

Features

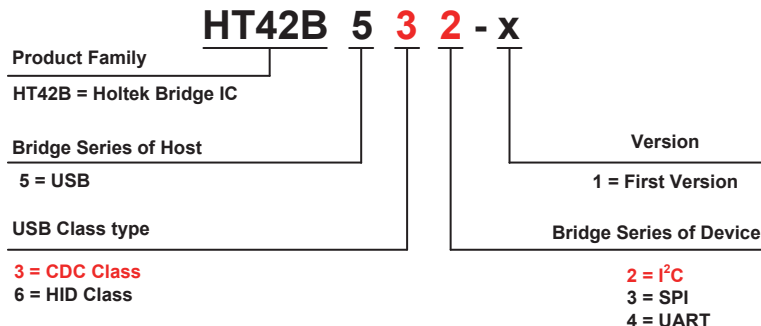
- Operating Voltage (V_{DD}): 3.3V~5.5V
- I²C pins Voltage (V_{DDIO}): 1.8V~ V_{DD} (Less than V_{DD} voltage)
- Fully integrated internal 12MHz oscillator with 0.25% accuracy for all USB modes which requires no external components
- USB interface
 - USB 2.0 Full Speed compatible
 - Implements USB protocol composite device:
 - Communication Device Class (CDC) for communications and configuration.
 - Human Interface Device (HID) for user configure USB VID, PID and device description strings
 - Integrated an internal 1.5k Ω pull-high resistor on D+ pin
- Serial Interface – I²C
 - Supports clock rate up to 400kHz
 - Supports Master and Slave modes decided by AP command
 - Supports maximum 62 bytes transmit buffer and 62 bytes receive buffer
 - Supports SDA (Master mode) and SCL or SDA (Slave mode) pins resume signal to request a remote wake-up
 - Supports VDDIO pin for I²C and A0~A1 pins power supply

- Support standard Windows® drivers for Virtual Com Port (VCP): Windows XP (SP2), Vista, Windows 7, Windows 8, Windows 8.1 (only an INF file is required) and Windows 10.
- Support Android 4.0 or later version and Mac OS X
- Integrated 256 bytes internal true EEPROM for user memory
- Power down and wake-up functions to reduce power consumption
- Package types: 8-pin SOP, 10-pin MSOP

General Description

The HT42B532-x is a high performance USB to I²C bridge controller with fully integrated USB and I²C interface functions, designed for applications that communicate with various types of I²C. The device includes a USB 2.0 full speed compatible interface which is used for PC communication. The device also includes a fully integrated high speed oscillator which is used for USB and I²C clock generator.

USB Bridge IC Naming Rules

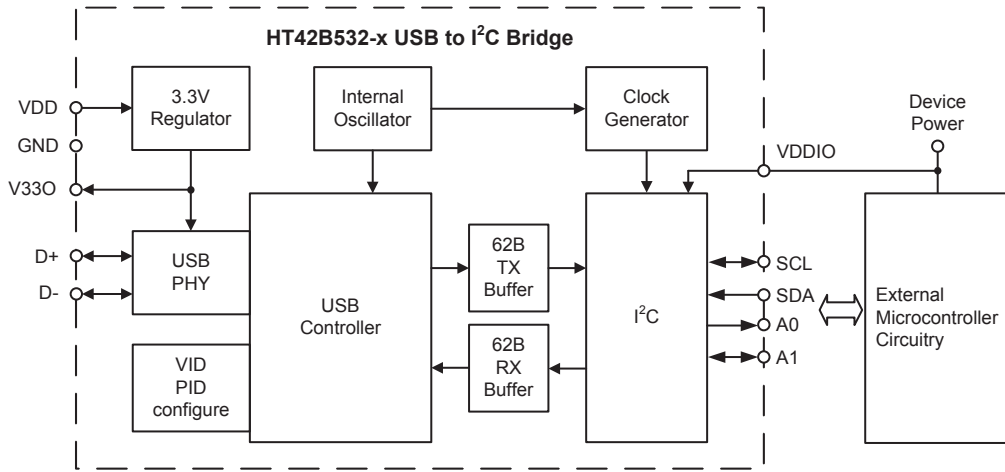


Selection Table

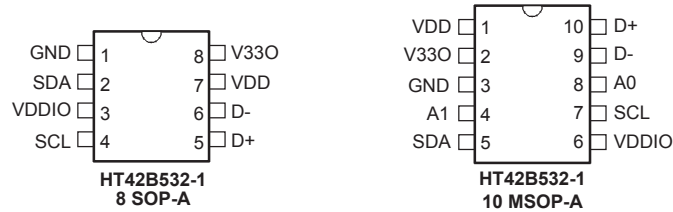
Most features are common to all devices. The following table summarises the main features of each device.

Part No.	Description	V _{DD}	USB	Virtual COM	HID	FIFO/Buffer	Interface Data Rate	I/O V _{DD}	Package
HT42B532-x	USB to I ² C Bridge	3.3V~5.5V	Full-Speed	√	–	TX: 62 bytes RX: 62 bytes	Up to 400kHz	√	8SOP 10MSOP
HT42B533-x	USB to SPI Bridge			√	–	TX: 128 bytes RX: 128 bytes	Up to 8MHz	√	10MSOP 16NSOP
HT42B534-x	USB to UART Bridge			√	–	TX: 128 bytes RX: 128 bytes	Up to 3Mbps Baud	√	8/10SOP 10MSOP 16NSOP
HT42B564-x	USB (HID) to UART Bridge			–	√	TX: 32 bytes RX: 32 bytes	Up to 115.2kbps Baud	√	10SOP

Block Diagram



Pin Assignment



Package type	Marking
8SOP	HT42B532-x
10MSOP	B532-x

Note: x=1 for version number.

Pin Descriptions

As the Pin Description table applies to the package type with the most pins, not all of the listed pins may be present on package types with smaller numbers of pins.

Pin Name	Type	Description
D+	I/O	USB D+ Line
D-	I/O	USB D- Line
SCL	I/O	I ² C clock line
SDA	I/O	I ² C data/address line
A0~A1	O	Master mode address configuration pins
V33O	O	3.3V Regulator output
VDDIO	PWR	Positive power supply for the SCL, SDA, A0, A1 pins
VDD	PWR	Positive power supply, USB bus power
GND	PWR	Negative power supply, ground

Absolute Maximum Ratings

Supply Voltage	$V_{SS}-0.3V$ to $V_{SS}+6.0V$	I_{OH} Total	-80mA
Input Voltage	$V_{SS}-0.3V$ to $V_{DD}+0.3V$	I_{OL} Total	80mA
Storage Temperature	-50°C to 125°C	Total Power Dissipation	500mW
Operating Temperature	-40°C to 85°C		

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to this device. Functional operation of these devices at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

D.C. Characteristics

 $T_a=25^{\circ}C$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{DD}	Operating Voltage	—	—	3.3	—	5.5	V
V _{DDIO}	VDDIO Input Voltage for I ² C Pins	—	—	1.8	—	V _{DD}	V
I _{DD}	Operating Current	5V	No load	—	11	16	mA
I _{SUS}	Suspend Current (USB)	5V	Suspend mode, No load, USB on, other peripherals off	—	360	450	μA
V _{IL}	Input Low Voltage for Input Pins	—	—	0	—	0.2V _{DDIO}	V
V _{IH}	Input High Voltage for Input Pins	—	—	0.8V _{DDIO}	—	V _{DDIO}	V
I _{OL}	Sink Current for I/O Pins	3V	V _{OL} = 0.1V _{DDIO}	4	8	—	mA
		5V		10	20	—	mA
I _{OH}	Source Current for I/O Pins	3V	V _{OH} = 0.9V _{DDIO}	-2	-4	—	mA
		5V		-5	-10	—	mA
R _{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ
I _{LEAK}	Input Leakage Current	3V	V _{IN} = V _{DD} or V _{IN} = V _{SS}	—	—	±1	μA
		5V		—	—	±1	μA
V _{V33O}	3.3V Regulator Output Voltage	5V	I _{V33O} = 70mA	3.0	3.3	3.6	V
R _{UDP1}	Pull-high Resistance between D+ and V33O	3.3V	—	-5%	1.5	+5%	kΩ

A.C. Characteristics

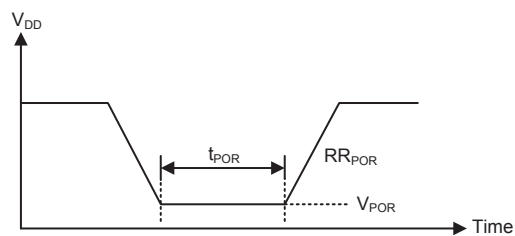
Ta=25°C

Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V _{DD}	Condition				
f _{HIRC}	High Speed Internal RC Oscillator	3.3V~5.5V	USB mode	-0.25%	12	+0.25%	MHz
t _{SST}	System Start-up Timer Period	—	I ² C pins Wake-up from power down mode SCL(Master mode)/SCL or SDA (Slave mode)	16	—	—	t _{HIRC}
t _{RSTD}	System Reset Delay Time	—	Power-on reset	25	50	100	ms

Power-on Reset Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{POR}	V _{DD} Start Voltage to Ensure Power-on Reset	—	—	—	—	100	mV
RR _{POR}	V _{DD} Rising Rate to Ensure Power-on Reset	—	—	0.035	—	—	V/ms
t _{POR}	Minimum Time for V _{DD} Stays at V _{POR} to Ensure Power-on Reset	—	—	1	—	—	ms



USB Interface

The USB interface, being USB 2.0 full-speed compatible, is a 4-wire series bus that allows communication between a host device and up to 127 max peripheral devices on the same bus. A token based protocol method is used by the host device for communication control. Other advantages of the USB bus include live plugging and unplugging and dynamic device configuration. As the complexity of USB data protocol does not permit comprehensive USB operation information to be provided in this datasheet, the reader should therefore consult other external information for a detailed USB understanding. The device includes a USB interface function allowing for the convenient design of USB peripheral products.

Power Plane

There are two power planes for the device and they are the USB bus power input (V_{DD}) and 3.3V regulator output (V_{V330}).

For the USB SIE VDD, it will supply power for all circuits related to USB SIE and is sourced from pin "VDD". Once the USB is removed from the USB and there is no power in the USB BUS, the USB SIE circuit is no longer operational.

USB Interface Operation

To communicate with an external USB host, the internal USB module has the external pins known as D+ and D- along with the 3.3V regulator output pin V330. A Serial Interface Engine (SIE) decodes the incoming USB data stream and transfers it to the correct endpoint buffer memory known as the FIFO. The USB module has 4 endpoints, EP0 ~ EP3. The endpoint 0 supports the Control transfer while the endpoint 1 ~ endpoint 3 support the Interrupt or Bulk transfer. The HT42B532-x Bridge IC supports the USB Communication Device Class (CDC) for communications and configuration.

Endpoint	Transfer Type
0	Control
1	Interrupt
2	Bulk Out
3	Bulk In

USB Endpoint Transfer Type

If there is no signal on the USB bus for over 3ms, the USB device will enter the suspend mode. The device enters the suspend state to meet the requirements of the USB suspend current specification. When the resume signal is asserted by the USB host, the device will be woken up and leave the suspend mode.

As the USB device has a remote wake-up function, the USB device can wake up the USB host by sending a remote wake-up pulse. Once the USB host receives a remote wake-up signal from the USB device, the host will send a resume signal to device.

USB VID and PID Configure

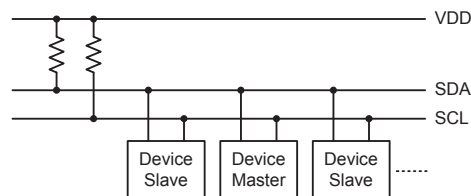
The device has configured the default Vendor ID (VID:0x04D9), Product ID (PID:0xB532) and product description strings of "USB to I²C Bridge". The user can update Vendor ID, Product ID, product description strings and remote wake-up setting using their application programs.

This device has been configured to the default USB configuration data as shown in the following table.

Parameter	Value (Hex)
USB Vendor ID (VID)	0x04D9
USB Product ID (PID)	0xB532
Remote wake-up	Default disable
Manufacturer Name	Holtek
Product Description	USB to I ² C Bridge
Serial Number	0000

I²C Interface

The HT42B532-x contains an I²C function. The I²C interface is often used to communicate with external peripheral devices such as microcontrollers, sensors, EEPROM memory etc. Originally developed by Philips, it is a two line low speed serial interface for synchronous serial data transfer. The advantage of only two lines for communication, relatively simple communication protocol and the ability to accommodate multiple devices on the same bus has made it an extremely popular interface type for many applications.

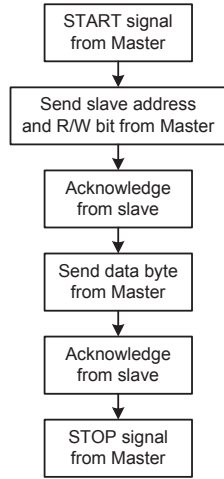


I²C Master/Slave Bus Connection

I²C Interface Operation

The I²C serial interface is a two line interface, a serial data line, SDA, and serial clock line, SCL. As many devices may be connected together on the same bus, their outputs are both open drain types. For this reason it is necessary that external pull-high resistors are connected to these outputs. Note that no chip select line exists, as each device on the I²C bus is identified by a unique address which will be transmitted and received on the I²C bus.

When two devices communicate with each other on the bidirectional I²C bus, one is known as the master device and one as the slave device. Both master and slave can transmit and receive data, however, it is the master device that has overall control of the bus. For these devices, which only operate in slave mode, there are two methods of transferring data on the I²C bus, the master transmit mode and the slave receive mode.



The I²C Serial Interface function includes the following features:

- Both Master and Slave mode
- Master mode serial clock frequency up to 400kHz
- 62-byte Deep FIFO Transmit Data Buffer
- 62-byte Deep FIFO Receive Data Buffer
- SCL pin (Master mode) wake-up function
- SCL or SDA pins (Slave mode) wake-up function.
- I²C and A0~A1 pins power supply by the VDDIO pin

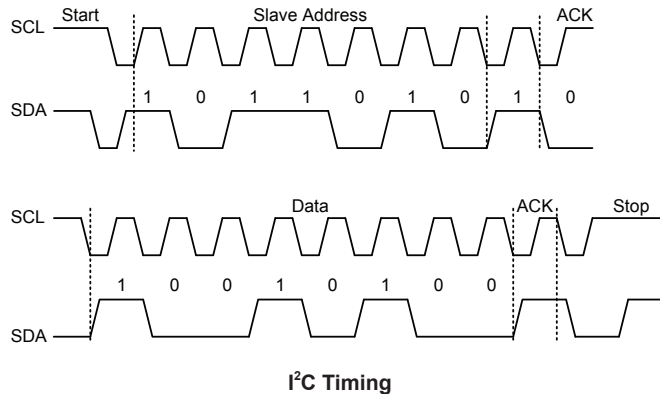
I²C Communication

Communication on the I²C bus requires four separate steps, a START signal, a slave device address

transmission, a data transmission and finally a STOP signal. When a START signal is placed on the I²C bus, all devices on the bus will receive this signal and be notified of the imminent arrival of data on the bus. The first seven bits of the data will be the slave address with the first bit being the MSB. If the address of the slave device matches that of the transmitted address, the communication will be generated. During a data transfer, note that after the 7-bit slave address has been transmitted, the following bit, which is the 8th bit, is the read/write bit. The 8th bit will be checked by the slave device to determine whether to go into transmit or receive mode. If the 8th bit is “1” then this indicates that the master device wishes to read data from the I²C bus, therefore the slave device must be setup to send data to the I²C bus as a transmitter. If the 8th bit is “0” then this indicates that the master wishes to send data to the I²C bus, therefore the slave device must be setup to read data from the I²C bus as a receiver. Before any transfer of data to or from the I²C bus, the application program (AP) must firstly initialise the HT42B532-x.

The transmitted data is 8-bits wide and is transmitted after the slave device has acknowledged receipt of its slave address. The order of serial bit transmission is the MSB first and the LSB last. After receipt of 8-bits of data, the receiver must transmit an acknowledge signal, level “0”, before it can receive the next data byte. If the slave transmitter does not receive an acknowledge bit signal from the master receiver, then the slave transmitter will release the SDA line to allow the master to send a STOP signal to release the I²C Bus. The corresponding data will be stored in the 62-byte FIFO.

When the slave receiver receives the data byte, it must generate an acknowledge bit on the 9th clock. The slave device, which is setup as a transmitter will check the 9th bit to determine if it is to send another data byte, if not then it will release the SDA line and await the receipt of a STOP signal from the master.



I²C Timing

I²C Clock

The SCL pin clock for the Master mode can be set using the Holtek Bridge API command to define the desired value, as shown below.

I ² C Mode	SCL Clock	Value (Hex)
Master	400KHz	00
Master	300KHz	01
Master	200KHz	02
Master	100KHz	03
Master	75KHz	04

I²C Power Down and Wake-up

If the USB host sends a suspend signal to the HT42B532-x USB device, it will enter the suspend mode. It is recommended to ensure that the I²C data transmission or reception has been finished before the device enters the suspend mode.

The I²C function contains the SDA pin (master mode), SCL or SDA pins (slave mode) wake-up functions. A falling edge on the SCL/SDA pin will wake up the device from the suspend mode.

Holtek Bridge DLL User Guideline

Holtek USB Bridge Program

Holtek has provided the DLL to build the HT42B532-x/HT42B533-x Bridge IC application programs for USB to I²C or USB to SPI data communication. The API descriptions are described as below.

- **HTB_API BOOL OpenDevice(int nCom);**
Holtek Bridge operates in the format of Virtual COM Port. Define the COM port number using the parameter.
- **HTB_API void CloseDevice();**
Used to turn off the Bridge Device.
- **HTB_API BOOL SetIICDataRate(int nDR);**
Used to configure the I²C Data Rate (for USB to I²C Bridge). Refer to HoltekBridgeDLL.h for parameter definition.
- **HTB_API BOOL SetSPIDataRate(int nDR);**
Used to configure the SPI Data Rate (for USB to SPI Bridge). Refer to HoltekBridgeDLL.h for parameter definition.
- **HTB_API BOOL SetSPIMode(int nMode, int nOrder, int nCSB);**
 - Refer to HoltekBridgeDLL.h for parameter definition.
 - Used to configure SPI mode 0/1/2/3, LSB/MSB, and whether to use CSB (for USB to SPI Bridge).
 - When in the SPI Slave Mode, CSB must be enabled.
- **HTB_API BOOL SetIICMode(int nMode, int nAddr);**
 - Used to configure I²C Master or Slave mode and address.
- **HTB_API BOOL SetIIC_ReceiverEnd(BOOL bAck);**
 - Used to configure when the Master Receiver ends transmission, I²C returns ACK or NACK.
bAck: When the data length defined by SetDirection or BRRead has been received, I²C will return ACK or NACK.
TRUE: ACK
FALSE: NACK
 - After the OpenDevice action, this parameter's default status is NACK. The parameter should be configured after OpenDevice and before Read/Write actions.
- **HTB_API BOOL SetIIC_Restart(BOOL bRestart);**
 - Used to define the signal behavior when in the Master Mode.
bRestart: When the data length defined by SetDirection or BRRead has been received or transmitted, I²C will generate a STOP signal or Restart signal.
TRUE: Restart
FALSE: STOP
 - After the OpenDevice action, this parameter's default status is STOP. The parameter should be configured before Read/Write actions.
- **HTB_API BOOL BRRead(char *p, DWORD nLen, DWORD&BytesRead, DWORD dwTimeOut);**
 - Read
If the I²C Receiver returns NACK, the return value is FALSE;
“BytesRead” indicates the actual byte count being read, if it is not enough, continue call Read function.
“dwTimeOut” indicates the waiting time for read, unit: ms.

- **HTB_API BOOL BRWrite(char *p, DWORD nLen);**
 - Write
If the I²C Receiver returns NACK, the return value is FALSE;
- **HTB_API BOOL Finalize();**
 - To end the current transmission.
 - When in the SPI master mode and CSB is enabled, calling this function will pull CSB high.
 - When in the SPI slave mode, calling this function will reset the bridge to its Receiver default status.
- **HTB_API BOOL ResetDevice();**
 - Reset Bridge. This action will clear the contents already stored in the Bridge FIFO.
- **HTB_API BOOL SlaveCSBFalling();**
 - When in the SPI Slave Mode, this function is used to detect whether the master has re-enabled CSB. If yes, it means the master will re-transmit commands, in which case call Finalize or ResetDevice to reset the bridge to its Receiver default status.
- **HTB_API BOOL SetDirection(BYTE ucDir, WORD ucLen);**
- **HTB_API BOOL PureRead(char* p, WORD ucLen,WORD &BytesRead,DWORD dwTimeOut);**
- **HTB_API BOOL PureWrite(char *p, WORD nLen);**
 - For both SPI and I²C, before switching between Read and Write, first to set Direction.
 - SetDirection(DIR_READ,LEN) + PureRead = BRRead
 - SetDirection(DIR_WRITE,LEN) + PureWrite = BRWrite
 - The SetDirection function defines the total length "LEN", PureRead or PureWrite supports any length but their total length can not larger than "LEN".
 - "dwTimeOut" indicates the waiting time for read, unit: ms.
- **HTB_API BOOL SetGPIOWakeUp(BYTE);**
 - Used to configure GPIO0/GPIO1/GPIO2/GPIO3 with or without wake up function.
 - This parameter is transmitted in the format of OR. For example, to enable GPIO0 and GPIO2 wake up functions, the parameter is set as shown below:
SetGPIOWakeUp(GPIO0|GPIO2);
- **HTB_API BOOL SetGPIOPullUp(BYTE);**
 - Used to configure GPIO0/GPIO1/GPIO2/GPIO3 with or without pull-high function.
 - This parameter is set in the same way as the previous one.
- **HTB_API BOOL SetGPIOInput(BYTE);**
 - Used to configure GPIO0/GPIO1/GPIO2/GPIO3 input/output direction.
 - SetGPIOInput(GPIO1|GPIO2) indicates GPIO1/GPIO2 are input, GPIO0/GPIO3 are output.
- **HTB_API BOOL SetPWM(BOOL bEnable,PWM *pwm=NULL);**
 - Used to configure the desired PWM value, then the PWM signal will be generated on GPIO3.
 - Refer to the associated Bridge user manual for more details about the setup value.
The PWM structure is described as below:

bPeriod	defines Period width, unit: clock
bClock	defines PWM clock frequency
bActiveLevel	defines active low or active high
bOutputMode	defines PWM signal output mode
bOpMode	defines PWM operating mode
wDuty	defines Duty width, unit: clock

Example
Setup PWM

```
PWM pwm;
pwm.bPeriod=PD_1024_CLK;
pwm.bClock=CLK_3M;
pwm.bActiveLevel=ACTIVE_LOW;
pwm.bOutputMode=PWM_OUTPUT;
pwm.bOpMode=PWM_OUTPUT;
pwm.wDuty=0x80;
BOOL bRet = SetPWM(TRUE, &pwm);
```

Read device which needs a ACK response

```
DWORD dw=0;
char szBuf[9]={0x10,0,0,0,0,0,0,0,0};
char szRead[16];
BOOL bRet=OpenDevice(3);
    bRet=SetIIC_ReceiverEnd(TRUE);
bRet=SetIICDataRate(IIC_200K);
bRet=SetIICMode(IIC_MASTER,0x51);
bRet=BRWrite(szBuf,9); //Write 8 bytes of 0 to address 0x10
bRet=BRWrite(szBuf,1); //Write the read address 0x10
bRet=BRRead(szRead,16,dw,50); //data read back, response with ACK after Read
//ends

Finalize();
CloseDevice();
```

Write to/Read from Holtek EEPROM HT24LC0x – using Restart signal

```
DWORD dw=0;
char szBuf[9]={0x10,0,0,0,0,0,0,0,0};
char szRead[16];
BOOL bRet=OpenDevice(3);
bRet=SetIICDataRate(IIC_200K);
bRet=SetIICMode(IIC_MASTER,0x51);
bRet=BRWrite(szBuf,9); //Write 8 bytes of 0 to address 0x10
bRet=SetIIC_Restart(TRUE); //Set before BRWrite
bRet=BRWrite(szBuf,1); //Write the read address 0x10, generate Restart
//signal after Write ends

bRet=SetIIC_Restart(FALSE); //Set before BRRead
bRet=BRRead(szRead,16,dw,50); //data read back, response with NACK after Read
//ends and generate a STOP signal

Finalize();
CloseDevice();
```

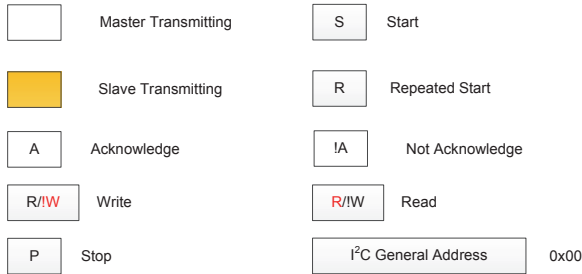
Write to/Read from SPI Flash

```
DWORD dw=0;
char szID[4]={0x90,0,0,0};
char szCmd[4]={0x03,0,0,0};
char szRead[16];
:
BOOL bRet=OpenDevice(3);
bRet=SetSPIDataRate(SPI_4M);
bRet=SetSPIMode(SPI_MODE0, SPI_MSB,SPI_EN_CSB);
bRet=BRWrite(szID,4); //read id
bRet=BRRead(szRead,2,dw,50);
Finalize();
bRet=BRWrite(szCmd,4);
bRet=BRRead(szRead,0x10,dw,50); //read 0x10 bytes from address 0x00
Finalize();
```

I²C Slave Mode Setup Description

When the I²C is configured to operate in the slave mode, it is to operate as a Slave Receiver which means to read from the USB host (BULK OUT). The I²C Master terminal should follow the protocol shown in the table for normal communications.

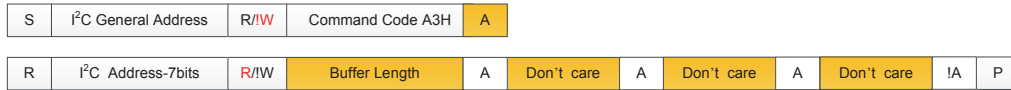
Master Transmitter (MT) Request	Master Transmitter (MT) Command Code	Slave Receiver (SR) Response
Write Data Available Request	0xA3	Buffer Length
Read Data Available Request	0xA4	Buffer Length



MT : Master Transmitter
SR : Slave Receiver
ST : Slave Transmitter

Write data : data from I²C interface to the host(BULK IN)
Read data : data from host to I²C interface(BULK OUT)

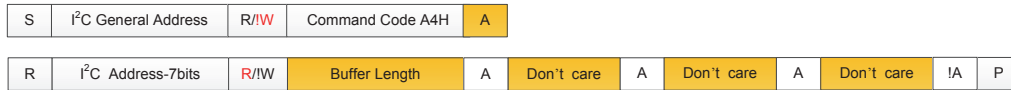
Write Data Available Request



Write Data



Read Data Available Request



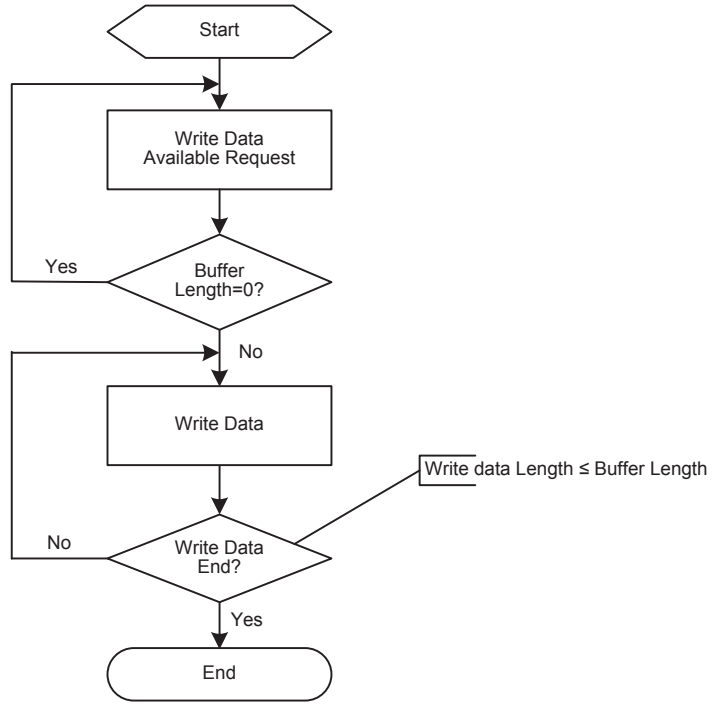
Read Data



Write Data flow (Data from I²C Interface to Host: BULK IN)

MT : Master Transmitter
SR : Slave Receiver
ST : Slave Transmitter

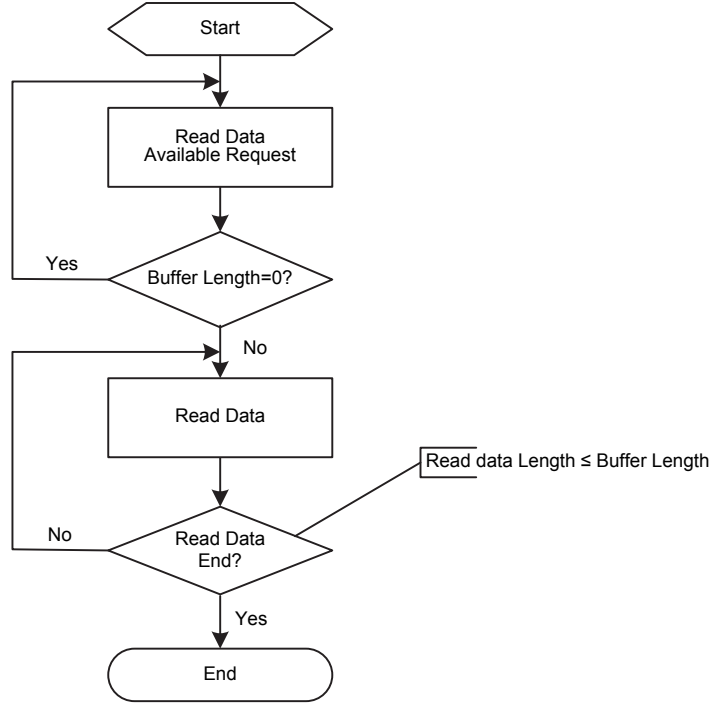
MT Write Data to SR Flow



Read Data Flow (Data from Host to I²C Interface: BULK OUT)

MT : Master Transmitter
SR : Slave Receiver
ST : Slave Transmitter

MT Read Data from SR Flow



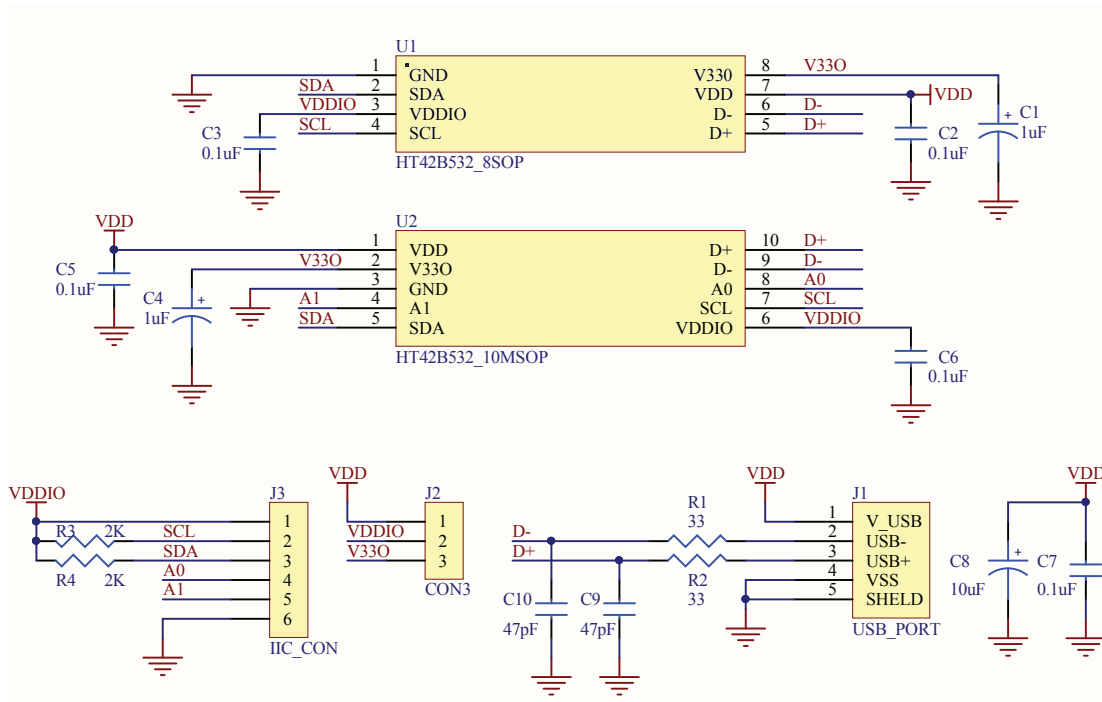
Product Description Update

To implement product description update first open the Holtek HT42B534-x Bridge IC demo AP, if the USB had been plugged into the host PC, it will show that the USB has been opened successfully on a new window. The user can use the application program to update the customer VID, PID, Manufacturer Name, Product Description, Serial Number and 256 bytes of user memory. It can configure the I²C bridge device hardware flow control and remote wake-up functions. In addition to the definable descriptions, a user memory area which is not used to store parameters is also provided for users to record data.

The configuration descriptor length table is shown as below.

Parameter	Length
USB Vendor ID(VID)	1 Word (hex)
USB Product ID(PID)	1 Word (hex)
Manufacturer Name	Support max. 16 characters
Product Description	Support max. 32 characters
Serial Number	Support max. 4 words

Application Circuits

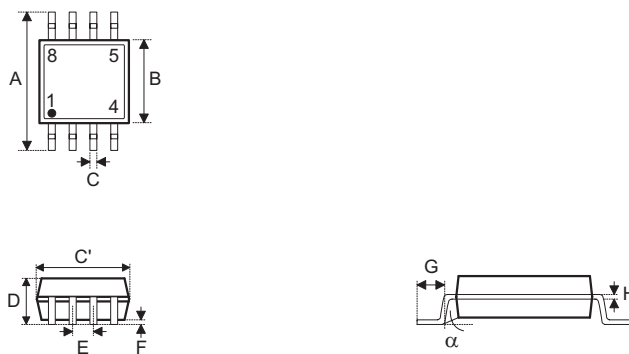


Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the [Holtek website](#) for the latest version of the [Package/ Carton Information](#).

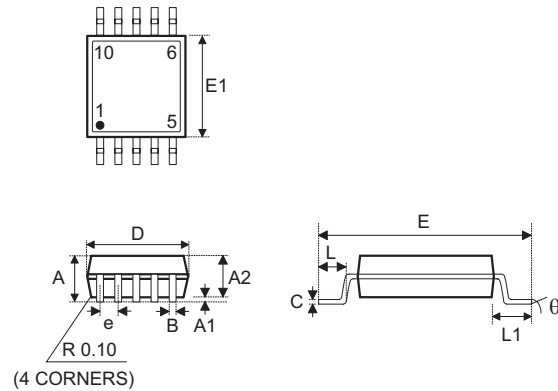
Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

- Package Information (include Outline Dimensions, Product Tape and Reel Specifications)
- The Operation Instruction of Packing Materials
- Carton information

8-pin SOP (150mil) Outline Dimensions


Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.193 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.00 BSC	—
B	—	3.90 BSC	—
C	0.31	—	0.51
C'	—	4.90 BSC	—
D	—	—	1.75
E	—	1.27 BSC	—
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°

10-pin MSOP Outline Dimensions


Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	—	0.043
A1	0.000	—	0.006
A2	0.030	0.033	0.037
b	0.007	—	0.013
c	0.003	—	0.009
D	—	0.118 BSC	—
E	—	0.193 BSC	—
E1	—	0.118 BSC	—
e	—	0.020 BSC	—
L	0.016	0.024	0.031
L1	—	0.037 BSC	—
y	—	0.004	—
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	—	1.10
A1	0.00	—	0.15
A2	0.75	0.85	0.95
b	0.17	—	0.33
c	0.08	—	0.23
D	—	3 BSC	—
E	—	4.9 BSC	—
E1	—	3 BSC	—
e	—	0.5 BSC	—
L	0.40	0.60	0.80
L1	—	0.95 BSC	—
y	—	0.1	—
α	0°	—	8°

Copyright© 2017 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com/en/>.