
Table 27. Frame Bytes for RTCData.req service 21
Table 28. Frame Bytes for RTCData.cnf service 22
Table 29. Frame Bytes for RTCData.ind service 23
Table 30. Standby mode changes 24
Table 31. Frame Bytes for SBYData.req service 24
Table 32. Frame Bytes for SBYData.cnf service 25
Table 33. Multiplex drive timing .29
Table 34. Frame Bytes for VFDData.req service - configuration 30
Table 35. Frame Bytes for VFDData.req service segment data 31
Table 36. Frame Bytes for VFDData.cnf service 32
Table 37. Limiting values 32
Table 38. Static characteristics .33
Table 39. Dynamic characteristics (12 MHz) 34
Table 40. Abbreviations 37
Table 41. Revision history 37

20. Figures

Fig 1. Block diagram 3
Fig 2. Pin configuration 4
Fig 3. Frame format for the data register protocol 10
Fig 4. Host reads TDA9951 status register after setting address pointer. .14
Fig 5. Host reads TDA9951 status register without setting address pointer (pointer was at 0 already) 14
Fig 6. Host reads address ACK registers after setting address pointer 14
Fig 7. Host writes configuration register after setting address pointer 15
Fig 8. Host writes last three data registers 15
Fig 9. VFD driver device 26
Fig 10. Static drive 26
Fig 11. Multiplexed drive 27
Fig 12. Static drive 28
Fig 13. Grid bits clocked out after the segment bits 28
Fig 14. Output of four multiplex cycles comprising four characters 29
Fig 15. Application diagram. 35
Fig 16. Package outline SOT361-1 (TSSOP28) 36

continued >>

21. Contents

1 General description . . . 1
2 Features . . . 1
3 Applications . . . 2
4 Quick reference data . . . 2
5 Ordering information . . . 2
6 Block diagram . . . 3
7 Pinning information . . . 4
7.1 Pinning . . . 4
7.2 Pin description . . . 4
8 Functional description . . . 5
8.1 Device addressing . . . 5
8.2 Configuring the TDA9951 . . . 6
8.3 Using the INT line . . . 7
8.4 Register descriptions . . . 7
8.5 Data register protocol . . . 9
8.5.1 CECData.req service . . . 11
8.5.2 CECData.cnf service . . . 11
8.5.3 CECData.ind service . . . 12
8.5.4 CECData.err service . . . 12
8.5.5 CECData.ier service . . . 12
8.6 Example communication sequences . . . 13
8.6.1 Notes on writing the CEC common data registers . . . 13
8.6.2 Notes on reading the CEC common data registers . . . 13
8.6.3 Communication use cases . . . 14
8.7 I^{2}C command examples . . . 15
8.7.1 Initialization . . . 15
8.7.2 Sending CEC messages . . . 16
8.7.3 Receiving CEC messages . . . 16
8.8 Infrared receiver interface . . . 16
8.8.1 Infrared hardware . . . 16
8.8.2 Protocol discrimination . . . 17
8.8.3 IRData.ind service . . . 17
8.9 Key matrix interface . . . 17
8.9.1 Key matrix hardware . . . 17
8.9.2 Key matrix decoding . . . 17
8.9.3 Key matrix encoding . . . 18
8.9.4 KEYData.req service . . . 18
8.9.5 KEYData.cnf service . . . 18
8.9.6 KEYData.ind service . . . 19
8.10 Real time clock interface . . . 20
8.10.1 Real time clock hardware . . . 20
8.10.2 Clock-calendar . . . 20
8.10.3 Alarm . . . 20
8.10.4 Timers . . . 21
8.10.5 RTCData.req service . . . 21
8.10.6 RTCData.cnf service . . . 22
8.10.7 RTCData.ind service . . . 23
8.11 Standby interface . . . 23
8.11.1 Standby hardware . . . 23
8.11.2 SBYData.req service . . . 24
8.11.3 SBYData.cnf service . . . 25
8.12 Vacuum fluorescent display interface . . . 25
8.12.1 VFD hardware . . . 25
8.12.2 VFD driver connection topologies . . . 26
8.12.2.1 VFD driver device . . . 26
8.12.2.2 Static drive . . . 26
8.12.2.3 Multiplexed drive . . . 27
8.12.3 Host request bit mapping to clocked output bits . . . 27
8.12.3.1 Static drive . . . 27
8.12.3.2 Multiplexed drive . . . 28
8.12.4 Multiplex drive timing . . . 29
8.12.5 Multiplex drive pitfalls . . . 30
8.12.6 VFDData.req service configuration . . . 30
8.12.7 VFDData.req service segment data . . . 31
8.12.8 VFDData.cnf service . . . 31
9 Limiting values . . . 32
10 Static characteristics . . . 33
11 Dynamic characteristics . . . 34
12 Application information . . . 35
13 Package outline . . . 36
14 Abbreviations . . . 37
15 Revision history . . . 37
16 Legal information . . . 38
17 Data sheet status . . . 38
17.1 Definitions . . . 38
17.2 Disclaimers . . . 38
17.3 Licenses . . . 38
17.4 Trademarks . . . 38
18 Contact information . . . 39
19 Tables . . . 40
20 Figures . . . 40
21 Contents . . . 41

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section ‘Legal information’.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 7 August 2008

founded by
PHILIPS