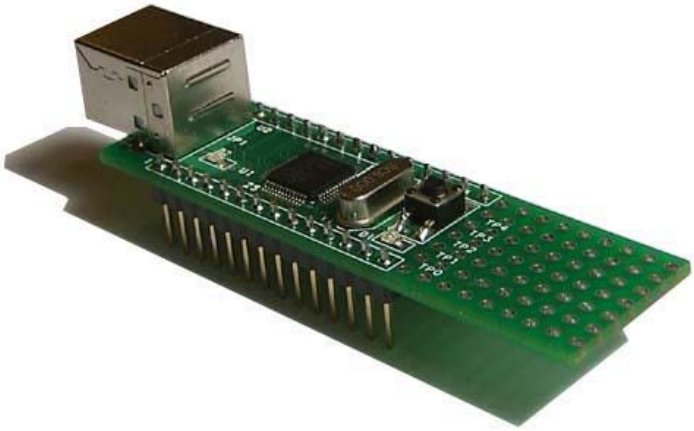
USB-LT

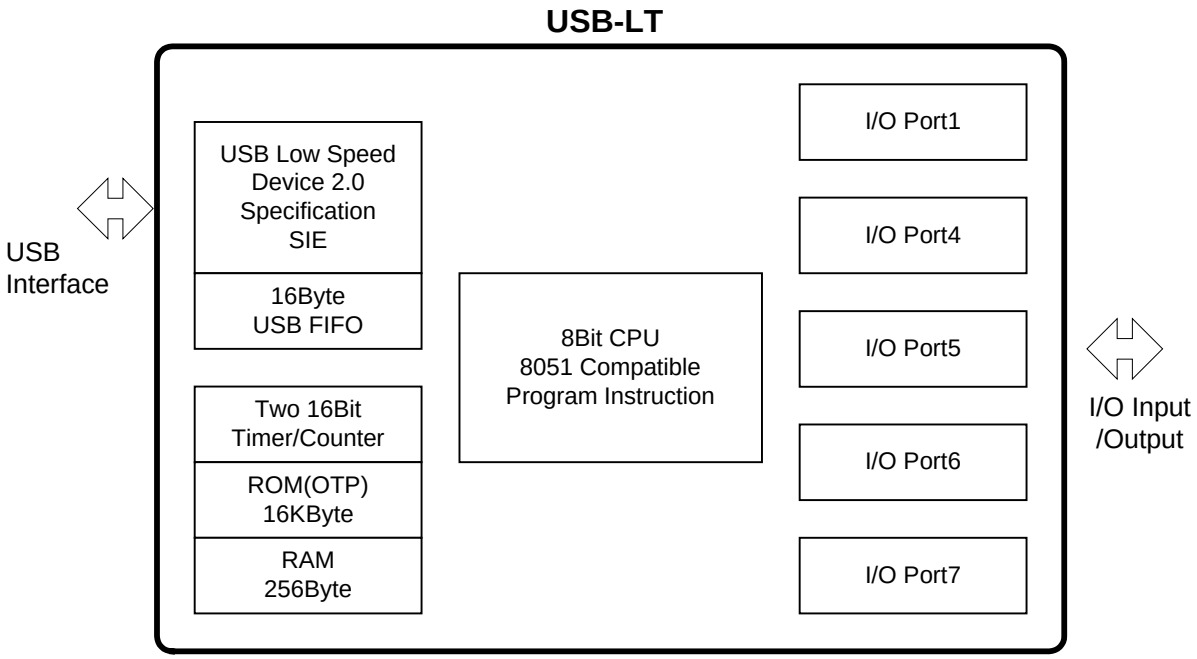
User's Manual



Contents

1. Introduction

The total control takes charge of the Micro controller in case of USB-LT as like Figure 1-1. To connection USB and I/O take charge of the 8Bit Micro Controller, the command set equally use the 8051 command set. Each I/O port can use wanted purpose whether or not set up input or output to user.



[Figure 1-1. USB-LT Block Diagram]

Internal program memory is taken possession of programmed firmware, program memory is OTP (One Time Programmable) type.

[Main Function]

- USB 2.0 Specification, Low speed(1.5Mbps)
- 16K Program memory (EPROM-OTP), 256B RAM
- 32 General Purpose I/O
- 8051 Compatible command set
- USB HID without device writing
- Easy USB Application product development
- Power Supply through USB (Max 500mA)
- Easy API to develop application program

- To development various application program with PC(ex : USB Jog & Shuttle, USB Remote control, USB Key-pad, USB Program etc..)

[Application Filed]

- USB Key-pad, Clock, Electronic display, Jog & Shuttle, Voltage meter
- USB Data acquisition (Temperature/Humidity/Voltage/ Current Measurement)
- USB I/O Control
- USB Remote Control (Electric product control through PC)
- USB Programmer (Serial Flash, ROM, EEPROM, etc..)

2. Installation

You confirm whether or not the packing contents are in good order before installation.

2.1 Confirm Product Contents

[Figure 2-1. USB-LT Product Contents]

Product Contents

1. USB-LT
2. USB (A-B) Cable
3. CD (Manual/API/Sample Source etc..)

2.2 Installation

For USB-LT board installation to PC is as follows. There isn't a Jumper to especially to set up for board installation in case of USB boards as it is Plug & Play devices.
Also, there is no need to install of device driver because of HID (Human Interface Device) connection in case of USB-LT.

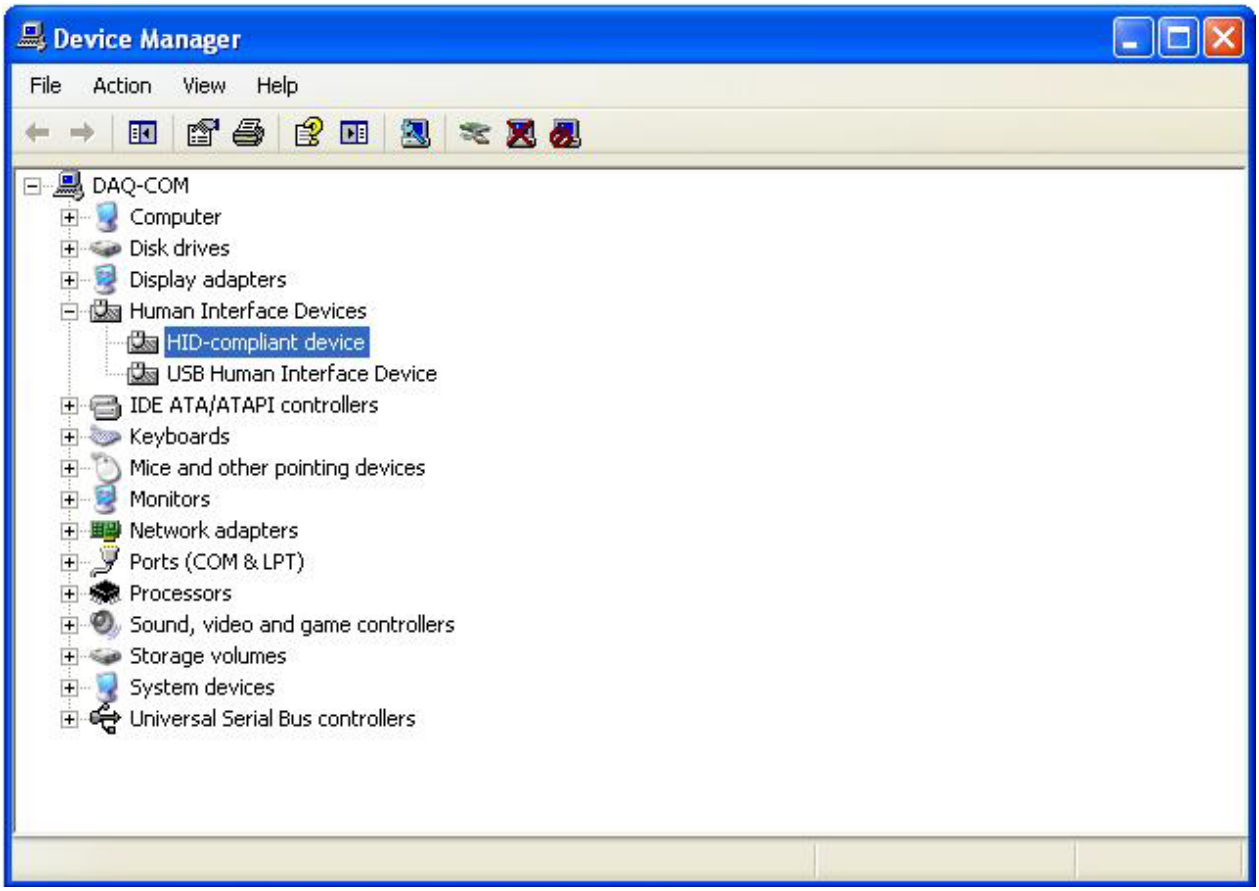
(1) First, open the box and connect to PC through USB cable.
(2) The opposite side of USB cable connects to USB-LT board.
(3) If cable connection succeeds, LED light on for indication of board supply.
(4) It confirms a driver normally installation in the following ways.

Do the following steps to show up the "Device Manager" window.

[My Computer -> Properties -> Hardware -> Device Manager -> Human Interface Devices -> "HID-compliant device"

[Figure 2-2. "System Properties" window]

[Figure 2-3. System Properties window]

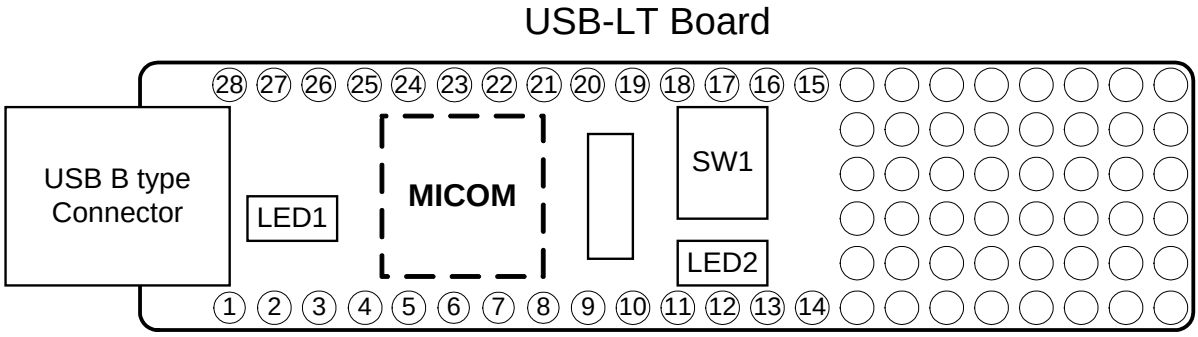


[Figure 2-4. Device Manager Window]

3. USB-LT Board

In this chapter, the primary functions of the board are described briefly. For more information, refer to the device specification.

3.1 USB-LT Concept



[Figure 3-1. USB-LT Out-side View]

USB-LT can easily use the functions of I/O, so simple circuit can be composed to universal board.

(1) **USB B type Connector**

The USB-LT board connects to the PC through the USB A-B cable. The power supply and data gives and takes through this cable.

(2) **LED1**

It confirms supply power to board.

(3) **MICOM**

It is 8bit Micro Controller which has the 8051 compatible command set. It takes charge of I/O and USB data communication.

(4) **PIN 1-28**

It supplies the power and I/O to external. The PC +5V power is supplied outside through the USB cable. If external board has special power, the power between each other shall not be connected.

(5) **SW1**

Board reset button.

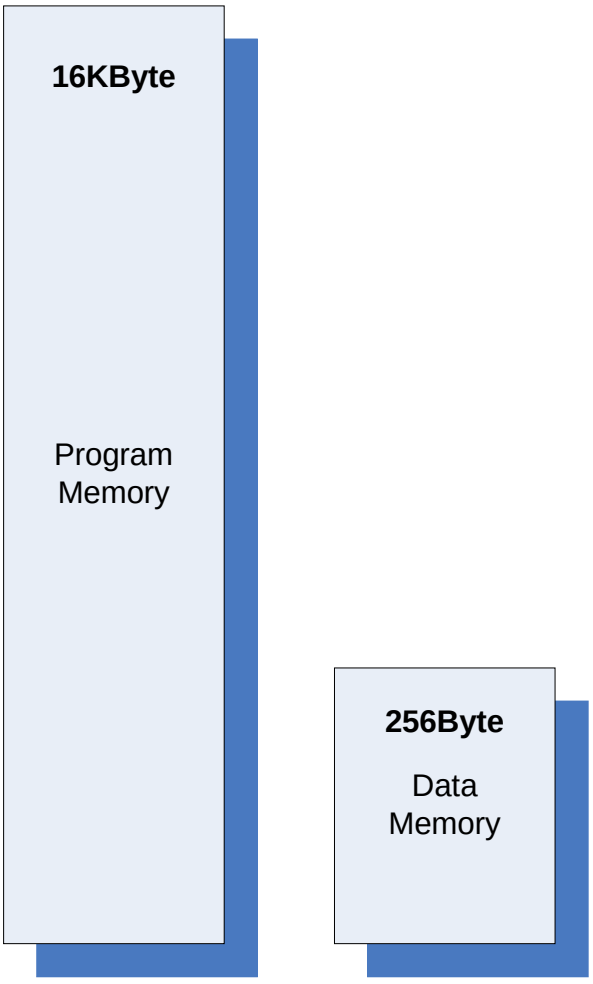
(6) **LED2**

It can confirm whether or not there is a stability of a board.

3.2 Memory Map

The board has a program memory and data memory as like general 8051. The program memory area is OTP type. User can not reprogram it because of the board was programmed to supply in case of shipment.

So, if you want special program, (You need more speedy response and private application) please contact to DAQ system.

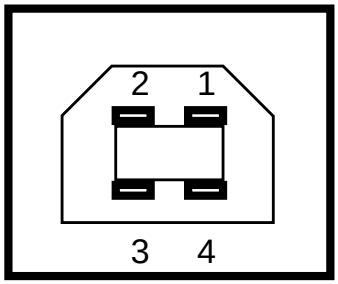


[Figure 3-2. Memory Map]

3.3 Connector Pin-out

The board has two connectors. There are the USB-B connector for USB communication, the PIN-OUT of DIP IC type connector for external I/O and power. And There is several test points.

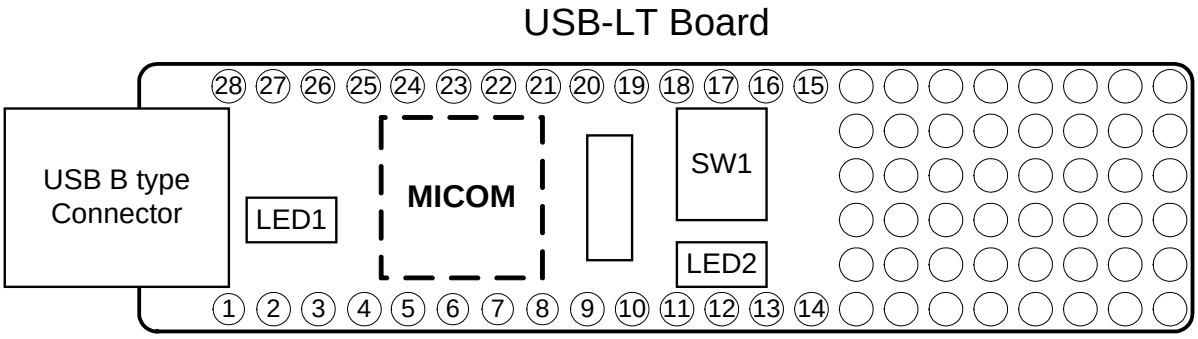
USB-B type is a connector for High speed USB connection. Figure 3-3 and Table 3-1 shows the connector and its pin description.



[Figure 3-3. USB-B connector (USB-B type Front View)]

[Table 3-1. USB-B Connector]

Pin	Signal Name	Description	Remark
1	**VCC**	USB power +5V	
2	**D-**	USB signal Minus(Negative)	
3	**D+**	USB signal Plus(Positive)	
4	**GND**	USB power GND	



[Figure 3-4. PIN-OUT of DIP IC type (1..28)]

[Table 3-2. DIP IC PIN-OUT]

No.	Name	Description	Remark
1	**P53**	Bit 3 of Port 5	
2	**P52**	Bit 2 of Port 5	
3	**P51**	Bit 2 of Port 5	
4	**P50**	Bit 0 of Port 5	
5	**P47**	Bit 7 of Port 4	
6	**P46**	Bit 6 of Port 4	
7	**P45**	Bit 5 of Port 4	
8	**P44**	Bit 4 of Port 4	
9	**P43**	Bit 3 of Port 4	
10	**P42**	Bit 2 of Port 4	
11	**P41**	Bit 1 of Port 4	
12	**P40**	Bit 0 of Port 4	
13	**P66**	Bit 6 of Port 6	2
14	**GND**	Power Ground, Supply power to external board.	
15	**P65**	Bit 6 of Port 6	
16	**P17**	Bit 7 of Port 1	3
17	**P16**	Bit 6 of Port 1	3
18	**P15**	Bit 5 of Port 1	3
19	**P14**	Bit 4 of Port 1	3
20	**P13**	Bit 3 of Port 1	3
21	**P12**	Bit 2 of Port 1	3
22	**P71**	Bit 1 of Port 7	
23	**P70**	Bit 0 of Port 7	
24	**P57**	Bit 7 of Port 5	
25	**P56**	Bit 6 of Port 5	
26	**P55**	Bit 5 of Port 5	
27	**P54**	Bit 4 of Port 5	
28	**+5V**	USB power +5V, Supply power to external board.	

[Table 3-3. Test Point PIN-OUT]

Name	Description	Remark
TP0	Bit 0 of Port 6	2
TP1	Bit 1 of Port 6	2
TP2	Bit 2 of Port 6	2
TP3	Bit 3 of Port 6	2
TP4	Bit 4 of Port 6	2

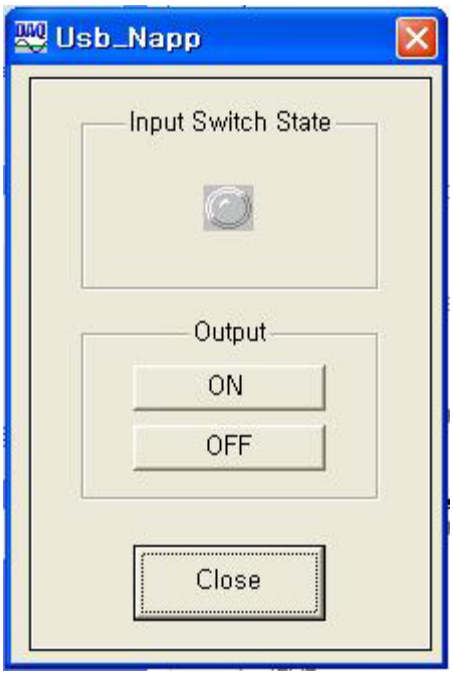
(Remark)

1. A bit 7 of Port6 is connected to LED2, this bit is set up to output. If output is “Low” and “ 0V”, LED2 light on.
2. There is a circuit to limit current to internal in case of Port6. If LED light on, there is no need to limit special current resistor to external.
3. The port1 has same specification of port1 of standard 8051. In other words, it can set up weak pull-up only.

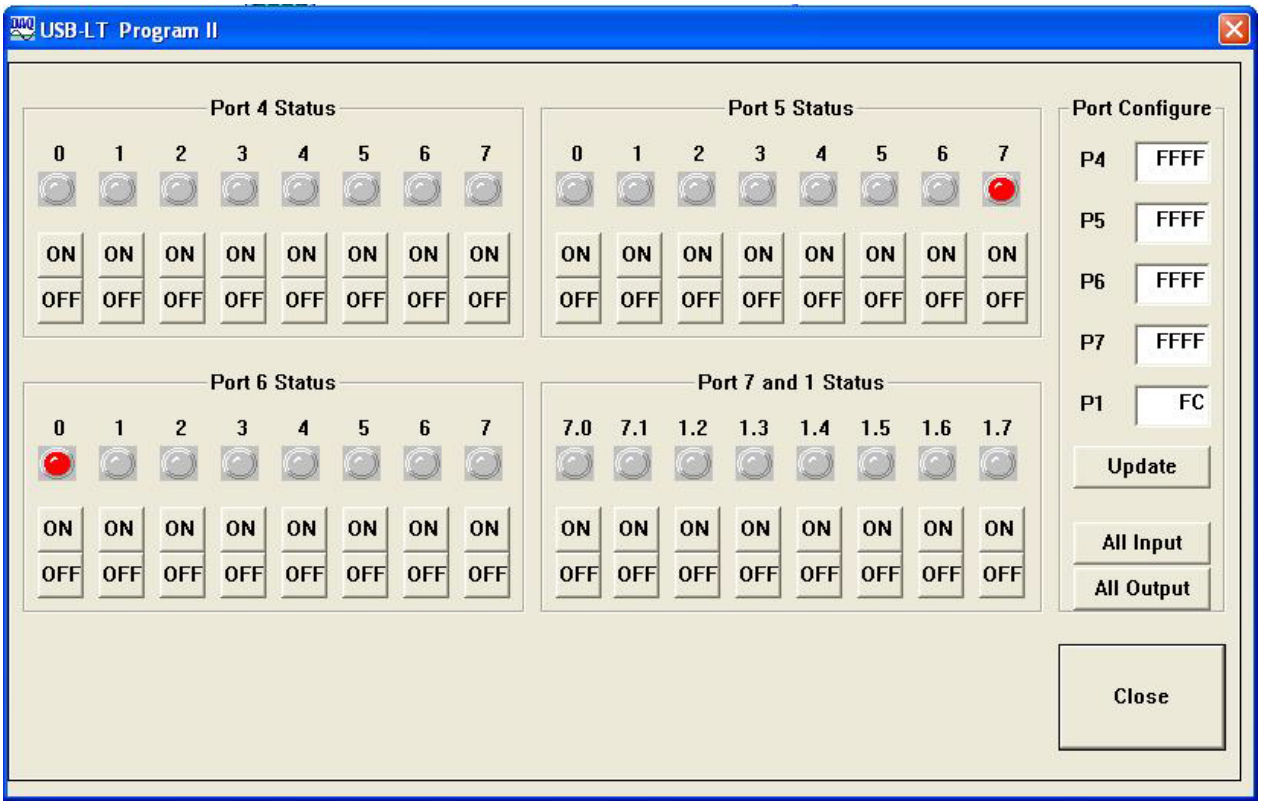
3.4 Sample program

A sample program is provided to make the user get familiar with the board operation. There is no installation of special driver because USB HID device is used to driver with supported Windows system.

The sample program has two programs. One is a program to decision whether the board is strange action or not. The other is a program to set up I/O of each port and to control output.



[Figure 3-5. Sample Program 1]



[Figure 3-6. Sample Program 2]

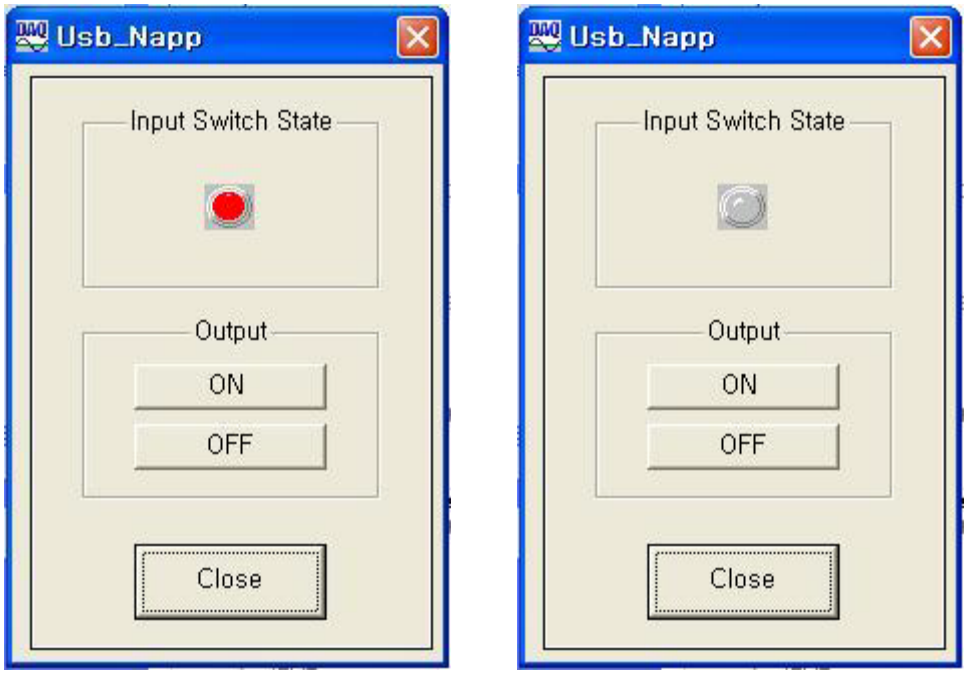
To run the sample application program, you need to use API (Application Programming Interface), which is a form of client DLL. To compile the sample source to make its executable file, you have to use Import Library files and Header files. You can find them in the CDROM.

4. Test

4.1 Input Test

This chapter is for test to learn how to operate a board and check the abnormality of a board. The test performs it at the PC which a USB-LT board was installed as it use a program of "sample1.exe" in an EXE folder of CDROM.

The execution file and source file of sample1 and sample2 is in App and App0 folder of CDROM. The execution file use a test, a user modifies a necessary a sample source file that provided it usable.



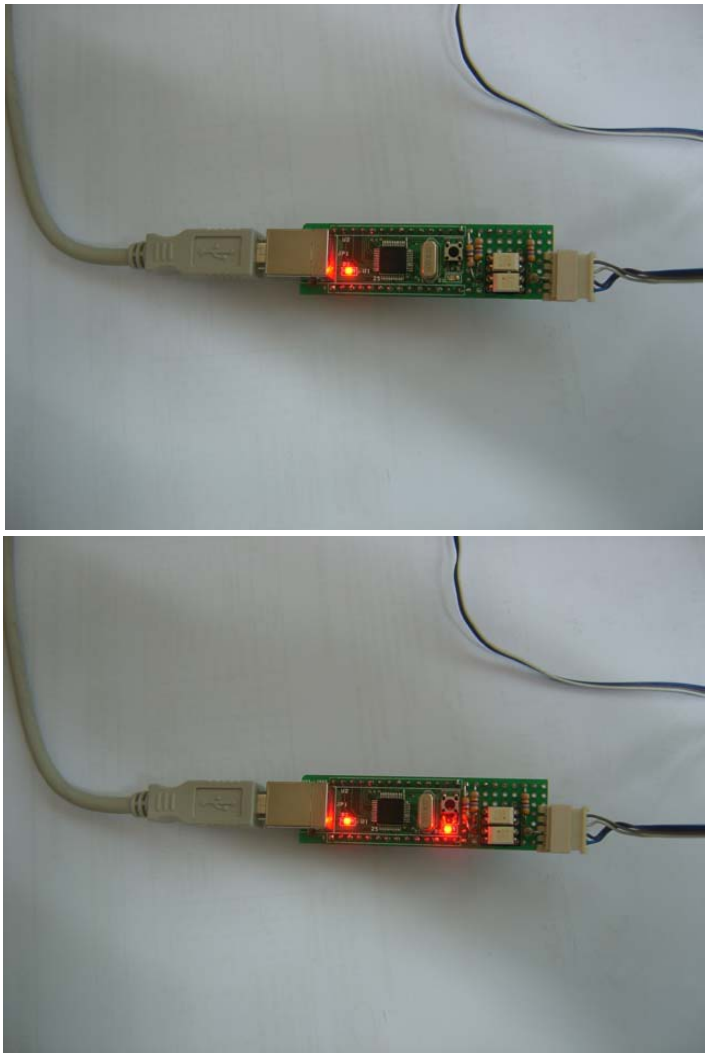
[Figure 4-1. "Sample1.exe" Window]

In the left LED is on at the above pictures, in the right LED is off. If an input of bit1 of port6 is "1', LED is on, on the contrary to this, LED is off.

In other words, If TP0 connect to VCC (+5V), LED is on. IF TP0 connect to GND, LED is off. At this time, the bit1 of port6 is used to set up by input at the above programs.

4.2 Output test

(1) You can test of output function through LED is on/off in a board at below pictures.



[Figure 4-2. LED on/off display]

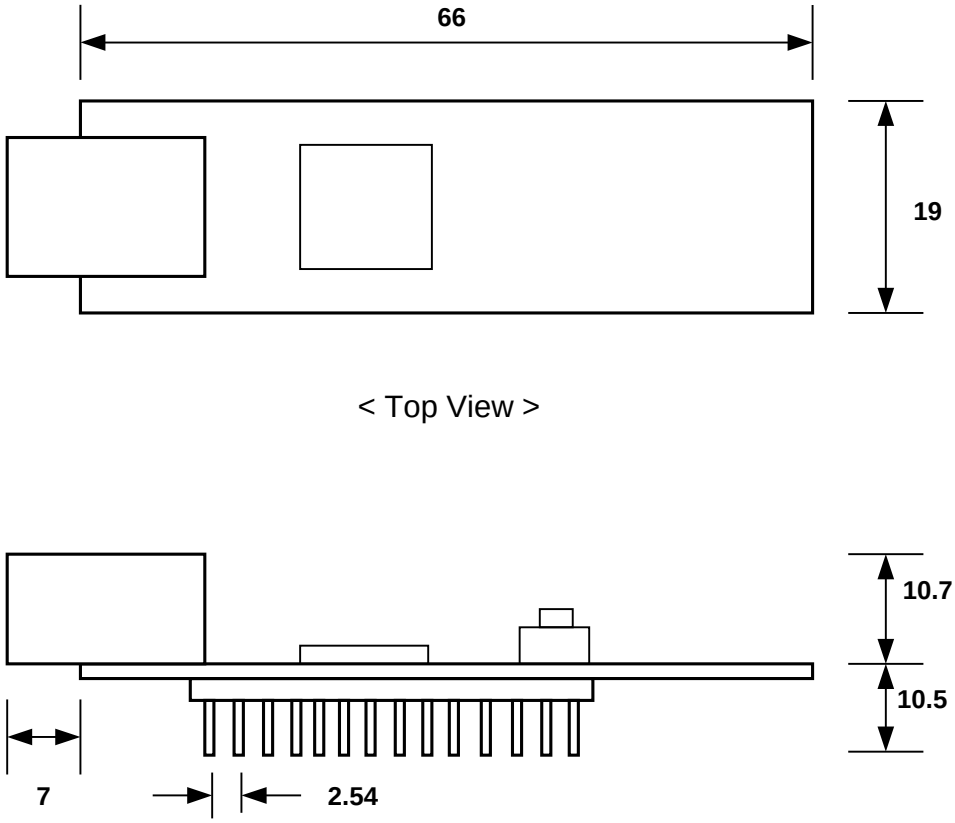
In the top LED is off, in the bottom LED is on. If an output of bit7 of port6 is “0’, LED is on, on the contrary to this, LED is off. User can test to press “ON/OFF” button at the Sample1 programs.

5. Specification

5.1 General Specification

Specification	
General	• USB 2.0 Specification • Low Speed Device 1.5Mbps • USB HID Device • 8051 Compatible Instruction Set, 4 Clock per Cycle • 16K EPROM, 256Byte RAM,
Interface	• USB Powered through USB-B Connector, Max 500mA • +5V Single Power operation • 32 General Purpose I/O • 28pin DIP-IC form factor.
Functions	• Two 16bit Timer/Counter • User configurable(Schmitt trigger Input/ N-ch open drain Output) I/O • Full-up option for each I/O. • 1 Control Endpoint, 2 Data Endpoints • 30 external interrupt source
Software	
Supported OS	Windows 2000/XP
API	Interface with Application through client DLL
Custom Software Design	If user want special program to have special function, user can inquiry to modify Firmware.

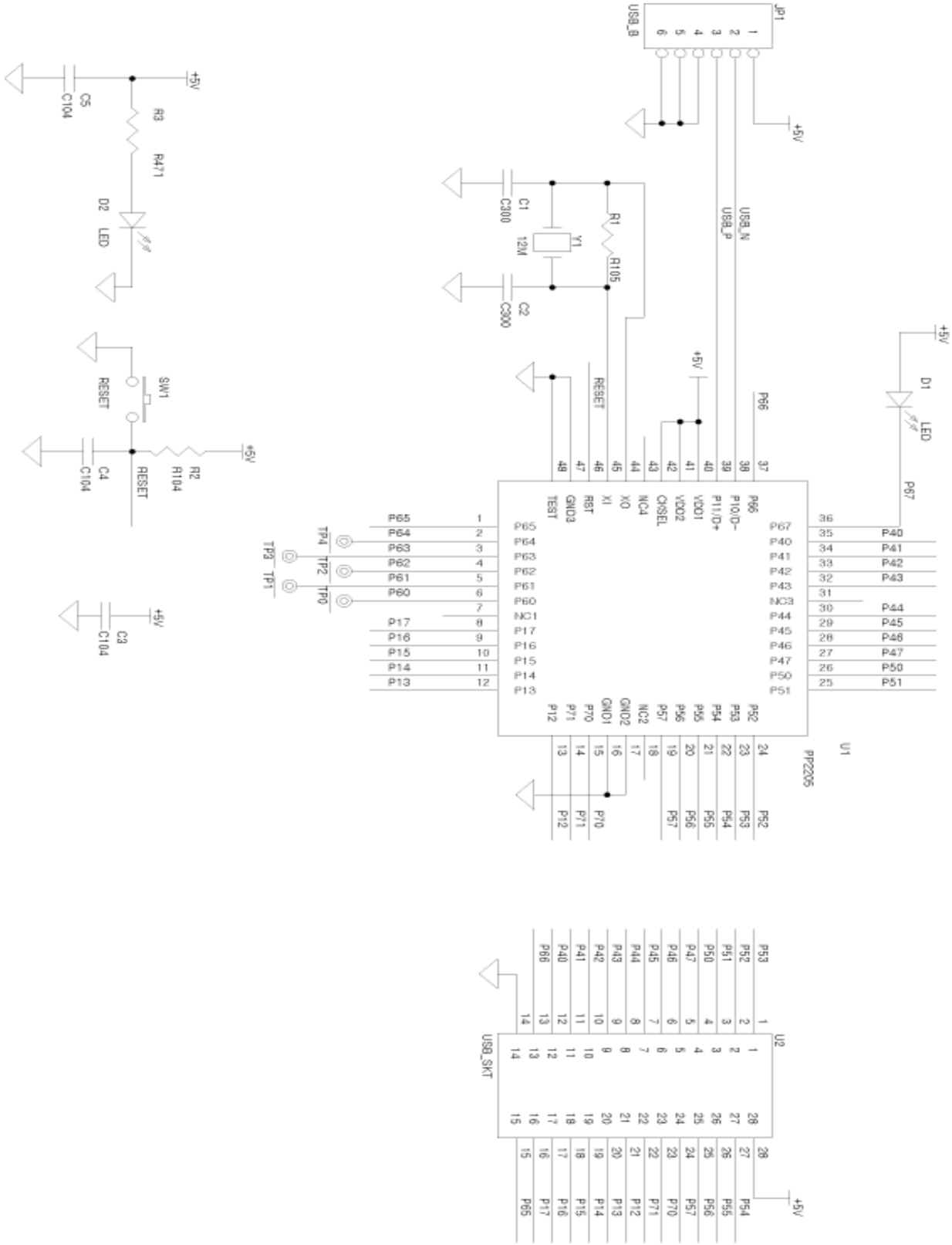
5.2 Physical Dimension



[Figure 5.1 Physical Dimension Diagram]

Appendix

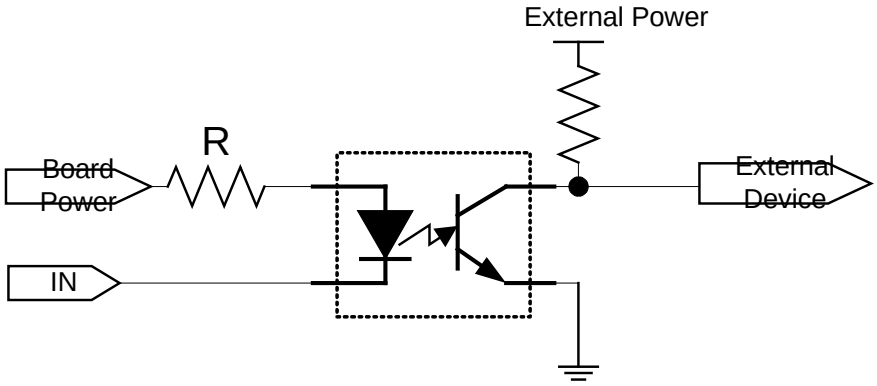
A.1 Schematic



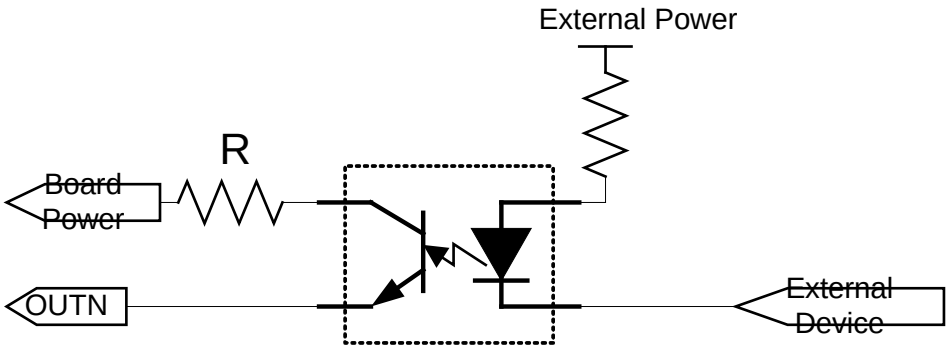
A.2 Application Circuit (Isolation Input/Output)

It is used in the automations that used PC in case of USB-LT in order to simply input/output that it used the established serial (RS232), parallel (printer) ports. The advantage of USB-LT has no program confliction because of using exclusive USB source, on the other hand, RS232 and printer ports have a confliction between programs because of using common source. Also, control Input/output is possible as provide 31 I/O.

If it exchange an input and output between different equipment like factory automations etc., it use a photo-coupler isolation in order to not to affect with mutual interference and influence at the below pictures.



< Photo-coupler Input Circuit>



<Photo-coupler Output Circuit>

A register value at the above pictures can use properly selection which it fit to the external power. The industry power mainly uses 5V, 12V, 24V, 48V.

A.3 API

Board Level API Functions

Overview

int **InitBoard**(void)

int **PortConfigure**(WORD P4, WORD P5, WORD P6, WORD P7, WORD P1)

int **PortRead**(BYTE *P4, BYTE *P5, BYTE *P6, BYTE *P7, BYTE *P1)

int **PortWrite**(BYTE P4, BYTE P5, BYTE P6, BYTE P7, BYTE P1)

InitBoard

int **InitBoard**(void)

It initializes USB-LT board. After power up, before using the board initialize first.

Parameters:

None

Return Value:

If the initialization fails, it returns “0”.

If the initialization succeeds, it returns “1”.

PortConfigure

int **PortConfigure**(WORD P4, WORD P5, WORD P6, WORD P7, WORD P1)

It decides whether or not to use input or output each port.

Parameters:

It decides input/output by each bit values of P4, P5, P6, P7, P1. The below pictures is sample of P4 value.

Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Port4.7

15/14

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-ch open drain output
1	1	N-ch open drain output with pull-up

Port4.6

13/12

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-ch open drain output
1	1	N-ch open drain output with pull-up

Port4.1

3/2

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-ch open drain output
1	1	N-ch open drain output with pull-up

Port4.0

1/0

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-ch open drain output
1	1	N-ch open drain output with pull-up

P4, P5, P6, P7 is same type, There is only port0 and port1 in case of P7 for reference. So, a value of bit 0,1,2,3 is meaning.

In case of P1 is as follows.

Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Port1.7 - Port1.0

7 -- 0

0	Pull-up disable
1	Pull-up enable

A P1 is same handling standard Port1, don't need to dedicate input/output, But it decide whether or not to stick an internal pull-up resister. It is meaning from bit0 to bit 7 of parameter P1, it is meaningless from bit15 to bit 8.

Return Value:

If the initialization fails, it returns "0".

If the initialization succeeds, it returns "1".

PortRead

int **PortRead**(BYTE *P4, BYTE *P5, BYTE *P6, BYTE *P7, BYTE *P1)

It returns current input state of each port.

Parameters:

None

Return Value:

If the initialization fails, it returns "0".

If the initialization succeeds, it returns "1".

PortWrite

int **PortWrite**(BYTE P4, BYTE P5, BYTE P6, BYTE P7, BYTE P1)

It outputs designated value of each port.

Parameters:

None

Return Value:

If the initialization fails, it returns "0".

If the initialization succeeds, it returns "1".

Port1Write

int **Port1Write**(BYTE act, BYTE pos)

It outputs designated value of each bit of port1. 0 and1 of port1 are used to USB data lines, it cannot record an entire value. You shall perform separately a writing command each bit (except bit0 and bit1).

Parameters:

act: If it is '0', it writes "0". If the value isn't '0', it writes '1'.

pos: It indicates a bit position(from 2 to 7). Last value is meaningless.

Return Value:

If the command function fails, it returns "0".

If the command function succeeds, it returns "1".

References

1. USB 2.0 System Architecture

-- Don Anderson, USB SIG (www.usb.org)

2. Universal Serial Bus Specification

-- Compaq/Intel/Microsoft/NEC, MindShare Inc. (Addison Wesley)