

TIGER560B

Application note on migrating existing Tiger560 designs to the Tiger560B

The Tiger560B is in high volume production and can directly replace the Tiger560 as it is fully software and pin and pin compatible.

Using the Tiger560B in place of a Tiger560 is simply a case of replacing the existing chip.

No PCB or software changes are required.

The additional new features provided by the Tiger560B are optional and are described below, to take advantage of the additional new features minor PCB and software changes may be required.

Introduction:

Tiger560B is pin to pin compatible with the Tiger560. For an existing Tiger560 based design, it is a direct replacement without any hardware or software modifications being required.

The Tiger560B includes USB HID support and can provide full functionality using Microsoft provided drivers that are embedded in Windows. In this situation no vendor provided drivers are required.

For USB Internet telephony products that will use the new features offered by the Tiger560B, hardware changes will be required to enable all of the new features to be used.

This document will describe the necessary hardware modification for specific new functions. Examples will be given for the VOIP handset design. For the other new designs, please refer to TigerJet Tiger560B reference design schematics.

We suggest that you email us your schematics for review (Acrobat or OrCAD); support@tjnet.com

Hardware changes required to support the USB HID interface:

Product ID[15:14] defines the device interface type. Tiger560B has two new values for the HID support. Below is the table showing the differences between the Tiger560 and Tiger560B.

Product ID [15:14]	Tiger560	Tiger560B
2'b00	Reserved	Audio + HID + Composite (Driver required)
2'b01	Audio	Same as Tiger560
2'b10	Audio + Composite (Driver required)	Same as Tiger560
2'b11	Reserved	Audio + HID (no driver required)

So, for a design that implements HID, change Product ID[15:14] to 2'b11 or 2'b00.

Using Microsoft drivers embedded in Windows – Product ID change

Previous table shows if the Product_id[15:14] is 2'b01 or 2'b11, the device will not require the vendor device driver.

It is recommend all the Tiger560B designs use the Product ID[15:14] = 2'b11.

Hardware keypad scanning – changes required

Tiger560B adds new functions for hardware keypad scanning. The Tiger560B can support keypads 4x8 using matrix scanning. The keypad scanning is performed by using 4 AUX pins for the row outputs and the 8 bit HD bus for the column inputs. The Tiger560B HD bus includes a 50K ohm pull up resistor inside the chip. The 74LS244 buffer and external pull-up resistors required by the Tiger560 are not required.

Below is the keypad matrix definition table.

Matrix	HD0	HD1	HD2	HD3	HD4	HD5	HD6	HD7
AUX0	1	2	3	Drop	Hold	Page Up	Return	Button1
AUX1	4	5	6	Send	Redial	PageDown	Home	Button2
AUX3	7	8	9	Delete	Flash	Up	Volume+	Button3
AUX4	*	0	#	Mute	Transfer	Down	Volume-	Button4

If you need to implement a key that is not defined in the table, use the Button 1 to 4 keys which can be mapped to your choice of key definition.

AUX5 can be used for a hook switch. Logic 1 is off-hook and 0 is on-hook.

To enable the automatic hardware keypad scanning, set the HA[3:2] bit according to the table below.

HA[3:2]	Function Description
2'b00	Enable Hardware keypad scanning with 4x4 matrix
2'b01	No Hardware keypad scanning
2'b10	Test Mode
2'b11	Enable Hardware keypad scanning with 4x8 matrix

Hardware change for enabling USB serial number:

Tiger560B implements a new function that will access an EEPROM (9346) after hardware reset. From the EEPROM the USB ID and serial number can be read by the Tiger560B. Before reading the data from the EEPROM the Tiger560B checks if there is pull-up on the EECS# pin and the check sum value is correct. Below are the connections for the EEPROM.

EEPROM(9346)	Tiger560B
CS	EECS#
SK	PID_13
DI	PID_12
DO	AUX_7

Below is the table for the EEPROM layout

Word Offset	Description
0	Must set to 0x0000
1	Must set to 0x0000
2	USB Vendor ID
3	USB Product ID
4	Serial number word 0
5	Serial number word 1
6	Serial number word 2
7	Serial number word 3
8	Serial number word 4
9	Serial number word 5
10	SUM from Byte 0 to Byte 17 + 0x1234

Examples:

Example 1: Migrate a Tiger560 based USB phone design to Tiger560B and enable HID support together with keypad scanning and use the Microsoft drivers embedded in Windows.

- Step1: Change Product ID to 0xF210 if GCI CODEC (Motorola or Winbond) is used.
Change Product ID to 0xF250 if Long Frame CODEC(Oki or Conwise) is used.
- Step2: Reroute the keypad matrix based on the Tiger560B keypad matrix definition. If the required keypad is not shown on the table, use keypad "Button1" to "Button4". AUX5 is for the hook switch. Note the hook switch is level triggered so do not use a keypad button for the hook switch.
- Step3: The keypad matrix does not require 74LS244 and either for pull-up resistors. If the matrix is bigger than 4x4, add pull-up resistor for HA[2] and HA[3]. If the matrix is less than 4x4, leave HA[2] and HA[3] open.
- Step4: Connect 93C46 to Tiger560B based on the table in the document for the USB serial number support. Add pull-up resistor on EECS# pin.

Example 2: Migrate a Tiger560 Phone (ProSlic) Design to a Tiger560B based design.

Step1: Change the Product ID to 0xC31C. If the design includes an Echo Canceller chip, change the Product ID to 0xC25C.

Step2: Connect 93C46 to Tiger560B based on the table in the document for the USB serial number support. Add pull-up resistor on EECS# pin.

Software changes to take advantage of the new Tiger560B features

Important These software changes will only work on hardware that has been updated to support the new Tiger560B features (please see above for more information).

For software written based on TigerJet Hardware Interface SDK 7.21 or above, please follow the steps below to upgrade your existing software:

- a. Download the latest TigerJet Hardware Interface SDK (8.00 or above) from TigerJet.
- b. Replace the following files in your source code with the ones included in SDK 8.00:
 1. TjlpApi.h and TjlpApi.cpp
 2. TjlpSys.h
- c. Remove the following files from your source code and Visual C++ 6.0 project:
 1. Tj580Handset.h and Tj580Handset.cpp
 2. Tj580Phone.h and Tj580Phone.cpp
- d. Add the following new files from the downloaded SDK to your source code and Visual C++ 6.0 project:
 1. Tj560BHandset.h and Tj560BHandset.cpp
 2. Tj560BPhone.h and Tj560BPhone.cpp
- e. In Tj560Phone.h, modify CTj560Phone class for Echo Canceller support:
 1. Change
CTj560Phone();
to
CTj560Phone(BOOL bEchoCanceller = FALSE);
 2. Change
virtual BOOL InitProSlic();
to
virtual BOOL InitProSlic(BOOL bEchoCanceller = FALSE);
 3. Add
BOOL m_bEchoCanceller;

f. In Tj560Phone.cpp, modify CTj560Phone implementations for Echo Canceller support:

1. Change

```
CTj560Phone::CTj560Phone()
{
}
```

to

```
CTj560Phone::CTj560Phone(BOOL bEchoCanceller) :
    m_bEchoCanceller(bEchoCanceller)
{
}
```

2. In CTj560Phone::InitTjHardware(), change the code

```
InitProSlic(); // set to default settings
```

to

```
InitProSlic(m_bEchoCanceller); // set to default settings
```

3. Modify InitProSlic function

From

```
BOOL CTj560Phone::InitProSlic(void)
```

to

```
BOOL CTj560Phone::InitProSlic(BOOL bEchoCanceller)
```

and inside InitProSlic function modify the code for writing ProSlic register 1
from

```
if (!WriteProSlicDirectReg(1, 0x2a)) // U-Law GCI 8-bit interface
    return FALSE;
```

to

```
if (!WriteProSlicDirectReg(1, bEchoCanceller ? 0x28 : 0x2a))
    // 0x28: U-Law PCM 8-bit interface
    return FALSE; // 0x2a: U-Law GCI 8-bit interface
```

g. In TjIpdemo.cpp, we no longer use Tj580 so change all 580 related code to 560B as below:

1. Change

```
#include "Tj580Handset.h"
#include "Tj580Phone.h"
```

to

```
#include "Tj560b0Handset.h"
#include "Tj60bPhone.h"
```

2. In CTjIpdemoDlg::PrepareTjDevice() and

```
CTjIpdemoDlg::UpdateDisplay(UINT uDevInfo, UINT uProductID),
```

modify the code according to your new product ID.

For example, if your device is 560B Phone using HID (product ID = 0xC31C),

you need to create an object of class CTj560BPhone_Hid() in

```
CTjIpdemoDlg::PrepareTjDevice().
```

if your device is 560B Handset/Keypad using HID (product ID = 0xF210),

you need to create an object of class CTj560BPhone_Hid() in

```
CTjIpdemoDlg::PrepareTjDevice().
```

For more details, please look at the sample code in latest Hardware Interface SDK

Additional questions

Please email support@tjnet.com if you have any questions.

Legal Words

TigerJet reserves the rights to make changes without further notice to any products herein. TigerJet makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does TigerJet assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in TigerJet data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. TigerJet does not convey any license under its patent rights nor the rights of others. TigerJet products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or any other application intended to support or sustain life, or for any other application in which the failure of the TigerJet product could create a situation where personal injury or death may occur. Should Buyer purchase or use TigerJet products for any such unintended or unauthorized application, Buyer shall indemnify and hold TigerJet and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, costs, damages and expenses and reasonable attorney fees arising out of, directly or indirectly any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that TigerJet was negligent regarding the design and manufacture of the part.

TigerJet and TJNET are registered trademarks of TigerJet Network Inc.

How to reach TigerJet

TigerJet Network Inc.
50 Airport Parkway
San Jose
CA 95110
USA

E-mail: info@tjnet.com
www.tjnet.com