

USB-PC/104 Bridge Board

HUSB-BRG1040

User's Guide

NC Board Series



Hivertec, Inc.

<http://www.hivertec.co.jp/>

This document applies to

HUSB-BRG1040 USB Interface

This document and programs may not be reproduced or copied, in whole or in part, without the prior consent of Hivertec.

This product is subject to change due to improvement without notice.

If you have any comments on this product, please contact us.

Windows 98, Windows 2000, Windows XP Home Edition, Windows XP Professional, Visual C++, Visual Basic, and Microsoft C/C++ are registered trademarks of Microsoft Corporation in the U.S. and other countries.

Other company names and product names are trademarks or registered trademarks of the respective companies.

Hivertec, Inc.
Mitsui Seimei Shin-Ohashi Bldg. 1-8-11
Shin-Ohashi, Koto-ku, Tokyo 135-0007, Japan
TEL: +81-3-3846-3801
FAX: +81-3-3846-3773
sales@hivertec.co.jp

Ver. 1.1E Issued on May 11, 2005
All rights reserved.

Extent of Warranty



1. The product warranty is valid for a period of one year from the date of purchase.
If a defect is acknowledged by Hivertec within the period of warranty, Hivertec will repair or replace the product upon return of the product to Hivertec.
2. Hivertec is not responsible beyond the purchase price of the product for any damages or loss of profit, direct, indirect, or secondary, caused by application, delivery, or failure of a Hivertec product either within or outside of the period of warranty.

Limitations to Liability



1. Hivertec is not responsible for any damages resulting from product installation, connections, settings, or operation that do not follow the contents of this manual.
2. This product uses semiconductor devices manufactured for general electronics equipment, such as machine tools, instrumentation, FA devices, OA devices, and communications equipment. They are not designed, conceived, approved for, or warranted for application in devices for which faulty operation or failure will have a direct affect on human life or result in personal injury or damage to property. The safety, quality, and performance of the product are not guaranteed explicitly or implicitly beyond those given in this manual or related catalogs.
3. Hivertec is not responsible for any damages resulting from modifications or repairs made to the product without the approval of Hivertec either within or outside of the period of warranty.
4. The contents of this manual do not guarantee or grant rights to patents, copyright, trademark rights, or any other rights to the intellectual property of Hivertec or any third party. Hivertec is not responsible for any problems that may occur concerning the rights to intellectual property of third parties resulting from the application of information provided in this manual.

Important Safety Instructions



Thank you for choosing a Hivertec NC Board. This manual contains information that is important for the safe and reliable operation of the NC Board. Read this section and understand the information contained before attempting to use the NC board. Furthermore, save this manual and store it in an easily accessible location near the installed NC Board, so that it can be referenced when necessary.

Safety Precautions	
<p>Always read this manual and any attached documents completely before attempting to use the product. Be sure that you understand the information provided and are using the product correctly. Do not use the product before having a complete understand of the product, product safety information, and precautions. In this manual, safety precautions are classified as either Warnings or Cautions.</p>	
 Warning	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
 Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.




1. Authorized For

 Caution
 The product and this manual are designed for those with the following knowledge. <ul style="list-style-type: none">▪ A basic knowledge of installing and wiring expansion boards.▪ A basic knowledge of electronic control devices and personal computers.



2. Applicable Bus

 Warning
 The product is applicable to Universal Serial Bus 1.1(Full Speed).










3. Applicable Operating System




 Warning
 The product can be used with OS that supports USB1.1 or USB2.0.
 The appended software is for WindowsXP Professional, WindowsXP Home Edition, Windows2000 Professional, and Windwos98 Second Edition.

4. Environmental Conditions



 Warning
 Store and use the product only under the following environmental conditions. <ul style="list-style-type: none">• Ambient operating temperature: 0 to +50 °C• Ambient operating humidity: 20% to 85% (with no condensation)• Ambient storage temperature: -15 to 75 °C• Ambient storage humidity: 10% to 90% (with no condensation)• Atmosphere: No corrosive gas, inflammable gas, oil mist, or dust.• Altitude: 3,000 m above sea level max. (Lower upper level of temperature ranges by 2 °C for every 300 m.)

5. Transportation and Installation

 Warning	
	Touch a metal object to discharge static electricity from your body before touching the board. Static electricity may cause board failure.
	Do not wrap the board in any wrapping material that easily carries a static charge, such as bubble wrap. Static electricity may cause board failure.
	Do not place any heavy object on the board. Heavy object may damage components on the boards, possibly causing board failure.
	Set the jumpers on the board before mounting it to the computer. If the jumpers are set with the power supply turned ON, the settings may not be recognized correctly possibly causing operating errors.
	Set the board jumpers correctly. Incorrect settings may cause operating errors.
	Do not detach the USB connector by power supply ON, because this product is an electronic control devices, though USB supports the hot-swap.
	Must connect accurately the USB connector, when the product is connected with PC. The connector must not disconnect, while the product is operating. The connector may come loose and faulty contact may cause operating errors.
	Always keep external power supply within ratings. Using a power supply that is not within ratings may cause board failure, burning, or operating errors.

 Caution	
	Do not drop the board or handle it roughly. Vibration or shock may cause board failure.
	Do not touch the solder surface of the board with your hands. Sharp points on the components may cause injury.

6. Trial Operation and Adjustment

 Warning	
	Always debug the program completely before using the board to drive a device. Any error in the program may cause unexpected operation.

7. Disposal



 Warning	
	Abide by all applicable laws and ordinances when disposing of the product.

Table of Contents

1. OVERVIEW.....	1
1.1 FEATURES AND FUNCTIONS	1
1.2 PHYSICAL LAYOUT.....	1
1.3 SAMPLE SYSTEM ARCHITECTURE	2
1.3.1 SAMPLE CODE(C LANGUAGE)	2
2. HARDWARE.....	3
2.1 CONFIGURING BOARD IDS.....	3
2.2 LED	3
2.3 EXTERNAL POWER SUPPLY INPUT TERMINAL	4
2.4 DIMENSIONAL DETAILS OF THE BOARD	4
2.5 J1 CONNECTOR (PC/104 BUS) SIGNALS	5
3. SOFTWARE SPECIFICATIONS.....	6
3.1 SOFTWARE ARCHITECTURE	6
3.2 INSTALLATION OF DEVICE DRIVER	7
3.3 USING MULTIPLE BRGs.....	8
3.4 RECOGNITION OF USB BOARDS	8
3.5 HOW TO ACCESS A BRG	9
3.5.1 DATA STRUCTURE FOR RECOGNIZING THE BRGs.....	9
3.5.2 SPECIFYING THE IRQ NUMBERS USED BY THE PC/104 DEVICES	9
3.5.3 USING DRIVER FUNCTIONS.....	10
3.5.4 USING DRIVER FUNCTIONS IN C++ APPLICATIONS.....	10
3.5.5 PREPARING TO ACCESS THE BOARD.....	11
3.6 LIST OF FUNCTIONS	11
3.7 SPECIFICATIONS OF THE FUNCTIONS	12
3.8 LIST OF RETURN VALUES	18
4. SPECIFICATIONS	18

1. Overview

This specification document describes USB-PC/104 bridge board HUSB-BRG1040 (BRG).

1.1 Features and Functions

- (1) Enables data input/output via USB from a PC host (*1) to a board that is compliant with PC/104 bus of 8-bit data width.
- (2) Compatible with Universal Serial Bus 1.1 Full Speed (12Mbps).
- (3) PC/104 standard size.
- (4) 8-bit data width, TTL-level bus signal.
- (5) Self-powered.
Equipped with external power supply terminal for PC/104 bus (+5V).
- (6) Provides board IDs to identify multiple BRG boards.
Up to 16 BRG boards available.

*1. PC with USB 1.1 or USB 2.0 ports

1.2 Physical Layout

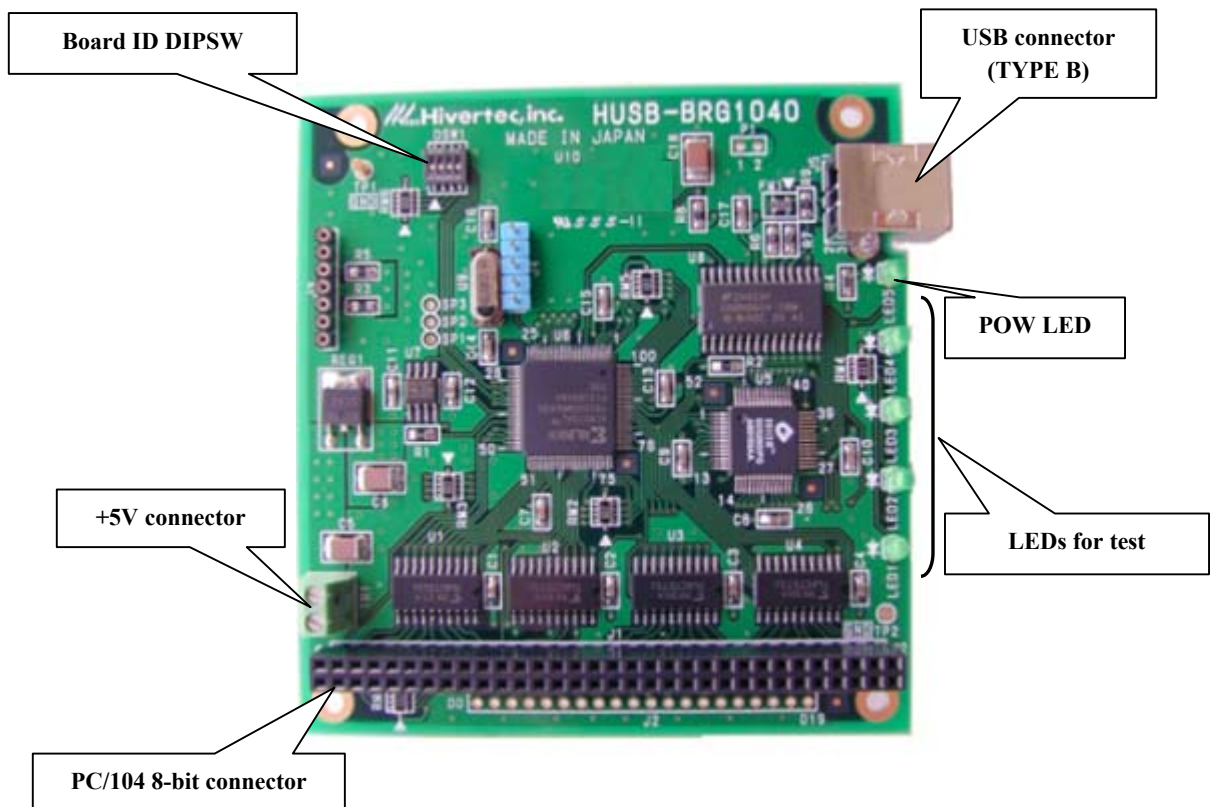


Fig 1. 2-1 USB-PC/104 Bridge Board

1.3 Sample System Architecture

Fig. 1. 3-1 shows a sample system with a two-channel encoder counter module.

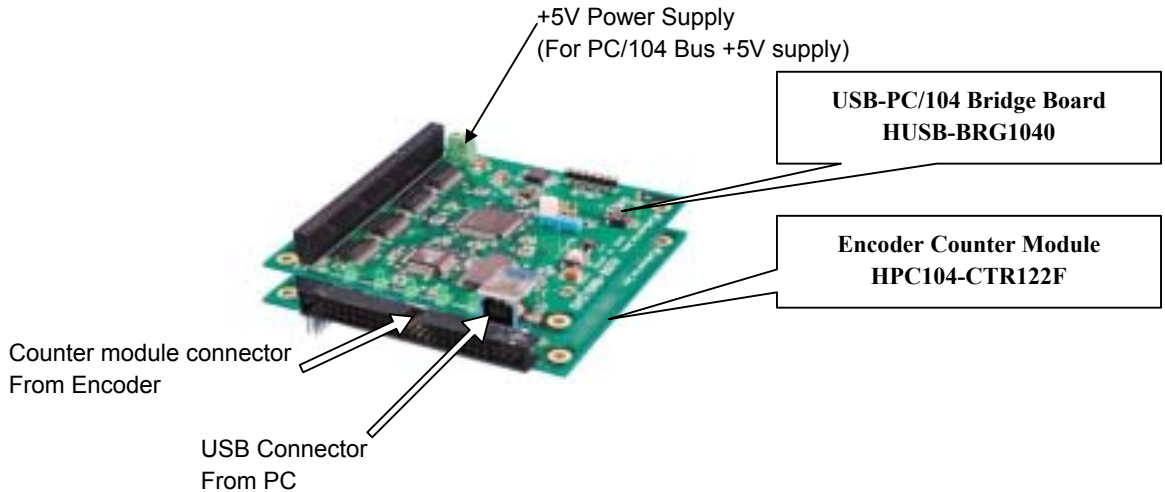


Fig. 1. 3-1 Sample System Architecture

1.3.1 Sample code(C language)

```
/* This is a sample code for a PC/104 module (HPC104-CTR122F) with IO ports access. */
#include <windows.h>
#include <stdio.h>
#include "hubr1040.h"

#define BASE_ADR      0x100          /* PC/104 module(HPC104-CTR122F) base address */
#define DI_PRT       BASE_ADR +0x10 /* general-purposed input address */
#define DO_PRT       BASE_ADR +0x12 /* general-purposed output address */

void main( void )
{
    unsigned long    ret, cnt, hBrg;
    unsigned char    prt;
    HUSBBRGINF      hBrgInfo[16];

    /* Initialize the BRG board */
    ret = br1040_GetDeviceCount( &cnt );          /* Checks the number of the board */
    ret = br1040_GetDeviceInfo( &cnt, &hBrgInfo[0] ); /* Gets the device information of the board */
    ret = br1040_OpenDevice( &hBrg, &hBrgInfo[0] ); /* Opens the board */

    /* Reads/Writes to PC/104 module(HPC104-CTR122F) */
    ret = br1040_inp( hBrg, DI_PRT, &prt );      /* Reads general-purposed input */
    printf( "%02x", prt );                       /* Prints to output the input data */
    ret = br1040_outp( hBrg, DO_PRT, 0x4 );      /* Writes general-purposed output */
                                                    /* (Outputs DO3 ON) */
    ret = br1040_outp( hBrg, DO_PRT, 0 );       /* Writes general-purposed output */
                                                    /* (Outputs DO3 OFF) */

    /* Finish the BRG board */
    ret = br1040_CloseDevice( hBrg );           /* Closes the board */
}
```

2. Hardware

2.1 Configuring Board IDs

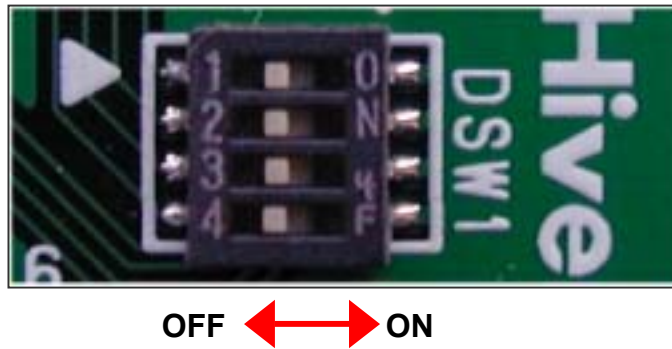


Fig. 2. 1-1 DIP Switches for Configuring Board ID

When you use multiple BRG boards, you can use board IDs (0 to 15) that are configured by DIP switches. Switches 1-4 correspond to bits 0-3. When a switch is ON, it represents 1. Otherwise, 0.

The following table shows the relations between the configured values for board IDs and status of the DIP switches (the board ID is configured to 0 when shipped).

Configured values for Board IDs	0	1	2	4	15
Status of the DIP Switches					

Fig. 2.1-2 Dip Switches and Configured board IDs

2.2 LED

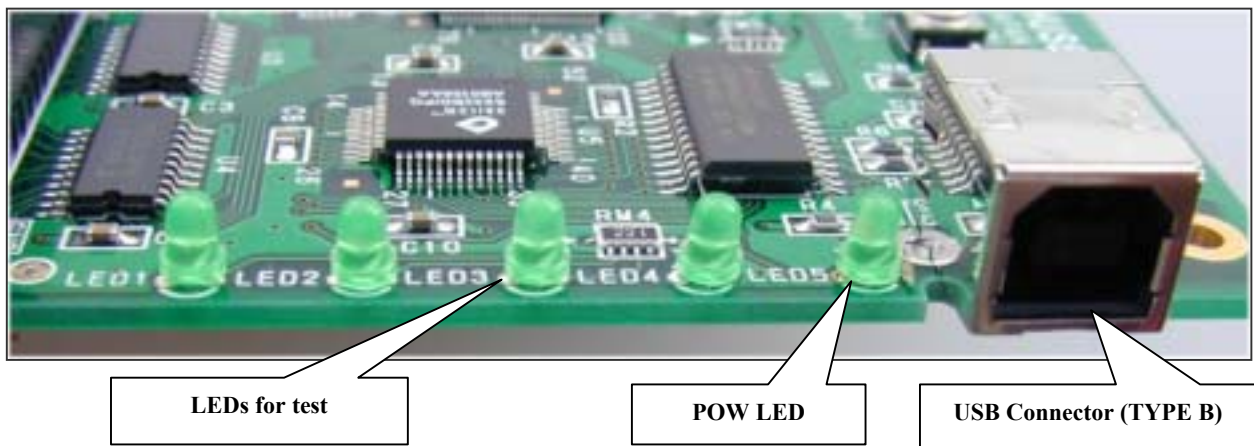


Fig. 2. 2-1 LED1-5 and USB Connector

POW LED (LED 5)

: Lit when power is on.

LEDs (LED 1-4) for test

2.3 External Power Supply Input Terminal

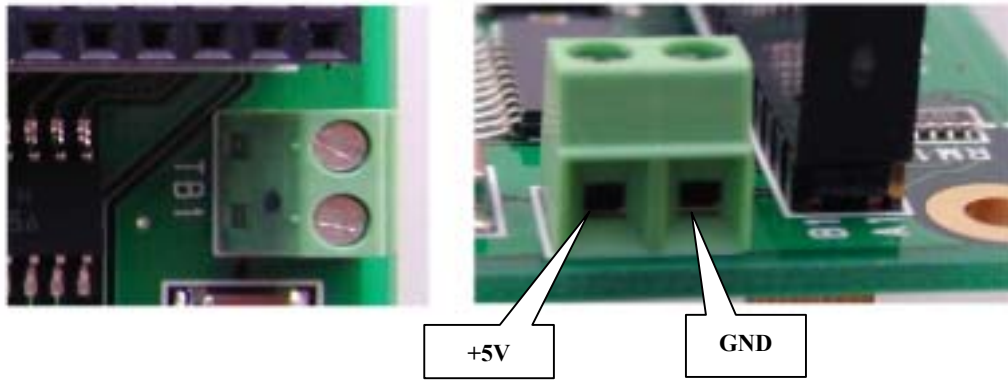


Fig. 2. 3-1 +5V Power Supply Input Terminal (For PC/104 bus +5V supply)

2.4 Dimensional Details of the Board

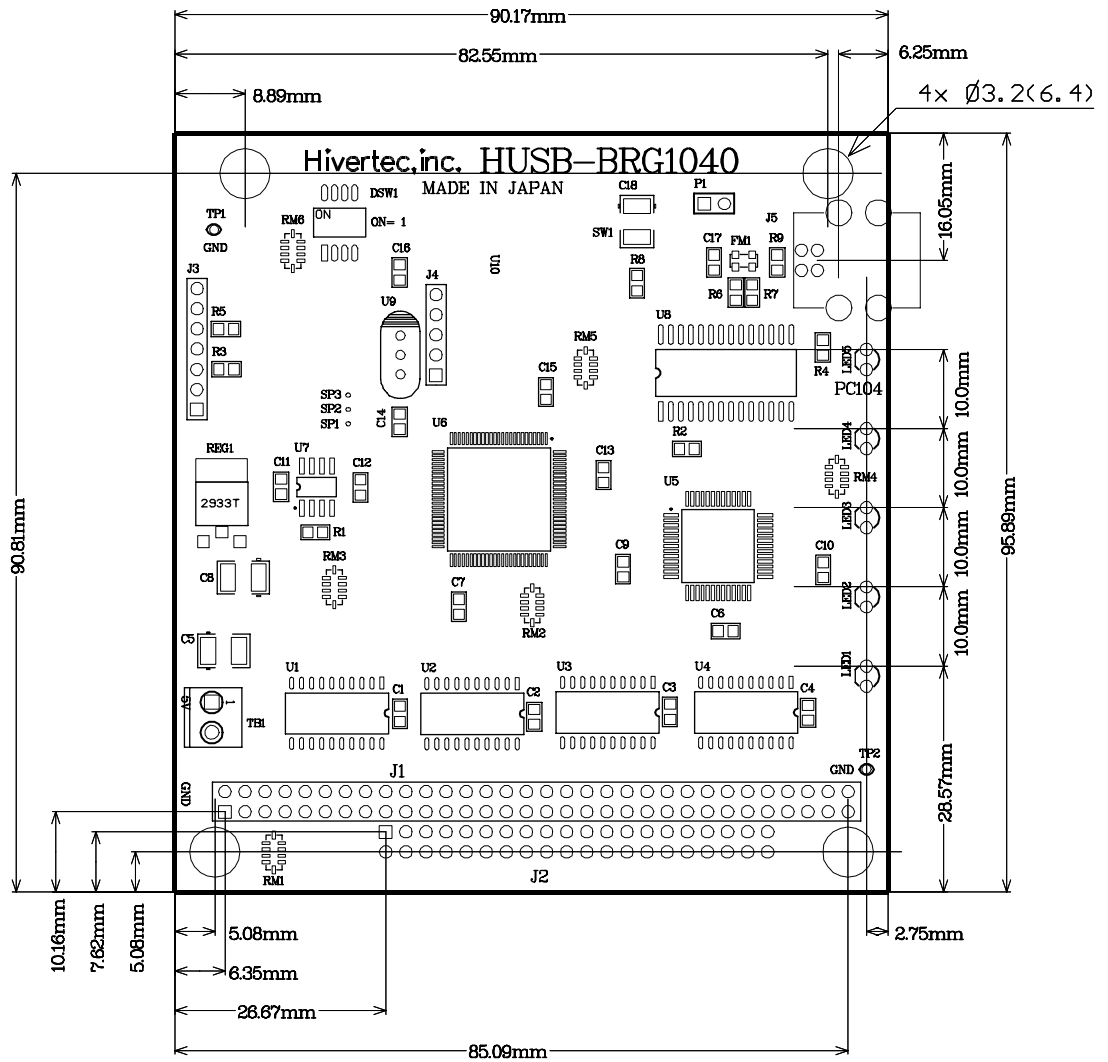


Fig. 2. 4-1 Dimensional Details of the Board

2.5 J1 Connector (PC/104 Bus) Signals

Table 2. 5-1 shows J1 Connector Signals

J1		
Pin No	Row A	Row B
1	IOCHK	GND
2	SD7	RESET
3	SD6	+5V
4	SD5	IRQ9
5	SD4	-5V
6	SD3	DRQ2
7	SD2	-12V
8	SD1	SRDY
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW
12	SA19	SMEMR
13	SA18	IOW
14	SA17	IOR
15	SA16	DACK3
16	SA15	DRQ3
17	SA14	DACK1
18	SA13	DRQ1
19	SA12	REFRESH
20	SA11	BCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2
27	SA4	TC
28	SA3	BALE
29	SA2	+5V
30	SA1	OSC
31	SA0	GND
32	GND	GND

: Used

: Not used

Table 2. 5-1 J1 Connector Signals

3. Software Specifications

"hubr1040.dll" is a driver interface DLL to control HUSB-BRG1040 and PC/104 modules stacked to BRG. This DLL is used on Windows XP, Windows 2000, and Windows 98 SE.

To use "hubr1040.dll", the following device drivers must be installed in the respective operating systems:

"hb1040xp.sys" for Windows XP

"hb10402k.sys" for Windows 2000

"hubr1040.sys" for Windows 98 SE

3.1 Software Architecture

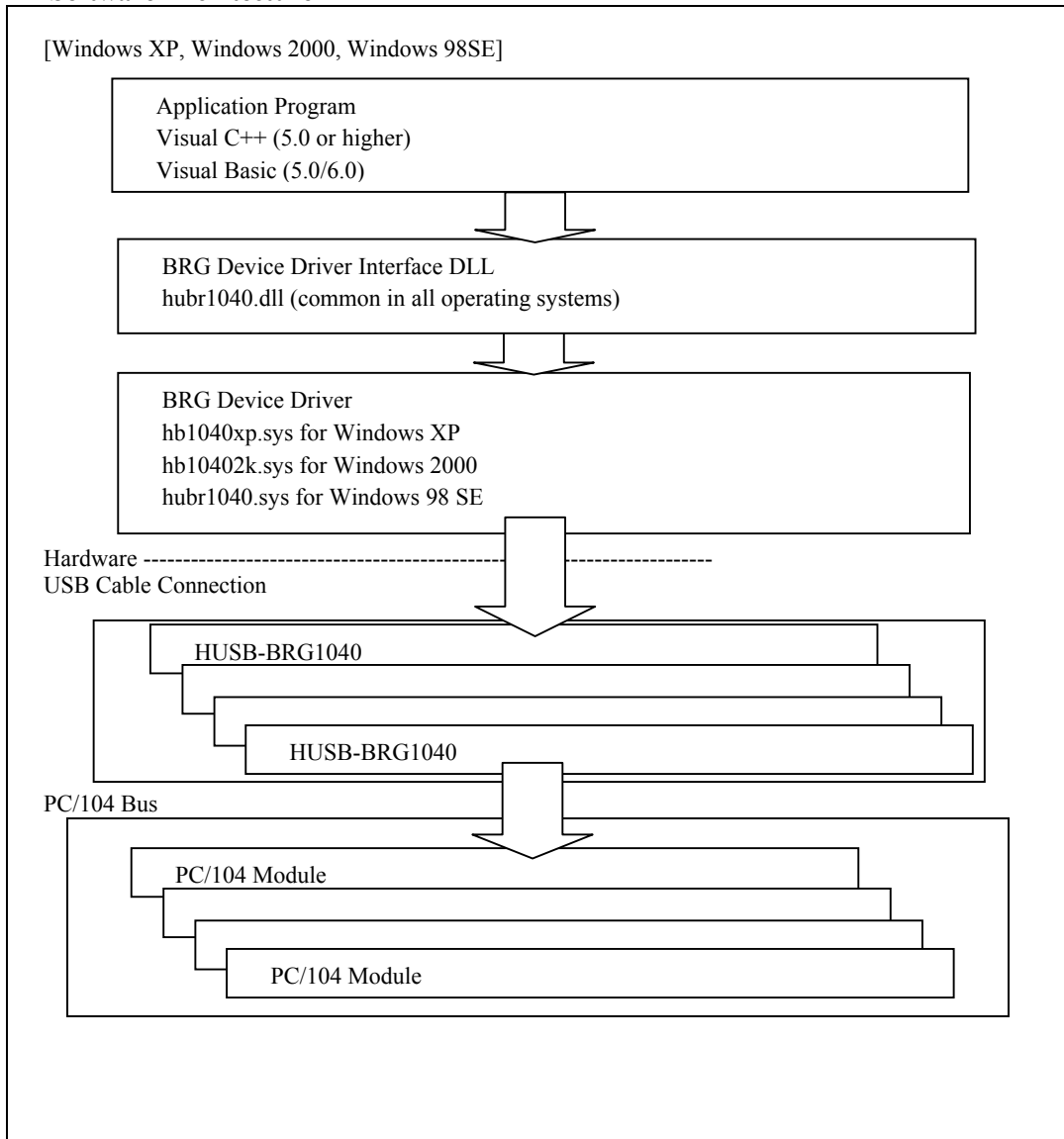


Fig. 3. 1-1 Software Architecture

3.2 Installation of Device Driver

1. Installation to Windows XP

- (i) Make sure that Windows is running. Insert the USB connector from the BRG board, with its power off, to the USB connector on the computer. Then, turn on the BRG board.
- (ii) Once the BRG board is detected by the system, the installation window for the required device driver is automatically displayed.
- (iii) Insert the installation disk into the floppy disk drive.
- (iv) Select "Install the software automatically (Recommended)"
- (v) Specify HUSB-BRG1040 (WinXP).

Follow the instructions displayed to complete the rest of the installation.

2. Installation to Windows 2000

- (i) Make sure that Windows is running. Insert the USB connector from the BRG board, with its power off, to the USB connector on the computer. Then, turn on the BRG board.
- (ii) Once the BRG board is detected by the system, the installation window for the required device driver is automatically displayed.
- (iii) Insert the installation disk into the floppy disk drive.
- (iv) Select "Search for a suitable driver for my device (recommended)"
- (v) Make sure to select the "Specify a location" checkbox.
- (vi) Specify A:\WIN2K (if it is in the A drive).

Follow the instructions displayed to complete the rest of the installation.

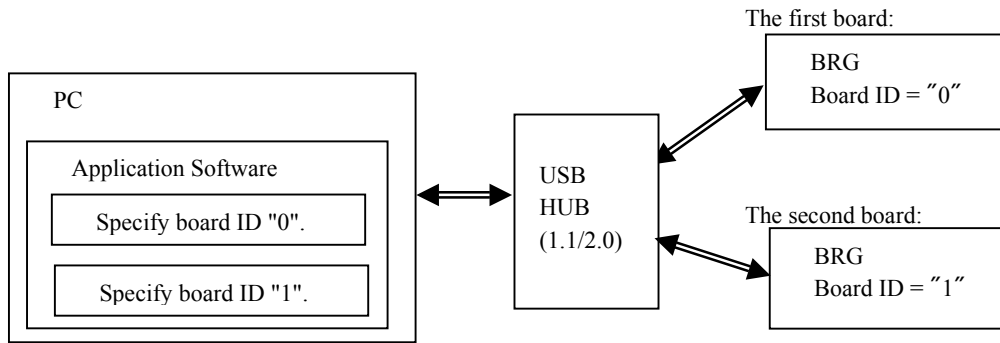
3. Installation to Windows 98 SE

- (vii) Make sure that Windows is running. Insert the USB connector from the BRG board, with its power off, to the USB connector on the computer. Then, turn on the BRG board.
- (viii) Once the BRG board is detected by the system, the installation window for the required device driver is automatically displayed.
- (ix) Insert the installation disk into the floppy disk drive.
- (x) Select "Search for the best driver for your device (Recommended)"
- (xi) Make sure to select the "Specify a location" checkbox.
- (xii) Specify A:\WIN98 (if it is in the A drive).

Follow the instructions displayed to complete the rest of the installation.

3.3 Using Multiple BRGs

This section describes how to connect multiple BRGs to the same computer and how to map each BRG board to external connections respectively.



(1) How to specify each HUSB-BRG1040

When multiple BRG boards are connected using USB cables, the computer internally assigns a fixed number to each board. However, this number can't be used by software. Therefore, to identify boards of the same model, a unique "board ID" needs to be configured to each BRG board (Board IDs should be 0 to 15).

(2) How to configure board IDs

For the configuration of board IDs, see "Fig. 2. 1-2 Dip Switches and Configured board IDs" in 2.1 Configuring board IDs.

3.4 Recognition of USB Boards

This section describes about recognition, connection, and disconnection of USB boards.

	Behavior/Operation	Description
1	Recognition of the BRG board (device)	Done when the application program (driver interface DLL) is started.
2	Newly connected devices AFTER the application program is started	The device is not recognized.
3	Disconnection of devices AFTER the application program is started	The device becomes uncontrollable.
4	Any USB connections WHILE the application is running	Affects USB communication of the application (communication stops).
5	Connection and disconnection of the BRG board (device)	Wait for at least five seconds before using (or communicating with) the connected device. Also, wait at least five seconds before reconnecting the device after disconnection.

3.5 How to Access a BRG

To access a BRG board, you have to open this device first to get its device handle.

3.5.1 Data Structure for Recognizing the BRGs

A HUSBRRGINF type structure as shown below is provided for recognizing the boards:

```
[C language: Visual C/C++]
typedef struct _HUSBRRGINF {
    WORD        BrdID;           // board IDs    (0 to 15)
    WORD        IrqEn;          // IRQ numbers used
    WORD        EPcnt;          // numbers of end points (>=2)
    WORD        EPspc[8];       // end point specifications (0x81,0x02,0x00,0x00)
    WORD        EPSiz[8];       // end points in byte (32, 32, 0, 0)
    WORD        EPopt[8];       // end point options (reserved)
    HANDLE      EPhdl[8];       // end point handles (pipe handle)
} HUSBRRGINF, *PHUSBRRGINF;
```

```
[Visual Basic]
Type spc_mat
    dt(0 To 7) As Integer
End Type
Type hdl_mat
    dt(0 To 7) As Long
End Type
```

```
Public Type HUSBRRGINF
    BrdID        As Integer      ' board IDs    (0 to 15)
    IrqEn        As Integer      ' IRQ numbers used
    EPcnt        As Integer      ' numbers of end points (>=2)
    EPspc        As spc_mat      ' end point specifications (0x81,0x02,0x00,0x00)
    EPSiz        As spc_mat      ' end points in byte (32, 32, 0, 0)
    EPopt        As spc_mat      ' end point options (reserved)
    EPhdl        As hdl_mat      ' end point handles (pipe handle)
End Type
```

3.5.2 Specifying the IRQ Numbers Used by the PC/104 Devices

To generate an interrupt by the PC/104 device that is controlled by the BRG, you need to specify the interrupt number to use.

Set the member variable "IrqEn" of the device information that is given when the device is opened.

- IrqEn (the initial value is 0)
 - bit n = '1': Interrupt at the specified IRQ number is enabled.
You can check the interrupt using the BRG's interrupt status.
 - n = '0': Interrupt at the specified IRQ number is disabled.

(Interrupt is not reflected to the BRG's interrupt status.)

15 - 6	5	4	3	2	1	0
Not used	IRQ9	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3

3.5.3 Using Driver Functions

- (1) Building applications with Microsoft Visual C++ (5.0 or higher)

Add the following files to your project:

- File to be added to the project
 - For driver functions --**hubr1040.lib** -- an import library for the driver functions
- Include file
 - For driver functions --**hubr1040.h** -- a header file for binding the driver functions

- (2) Building applications with Microsoft Visual Basic (5.0/6.0)

Add the following files to your project:

- For driver functions --**hubr1040.bas** -- a standard module defining DLL functions for driver interfaces
External functions are declared (by Declare statements) and user-defined types are defined in this file.

3.5.4 Using Driver Functions in C++ Applications

Driver functions are coded in "C language". These functions are explicitly defined as "C language" in the header file for the driver functions as follows so that they can be used in C++ applications.

```
//-----  
// The prototype declaration for the functions  
//-----  
#ifdef __cplusplus  
extern "C"  
{  
#endif  
  
    A prototype declaration for an individual function  
  
#ifdef __cplusplus  
}  
#endif
```

- (1) #ifdef __cplusplus ---- These are the statements for Microsoft Visual C++ definition.
#endif In C++ coding, library functions are explicitly interpreted as "C modules"
- (2) extern "C" ---- This statement represents a function defined in a C module

3.5.5 Preparing to access the board

- (1) Obtaining device information of all boards that are to be used
 First, obtain device information of all BRG boards and store it into (an array of) the "HUSBRRGINF" type structure area.
 - br1040_GetDeviceCount() ----- Checks the number of BRG boards
 - br1040_GetDeviceInfo() ----- Obtains the device information of the BRG boards

- (2) Opening device for each board
 Pass the device information of the BRG to be controlled to the device open function.
 If you use a PC/104 device interrupt that is stacked to the BRG, set "IrgEn" (a member variable in the device information) at this time.
 The BRG is opened when the device open function is called, and then the function returns a device handle value used to access this board.
 When there are more than one boards, repeat this process for each board.
 If you need to select a specific board, identify it by the board ID in the device information.
 - br1040_OpenDevice() ----- Opens the BRG board
 Now the board is accessible using this "device handle".

- (3) When all processes are finished, call the "Close" process for each opened device before exiting from the application.
 - br1040_CloseDevice() ----- Closes the board

3.6 List of Functions

No.	Name of the function	Description
1	br1040_GetDeviceCount ()	Obtains the number of boards
2	br1040_GetDeviceInfo ()	Obtains the device information
3	br1040_OpenDevice ()	Opens a device
4	br1040_CloseDevice ()	Closes a device
5	br1040_rInt1()	Reads the interrupt status (updates interrupt status)
6	br1040_rInt2()	Reads the interrupt status (use this after br1040_inpw,br1040_inpw,br1040_rInt1)
7	br1040_rBid ()	Reads the board ID
8	br1040_inpw ()	Reads 1 byte data
	br1040_outpw ()	Writes 1 byte data
	br1040_inpw ()	Reads 2 byte data
	br1040_outpw ()	Writes 2 byte data (from lower byte to upper)
	br1040_outpw_hl ()	Writes 2 byte data (from upper byte to lower)
9	br1040_wLed ()	Controls LEDs for test

Table 3. 6-1 List of Functions

3.7 Specifications of the Functions

No.	1	br1040_GetDeviceCount()	Obtains the number of boards
Description		Obtains the number of BRG boards that are currently equipped in the computer.	
VC++	Format	DWORD WINAPI br1040_GetDeviceCount(DWORD* count);	
	Arguments	DWORD* count ---- the area to store the obtained number of BRG boards	
	Example usage	DWORD* count; // number of BRG boards DWORD ret; // return value of the function ret = br1040_GetDeviceCount(&count);	
VB	Format	Declare Function br1040_GetDeviceCount Lib "hubr1040.dll" (ByRef count As Long) As Long	
	Arguments	ByRef count As Long ---- the area to store the obtained number of BRG boards	
	Example usage	Dim count As Long ' number of BRG boards Dim ret As Long ' return value of the function ret = br1040_GetDeviceCount(count)	
Notes			

No.	2	br1040_GetDeviceInfo()	Obtains the device information
Description		Obtains the device information of the specified number of BRG boards currently equipped in the computer. The device information is stored in the array of the device information structure. This device information is used when opening the device.	
VC++	Format	DWORD WINAPI br1040_GetDeviceInfo(DWORD* count, HUSBRRGINF* HusbInf);	
	Arguments	DWORD* count ---- number of BRG boards (Number of sets of device information to be obtained) HUSBRRGINF* HusbInf ---- the start address of the area to which the obtained device information is set	
	Example usage	DWORD ret; // return value of the function DWORD count = 2; // the maximum number of boards is two HUSBRRGINF HusbInf[2]; // the area to which device information of the two BRG boards ret = br1040_GetDeviceInfo(&count, &HusbInf[0]);	
VB	Format	Declare Function br1040_GetDeviceInfo Lib "hubr1040.dll" (ByRef count As Long, HusbInf As HUSBRRGINF) As Long	
	Arguments	ByRef count As Long ---- number of BRG boards (Number of sets of device information to be obtained) HusbInf As HUSBRRGINF ---- the start address of the area to which the obtained device information is set	
	Example usage	Dim ret As Long ' return value of the function Dim count As Long ' number of boards Dim HusbInf(2) As HUSBRRGINF ' device information area count = 2 ' the maximum number of boards is two ret = br1040_GetDeviceInfo(count, HusbInf(0))	
Notes			

No.	3	br1040_OpenDevice()	Opens a device
Description		Opens a BRG device that has the specified device information and obtains its device handle to identify it from others. This device handle can be used in the subsequent processes as the handle to access the BRGs. This function also sets the member of the device information, "IrqEn", and enables interrupt.	
VC++	Format	DWORD WINAPI br1040_OpenDevice(DWORD * hDev, HUSBBERGINF* HusbInf);	
	Arguments	DWORD hDev; ---- the area to store the obtained device handle HUSBBERGINF* HusbInf; ---- the address to store the device information of the BRG to be opened	
	Example usage	DWORD ret; // the return value of the function DWORD hDev[2]; // the area to store the device handle ret = br1040_OpenDevice(&hDev[0], &HusbInf[0]); // the device information of the first board ret = br1040_OpenDevice(&hDev[1], &HusbInf[1]); // the device information of the second board	
VB	Format	Declare Function br1040_OpenDevice Lib "hubr1040.dll" (ByRef hDev As Long, HusbInf As HUSBBERGINF) As Long	
	Arguments	ByRef hDev As Long ---- the area to store the obtained device handle HusbInf As HUSBBERGINF ---- the address to store the device information of the BRG to be opened	
	Example usage	Dim ret As Long ' the return value of the function Dim hDev(2) As Long ' the area for the obtained device handle ret = br1040_OpenDevice(hDev(0), HusbInf(0)) ' the device information of the first board ret = br1040_OpenDevice(hDev(1), HusbInf(1)) ' the device information of the second board	
Notes			

No.	4	br1040_CloseDevice()	Closes a device
Description		Closes a BRG board. The device handle becomes invalid, which makes access using this handle unable in later processes.	
VC++	Format	DWORD WINAPI br1040_CloseDevice(DWORD hDev);	
	Arguments	DWORD hDev ---- the device handle	
	Example usage	DWORD ret; // the return value of the function ret = br1040_CloseDevice(hDev);	
VB	Format	Declare Function br1040_CloseDevice Lib "hubr1040.dll" (ByVal hDev As Long) As Long	
	Arguments	ByVal hDev As Long ---- the device handle of the board to be closed	
	Example usage	Dim ret As Long ' the return value of the function ret = br1040_CloseDevice(hDev)	
Notes			

No.	5	br1040_rInt1()	Reads interrupt status															
Description		Reads the interrupt status of the BRG board. This function updates the interrupt status.																
VC++	Format	DWORD WINAPI br1040_rInt1(DWORD hDev, WORD* sts);																
	Arguments	DWORD hDev	---- the device handle															
	WORD* sts	---- the interrupt status																
VB	Format	Declare Function br1040_rInt1 Lib "hubr1040.dll" (ByVal hDev As Long, _ ByRef sts As Integer) As Long																
	Arguments	ByVal hDev As Long	---- the device handle															
	ByRef sts As Integer	---- the interrupt status																
Notes	Example usage	Dim ret As Long ' the return value of the function Dim sts As Integer ' the interrupt status ret = br1040_rInt1(hDev, &sts);																
	Interrupt status																	
	<table border="1"> <tr> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>---</td> <td>IOCHCHK</td> <td>Interrupt at IRQ9 exists</td> <td>Interrupt at IRQ7 exists</td> <td>Interrupt at IRQ6 exists</td> <td>Interrupt at IRQ5 exists</td> <td>Interrupt at IRQ4 exists</td> <td>Interrupt at IRQ3 exists</td> </tr> </table>			7	6	5	4	3	2	1	0	---	IOCHCHK	Interrupt at IRQ9 exists	Interrupt at IRQ7 exists	Interrupt at IRQ6 exists	Interrupt at IRQ5 exists	Interrupt at IRQ4 exists
7	6	5	4	3	2	1	0											
---	IOCHCHK	Interrupt at IRQ9 exists	Interrupt at IRQ7 exists	Interrupt at IRQ6 exists	Interrupt at IRQ5 exists	Interrupt at IRQ4 exists	Interrupt at IRQ3 exists											

No.	6	br1040_rInt2()	Reads interrupt status
Description		Reads the interrupt status at the last time following functions (br1040_inpw, br1040_inpw, and br1040_rInt1) were called. This function can process software early, because it doesn't use USB communication.	
VC++	Format	DWORD WINAPI br1040_rInt2(DWORD hDev, WORD* sts);	
	Arguments	DWORD hDev	---- the device handle
	WORD* sts	---- the interrupt status	
VB	Format	Declare Function br1040_rInt2 Lib "hubr1040.dll" (ByVal hDev As Long, _ ByRef sts As Integer) As Long	
	Arguments	ByVal hDev As Long	---- the device handle
	ByRef sts As Integer	---- the interrupt status	
Notes	Example usage	Dim ret As Long ' the return value of the function Dim sts As Integer ' the interrupt status ret = br1040_rInt2(hDev, sts);	
	1. See br1040_rInt1() for the interrupt status.		
	2. If br1040_rInt2 is executed right after the device is opened, there should be no readable interrupt status.		

No.	7	br1040_rBid ()	Reads the board ID
Description		Reads the board ID of the BRG.	
VC++	Format	DWORD WINAPI br1040_rBid (DWORD hDev, WORD* dsw);	
	Arguments	DWORD hDev	---- the device handle
	WORD* bid	---- the board ID	
Example usage	DWORD ret; // the return value of the function WORD bid; // the board ID ret = br1040_rBid(hDev, &bid);		
VB	Format	Declare Function br1040_rBid Lib "hubr1040.dll" (ByVal hDev As Long, _ ByRef bid As Integer) As Long	
	Arguments	ByVal hDev As Long	---- the device handle
	ByRef bid As Integer	---- the board ID	
Example usage	Dim ret As Long ' the return value of the function Dim bid As Integer ' the board ID ret = br1040_rBid(hDev, bid)		
Notes	The board ID at the point the application was started is read. Even if the DIP switch is changed while the application is running, the value this function reads does not change.		

No.	8	br1040_inp ()	Reads 1 byte data
		br1040_outp ()	Writes 1 byte data
		br1040_inpw ()	Reads 2 byte data
		br1040_outpw ()	Writes 2 byte data (from lower byte to higher)
		br1040_outpw_hl ()	Writes 2 byte data (from higher byte to lower)
Description		<p>Reads or writes data at the specified board address on the PC/104 bus of the BRG board specified by the device handle as follows:</p> <p>Read 1 byte data ---- reads 1 byte data and stores it in the specified area.</p> <p>Write 1 byte data ---- writes 1 byte data.</p> <p>Read 2 byte data ---- reads 2 byte data and stores it in the specified area.</p> <p>Writes 2 byte data (from lower byte to upper) ---- writes 2 byte data from lower byte to upper.</p> <p>Writes 2 byte data (from lower byte to upper) ---- writes 2 byte data from upper byte to lower.</p>	
VC++	Format	DWORD WINAPI br1040_inp (DWORD hDev, WORD adrs, BYTE* brdt); DWORD WINAPI br1040_outp(DWORD hDev, WORD adrs, BYTE bwdt); DWORD WINAPI br1040_inpw(DWORD hDev, WORD adrs, WORD* wrdt); DWORD WINAPI br1040_outpw (DWORD hDev, WORD adrs, WORD wwdt); DWORD WINAPI br1040_outpw_hl(DWORD hDev, WORD adrs, WORD wwdt);	
	Arguments	DWORD hDev ---- the device handle WORD adrs ---- the port address of the PC/104 module stacked to BRG BYTE* brdt ---- the address for the 1 byte area to which the read data is stored. BYTE brdt ---- the 1 byte data to write WORD* brdt ---- the address of the 2 byte area to which the read data is stored. WORD wwdt ---- the 2 byte data to write	
	Example usage	DWORD ret; // the return value of the function BYTE dt; // the area to store data ret = br1040_inp(hDev, 0x100, &dt); // reads 1 byte data at address 0x100	
VB	Format	Declare Function br1040_inp Lib "hubr1040.dll" (ByVal hDev As Long, ByVal adrs As Integer, _ ByRef brdt As Byte) As Long Declare Function br1040_outp Lib "hubr1040.dll" (ByVal hDev As Long, ByVal adrs As Integer, _ ByVal bwdt As Byte) As Long Declare Function br1040_inpw Lib "hubr1040.dll" (ByVal hDev As Long, ByVal adrs As Integer, _ ByRef wrdt As Integer) As Long Declare Function br1040_outpw Lib "hubr1040.dll" (ByVal hDev As Long, _ ByVal adrs As Integer, _ ByVal wwdt As Integer) As Long Declare Function br1040_outpw_hl Lib "hubr1040.dll" (ByVal hDev As Long, _ ByVal adrs As Integer, _ ByVal wwdt As Integer) As Long	
	Arguments	ByVal hDev As Long ---- the device handle of the target device ByVal adrs As Integer ---- the port address of the PC/104 module stacked to BRG ByRef brdt As Byte ---- the 1 byte area to which the read data is stored. ByVal bwdt As Byte ---- the 1 byte data to write ByRef wrdt As Integer ---- the 2 byte area to which the read data is stored. ByVal wwdt As Integer ---- the 2 byte data to write	
	Example usage	Dim ret As Long ' the return value of the function Dim dt As Byte ' the area to store data ret = br1040_inp(hDev, &H100, dt) ' reads 1 byte data at address &H100	
Notes			

No.	9	br1040_wLed ()	Controls LEDs for test										
Description		Turns on/off the LEDs for test on the BRG board.											
VC++	Format	DWORD WINAPI br1040_wLed(DWORD hDev, WORD wdt);											
	Arguments	DWORD hDev	---- the device handle										
	WORD wdt	---- the LEDs on/off data											
Example usage	DWORD ret; // the return value of the function ret = br1040_wLed(hDev, 0xf); // turns on LEDs 1-4												
VB	Format	Declare Function br1040_wLed Lib "hubr1040.dll" (ByVal hDev As Long, _ ByVal wdt As Integer) As Long											
	Arguments	ByVal hDev As Long	---- the device handle										
	ByVal wdt As Integer	---- the LEDs on/off data											
Example usage	Dim ret As Long ' the return value of the function ret = br1040_wLed(hDev, &HF) 'turns on LEDs 1-4												
Notes	LEDs on/off data <table border="1" style="margin-left: 20px;"> <tr> <td>bit</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td></td> <td>LED4</td> <td>LED3</td> <td>LED2</td> <td>LED1</td> </tr> </table> <p>The corresponding LED is turned on when the value is set to '1', and turned off when set to '0'.</p>			bit	3	2	1	0		LED4	LED3	LED2	LED1
bit	3	2	1	0									
	LED4	LED3	LED2	LED1									

3.8 List of Return Values

No.	Notation	Return value (HEX)	Error description and cause
1	NO_ERROR	0x000	Normal
2	NOT_FOUND	0x001	Device driver does not exist <ul style="list-style-type: none"> • Device driver is not installed. • Device driver is not stored in the appropriate folder.
3	ALREADY_OPENED	0x002	Tried to re-open an opened device <ul style="list-style-type: none"> • Ordered an open function for a device that is already open. <ul style="list-style-type: none"> ○ Keep using the device you have opened until you close it (i.e. multiple open processes are prohibited). • If you use two or more boards, make sure that the device information of the device to be opened has been updated.
4	NOT_MEMORY	0x004	Insufficient memory to store device information <ul style="list-style-type: none"> • Insufficient application memory <ul style="list-style-type: none"> ○ Insufficient memory for the primary storage of the computer • Insufficient system resource (memory for the operating system) <ul style="list-style-type: none"> ○ Too many applications are running. ○ Too many windows are open.
5	INVALID_HANDLE	0x008	An invalid device handle is specified <ul style="list-style-type: none"> • You are using another "device handle" than the one obtained when the device was opened. • This device has been already closed.

Table 3. 8-1 List of Return Values

4. Specifications

[USB specifications]	USB specifications	Universal Serial Bus 1.1 Full Speed (12 Mbps) Self-powered device (+5V external power supply)
[Data bus width]		8-bit
[Supported PC/104 module]		I/O map'ed port board
[Environment conditions]	Current consumed	Max. 100mA
	Temperature conditions	0° to 50°C, non-condensing
	Board size	90.17 mm x 95.88 mm

Table 4. 1-1 HUSB-BRG1040 Specifications